

統計数理研究所

研究レポート48

社会調査データの質の統計的評価のための
数理的処理および管理体系の開発

統計パッケージ MINERVA

— 利用の手引 —

(第1版)

1980年3月

統計数理研究所

社会調査データの質の統計的評価のための 数理的処理および管理体系の開発

この研究リポートは、昭和52・53・54年度特別事業「社会調査データの質の統計的評価のための数理的処理および管理体系の開発」の研究成果報告書として作成されたものである。

当研究所では、現在、

Annals of the Institute of Statistical Mathematics

統計数理研究所彙報

Computer Science Monographs

を発行している。

このリポートは研究調査のデータの発表等を目的とし、必要に応じて発行する。

統 計 数 理 研 究 所

東京都港区南麻布4-6-7

電話(03)446-1501

(★) 本レポート内に掲載の内容を，ご利用希望の方は，事前に書面にて第6研究部までご連絡下さい。

ALL RIGHT RESERVED.

NO PART OF THIS PUBLICATION MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY, RECORDING, OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM, WITHOUT PERMISSION IN WRITING FROM THE INSTITUTE.

目 次

1. はしがき	3
1.1 開発の意図と経過	3
1.2 MINERVA とは	4
1.3 MINERVA の特徴	5
2. MINERVA の利用法	7
2.1 基本的な使い方とその一般規則	7
2.1.1 入力データの形式	7
2.1.2 MINERVA 命令文	7
2.1.3 MINERVA ジョブと MINERVA ファイル	7
2.1.4 MINERVA ジョブの基本規則	8
2.1.5 記号・記法の一般規則について	9
2.2 MINERVA 命令文	10
2.2.1 MINERVA ジョブ制御用命令文	10
2.2.2 データの加工および検証用命令文	17
2.2.3 統計分析用の命令文	30
3. MINERVA の構成	52
3.1 MINERVA プログラムの構成図	52
3.2 MINERVA ジョブの処理の概略	53
3.2.1 基本的な処理の流れ	53
3.2.2 いくつかの分析処理の流れ図	53
3.2.3 その他の事項	58
3.3 出力メッセージ一覧表	63
3.4 MINERVA 命令文の一覧表	64
3.5 例題集	66
3.6 MINERVA プログラム・リスト	70
参考資料（参考文献、パッケージ・マニュアルのリスト）	231



1. は し が き

「MINERVA 利用の手引」は、特別事業「社会調査データの質の統計的評価のための数理的処理および管理体系の開発」(昭和52~54年度)の表題のもとに進めてきた研究成果の一部である。

MINERVA (MINi-package for Evaluating and Rating the VAlidity of survey data)は、主として社会調査データのようなカテゴリカル・データの統計処理を目的として開発したパッケージである。そして、「MINERVA 利用の手引」は、このパッケージの機能を紹介するために用意したものである。

1. 1 開発の意図と経過

諸外国に比べて立ち遅れているといわれてきた国内の統計パッケージに対する関心は、SPSS や BMDP などの普及により、その利用価値が認識され、最近是一段と強まりつつある。

それと同時に国内にも、独自の統計パッケージを開発しようとする動きが目立ってきた。現に、かなりの規模のものを作成し、公表しているグループがみられる。われわれの知る限りでは、たとえば、次のようなパッケージがすでに稼働を初めている。

- SALS (Statistical Analysis with Least-Squares Fitting)
- TIMSAC-78
- UMS (Universal Mathematical Software-system)
- ETPS (Editing and cross Tablation Programs for Survey analysis)
- SPMS (Statistical Package for Medical Science)
- MAP (Multivariate Analysis Program Package)

これらは、サブルーチン集合に近い形のものから、専用言語を備えた、いわゆるコマンド方式をとるものまで様々である。

ところで、これらのパッケージは、ETPS の例を除き、一般に社会調査データに多いカテゴリカル・データの取り扱いには不向きといわねばならない。唯一の例外である ETPS にしても、その使用言語が ALGOL であることやその他の制約から、他の計算機への移植可能性は薄く、もっぱら開発した企業自体の営業用のソフトウェアとして稼働している。

一方、諸外国とくに米国に目を移すと、様々の目的を持つ(と宣伝されている)パッケージが目白押しに次から次へと誕生している。また既に長い歴史をもつ SPSS, BMDP, OSIRIS-III, SAS などのパッケージを見ても、その規模の増大や改編作業は留まるところを知らず、それに伴って開発当初の設計意図あるいは主張が薄れて次第に汎用化あるいは網羅的になる傾向がみられる。

ところで、われわれが日常接するデータは主として、カテゴリカル・データと総称される、社

会調査や意識調査などのそれである。そして過去の経験則からいえば、高度で個別的な統計解析手法に到る過程として、その解析手法自体に要する何層倍もの労力がデータの整合、検証その他の事前処理加工に投入されている。これは、国内における SPSS の利用状況をみても十分にうなづけることである。すなわち、その内容を仔細に点検すると利用頻度の大部分がデータ加工、編集などのサービス機能や単純なクロス集計などに集中していることがわかる。

それにもかかわらず、国内にはデータ検証やサービス機能に重点を置くカテゴリカル・データ向きのパッケージは依然として例が少ない。われわれの意図したことは、ここに力点を置いた上で、なお簡単な統計解析を可能にする自前のパッケージを開発することにある。3年間の予定で開発を始めたが、初めの1年間はもっぱら「統計パッケージとはいかなるものか」を探ることに重点を置き、2年目に実際の作業に入った。(この時点までの統計パッケージに関連した調査内容については文献〔17〕,〔18〕などにまとめて報告してある)。

その後約1年半を要して MINERVA の原型の製作を終え、この「MINERVA 利用の手引」をもってその第1版を公開できるまでに到った。

なおこの MINERVA 開発に関与した人々は次のとおりである。

鈴木 達三, 水野 欽司, 鈴木 義一郎, 大隅 昇, 桂 康一, 長坂 和子 (以上, 統計数理研究所), 柳澤 幸雄 (日本科学技術研修所), 大内 道夫 (トーエイ B M(株)), 村石明彦 (日本大学)。

とくに、実際のプログラムの設計, コーディング, 点検・検査などは次の MINERVA 作業班が担当した。

大隅 昇, 桂 康一, 柳澤 幸雄, 大内 道夫, 村石 明彦。

1.2 MINERVA とは

MINERVA は、主としてカテゴリカル・データの事前検証, 簡単なサービス機能, 統計分析などを目的として試作されたパッケージである。

類似目的のパッケージとしてよく知られまた広く利用されているものに SPSS, OSIRIS-III などがある。これらはいずれも規模が大きくどちらかといえば汎用的であるが、MINERVA はこうした多目的機能を備えたものではなく、むしろ限定された範囲のデータ処理を行う専用のパッケージであり、規模もまた小さい。

このパッケージは、既存の多くの統計パッケージやサブルーチン集合と、原データとの接触部にあって、本格的な統計解析への導入部におけるデータ事前検証の役割に重点を置いたパッケージであるといえよう。

ところで近來のパッケージの特徴の一つとして、固有の命令語体系を備えているということを指摘できる。またこれ을押し進めて、会話型のコマンド方式や、さらにコンパイラ型の自己言語をもつパッケージも現われている。

こうした傾向はもはや常識とされているが、MINERVA も、専用の“簡略化命令文”である

“MINERVA 命令文”を備えており、利用者はこれを用いてパッケージを容易に利用できるようになっている。この“利用の手引”は、いわばこの命令文体系の使い方を説明したものである。利用者の立場から考えると、個々の解析プログラムの功緻性もさることながら、“使い易さ”がきわめて重要であろう。MINERVA はなるべくこの点に配慮したつもりであるが、結果として出来上ったものを眺めると必らずしも十分とはいえない。これは、作業班自体の開発努力の不足もあるが、他機種への移植の可能性などまで考慮すると、この程度のものに留らざるを得なかったという事情によっている。

1.3 MINERVA の特徴

データ解析において、データ編集を初めデータ・クリーニング（あるいはランドリー）やデータ・スクリーニングなどの事前処理が重要であり、事実、最も労力と注意を要するところである。

たとえば、

- オフ・コード・チェック
- データのコード許容範囲の検証
- 項目間の論理性の検証（回答の脈絡の矛盾性の検証）
- 項目間の比率のバランス・チェック

などがそれである。こうした作業を十分に行わずに統計解析に入るとはその分析結果の信頼性自体をも危うくする。

こうした検証と同時に、データを統計解析が可能な形に整理する機能も必要である。いわゆるデータ・サービス機能であり、必要データの選出、コードの変換、解析手法間での自在なデータの授受機能などをさす。

さらに、従来の計量的データ向きの解析手法では十分な結果が得られなかったカテゴリーカル・データに適した統計解析手法も必要となろう。とくに項目間の関連性分析を合理的に進めるモデル解析法が望まれる。

MINERVA はこうした役割の一端をになうもので、具体的に次の特徴を備えている。

- (1) 分析対象とする原データ（カード・データ、ファイル・データ）から、利用者が当面必要とするデータを必要な時に取り出す。
- (2) 簡単な命令文により利用できる（処理の自在性）。
- (3) 複数個の分析処理機能を連結・一括処理できる。
- (4) データ検証の機能をもつ。
- (5) カテゴリーカル・データ向きのモデル解析機能が含まれている（対数線形モデルを中心とする項目間の関連分析）。
- (6) 検証済みあるいはサービス機能処理済みデータを他の解析プログラムに引き渡すことができ

る。

(7)プログラムの増殖可能性をある程度考慮してある（新プログラムの追加が容易である）。

この「利用の手引」作成中にも、改編の作業は進行しており、パッケージの機能は変容が続いている。手引の中でも部分的に予定の事項がいくつか含まれているのはこのためである。

また、ここではあくまで「利用の手引」に焦点をあて、各分析手法の詳細な解説に立ち入ることは避けた。これは、扱っている内容が主としてデータ処理に関わるものでさほど難解なものではないということもあるし、また立ち入った解説を付与して利用者の混乱を招くということは避けたいという心積りもあったためである。

しかし、関連係数や対数線形モデルなどに関連した部分は、やはり若干の解説を必要とすると思われるがこれらについては関連資料を参考文献としてあげてあるのでこれを参照されたい。

こうしたパッケージ作成はつねに無窮の作業であるが、とりあえずこの手引を公開することにより一つの区切りとしたい。

2. MINERVA の利用法

2. 1 基本的な使い方とその一般規則

2.1.1 入力データの形式

MINERVA では、いわゆる (サンプル) × (項目) あるいは (個体) × (変数) の **多変量特性データ行列** を扱う。行側は測定対象を表わし、列側は特性項目を示している。また原則として、**数値型データ** だけを扱う。そして分析対象として与えられたデータ行列は 1 サンプルを 1 単位として扱う。

2.1.2 MINERVA 命令文

分析処理内容の指示はすべて MINERVA 独自の **命令文** により行う。この命令文は必ずカードで与える。MINERVA はこれらのカード群をデータとして読み取り、内蔵した解読サブプログラムにより解読照合の上指定の順に従って処理をすすめる。

命令文は、カード上の 80 カラムを 1 ～ 15 カラム (第 1 区分) と 16 ～ 80 カラム (第 2 区分) の 2 つの区域に区分して利用する。**第 1 区分** で指定する内容を **命令語** または **コマンド** と呼ぶ。また、**第 2 区分** は命令語で指示した内容の明細を示し **指定文** と名づける。

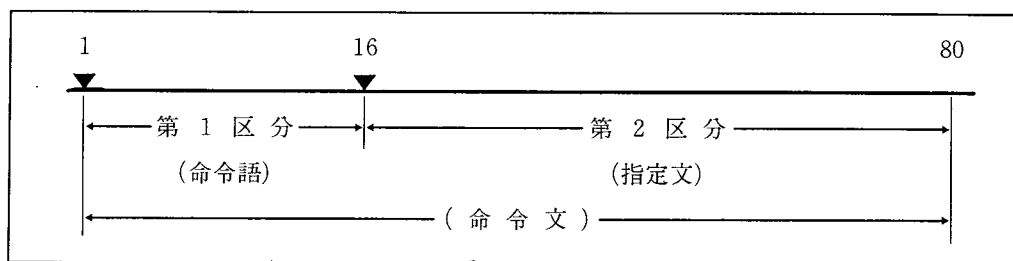


図 1 命令文カードの書式

2.1.3 MINERVA ジョブ と MINERVA ファイル

命令文の集まりから構成される 1 つのジョブの流れを **MINERVA ジョブ** と呼ぶ。また、入力用の命令文により原データ行列から必要項目だけを抽出し、磁気ディスク・ファイル上に格納した MINERVA ジョブ実行用のデータ行列を **MINERVA ファイル** (データ) と名づける。

(→ READ 文, FORMAT 文, ITEM 文の項参照)

2.1.4 MINERVA ジョブの基本規則

命令文の指定にあたって少くとも次の各項を守ることが必要である。

- 1) 1つの命令文の効力の範囲は命令語に初まりスラッシュ(/)で終わる(いくつかの例外はある)。
- 2) 1つの命令文内でのカードの継続は許される。しかしその枚数には制限がある(分析内容により少しずつ異なる)。
- 3) そのときの継続カードの記入は指定文区域(第2区分)に対して有効である。
- 4) ブランクカードの使用が許される。

命令語について

命令語とは分析処理の内容を具体的に指示するキーワードのことである。これには大別して次の3つがある。

- a) MINERVA ジョブ制御用
- b) データの加工および検証用
- c) 統計分析手法用

命令語の記法は次の規則に従って指定する。

- 1) 必ず第1区分内に、左づめで第1カラムから穿孔すること。
- 2) 先頭の4文字が解読のためのキーワードであるから、あらかじめ決められた綴りでなければならない。

命令語の種類

命令語として現在、表1に挙げたものがある。機能によって制限事項に多少の違いがあるがこれらの詳細は順を追って個別的に説明する。また、命令文の一覧表は3.4節にあげてある。

指定文について

指定文とは、命令語で指示した処理内容の明細を与えるものである。

- 1) 指定文内に定義する項目名、数値、演算子などは少くとも1つ以上の区切記号により区分する必要がある。ここで区切記号としてはスラッシュ(/)、波(～)、左右括弧((,)), 数値符号(一, +)を除くほとんどの特殊文字が利用できるが、基本的には、空白(), カンマ(,), コロン, セミコロン(:, ;), 等号(=)などを利用するのがよい。
- 2) 指定文が複数枚のカードにまたがるとき、区切記号部以外の部分で切断・継続してはいけない。
- 3) 指定文は第2区分内のどこに位置してもよい。
- 4) 指定文の継続は必ず次のカードの第2区分内に書く。

表1 命令語の一覧表

MINERVA ジョブ制御用	データ加工・検証用	統計分析用
<u>READ</u> <u>FORMAT</u> <u>ITEM</u> <u>INPUT DEVICE</u> <u>OUTPUT DEVICE</u> <u>OPTION</u> <u>END</u>	<u>RANGE</u> <u>CONSISTENCY</u> <u>PATTERN</u> <u>FIND</u> <u>RECODE</u>	<u>ASSOCIATION</u> <u>BREAK DOWN</u> <u>MULTIWAY TABLES</u> <u>QCHART</u> <u>CROSS TABLES</u> <u>BMDP</u> <u>FINISH</u> /

2.1.5 記号・記法の一般規則について

この「利用の手引」を通じて使用する記号と記法の概略をまとめておく。

- 1) 命令文カードは図1の書式で統一的に表わす。
- 2) 命令文中の下線部は必ず指定すべき部分で、解読のキーワードとなる（ただし、項目名は必ず指定しなくてはならないので特別の場合を除いて下線は省略する）。
- 3) [] は省略可能であることを示す。
- 4) { } により指定の内容を説明する。
- 5) *n*, *nn*, *m*, *mm*, *l*, *u*, *k* などの小文字のアルファベットにより数値を表わす。個数が桁数を示す。
- 6) 項目名としてA, B, C, ……., X, Y, Z など、アルファベットの大文字を使うことがある。

2.2 MINERVA 命令文

2.2.1 MINERVA ジョブ制御用命令文

MINERVA ジョブ制御用の命令語として次の6種類がある。

表 2

命 令 語	機 能
<u>READ</u>	データの入力指示。機器・機番の指定。
<u>FORMAT</u>	データの入力書式の指定。
<u>ITEM</u>	必要データの項目名の付与。
<u>INPUT DEVICE</u>	加工済みデータの再入力指定。
<u>OUTPUT DEVICE</u>	加工済みデータの出力用作業ファイルの指定。
<u>END</u>	MINERVA ジョブの終了の指示。

これらを順を追って説明しよう。

READ 文

〔機能〕

分析データの入力機器および入力機番の指定を行う。

入力機器あるいはデータの入力媒体としては磁気テープ、磁気ディスク、およびカード・データの3種が利用できる。

〔指定方法〕

入力媒体の種類に対応して次の3通りの指示方式がある。

▼	▼
<u>READ</u>	[<u>TAPE</u> <u>nn</u>]
<u>READ</u>	<u>DISK</u> <u>nn</u>
<u>READ</u>	<u>CARD</u>

上から順に入力媒体が、テープ、ディスク、カードの場合にそれぞれ相当する。

磁気テープまたは磁気ディスクからデータを入力する場合には、入力機番 (*nn*) を指定する。

ただし磁気テープの場合に限って、機番指定の省略が可能である。

〔オプションおよび標準値〕

- 1) 磁気テープからのデータ入力が標準値である。したがって、磁気テープ・データの入力時は、READ 文は省略してよい。このとき、機番は $nn = 09$ が与えられる。
- 2) 命令語 READ だけを指定して、指定文を省略した場合、標準値を適用する。

〔制限事項〕

- 1) 磁気ディスクからのデータ入力するとき、機番の指定は不可欠である。
- 2) カード・データを入力の場合、そのカードは命令語 END の次に置く (→END 文の項参照)。
- 3) 指定文をスラッシュ (/) で閉じてはいけない。

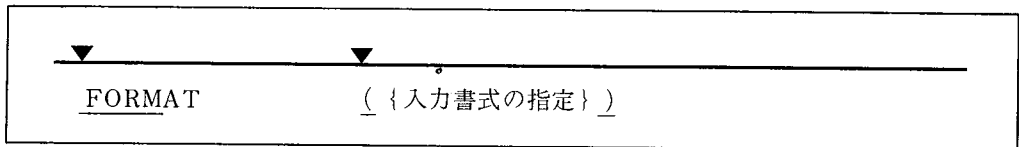
FORMAT 文

〔機能〕

分析対象データから解析に必要な特性項目を指定する入力書式の命令文である。

〔指定方法〕

基本的な書式は次のとおりである。MINERVA ジョブにおいては、この命令文の省略は許されない。



〔制限事項〕

- 1) 命令文カードの継続は10枚まで許される。
- 2) 入力データの型は実数型入力 (Fタイプ) に限られる。
- 3) 指定する項目数 (変数特性の数) は ITEM 文で指定する項目名の総数に一致しなければならない。
- 4) 指定文をスラッシュ (/) で閉じてはいけない。

ITEM 文

〔機能〕

FORMAT 文で指定の各項目に名称を付与する。これを指定することにより MINERVA ジョブ内のデータの参照は、すべて与えた項目名により行うことができる。また分析結果の出力リストには必ずこの項目名が表示される。

〔指定方法〕

この命令文の省略はできない。項目名は **8 文字以内の英数字** および区切記号以外の特殊文字を使って付与する。各項目名の間は 1 個以上の区切記号 [ブランク (), カンマ (,) など] により仕切る。また ITEM 文は FORMAT 文に続いて置かなければならない。

<div style="text-align: center;">▼</div> <div style="text-align: center;">▼</div> <div style="display: flex; justify-content: space-between;"><div><u>ITEM</u></div><div>{項目名のリスト} /</div></div>
--

〔制限事項〕

- 1) 項目名の先頭文字は英字だけが許される。
- 2) 命令文カードの継続枚数には制限はない。
- 3) 項目数の数は最大50個である。これは原則として、すべての命令文に対して適用される（ただし、MINERVA 作業領域の初期割当ての大きさにより可変である。また、課題内容によって若干項目数の制限が異なる）。
- 4) 項目名が 8 文字を越えた場合、先頭から 8 文字だけが採用される。残りの部分は無視される。
- 5) ITEM 文はつねに READ 文および FORMAT 文と合わせて用いなければならない。
- 6) ITEM 文は必ずスラッシュ (/) で閉じる。

〔注意事項〕

一般に、項目名を個別に与えることはわずらわしい。この手当てとして次の方式が考えられる。

- 1) X 1 - X 10, X 1 _ TO _ X 10 などのように指定すると X 2 から X 9 までの項目名を自動的に付与する。この方式は、既に SPSS, OSIRIS-Ⅲ など多くのパッケージで採用している。
- 2) FORMAT 文で入力した項目名数を解読し、自動的に適当な名称を付与する。たとえば FORMAT 文で、(4 F 3.0, 2 F 5.0) と指定すると自動的に V 1 から V 6 まで、あるいは X 1 から X 4, Y 1, Y 2 と区分に対応して、それぞれ 6 個の項目名を生成する、という方式、も考えられる（現在、この方式 2）を開発予定中である）。

以上3種の命令文 READ, FORMAT, ITEM を指定することで, MINERVA ジョブの実行に必要な分析用データが磁気ディスク上に用意される。この作業用データファイルを **MINERVA ファイル** と名づける。

〈注〉 MINERVA ファイルの構成は次のとおりである。

- 一時作業ファイルの標準機番は11である。
- 出力書式はリストつきの binary data である。
- 各サンプルには、先頭にサンプル番号が自動的に付与される。
- RECODE 文を適用すると標準機番の変更がおこることがある (→ RECODE の項参照)。

ここでいくつか簡単な例を挙げておく。

〈例題1〉

▼	▼
READ	TAPE 8
FORMAT	(1 X, 2 F3.0, F4.3)
ITEM	SCORE, PERCENT, JOBA /
▼	▼
FORMAT	(F2.0, 2 X, 4 (2 X, F1.0))
ITEM	IDCODE, NATION, AGE, Q1, Q8.3 /
▼	▼
READ	DISK = 15
FORMAT	(5 F10.2)
ITEM	INCOME, OCCUPY, EDUCATN, V4.2-5, X3:8 /

INPUT, OUTPUT 文 [一時作業ファイルの指定]

〔機能〕

検証, 加工など, 分析処理を終えたデータを他の分析処理で利用するために一時的に格納したり (OUTPUT), 先に格納したデータを取り出して利用する (INPUT) 場合に, この命令文を使う。一時的に作業用ファイルとして利用するための指定であるから省略が可能である。

〔指定方法〕

次の書式で指定する。

▼	
OUTPUT DEVICE	<u>nn</u>
INPUT DEVICE	<u>nn</u>

なお, これを利用する場合には, 適用したい分析の命令語の前または後に必ずおく。このとき INPUT は必ず適用課題の前に, OUTPUT はその後に置く。

ここで例題により説明する。

〈例題 2〉

▼	
①	[FORMAT (3 F 4.1) ITEM SEX, AGE, Q2.1 /
②	[RANGE SEX (1, 2), AGE = 1 ~ 8 / OUTPUT 12
③	[INPUT 12 ASSOCIATION SEX, AGE, Q2.1 /
END	

初めに FORMAT 文, ITEM 文の指定にしたがって, 磁気テープ上のデータから 3 つの項目を指定し, SEX, AGE, Q2.1 と名称を付与する (処理①)。

続いてデータの範囲の検証を RANGE 文で指示する。性別 (SEX) はコードが 1 と 2, 年齢 (AGE) はコードが 1 ~ 8 の範囲にあるか否かを, この命令文で指示する。そして検証結果の情

報をリスト上に出力すると同時に、条件を満足した適切データをファイル機番12の作業用ファイル上に出力する（処理②）。

さらに、この範囲の検証済みデータを使って、3つの項目間の関連係数を算出するために処理の③を指示する。

最後に END 文で MINERVA ジョブの終了を指示する。ここで仮りに次のような指示を命令文で与えた場合の処理についても説明しておこう。

（1）RANGE の前に INPUT 12 を挿入した場合。

このときには、機番12にはデータが存在しないのでエラーメッセージを出力した上で次の処理すなわち③へ移る。ここでも②の処理をおえていないので機番12のファイルにはデータが未定義である。よってエラー表示となる。

（2）RANGE の後の「OUTPUT 12」を削除した場合、RANGE の処理をおえ検証結果をリストに表示する。次に、③の処理に移るが、機番12にデータが存在しないのでエラーを表示して、ジョブを終了する。

（3）処理③で、INPUT 文を削除した場合。

RANGE の処理を終えたのち、合格データをファイル12へ出力する。続いて、ASSOCIATION は初めに生成した MINERVA ファイルに対して適用される。

そして、機番12のファイル上には RANGE 検証済みのデータが MINERVA ジョブの終了時まで格納保存されている。したがって、別の分析で改めて INPUT 文により呼び出した上、再利用することができる。

〔制限事項〕

- 1) 機番の指定は不可欠である。機番コード 'nn' は番号12から20の範囲で指定することが望ましい。とくに機番10以下の指定は望ましくない。
- 2) 同一ファイル機番の反復利用が許される。ただし、この場合順次ファイル内容が更新されるので、前の格納データは保存されない。

〔オプションおよび標準値〕

- 1) 入出力機番の指定がないとき、標準値として入力機番は $nn = 11$ 、出力機番は $nn = 00$ が与えられる。したがって、出力の場合はデータの出力は行なわない。
- 2) 一時作業ファイルは磁気ディスクが標準機器である。

〔注意事項〕

MINERVA ジョブの結果を他の解析プログラムに利用したいとき、その処理済みのデータを、上の命令文を使ってファイル上に一時格納し、これを他の解析プログラムに引き渡すことができる。すなわち MINERVA ジョブとその解析プログラムとを、マルチ・ジョブ・タスクとして、1つのジョブにまとめて実行すればよい（→61頁を参照）。

END 文

〔機能と指定方式〕

MINERVA ジョブの終了を指示する命令文である。必ずジョブの最後に置く、ただし、入力データがカードである場合には、この文の後にデータ・カードをつける。

END 文を省略した場合、それ以前に指定した命令文をすべて終えた上で、この命令文の解読が不可能であることを示すエラー・メッセージを出力する。

ジョブ制御命令文の一般的注意事項

- (1) ジョブ制御用命令文の多くは、スラッシュで閉じる必要はない。必要があるのは、ITEM 文、OPTION 文の 2 つである。
- (2) OPTION 文の説明は分析課題との関連で指定文が異なるので各関連分析課題の解説部分にある。
- (3) ITEM 文で指定の項目名は各分析課題の出力リストの中で必ず利用されるので、出来るだけ制限の 8 文字をすべて使って、理解の容易な名称を付与しておくと便利である。

2.2.2 データの加工および検証用命令文

RANGE 文 [範囲の検証, データの選出]

[機能]

各項目のとりうるカテゴリーの許容範囲の検証を行う。とくに次の処理に適用できる。

- 1) ある項目のカテゴリーが事前に定めた規定コードの範囲に入っているか否かの検証。
- 2) その場合の, サンプルと不適切データのサンプル数の確認。
- 3) 希望するカテゴリーに該当するサンプルの選出 (いわゆるデータ選出機能)。
- 4) 3) の応用として, 特定なコードの組み合わせを持つサンプルの選出。

[指定方法]

命令語 RANGE に続き, 指定文として, 範囲の検証を行う項目名と, その範囲を指定する。範囲はその上限 (u) と下限 (l) とを数値で指示する。数値は実数, 整数 (正, 負) いずれも許される。多数項目名に対して検証する場合には, これを1単位として, 反復指定すればよい。一般的な書式は次のとおりである。

RANGE	{ {項目名 A, A の検証範囲の指定}, {項目名 B, B の検証範囲の指定}, {項目名 C, C の検証範囲の指定}, } /
-------	---

ただし, 次の点に留意することが必要である。

- 1) {項目名, 下限 (l), 上限 (u)} の順に指定すること。

〈例題3〉

- | | |
|----------|--------------------------------------|
| ① [RANGE | NATION = 1 ~ 2, INCOME = (10~20.5) / |
| ② [RANGE | AREA - 3, 6 CLASS (1, 3) / |

①は, NATION が1~2の範囲にあるかを検証する。INCOME についても同様である。

②は, AREA が-3から6の範囲にあるか, CLASS が1から3の範囲にあるかを検証する。

- 2) 項目名と l , u との間は少なくとも1つ以上の区切記号で仕切ること (なるべく, 空白 (),

カンマ (,), 等号 (=), などの利用が望ましい。

- 3) ある1つの特定コードだけの検証あるいはデータ選出を行いたい場合には, l , u の区別は不要であるから次の例のように指定すればすむ。

〈例題4〉

	▼	▼	
①	[RANGE	NATION 1 /
		RANGE	NATION (1, 1) /
②	[RANGE	AGE = 2 /
		RANGE	AGE = 2 ~ 2 /

①の2つは同じ機能をもつ。同様に②の2つも処理の内容は同じである。

〔制限事項〕

- 1) l および u の数値が負の場合に, 負記号 (→) が許されるが, 正数値のときには正記号 (+) を与えることはできない。
- 2) 1つの命令文内で指定された内容の論理処理は, それらの共通部分 (and) として扱う。

〈例題5〉

▼	▼	
RANGE	AGE (1, 8), NATION	1 ~ 2 /

このとき, 「AGE が1から8の範囲にありかつ NATION は1, 2である」を満たすサンプルが適切データであり, これを満たさぬサンプルはすべて不適切データとして排除し, 出力情報として表示する。

- 3) したがって, 項目を個別的に, ある範囲の検証やデータ選出を行う場合には, 1項目名ずつ処理するか一時ファイルを利用するかしなくてはならない。

〈例題6〉

	▼	▼	
	[RANGE	QA (3 ~ 4), QB (3, 5), CLASS = 1, 2 /
		OUTPUT	12
①		INPUT	12
		CROSS TABLE	SIDE = QA, QB, HEAD = CLASS /

②	RANGE	QA (3, 4), CLASS (1, 2) /
	OUTPUT	13
	INPUT	13
	CROSS	SIDE = QA, HEAD = CLASS /

①のように指示すると、

「QA が3～4にあり、かつ、QB が3～5で、かつ、CLASS が1または2」を満たすサンプルが適切データとして選出される。次にこのデータに対して、2つのクロス表QA×CLASS, QB×CLASS のクロス表を作成するが、このとき QA はコードが3～4 を満たすサンプルのすべてではない。つまり QB, CLASS の条件つきであることを注意しておく。これは QB についても同様である。かりに QA が3, 4 を満たすサンプルのすべてを取り出して、CLASS とのクロス表を出力する場合には②のように与えなければならない。

CONSISTENCY 文

〔項目間の論理性の検証〕

〔機能〕

ふつう、調査データには、項目の個別的検証だけでは発見できない、項目間の論理矛盾による誤りが含まれている。

たとえば、典型的な例として、「性別は男である」と回答しながら、「職業は主婦」と回答しているような場合がそれである。これは明白な誤りであるがこれ以外にも、「年齢は60才代で、職業が学生」とか「年齢は10代であって所得が1000万円」などの場合、絶対にあり得ないことではないが、一応誤りではないかと、疑いを持つ必要があろう。

これらはいずれも実在の例であるが、こうしたことを素早く検出しその該当サンプルを検索・表示して丁寧に見直す必要が出てくる。こうした機能を「論理性の検証 (consistency checking)」と呼んでいるが、これを行うことが、この命令文の目的である。

指定書式は次のように3つの型がある。

▼ ▼

CONSISTENCY IF (P) /

CONSISTENCY IF (P), THEN (Q) /

CONSISTENCY (P) IFF (Q) /

(※) P, Qには項目名間の論理式を代入する。

ここで、上から順にⅠ型、Ⅱ型、Ⅲ型と名づける。

(Ⅰ型) : CONSISTENCY IF (P) /

括弧内に適当な論理式Pを指定する。この条件を満たさぬサンプルを不適切データとして検出する。逆に条件Pを満たすサンプルだけの選出も可能である。

(Ⅱ型) : CONSISTENCY IF (P), THEN (Q) /

この場合、「条件Pを満たすとき、条件Qも満たさねばならない」という論理検証を行う。これを満足しないサンプルを不適切データとして出力表示する。

いま、次の真理表を考えよう。

表 3	条件P \ 条件Q	真 (T)	偽 (F)
	真 (T)	a	b
	偽 (F)	c	d

このとき、a 以外はすべて論理矛盾である。Ⅱ型の指定により、まずbに相当する情報を出力する。かりに条件Pが初めから偽(F)のとき、全サンプルが不適切データであると表示する。このとき、Pの内容を確認することが必要となる(上の表でc, dの場合)。

Ⅱ型をデータ選出に利用すると、aを満たすサンプルだけを適切データとして作業ファイル上に格納する。

(Ⅲ型) : CONSISTENCY (P) IFF (Q) /

「条件Pであれば条件Qである、また条件Qであるためには条件Pでなければならない」という必要十分条件の関係を検証する(IFFはif and only ifの略)。前掲の真理表で、bおよびcに相当するサンプルを検出する。条件Pが初めから偽(F)のときには、Ⅱ型と同様に全サンプルが不適切であることをエラー・メッセージとして表示する。

(Ⅲ型)をデータ選出機能として利用すると、aを満たすサンプルが作業ファイル上に格納される。したがってこのとき、(Ⅱ型)をデータ選出に適用した場合に結果が一致する。

[指定方法]

基本的な規則として次の事項がある。

1) 関係演算子として、次の6種が許される。

LT (<), LE (≤), EQ (=), NE (≠), GE (≥), GT (>)

2) 論理演算子として、OR, AND が許される。

3) 関係式の中で利用できる数値は、整数および実数である。

4) 項目名、数値、演算子の間には必ず1個以上の区切記号が必要である。

5) 関係式の多重連結が可能である。

6) 多重括弧を使うことができる。

7) したがって、関係式の処理優先順を括弧により指定できる。通常、内側におかれた括弧内の条件から優先的に処理する。

〈注〉FORTRAN 文法による記法にほとんど同じである。異なる点は演算子の前後にピリオド(.)を必要としないことである。

例題7は次の処理が行われる。

①:「年令(AGE)がコード8以下でかつ、国名(NATION)がコード1である」ことの真偽を検証する。

②:「職業(OCCUPTN)がコード1ではなくかつ、地位(STATUS)が3である」をまず検証し、これを満たすサンプルについて、さらに「または、収入(INCOME)が100.5(万円)以下である」ことの検証を行う。

③：まず「年令 (AGE) がコード 5 であるかまたは職歴 (CAREER) がコード 8 以上である」ことを調べ、これを満たすと同時に「性別 (SEX) が 1 であるか否か」を検証する。

いま (AGE EQ 5 ……GE 8) の前後の括弧を削除すると、
「性別 (SEX) が 1 でありかつ年令 (AGE) が 5 である、または職歴 (CAREER) が 8 以上である」ことを満たすか否かを検証するので上の場合と内容が異なる。

〈例題 7〉

- | | | |
|---|----------------|-------------------------------------|
| ▼ | ▼ | |
| ① | [CONSISTENCY | IF (AGE LE 8 AND NATION EQ 1) / |
| ② | [CONSIST CHECK | IF ((OCCUPTN NE 1 AND STATUS EQ 3) |
| | | OR INCOME LE 100.5) / |
| ③ | [CONSIST TEST | IF (SEX EQ 1 AND (AGE EQ 5 OR |
| | | CAREER GE 8)) / |

〈例題 8〉

▼	▼	
CONS		IF (SEX EQ 1), THEN (OCCUP NE 7) /

「性別 (SEX) が男 (コード 1 とする) であると、職業 (OCCUP) は主婦 (コード 7 とする) であってはならない」の検証は、上のように指定すればよい。

▼	▼	
CONSIST		IF (AGE GT 7 OR OCCUPTN EQ 5),
		THEN (INCOME NE 5 OR INCOME NE 6) /
OUTPUT		12

「年令 (AGE) が 70 才代 (コード 7) であるか、または職業 (OCCUPTN) が学生 (コード 5) であるとき、年間所得 (INCOME) は 500 万円台 (コード 5) から 600 万円台 (コード 6) であることはない」の検証を行う。そして、これを満たすサンプルが作業ファイル上に格納され、条件を満たさぬ不適切データの情報はリストに表示する。

〈例題 9〉

CONST TEST IF (QA LE 2) IFF (QA.1 EQ 1) /

「質問 QA で選択肢 2 以下に回答したとき、副質問 QA.1 で 1 であること」また「QA.1 で 1 と答えたときには、QA では 2 以下でなければならない」という内容を検証する。

この例にみるように（Ⅲ型）は、副質問に対する分枝選択肢をもつ質問項目の回答矛盾の検証に役立つ。

CONSIST TEST (TRIP EQ 1) IFF (TIMES GE 5 AND
TIMES LT 10) /

「旅行 (TRIP) に行ったことがある (コード 1)」, しかも, 回数 (TIMES) は 5 回以上 10 回未満である」を上のように指示すると, 旅行に 6 回行ったと答えていながら, 旅行には行かなかった (TRIP がコード 0) と記録してあると, 誤りである。これを検出して, 回答者の間違いか, 記入のミスかなどを調べることができる。

〔制限事項〕

1) 関係式の区切部で命令文が継続カードにまたがることは許されるが項目名, 数値, 演算子そのものが切断, 継続することは許さない。

CONSISTENCY IF (NATION EQ 2 AND EDUC
AT EQ 3 OR Q 2.3 E
Q 1) /

EDUCAT が項目名として定義してあるとき, 上の例のようにそれが継続カードにまたがるとエラーになる (これを EDUC と AT の 2 つの項目名として解釈照合する)。また, EQ が, 例のように切断されても同様にエラーとなる。

2) 右括弧の総数と左括弧のそれとが一致すること。

FIND 文

〔サンプル参照機能〕

〔機能〕

この命令文は次の機能を持つ。

- 1) ある特定サンプルのデータ内容の参照。
- 2) 別の検証機能を実行して、不適切と判定されたサンプルの全データを参照したいとき。

〔指定方法〕

命令語 FIND に続いて、指定文として、内容参照したいサンプルの番号を任意に指定する。この番号は、MINERVA ファイル上のサンプル番号である。

FIND 文の書式は次の通りである。

▼
FIND DATA

▼
{ 参照したいサンプルの番号 } /

〈例題 10〉

▼
FIND DATA

▼
10, 5, 21 ~ 100, 998, 201 ~ 205 /

なお、基本的な規則として次の事項がある。

- 1) 参照番号は必ず正の整数で与える。
- 2) 各数値の間は、少なくとも 1 つ以上の区切記号が必要である。
- 3) 参照番号は大きさの順に指定する必要はない。
- 4) 連続した番号部分を参照したいとき、連続区間指定ができる。このとき区間を示す数値の間に記号 ‘～’ を挿入する (→ 上の例を参照)。

〔制限事項〕

- 1) 1 回の命令文で指定できるサンプル数は 500 個までである。
- 2) 連続区間指定のとき、‘～’ 記号で指定した 1 区分が複数の命令文カードにまたがってはいけない。
- 3) 1 回の命令文の中で参照番号の重複指定は許さない。

	▼	▼	
①	[FIND	10 ~ 11, <u>15 ~</u>	
		<u>18</u> , 19, 100 /	
②	[FIND FILE	10 ~ 15, 13 /	

これらの例はいずれもエラーとなる。

①は、上の事項 2) に該当する。②は事項 3) にあたる（番号13が重複指定である）。

PATTERN 文

〔回答パターンの確認とその度数表の作成〕

〔機能〕

たとえば回答がすべて2項選択肢であるような場合を考えよう。このとき項目数を m とすれば 2^m 通りの回答パターンがありうる。しかし、実際にはこれらすべてのパターンが出現することはまれである。むしろ、類似の回答パターンをとる回答者が多いであろう。そして回答者がどのような反応パターンをとる傾向にあるかを大まかに把握しておきたいというケースがよくある。また、数量化や POSA あるいはさらに別のカテゴリカル・データの解析手法の適用に際して、やはり回答パターンの傾向を知っておくことは有用である。こうした場面でこの機能の利用が考えられる。

〔指定方法〕

命令語 PATTERN の後に指定文として項目名を列挙する。この項目名のならびに従って回答パターンを出力表示する。なお回答パターンは出現度数が最大のものから度数の大きさに従って表示する。

一般形式は次のとおりである。

PATTERN

{項目名のリスト} /

次の例は、ITEM 文で定義した4つの項目のうちの3つの項目を SUBA, Q1, Q3 のならびで回答パターンのチェックを行う場合の指定方法である。

〈例題 11〉

```
FORMAT      (3 X, 2 F 5.1, 2 X, 2 F 2.0)
ITEM         Q1, Q3, SUBA, AGE /
PATTERN      SUBA, Q1, Q3 /
END
```

〔制限事項〕

- 1) パターンの総数は 1024 ($= 2^{10}$) 通りまで許される。これを越えた場合、MINERVA ジョブがその時点までに検出した 1024 通りのパターン内の度数だけについて、大小順をソートして出力表示し、次の命令文の処理に移行する。
- 2) データの数値の桁数が大きい場合、RECODE 文で数値変換を行ったうえでパターンの検証を指定することが望ましい。

RECODE 文

〔数値変換, 新コードの付与〕

〔機能〕

データの事前処理機能の一つであって、数値コードの変更、合併、連続量データのカテゴリー化などが、この命令文により指定できる。これをより具体的に整理しておく。

- 1) コードの変換・新コードの付与。
- 2) 連続量データのカテゴリー化。
- 3) 加工データの格納と他の分析での再利用。
- 4) データの一時変換機能と永久変換機能。

これらについて次の指定方法の説明の中で詳しく述べる。

〔指定方法〕

一般書式は次のとおりである。

RECODE

{ {項目 A (Aのコード変換指定) },
{項目 B (Bのコード変換指定) },
{項目 C (Cのコード変換指定) },
..... } /

命令語 RECODE の後に、指定文として、まずコード変換あるいはコード付与を行う項目名を与え、次にそのコード変換の内容を括弧内に指定する。変換対象の項目が多数個のとき、これを反復指定する。このとき、次の規則がある。

- 1) 変換の指示は括弧内に必ず次の3つの形式のいずれかにより書く。

(a) $n_1, n_2, \dots = l_1, m_1, m_2, \dots = l_2, \dots$

(b) $n_1 \sim n_2 = l_1, m_1 \sim m_2 = l_2, \dots$

(c) $n_1 \sim n_2, n_3, n_4 = l_1, m_1 \sim m_2, m_3 = l_2, \dots$

ここで(c)は(a), (b)の混合利用である。(a)はコードの数値変換に相当し, (b)は連続区間量をカテゴリー化する機能である。

なお、これらの指定において、等号の前後にそれぞれ変換前と変換後の数値を与える。したがってこの等号の省略は許されない。

- 2) OTHER による変換指定ができる。
- 3) 項目名、両括弧、等号、数値の前後には少なくとも1個以上の区切記号を置く。
- 4) 変換データの格納保存条件は次のとおりである。

- (a) RECODE 文の指定により MINERVA ジョブ内のそれ以後の分析は、変換データに対して適用される（永久変換機能）。なお変換後のデータファイルは変換指定部だけが変更された形として未変換の部分も含めた全項目に渡って、格納される。
- (b) しかし、この変換データはファイル機番10の作業ファイル上に格納されるので、初めに用意した MINERVA ファイルは依然として機番11の作業ファイル内に格納されている。
- (c) したがって、RECODE 実行後に再びもとの MINERVA ファイルのデータを利用する場合には、その分析の直前に INPUT 文を置いて機番11を指定する。
- (d) RECODE 文により一時的にデータ変換を行う場合には、RECODE 文の処理に続いて、OUTPUT 文を指定すればよい。これにより一時変換機能が適用できる。

〈例題 12〉

```

① [RECODE          QA (1, 3=5,   2, 8, 6=1),
    [RECODE          Q2.1 (1~4=1,   OTHER=2) /
② [RECODE          NATION (1=2 ; 2=1),
    [RECODE          RANK (11~19=1, 20~29=2, OTHER=9) /
    [FORMAT          (2 F 3. 0)
③ [ITEM            XX, YY /
    [RECODE          XX (1, 2=1, 4, 5=5) /
    [OUTPUT          13
  
```

①：項目 QA に対して、

{ カテゴリー 1, 3 を → 5 に変換
 { カテゴリー 2, 6, 8 を → 1 に変換

このとき、初めにコードが 1 であったデータをすべて 5 に変換し、2, 6, 8 であったものに新たに 1 とコードを付与する。

また項目 Q 2. 1 に対しては、

{ カテゴリー 1 から 4 までを → 1 に変換
 { その他のコード (OTHER) を → 2 に変換

ここで QA のとりうるカテゴリーが初め、かりに 1 から 8 までの正整数であるとき、上の変換に対して、コード 4, 5 は未変換のまま残される。

②：NATION はコード 1 を 2 に、2 であったものを 1 に、それぞれ変換する。すなわち、コードの入れ換え（置換機能）としても利用できる。

RANK に対しては、数値区間を区分して、カテゴリーを付与するという変換が行なわれる。

③：XX に変換をほどこした結果を 13 番 ファイル に格納する。このとき変換を行わない YY の

データも同時に格納する。

〔制限事項〕

- 1) 項目数は最大20項目まで許される。
- 2) () 内の変換指定の組数は1項目名当り最大10組である。
- 3) 1項目名に対し、変換指定部の括弧は1組だけ指定する。
- 4) 変換前の数値(等号の左側にくる数値)は正・負の整数値、変換後のカテゴリーは正の整数値だけが許される。
- 5) 変換指定を指示する等号(=)の省略はできない。

2.2.3 統計分析用の命令文

ASSOCIATION 文

〔項目間の関連係数の算出〕

〔機能〕

項目間の関連性の程度を測る係数を関連係数 (measure of association) と呼んでいるが、これには数多くのものがある。これらのうち表 4 に挙げた 12 種の係数が任意に指定、算出できる。

項目数を 3 項目以上指定すると、項目間関連係数行列を出力表示する。またこのとき、オプションにより、この行列をカードに出力できる。したがって、この行列を多次元尺度解析やクラスター分析など他のデータ解析の入力データとして利用することができる。項目数が 2 項目の場合には、2 元クロス表と全ての関連係数値を表示する。

〔指定方法〕

命令文の一般書式は次のとおりである。

▼

▼

ASSOCIATION

{項目名のリスト} /

[OPTION

COEFFICIENT = {関連係数のキーワード},
PUNCH /]

1) 標準的な使い方

次のように、命令語 ASSOCIATION に続いて指定文として算出したい関連係数と項目名とを指定する。この順序は任意であるが、各項目名の間に、少なくとも 1 つ以上の区切記号が必要である。

〈例題 13〉

- ▼

▼

① [ASSOCIATION

SUBQ, SCORE, EDUCTN Q3.2 /

② [ASSOC-MEASURE

STATUS, LIFE /

①は、4 つの項目名のすべての 2 項目名間の関連係数を算出し、これを関連係数行列として印刷表示する (したがって、 $4(4-1)/2 = 6$ 通り)。

②の場合には、STATUS, LIFE の 2 項目間の 2 元クロス表を印刷し、続いてすべての関連係数を表示する。(→詳しくは標準値とオプションの項を参照)。

2) オプションを指定するとき

- a) 特定の関連係数だけを算出したいとき。
- b) すべての関連係数を算出するとき。
- c) 関連係数行列をカード上に出力したいとき。

上のような場合にオプション機能が利用できる。

〈例題 14〉

①	[ASSOCIATION OPTION	SCOREA, SCOREB, AREA EDUC / COEFFICIENT = ALL, PUNCH /
②	[ASSOCIA OPTION	AGE, INCOME, TOTAL / COEFF = TAU, GOODMAN /
③	[ASSOC MEASURE OPTION	STUDY, CLASS, TEST, CAREER / PUNCH, COEF=KENDALL, YATES, SOMERS /

上の3つの例を説明する。

- ①：指定した4つの項目間の関連係数のすべての種類を算出し、同時にそれらをカード上に出
力する。
- ②：3つの項目について、Kendall の τ 係数と Goodman - Kruskal の λ 係数の2種類の関連
係数行列を算出する。
- ③：4つの項目を指定し、係数として、Kendall のコンティンジェンシー係数、Yates の補正
済み χ^2 値、Somers 係数を印刷表示し、同時にカードにこれらを出力する。

〔標準値とオプション〕

1) 標準値について

- a) 項目名が2個の場合には、その2項目間の2元クロス表と全係数を印刷表示する。ま
た各係数値の他に次の諸量を印刷する。
 - 相関係数 (CORRELATION) と共に、回帰係数を出力。
 - χ^2 値 (PEARSON) と共に、残差表 (2元クロスの期待度数と実現度数の差の適合表)、
標準化残差およびその分散の推定値、標準偏差で調整済みの残差などを印刷する。
- b) 項目数が3個以上の場合には、Cramer の係数と、Kendall-Stuart のコンティンジェン
シー係数が標準値として与えられる。

2) オプションについて

オプションには次の2つの機能がある。

- 関連係数の指定 (COEFFICIENT を指定する)
- 関連係数行列のカードへの出力指定 (PUNCH を指定する)

命令語 OPTION を与え、その指定文として COEFFICIENT または PUNCH を指定する。指定する係数の数と指定順は任意である。また、係数の指定と、カード出力の指定の順位も任意である (→ 前述の〈例題14〉を参照)。またいずれか一方だけ指定してもよい。

1) 関連係数の指定

- a) 表 4 の12通りの指示が可能である。これらの複数個を同時に指定してもよい。なお指定した係数行列はその順に従って印刷表示する。

指定方法は、COEF の後に1つ以上の区切り記号をおいて、表 4 のキーワードに従って指示すればよい。

たとえば次の4つはいずれも同じ指示内容をもつ。

16
▼

COEFFICIENT = PEARSON, CORR /
COEF = PEAR CORREL /
COEFCNT, PEARS, CORRN /
COEFCT (PEARSN, CORRTN) /

- b) 項目数の数が2個のとき、2元クロス表と、全ての係数を出力するので係数の指定オプションは無効である。

- c) Yates の補正済み χ^2 値と McNemer の χ^2 値は、それぞれ 2×2 次のクロス表に対してのみ有効であるので、これを指定した場合、関連係数行列の要素のうち 2×2 次のクロス表の条件をみたす項目の組み合わせに対してだけ数値を出力印刷し、行列の他の要素の数値は算出しない (記号 '*****' を表示する)。

かりに行列のすべての要素がこの条件を満たさぬとき、つまり指定した項目内に 2×2 次のクロス表が全くないときに上の2つの係数を指定するとその旨をメッセージとして印刷し、行列の印刷は行わない。

- d) キーワードとして ALL を指定すると全係数を算出する。この場合でも上記 b), c) の事項が適用される。

- e) ALL に加えて、さらに係数名を同時に指定すると、これらはどちらも算出される。

表4 関連係数の一覧表

キーワード	算出する関連係数名
<u>PEARSON</u>	Pearson の χ^2 値
<u>YATES</u>	Yates の補正済み χ^2 値 〔 2×2 次の2元クロス表に対してのみ有効〕
<u>MCNEMER</u>	McNemer の χ^2 値 〔 2×2 次の2元クロス表に対してのみ有効〕
<u>CONTINGENCY</u>	Pearson のコンティンジェンシ係数
<u>SQUARE</u>	Mean — square コンティンジェンシ係数
<div style="border: 1px solid black; padding: 2px;"><u>CRAMER</u></div>	{ Cramer の係数 Kendall — Stuart のコンティンジェンシ係数
<u>GOODMAN</u>	Goodman — Kruskal の λ 係数 〔 λ , λ_a , λ_b の3種類を算出する〕
<u>CORRELATION</u> (*)	相関係数および回帰係数のときだけ有効 〔回帰係数は2元クロス表のときだけ有効〕
<u>TAU</u> (*)	Kendall の τ 係数 (順位相関係数) 〔 τ_a , τ_b , τ_c を算出〕
<u>GAMMA</u> (*)	Goodman — Kruskal の γ 係数
<u>SOMERS</u> (*)	Somers の非対称係数 〔2種類算出する〕
<u>ALL</u>	上の全係数を算出する

〈注〉 i) (*)印の係数は、順位尺度データに対して意味がある。

ii) の2つの係数が標準値である。

2) 関連係数行列のカードへの出力指定

- a) 関連係数行列をカード上へ出力したいとき、指定文として PUNCH を指示する。このとき、項目名とそれに対応する関連係数行列の行ベクトルをカード上にパンチする。
このときのカード出力書式は、項目名と各要素を行ベクトルとして穿孔出力する（→サブルーチン CROPRI, CARD XX を参照）。
- b) 項目名が2個の場合には関連行列のカード出力機能は無効である。
- c) オプションとして PUNCH のみ指定すると、関連係数の標準値である Cramer の係数と Kendall-Stuart の係数とをカードに出力する。

〔制限事項〕

- 1) OPTION カードの継続枚数は2枚まで許される。また文の終了は必ずスラッシュ (/) で閉じる。
- 2) 前述のように指定した項目名が2個のとき PUNCH オプションは無効である。

〔注意事項〕

- 1) 関連係数のうち、順序尺度データに対して意味があるような係数を利用するとき、カテゴリーが順序データではあるがその順が乱れているようなとき RECODE 文により、事前にコード変換を行ってデータを順序尺度化しておくといよい。
- 2) カテゴリーが多値のとき必要に応じて、RECODE 文で適当に2値化した上で 2×2 次のクロス表に対してだけ有効である Yates, McNemer の係数を求めることが出来る。
- 3) 項目数は標準の50項目まで許される。しかし、現用の計算機の処理時間を考慮すると20~30項目以内に留めることが望ましい（項目数 m に対し、 $m(m-1)/2$ 通りの関連係数を算出なくてはならない）。

BREAK DOWN 文

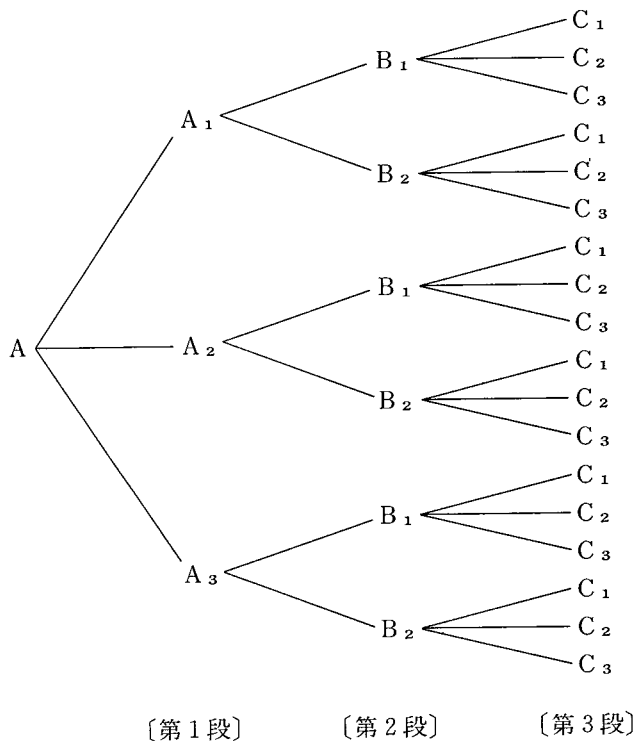
〔データの多段階別化〕

〔機能〕

ある項目名Xのデータを、別の項目A, B, C, D, ………のカテゴリーの組み合わせ別に多段階別化することを目的とした命令文である。いま一例として、3つの項目A, B, Cを考え、それぞれのカテゴリーが次のように与えられるとしよう。

$$A = \begin{cases} A_1 \\ A_2 \\ A_3 \end{cases}, \quad B = \begin{cases} B_1 \\ B_2 \end{cases}, \quad C = \begin{cases} C_1 \\ C_2 \\ C_3 \end{cases}$$

そして、ある項目Xのとりデータ値を上の3項目で次のように多段階別する。



この各段ごとに、その分類層内にあるXのデータについて次の初等統計量を算出する。

平均、標準偏差、変動係数、全データ数に対する層内データの構成比率(%), 最大値および最小値。

〔指定方法〕

命令語 BREAK のあとに指定文としてまず層別対象としたい項目名を書き続いて WITH を書く。そのあとに層別のキーとしたい項目名 (ブレイク・キー) のリストを、層化したい段の順に

指定する。

一般書式は次のとおりである。

▼	▼
<u>BREAK</u> DOWN	{層別対象項目名} <u>WITH</u>
	{ブレイク・キーにする項目名のリスト} /

〈例題 15〉

▼	▼
① [BREAK DOWN	X WITH A, B, C, D /
② [BREAK	C WITH B, D, A /

上の例で、①はXのデータを、A、(A, B)、(A, B, C)、(A, B, C, D)の順に第4段までブレイク・ダウンする。②は、Cの内容を、B、(B, D)、(B, D, A)の順に3段層別する。

指定にあたって、次の諸点に留意すること。

- 1) WITH の省略はできない。
- 2) WITH および各項目名間に1つ以上の区切記号をおくこと。
- 3) ブレイク・キーとする項目のデータは原則として、カテゴリー・データであること(連続量データの場合には、適当にコード化する必要がある)(→ RECODE 文参照)。

またカテゴリー・データであっても、その取りうる値の範囲が広い数値にわたる場合、カテゴリーの組み合わせ総数が膨大になる。この場合にもコード変換を行う必要がある(→制限事項の4)参照)。

〔制限事項〕

- 1) 層別対象項目名の指示は1個のみ許される。
- 2) 層別に使うブレイク・キーの項目名は最大9個まで許される。
- 3) 命令文カードの継続は許されない(1枚である)。
- 4) ブレイク・キーの項目名のカテゴリーの組み合わせの総数が3000通りをこえたとき、3000通りまでを印刷しそれ以上は表示しない。

〔機能〕

主に多変量の計量的データに適した手法であるが、これらのデータから多変量管理図を作成する。とくに次の場面で有効である。

1) 項目間に強い相関があるようなデータの管理状態（データが適切な範囲内に入っているか）の掌握。

2) したがって異常値の検出 (outlier detection) に利用できる。

3) とくに2項目（2変量）データのときには、散布図によりデータの分布状況を把握できる。

同時に図中の2次元特性管理楕円（95%管理限界）と1項目ごとの3シスマ限界線を利用して、データの傾向を吟味することができる。

4) 次の内容を印刷表示する。

- 多変量管理図 (Q-chart)
- サンプルごとのQ値^(*)
- 管理はずれとなった (Q-chart の管理線外に出た) データのサンプル番号のリスト

(*) ここでいうQ値とは、次の値をさす。

いま、第 i サンプルの観測ベクトルを \mathbf{x}_i ($p \times 1$ 次のベクトル ; $i = 1, 2, \dots, n$) とかくと、平均ベクトル $\bar{\mathbf{x}}$ 、分散・共分散行列 \mathbf{V} はそれぞれ次のようにかける。

$$\bar{\mathbf{x}} = \sum_{i=1}^n \mathbf{x}_i / n, \quad \mathbf{V} = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})' / n$$

このとき $Q = (\mathbf{x}_i - \bar{\mathbf{x}})' \mathbf{V}^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})$ である。

〔指定方法〕

一般書式は次のとおりである。

<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> ▼ ▼ </div> <div style="display: flex; justify-content: space-between;"> <u>QCHART</u> { 項目名のリスト } / </div>
--

項目名間には1つ以上の区切記号をおく。

項目名が2個のとき、管理線の入った散布図を出力表示するが、このとき、第1の項目が横軸に、第2の項目が縦軸に対応する。

〔機能〕

与えられた項目間の多重クロス表を作成するとともにそれらの表に対する対数線形モデルの適合度検定と、その最適なモデルの選出を行う。

とくにモデル選択の方式として次の2つがある。

- (1) 総当り法により、指定した項目についてのすべての対数線形モデルの計算と適合度の検証を行う。
- (2) 最適モデルの選択を逐次選択方式、組み合わせ方式により行う。

いずれの方式も、モデルの評価基準として、従来の有意性検定だけではなく赤池の情報量基準(AIC)を使うところに特色がある。とくに、方式(2)ではAICにもとづく、項別情報量という概念を導入し、これを利用して逐次選択を行う。この方式は独自に開発したもので従来の方式に比してきわめて客観的にモデルの選択ができる。また総当り法にくらべてその計算量を加速的に減らすことができる(→詳細は文献〔12〕,〔13〕)。

〔指定方法〕

▼	▼
MULTIWAY	{項目名のリスト} /
[OPTION	STEPWISE, COMBINATORIAL, PRINT = <u>n</u> /]

一般書式は上のとおりである。項目名の間は1つ以上の区切記号で仕切ればよい。

〈例題 17〉

▼	▼
MULTIWAY	SEX, AGE /
MULT-TABLES	STATUS, INCOME, LIFE /

〔標準値とオプション〕

オプションとして、次の2つの機能がある。

- モデル選択方式の指定
- 印刷出力内容の制御

なお、オプションを指定する場合には OPTION 文を用いる。指定がない場合には標準値が与えられる。

1) モデル選択方式の指定

これには次の3つの方式がある。

- 総当り方式 (ordinary method by all possible combinations)
- 逐次選択方式 (stepwise method)
- 組み合わせ方式 (combinatorial method)

このうち a) が標準値である。b), c) の場合、指定文として、次のいずれかを指定する。

- に対して、STEPWISE
- に対して、COMBINATORIAL

〈例題 18〉

	▼	▼	
①	MULTIWAY	SCORE, QA, Q3.1, Q9.2 /	
	OPTION	STEPWISE /	
②	MULTTAB	DRUG, TICS, HEART, DISEASE /	
	OPTION	COMBINATION /	

①は、逐次選択方式の指定例である。②は、組み合わせ方式の指定例である。

2) 印刷出力条件の指定

命令文 PRINT により、出力する印刷内容の条件をかえることができる。これは次の表のように5段階にわかれる。また、モデル選択方式の指定によって出力内容が多少異なる。これらは、表5-1, 5-2に一覧としてまとめている。表5-1は、PRINT オプションに対する出力説明である。仮りに PRINT のみ指定して、数値キー n を与えない場合には $n=0$ が与えられる。表5-2は、モデル選択方式の指定による出力内容の違いを示す表である。2つの表の組み合わせによって出力内容の組み合わせが異なることに注意を要する。

3) モデル選択方式の指定と、出力内容の指定とを同時に行うことができる。また指定の順位は任意である。

〈例題 19〉

- | | | |
|---|----------------------|--|
| ① | [MULTIWAY
OPTION | AGE SEX OCCUPTN /
PRINT = 3 / |
| ② | [MULT
OPTION | Q2.1 Q3 Q8, NATION /
PRINT = 1 / |
| ③ | [MULT-TAB
OPTIONS | SCORE, THERMAL, DRUG, BLOOD /
STEPWISE, PRINT = 2 / |
| ④ | [MULTAB
OPTIONS | Q1, Q3, Q2, DECIDE /
PRINT 4, COMBINAT / |

表 5 - 1

PRINT 指定キー	出力内容
$n = 0$ (標準値)	<ul style="list-style-type: none"> ● 多重クロス表の表示
$n = 1$	<ul style="list-style-type: none"> ● 多重クロス表の表示 ● モデルの構成一覧表 (いわゆるコンフィギュレーションの表) ● 収束判定情報 ● 各モデルに対する統計量 (Pearson のχ^2 値, 尤度比χ^2 値, χ^2 に対する裾の確率, AIC の値など)
$n = 2$	<p>$n = 1$ で出力表示の情報に加えて, 次の内容を表示する。</p> <ul style="list-style-type: none"> ● モデルの適合に利用した周辺度数の表 ● 周辺度数表の管理テーブルの表示
$n = 3$	<p>$n = 2$ の出力内容から多重クロスを除外したものに加えて, 次の情報を表示する。</p> <ul style="list-style-type: none"> ● 各モデルごとの期待度数を含んだ多重クロス表
$n = 4$	<p>$n = 3$ の内容に, 多重クロス表を加えたものを出力表示する。</p>

表 5 - 2

モデル選択の方式	出力内容
総当り方式	<ul style="list-style-type: none"> ● モデルのコンフィギュレーション一覧表 (AIC の値の順に, 上位25個のモデルだけ出力する) ● 全モデルに対する各種統計量の一覧表 (Pearson の χ^2 値, 尤度比 χ^2 値, χ^2 値に対する裾の確率, AIC の値) ● このとき, AIC の小さい値のモデルから順に表示する
逐次選択方式 } 組み合わせ方式 }	<ul style="list-style-type: none"> ● AIC 項別情報量の表 ● モデル選択の履歴情報一覧表 (選択過程で現われた各モデルのコンフィギュレーション表, Pearson の χ^2 値, 尤度比 χ^2 値, χ^2 値に対する裾の確率, AIC の値)

〔制限事項〕

- 1) 項目名は最大7個まで許される (7重クロス表まで)。
- 2) 命令文カードの継続は許さない (1枚である)。
- 3) 項目名を5項目以上指定した場合, 自動的に逐次選択方式に切りかわる (5重クロス表以上は逐次選択が標準値である)。これは, 5元以上では, 現用の計算機による総当り法の実行は不可能であるという物理的な理由による。
- 4) モデル選択の2つの方式を同時に指定は出来ない (改めて MULT, OPTION 文を指定することが必要である)。

〔注意事項〕

- 1) 対数線形モデル解析におけるモデル選択方式は, 過去に多くの研究者によって試みられてきたが, MINERVA に編入された方式は, われわれがまったく独自に開発したもので, AIC を利用してきわめて合理的にしかもほとんど自動的に最適モデルの候補を選ぶことができる。また坂元・赤池による多重クロス分析に相当するモデルの分析を行うことも可能である (→ 文献〔10〕, 〔13〕)。
- 2) われわれが日常取り扱うデータの量 (多くは数千サンプルから数万サンプル程度) を考慮すると, 5~6元程度の多重クロス表分析が限界であろう。これ以上データをブレイク・ダウンすると, クロス表の各セル内の度数が極端に粗になってモデル解析の安定性そのものが疑わしく

なる。こうした点を考慮して7元クロス表まで利用できれば十分であると判断して制限を付与した。しかし、こうした場合の、ロバストネスを確保する手当ての方法は今後の課題として残されている。

3) 次のような場合、RECODE 文によりコード変換を行った上で多重クロス分析を行うとよい。

- カテゴリーの数が多すぎるとき
- カテゴリーのつけかえをしたいとき
- 連続量データを区分して、適当にカテゴリー化したいとき

CROSS TABULATION 文 [クロス表の作成と項目間の比率検定]

[機能]

調査データのもっとも基本的な分析手法であるクロス表を作成する。この命令文を利用して次のデータ処理が可能である。

- 1) 任意の項目間の2元クロス表の作成 (I型と呼ぶ)。
- 2) ブレイク・ダウンの機能を持った2元クロス表の作成 (これをII型と名づける)。
- 3) 度数だけではなく、比率のクロス表も同時に作成する。
- 4) 比率クロス表の図表示の機能 (プロファイル表示機能)。
- 5) 比率の検定 (周辺分布に対する項目別比率, および項目の水準間での比率, それぞれの比率検定を行う)。

[指定方法]

- 1) 項目間の2元クロス表の作成 (I型の場合)

標準的な書式は次のとおりである。

▼	▼
<u>CROSS</u> TABLES	<u>HEAD</u> = { 表頭におく項目名のリスト } ,
	<u>SIDE</u> = { 表側におく項目名のリスト } /
または,	<u>SIDE</u> = { 表側におく項目名のリスト } ,
	<u>HEAD</u> = { 表頭におく項目名のリスト } /
[<u>OPTIONS</u>	<u>MARGINALTEST</u> または, <u>ENTRYTEST</u> ,]
	<u>PRINT</u> = <u>n</u> /

基本の命令語は CROSS である。指定文として、表頭におきたい項目名のリストを HEAD の後に続ける。同様に、表側におきたい項目名リストを SIDE の後に書く。

OPTION 文を利用する場合は、CROSS 文のブロックに引き続いて与える。これには比率検定を指定する機能 (MARGINALTEST, ENTRYTEST) と印刷内容の条件を指定する機能 (PRINT) がある。この2つの指定にとくに優先順位はない。

次のように指定すると、下の説明にあるような内容を入力する。なお、ここで表側を項目 A、表頭を項目 B のクロス表を、記号 A × B で表わすことにする。

〈例題 20〉

- | | | |
|---|---------------|---|
| ① | [CROSS TABLES | HEAD = A, B, C, SIDE = X, Y / |
| ② | [CROSS | SIDE = Q 1, Q 2.1 ; HEAD = SCORE / |
| ③ | [CROSS TAB | HEAD (RISK, PERCENT, CONDITIN),
SIDE (RACE, PLACE, CARE) / |

①：次の順にクロス表を出力する。

$X \times A, Y \times A, X \times B, Y \times B, X \times C, Y \times C$

すなわち表頭をまず固定し、表側について順に出力し、表側側の項目がつきると、表頭の次の項目に移り、同じことを反復する。

②：Q 1 \times SCORE, Q 2.1 \times SCORE の2表を出力する。

③：①、②の例で、‘=’、‘,’、‘_’は区切記号であるから、この例のように‘(’、‘)’により区切ってもよい。

〈例題 21〉

- | | | |
|---|--------|--|
| ① | [CROSS | HEAD = A, B : SIDE = A, B / |
| ② | [CROSS | SIDE = X, Y, Z , HEAD = Y, Z, X / |
| ③ | [CROSS | SIDE = (SEX, Q1, Q3),
HEAD = (Q 1, Q 4) / |

表側、表頭の項目名リストが全くの同一項目である場合や、一部に同じ項目名が含まれている場合には同一項目名の組み合わせに対して出力制限が働く。

①：A \times A, A \times B, B \times A, B \times B の4通りが考えられるが、このうち A \times B のみ印刷表示する。

②：①と同様の処理が行われ、9通りの組み合わせのうち X \times Y, X \times Z, Y \times Z のみ表示する (X \times Y に対する Y \times X, X \times Z に対する Z \times X などは重複するので出力しない)。

③：この例では Q 1 \times Q 1 は印刷しない。

2) ブレイク・ダウンを伴うクロス表の作成 (Ⅱ 型の場合)

Ⅱ 型の標準書式は次のとおりである。

<div style="text-align: left;">▼</div> <u>CROSS</u> TABLES	<div style="text-align: left;">▼</div> <u>MULTIPLE</u> = {項目名のリスト} <u>BY</u> { (項目名の対) のリスト } /
<div style="text-align: left;">[</div> <u>OPTIONS</u>	<div style="text-align: left;">]</div> <u>MARGINALTEST</u> または <u>ENTRYTEST</u> , <u>PRINT</u> = <u>n</u> /

命令語はⅠ型と同じ CROSS である。指定文としてまず MULT によりブレイク・キーに使う項目名のリストを書く。これにより指定した項目をキーとして、BY の後にクロス表の表側、表頭にしたい項目の対を括弧によりくくって指定する。

この項目の対のリストを順次指定する。

〈例題 22〉

<div style="text-align: left;">▼</div> CROSS	<div style="text-align: left;">▼</div> MULT = X, Y BY (A, B), (C, D) /
--	--

上の例では、

- a) X をブレイク・キーとする $A \times B$ の 2 元クロス表群
 - b) X をブレイク・キーとする $C \times D$ の 2 元クロス表群
 - c) Y をブレイク・キーとする $A \times B$ の 2 元クロス表群
 - d) Y をブレイク・キーとする $C \times D$ の 2 元クロス表群
- を順に印刷する。a) についてさらに詳しく説明しよう。

ここで、X は X_1, X_2, \dots, X_r と、 r 個のカテゴリーをとるものとする。このとき、次の $(r+2)$ 個のクロス表を出力する。

- i) まず $A \times B$ のクロス表を印刷する。
 - ii) 次に、 $X = X_1$ と固定して (この条件を満たすデータについて)、 $A \times B$ のクロス表を印刷する。以下 $X = X_2$ と固定して $A \times B$ を、……と r 個の $A \times B$ に関するクロス表を印刷する。
 - iii) そして、最後に $X \times B$ のクロス表を印刷する。
- b) ~ d) に対しても同様の方式でクロス表を出力する。

〔標準値とオプション〕

I 型、II 型に共通のオプションとして、次の 2 つの機能がある (→標準書式を参照)。

- 比率検定の指定
- 印刷出力内容の条件指定

1) 比率検定の指定

これには次の 2 種類がある。

- MARGINALTEST
- ENTRYTEST (*)

(*)ENTRYTEST の部分の機能はプログラムとしては組み込まれているが試験中の部分であるので、VERSION-01 では標準仕様に入っていない。

a) は比率クロス表によりその表頭の項目の周辺相対度数に対して、表側の各行ベクトルの比率の有意性をテストする (2 シグマ限界を用いる)。

〈例題 23〉

▼ ▼
CROSS HEAD = SEX, SIDE = AGE /
OPTION MARGINAL /

この例で性別 (SEX) がコード 1 (男) および 2 (女), 年齢 (AGE) がコード 1 ~ 5 と与えられるとき、次のクロス表が出力される (2 シグマ限界を用いる)。

表 6 クロス表の出力形式 (I 型)

(度数クロス表)

SEX \ AGE	1	2	計
1	n_{11}	n_{12}	n_{1+}
2	n_{21}	n_{22}	n_{2+}
3	\vdots	\vdots	\vdots
4	\vdots	\vdots	\vdots
5	n_{51}	n_{52}	n_{5+}
計	n_{+1}	n_{+2}	n_{++}

(比率クロス表)

SEX \ AGE	1	2	比率計
1	p_{11}	p_{12}	$p_{1+} (100)$
2	p_{21}	p_{22}	$p_{2+} (100)$
3	\vdots	\vdots	\vdots
4	\vdots	\vdots	\vdots
5	p_{51}	p_{52}	$p_{5+} (100)$
比率計	p_{+1}	p_{+2}	$p_{++} (100)$

比率クロス表は必ず行和が 100% となるように算出する。そこで、上の例では、性別の周辺相対度数（比率）（ p_{+1} , p_{+2} ）に対して、AGE の各カテゴリーに対する比率（ p_{i1} , p_{i2} ）（ $i = 1, 2, \dots, 5$ ）が有意であるか否かを検定する。すなわち、年齢別にみた男女比率が全サンプルの男女比率に比して有意か否かを検定する。仮に、性別からみた年齢比率間の検定を行う場合には、表側、表頭の指定を入れかえればよい。

次に b) は、特定の 2 つの行ベクトル間の比率検定だけを行う。たとえば上の表で AGE のカテゴリー 1 と 3 との男女比率間に有意性があるか否かを検定するなどが一つの例である。

2) 印刷出力内容の条件指定

OPTION 文の指定文として、PRINT により印刷内容を指定する。これには次の 3 種類がある。なお指定のキーは下表のように正整数で与える。

表 7

PRINT 指定キー	出力内容
$n = 1$	<ul style="list-style-type: none"> ● 1 ページ当り 1 表のクロス表を印刷する。このとき度数と比率を同一クロス表内に表示する。 ● 同時に比率プロフィルの図を表示する。
$n = 2$	<ul style="list-style-type: none"> ● 連続して、クロス表を印刷する。 ● 度数クロスと比率クロスは分離して出力する（左側に度数、右側に比率のクロス表を印刷する）（→ 表 6 参照）
<div style="border: 1px solid black; padding: 2px;">$n = 0$</div> (標準値)	$n = 1, 2$ の両者を印刷する。

印刷内容の標準値は $n = 0$ のときである。したがって PRINT の指定を行わぬときには、これが与えられる。また数値キー n を与えずに PRINT とだけ与えた場合にも標準値が与えられる。

3) オプション機能の同時指定

比率検定、印刷指示の 2 つの機能の同時指定が許される。またその指定順に優先度はない。ただし、比率検定の 2 つの機能（MARG, ENTR）の同時指定は許さない。

〈例題 24〉

①	[CROSS OPTIONS	HEAD = QA, LIFE, SIDE = AGE / MARGINAL, PRINT 2 /
②	[CROSS OPTIONS	SIDE = HEIGHT, BLOOD, HEAD = WEIGHT / PRINT = 1, MARG /
③	[CROSS OPTIONS	MULTIPLF = (AREA, CLASS) BY / (INCOME, OCCUPY), (SEX, CAREER) / MARGIN PRINT = 2 /

上の例のように混合指定が許される。

〔制限事項〕

1) 項目名の数の指定制限は次のとおりである。

I 型: HEAD, SIDE とも約15~20項目^(*)である。

II 型: プレイク・キーとする項目が約15~20項目^(*)まで許される。

(*)ここで項目名の数が厳密に確定できない理由は各項目のとりうるカテゴリー数によって計算のための作業領域の大きさが左右されるためである。

2) 命令文カードの継続に制限はないが、実際には1) との関連から若干の制約を受ける。

〔注意事項〕

1) クロス表の表頭のカテゴリー数が10個をこえた場合、PRINT オプションは $n=1$ に固定される。

2) カテゴリー値が不連続的に変化する場合には、データ中に存在するカテゴリー値だけがクロス表の中に現われる。

3) カテゴリー値が大きい数値のとき、連続的に変化させたいときなどは、RECODE 文でコード変換した上でクロス表を作成すればよい。

BMDP 文**[BMDP の連結処理]****[機能]**

典型的な統計パッケージとして知られている BMDP (1978年1月版) のうち、初等的な統計解析に関連した 8 手法を、MINERVA ジョブ内で連結利用できる。これの連結利用を指定する命令文が BMDP である。8 つの手法とは、BMDP の課題名のうち次のものである。

表8 BMDP の一覧表

課題名	処 理 内 容 の 概 略
P 1 D	データの初等統計量の算出。
P 2 D	データの記述。度数表、ヒストグラムなどを出力表示する。
P 4 D	データの数値検証。カラム・チェック、度数チェックなど。
P 5 D	1 変量データに関する統計分析 (ヒストグラム、正規プロットなど)。
P 6 D	2 変量データに関する統計分析 (散布図など)。
P 1 F	2 元クロス表の作成。関連係数の算出。
P 2 F	2 元クロス表の検定、適合度検定など。
P 3 F	多元クロス表の分析、対数線形モデル解析。

MINERVA ジョブにより適当に加工あるいは検証したデータを他の統計パッケージに連結利用することを実験的に試みるために付け加えた機能である。実際には、上の表のかんりの部分が MINERVA により処理可能である。

[指定方法]

命令語 BMDP に続いて、指定文として BMDP のうち 使用したい課題名 (表 8) を指示する。次に BMDP 用のコントロール・カード (コマンド文) をそのまま挿入し、最後に FINISH 文で BMDP 文の効力の範囲の終了を指定する。一般書式は次のとおりである。

▼	▼
<u>BMDP</u>	{ BMDP の課題名 }
<u>BMDP のコントロール・カードの指定</u>	
<u>FINISH</u>	/

〔制限事項〕

- 1) BMDP は MINERVA に常時、標準装備されていない。利用する場合には、BMDP の原ファイルから、上の 8 手法の実行に必要な部分を抜き出し編集した上 MINERVA に付加することが必要である。
- 2) BMDP へのデータの受け渡しは MINERVA ファイルまたは MINERVA ジョブ内で生成した一時作業ファイル上のデータを介して行う。
- 3) BMDP のコントロール・カードの作成法については、BMDP ユーザー・マニュアルを参照すること（→ マニュアル・リスト〔5〕）。

3. MINERVA の構成

MINERVA は、プログラム・リストにみるように (→3.6 節), ごく標準的な FORTRAN 文法により書かれている。最近の統計パッケージには、PL/I, BASIC, などを使ったものもみられるが、MINERVA 作業班が日常使い慣れてきた FORTRAN 文法だけを利用することにした。データの入・出力部の扱いを容易かつ迅速にするためには PL/I やアセンブラー言語を用いたほうが有利であると思われるが、他機種への移植の可能性、互換性などを考慮すると、やはり FORTRAN 文法に頼らざるをえない。

プログラム構造は、いわゆる調整可変型配列方式 (adjustable arrays) を採用しているので、プログラムが、若干冗長的にならざるをえないが、利用者が計算機の規模にあわせて自由に作業領域の大きさを変えられるという利点もある。このため主プログラムは 1 個とし、その他のプログラムはすべて副プログラムとして、その下位に位置づけられている。この点は SPSS を初め MINITAB-II など多くのパッケージと同じ方式を採用している。

コモン・ブロック文の徹底的導入、サブルーチンの引数省略など、多くの改良が考えられるが、今のように開発途上にあつては、多少冗長的であっても、点検・検査が容易であるような方式を優先してコーディングを図った。

また、実際のジョブ処理内容は、複雑な数値計算処理よりも、もっぱら、データの読み出し、加工、出力といった I/O 処理に仕事が集中するのであるが、FORTRAN 文法だけでは、この点の効率向上には限界がある。他の言語の活用が有利であるが、前述の理由から、この方式を避けた。このため極力データをバイナリー型処理、ランダム・アクセス処理により扱うことで、I/O 処理の効率化を図るように努めた (ただし、標準版はランダム・アクセスの機能を持たない)。

3. 1 MINERVA プログラムの構成図

MINERVA-V01 は、基本的には 1 つの主プログラムの下に多数個の副プログラムを連結するという構成をとっている。現在、副プログラムには 93 のサブルーチンがある。その他、ユーティリティー・ルーチンとして文字の転送・位置変更を行う MOVEC (Move-character) を利用している。これらの副プログラム群の間の関連を流れ図として示したものが図 2 である。図中の箱の数は主プログラム (MAIN) を含めて、119 ある。上記のサブルーチン数と一致しないのは、重複利用のサブルーチンがあるためである。なお、各サブルーチンの名称のリストは、プログラム・リストの先頭ページにも付与してある。

図 2 の(a)にみるように、副プログラムは 3 つに大別される。それぞれ各ブロックは、

- (I) 主として、命令文の解釈やジョブ制御に関連したルーチン群
- (II) 主に、データ加工や検証に関連するルーチン群
- (III) 統計解析に関連したルーチン群

と対応している。共通利用のサブルーチンもあるので、必ずしも明確に仕切ることが出来ないが、ほぼ上のようになっている。

〔注〕現在、MINERVA を稼働させている機種は、M-180 であり、そのシステムは VOS-3 (Version 05-00) である。

3.2 MINERVA ジョブの処理の概略

3.2.1 基本的な処理の流れ

MINERVA ジョブの処理の流れは、およそ図 3 の流れ図に示すとうりである。

今後の改編作業により、変更がおこることが考えられるが、基本的には、

- (1) 命令文の解説
- (2) 入・出力データの取り扱い方式の確認
- (3) 具体的な分析・処理の実行

の繰り返しである。一旦、ある部分でエラーが発生した場合でも、ジョブは終了せず、次の分析処理に移行するというエラー回避ジョブ続行方式をとっている点が特徴の 1 つである。これにより、ジョブ・ストリームが途切れることなく遂行できる。

プログラムの機能は、基本的にはデータの検証・確認、モデル分析などにあり、副次的にデータの選出、加工を行うことが出来るが、利用のしかたによっては、これらのサービス機能が有効であるという点も特徴の 1 つである。

その他の処理の流れの特徴は、流れ図を追ってみることで明らかであろう。

3.2.2 いくつかの分析処理の流れ図

すべての分析課題の処理の流れを詳細に説明することはできないので、ここに、いくつかの例の流れ図の概略をあげておこう (→図 4 参照)。

(1) 関連係数の算出 [ASSOCIATION] の処理 (例 1)

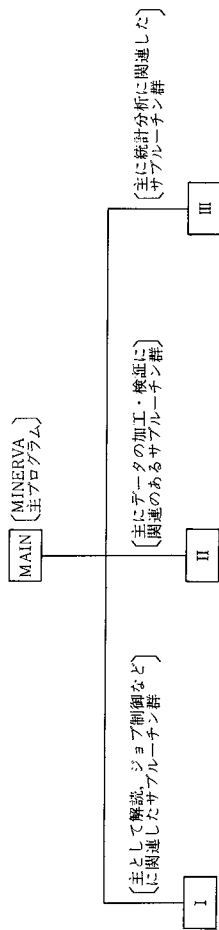
例 1 の 1.1 が、この課題のルーチン内に入ってから主な流れである。この中の、主ルーチン (AMAIN) をやや詳しく表わしたものが例 1 の 1.2 である。いずれも流れ図を追うことでその概要は理解できよう。

(2) コード変換ルーチン [RECODE] の処理 (例 2)

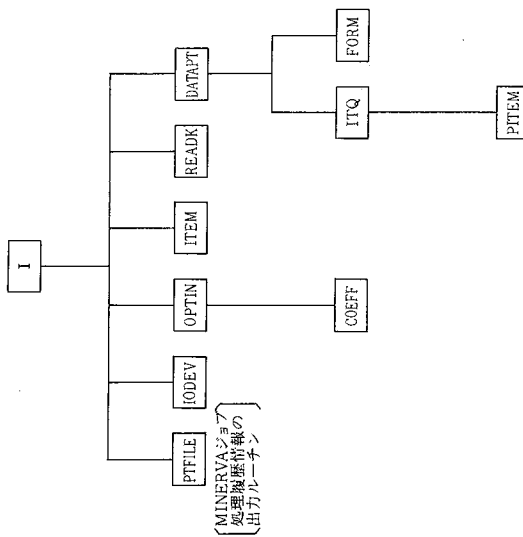
例 2 が、その流れ図である。RECODE の特色の 1 つに、データの一時変換および永久変換機能がある。これは、流れ図にみるように加工処理済みのデータの出力ファイルの指定がとくにないときには、RECODE を指定した以後の処理は、すべて変換加工済みデータに対して行う永久変換と、出力先ファイルを任意に指定すると一時的に、そのファイル上に加工データを格納して、それに対して処理を行う、いわゆる一時変換とが可能である。

図 2 MINERVA の構成 (その 1)

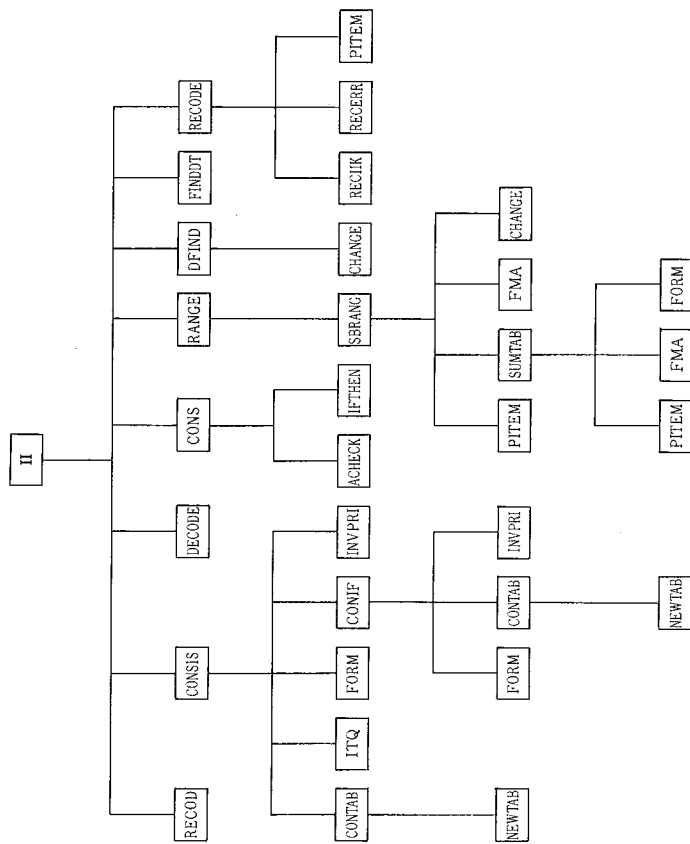
(a) MINERVA の全体の構成



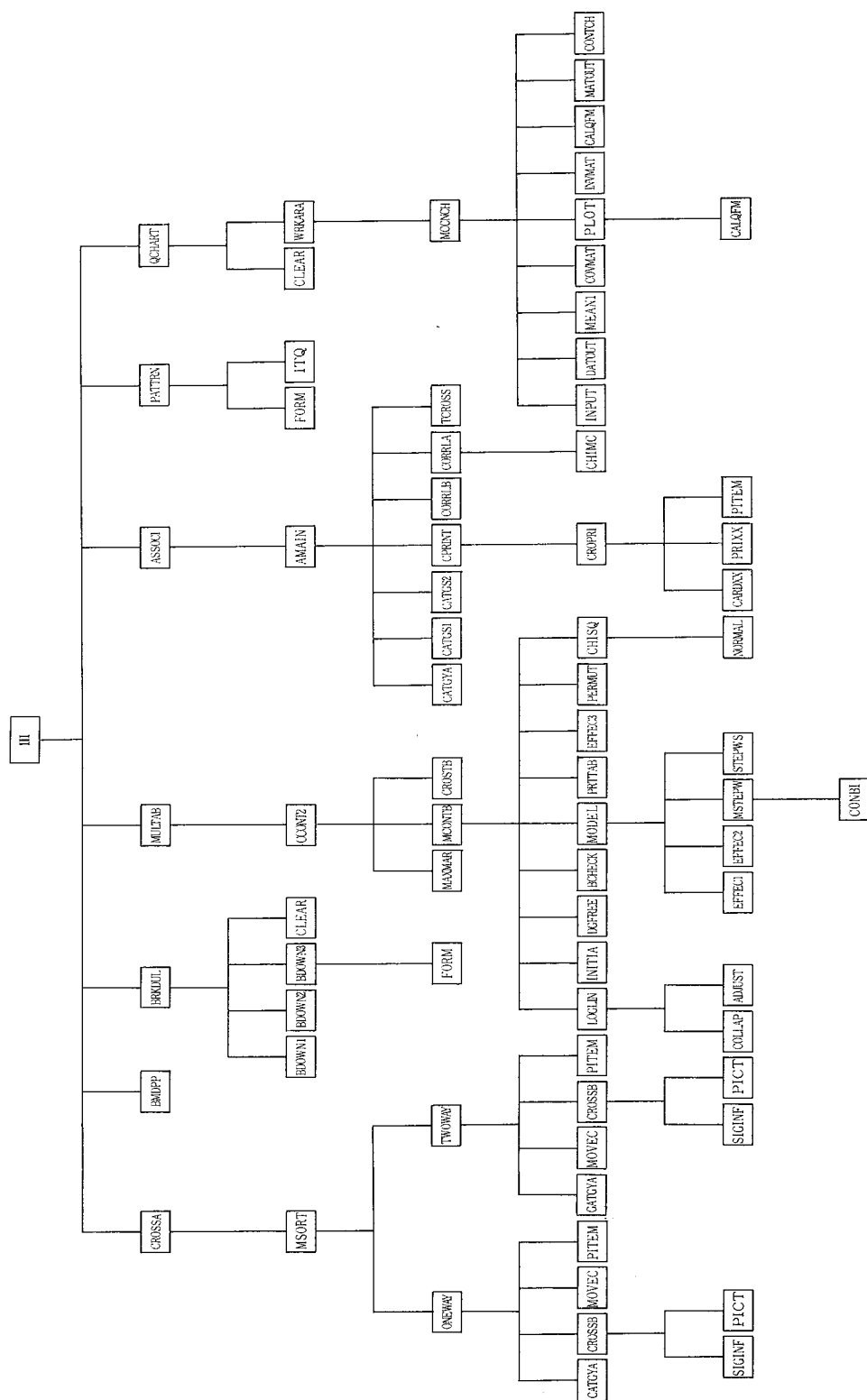
(b) ブロック I の構成



(c) ブロック II の構成

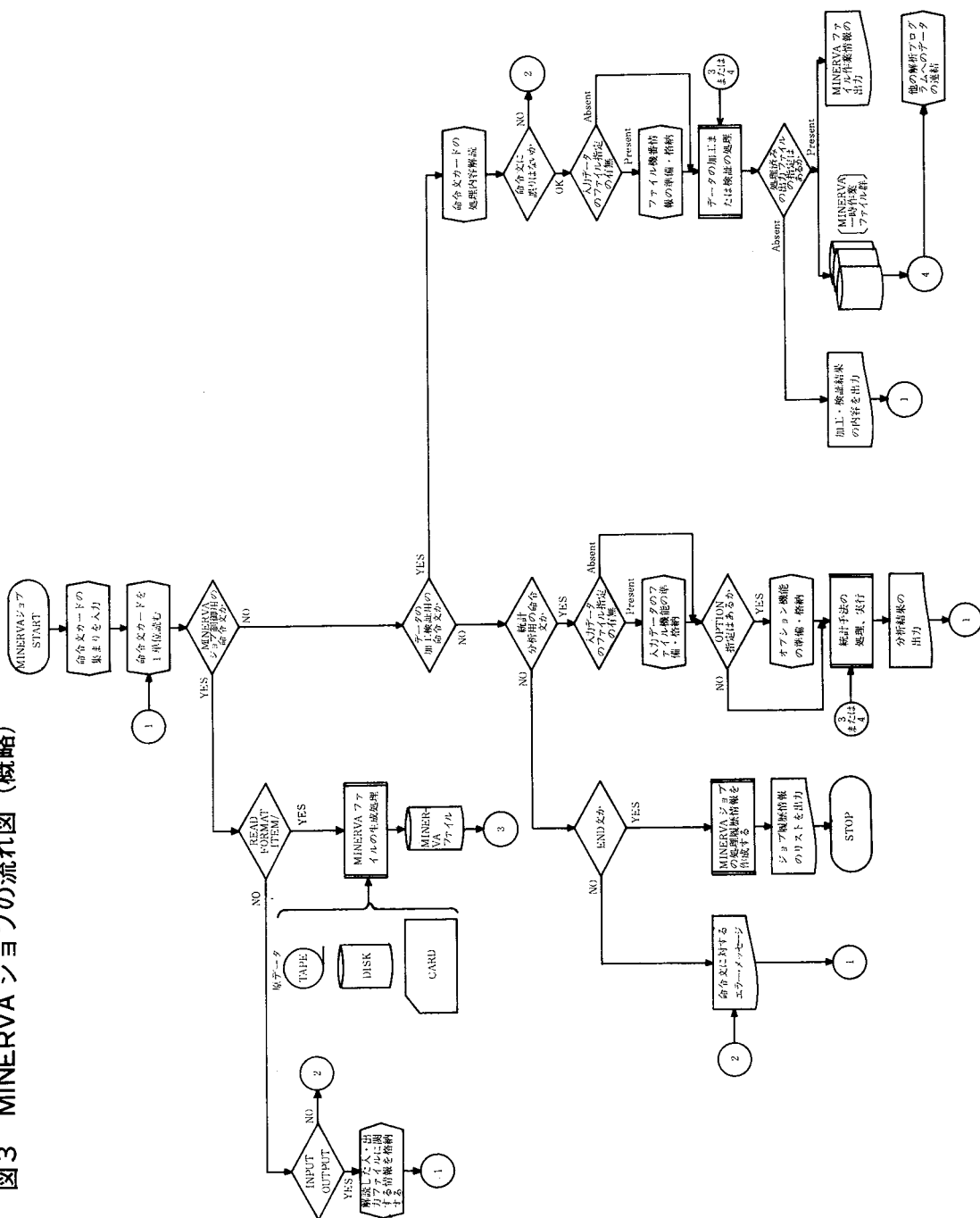


(d) ブロックⅢの構成



＜注＞ MOVECはM-180ユーティリティ・ルーチンを利用している。

図3 MINERVA ジョブの流れ図 (概略)



例1 ASSOCIATION (関連係数の算出) ルーチンの流れ図

1. 1 ASSOCIATIONの全体流れ図

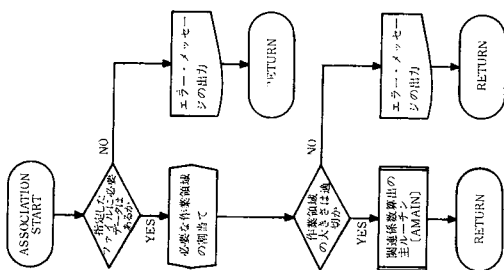
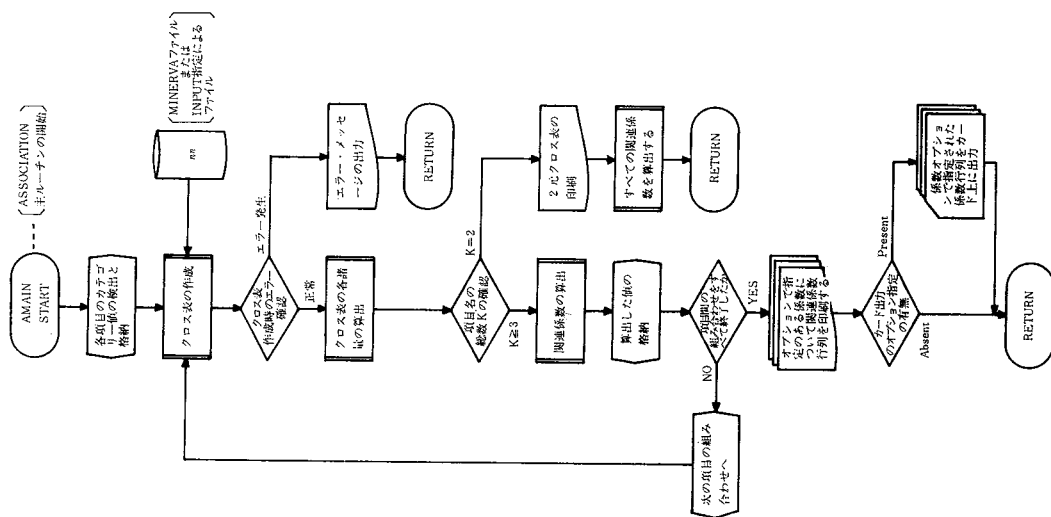
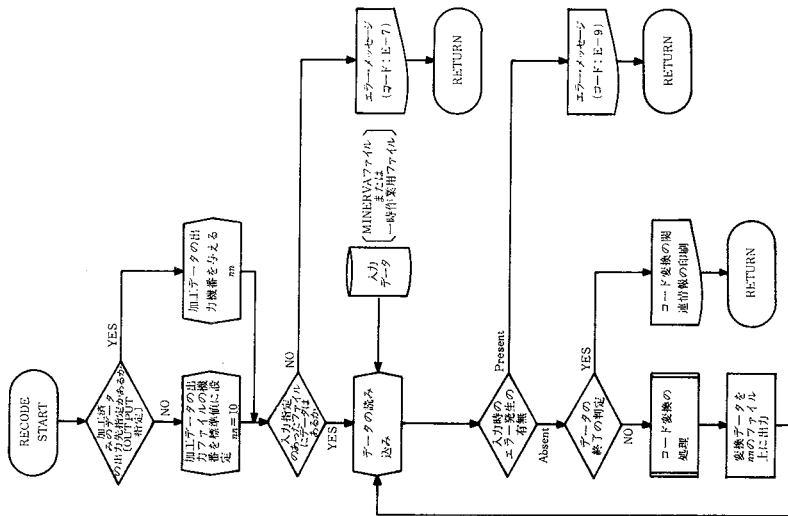


図4 いくつかの分析処理の流れ図

1. 2 ASSOCIATIONの解析部分 (AMAIN) の流れ図



例2 RECODE (コード交換) ルーチンの流れ図



3.2.3 その他の事項

(1) MINERVA ジョブの処理内容の表示

数多くの分析課題を命令文カードで与えると、出力も多くなり、処理の流れを追跡することが困難となる。

MINERVA では、処理の開始時と終了時に次の情報を表示する（→図 5 参照）。

- a) 分析結果の出力の最初に、命令文カード・ブロックを印刷表示する。例 1 は、後出の例題集の中の〔例題 1〕に対する出力例である。
- b) ジョブの終了時に、最終情報として、処理履歴情報のリストを表示する。たとえば、後出の例題 1, 2 の場合を例にとると、それぞれ例 2, 例 3 のような内容が表示される。

〔例 2 の説明〕

MINERVA ファイル（機番 11）に 8145 サンプル登録されている。これに CONS, CONS, …, FIND の処理を行った。機番 12 のファイルに 255 サンプルがあって、これに BREAKDOWN の処理を行った。また、機番 13 のファイルに 2527 サンプル分のデータがあり、これに CROSS を適用した、ことなどが読みとれる。

〔例 3 の説明〕

この場合も同様に、各作業ファイル内のサンプル数や、処理の内容を参照することができる。

〔注〕同一機番ファイルを連続的に反復利用した場合には、このリストの情報は最終処理課題に対してのみ有効である。すなわち、次々と、前の処理内容を更新するので、履歴が残らない。

(2) 他の解析プログラムへのデータ連結機能

MINERVA ジョブでは、データの加工・検証用の命令文については、一部を除いて、その処理ずみのデータを作業ファイルに出力できることは、既に指摘してきた。

したがって、この作業ファイルを介して、加工処理済みデータを、他の解析プログラムで利用することが可能である（いわゆる、1つのジョブ内で複数個のジョブを処理するマルチ・ジョブ方式のジョブ・コントロール・カードを用意しておけばよい。そして、MINERVA の実行後に、利用者の解析プログラムを実行すればよい）。ただし、現在では MINERVA ジョブではすべてバイナリー型データとして扱っているので、解析プログラムのデータ入力部に若干の変更が必要である。これを避けて、自由に、加工済みデータの受け渡しが可能となるようにデータ出力の機能を改編中である（→次のオーバーレイ構造の項の例を参照）。

図5 MINERVA ジョブの処理内容の表示

〔例1〕

INPUT INFORMATION BY CARD IMAGE

```

READ      TAPE
FORMAT    (F2,0,F4,0,BF2,0,4X,F2,0,2X,F2,0,30X,3F2,0,10X,F2,0)
ITEM      NO,SURVY,NO,SAMPL,NO,CARD,AREA,PREFEC,CITYSCL,SEX,AGE1,
          AGE2,UCCUPATN,INCOME,CAREER,TIMES,NDANS,STGNT,TLSCORE /

CONSIST-CHECK IF(AGE2 LE 2)+THEN(AGE2 GT 2 OR INCOME GE 7) /
CONSIST      IF(SEX EQ 2 AND AGE2 GE 6)+THEN(CAREER LE 3) /
CONSIST TEST IF(AGE2 GE 5)+THEN(OCCUPATN NE 7) /
CONSISTENCY  IF(AGE2 GT 2 OR (INCOME LE 7 OR INCOME GE 10)) /

FIND DATA   1,4,9,14,20,32,38,59,70,84,108,1,1580,2123,2917,3823,3539 /
FIND DATA   245,445,2062,2740,4473,5730,6869,7170,7214 /

RANGE       AGE2 1 /
OUTPUT DEVICE 12

CONSISTENCY IF((NO,SURVY GE 2 AND NO,SURVY LT 6) AND SEX EQ 1) /
OUTPUT DEVICE 13
INPUT DEVICE 13
CROSS TABLES SIDE=AGE2,AREA HEAD=INCOME,CAREER,AREA /

INPUT DEVICE 12
BREAK DOWN INCOME WITH NO,SURVY,SEX,AGE2 /

END

```

〔例2〕

JOB EXECUTING INFORMATION

SPECIFIED NUMBER OF FILE	SIZE OF DATA IN ITS FILE	HISTORY OF PROCEDURES
11	8149	CONS CONS CONS CONS RANG CONS FIND
12	255	BREA
13	2527	CROS

--- LIST ---

```

RANG = RANGE CHECK
CONS = CONSISTENCY CHECK
FIND = FIND DATA
PATT = PATTERN CHECK
MULT = MULTIWAY TABLES
QCHA = Q-CHART
BREA = BREAK DOWN
RECO = RECODE
ASSO = ASSOCIATION
CROS = CROSS TABLES
BMOP = P10, P20, P40, P50, P60, P1F, P2F, P3F
BLANK = NO EXECUTING PROCEDURE

```

〔例3〕

JOB EXECUTING INFORMATION

SPECIFIED NUMBER OF FILE	SIZE OF DATA IN ITS FILE	HISTORY OF PROCEDURES
10	3354	RANG
11	3354	RECO
12	3311	RANG RANG RANG
13	1869	MULT CROS
14	3109	MULT
17	3080	MULT

--- LIST ---

```

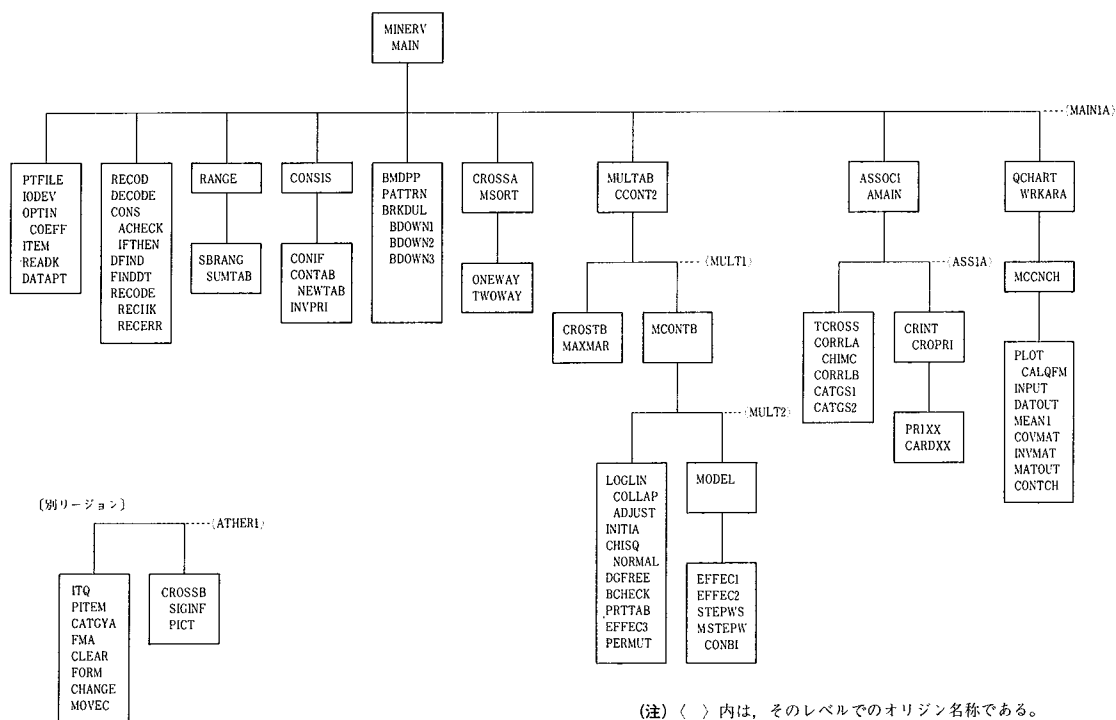
RANG = RANGE CHECK
CONS = CONSISTENCY CHECK
FIND = FIND DATA
PATT = PATTERN CHECK
MULT = MULTIWAY TABLES
QCHA = Q-CHART
BREA = BREAK DOWN
RECO = RECODE
ASSO = ASSOCIATION
CROS = CROSS TABLES
BMOP = P10, P20, P40, P50, P60, P1F, P2F, P3F
BLANK = NO EXECUTING PROCEDURE

```

(3) MINERVA におけるオーバーレイ構造について

MINERVA ジョブの実行時に、記憶容量を節約するための方策として、オーバーレイ構造の利用が考えられる。オリジン、リージョン等の構成の仕方はいろいろ考えられるが、現在は、下の図のように設定してある。それぞれ〈 〉内に示した名称が、その水準におけるオリジン名称である。実験によるとオーバーレイ構造を導入することにより、記憶容量を約30%節約できる。

図6



(4) MINERVA ジョブ実行のための JCL の与え方

実際にジョブを実行するためには、システムを制御するための JCL (Job Control Language) カードが必要である。多くの場合、JCL カードの指定は、不慣れた利用者にとってはわずらわしいものであるから、MINERVA では、そのほとんどのものをカタログ化して繁雑さをさけている。ここでは、標準的な指定法と、MINERVA ジョブに別の解析プログラムを連結して一括処理を行うマルチ・タスクの例との2つをあげておこう。

標準的な JCL の与え方

```
//DHSUMIXB JOB ,MINERVA01,  
// EXEC DHSUMI  
//GO,SYSLN DD #  
-----  
MINERVA COMMAND CARD BLOCK  
-----  
//
```

[MINERVA 命令文カード・ブロック]

(*) 出力ファイルとして機番12を指定

マルチ・タスクの例

```
//DHSUMIXB JOB ,MINERVA01,  
// EXEC DHSUMI  
//GO,SYSLN DD #  
-----  
MINERVA COMMAND CARD BLOCK  
-----  
READ TAPE  
-----  
OUTPUT DEVICE 12  
-----  
END  
-----  
//  
//FLG EXEC FLG  
-----  
PROGRAM PROCRUSTES  
-----  
READ(12) KK,(DATA(1),1=1,N)  
-----  
END  
-----  
//GO,FT12F001 DD USN=8B,SPACE=(CYL=(8*2))+UNITS=015K,015P=0LD,  
// DCB=(RECFM=VBS,LRECL=13026,BLKSIZE=13030)  
//GO,SYSLN DD #  
//
```

[別の解析プログラムの実行]

(*) MINERVA で出力したファイル上のデータを
ここで再利用する場合の例である

ここでは、初めに MINERVA ジョブを実行し、その中で加工処理済みデータを機番12のファイルに出力し、これを、次の別の解析プログラム (PROCRUSTES) 内で呼び出して再利用する、という例になっている。このように、MINERVA で処理をすませたデータをファイルを介して、他の解析プログラムに引き渡すことが出来る。

(5) MINERVA 作業ファイルについて

MINERVA では、データの選出や、加工済みデータの格納などの処理を行う場合には、一時作業ファイル（磁気ディスク）を多用する。

作業ファイルを初め、入・出力機番などは、現在は、次のように割りあててある。とくに、機番 09, 10, 11 の扱いについては注意を要する。

表 9 MINERVA 作業機器・機番一覧
(ファイル定義名一覧表)

ファイル定義名	機 能
FT 03 F 001 } FT 04 F 001 }	作業用予備ファイル
FT 05	カード入力機番
FT 06	ラインプリンター機番
FT 07	カード出力機番
FT 08 F 001	作業用予備ファイル
FT 09 F 001	データ入力標準機番（磁気テープ）
FT 10 F 001	データ入・出力機番（RECODE 出力標準機番）
FT 11 F 001	MINERVA ファイル
FT nn F 001	nn = 12 ~ 20 として、一時作業用ファイル

(6) 今後の改編作業について

現用の MINERVA には、不十分な点が数多く散見される。これらは、勿論、一挙に改善することは困難ではあるが、今後の改良事項として次のようなことを考えている。

- 命令文解読ルーチンの充実。
- エラー・メッセージ管理ルーチンの強化。
エラールーチンを独立させて、ジョブの流れを円滑にする。
- 作業領域の縮元をはかるための改良。
- 出来れば会話型に移行できるような命令文方式に変更すること。
- データのサービス機能の充実を図ること。

この点は、カテゴリーカル・データの処理分析では必須の機能である。一般のパッケージはこの部分の機能が不足しているので、これに特色ある方式を取り入れる。

その他の細かい点については、本文中でも若干指摘のあるように、逐次改良を進めている。

3. 3 出力メッセージ一覧表

[エラー・メッセージ]

コード	エラー・メッセージ	意 味	関連のあるサブルーチン名
E-1	WORKING AREA OVER.	MINERVA ジョブの全作業領域の指定時にオーバーフローが発生した。	MINERV
E-2	INPUT COMMAND CARD IS INVALID.	命令語とキーが一致しない。 解読が不可能である。	MAIN
E-3	INPUT OR OUTPUT DEVICE CODE WAS MISUSED, THUS CHECK AND SPECIFY AGAIN.	入・出力の一時作業用ファイルの指定に誤りがある。	IODEV
E-4	THIS COMMAND WAS MISUSED.	指定文の解読が不可能である。	OPTIN, BMDPP
E-5	COMMAND WAS MISUSED. READ TAPE CARD DISK <i>nn</i>	READ文の書式に誤りがある。 磁気テープ指定に誤りがある。 データ・カード入力指定に誤りがある。 磁気ディスク指定(機番 nn)に誤りがある。	READK, DATAPT
E-6	THIS ITEM UNDEFIND. ERROR ITEM={A}	ITEM 文で定義した項目名リストに項目名Aが一致しない。	DECODE, RECOD
E-7	FILE NUMBER IS { <i>nn</i> }, THIS FILE DOES NOT INCLUDE DATA.	指定した入力機番ファイル nn にデータがない。	ASSOCI, RANGE, CONSI, PATTERN, MULTAB, QCHART, BRKDUL, RECODE
E-8	ERROR FOR THE EXECUTION ABOUT WRITE TO DISK.	作業ファイルにデータを出力中に誤りが発生した。	RANGE
E-9	IT IS IMPOSSIBLE TO FIND DATA SPECIFIED BY COMMAND.	ファイル上のデータ入力中に異常が発生した。	RANGE, CONSI, CONIF, PATRN, CCONT2, CROSTB, BRKDUL, DATAPT, BDOWN1, BDOWN2, FINDDT, CRCSSB, RECODE, CATGYA, ONEWAY, TOWAY
E-10	LOGICAL EXPRESSIONS ARE ERROR, THEN CHECK AGAIN EACH ITEM AND LOGIC. ERROR={A}	指定した論理式の解読ができない。それは項目名Aに発生した。	IFTHEN
E-11	ALL DATA ARE INSUFFICIENT, THEREFORE IT IS IMPOSSIBLE TO STORE ON DISK.	全サンプルが不適切データである。 よって適切データをファイル上に出力することができない。	CONSI, CONIF
E-12	TOTAL NUMBER OF PATTERNS IS MORE THAN 1024.	回答パターンの総数が1024通りをこえた。	PATRN
E-13	USED WORKING AREA WORK (1)~WORK (<i>nnn</i>) USED WORKING AREA IWORK (1)~IWORK (<i>nnn</i>)	作業領域がオーバーフローした。	CCONT2, ASSOCI
E-14	YOU MUST CHECK YOUR DATA.	データに予備がある再検査せよ。	CROSTB, BRKDUL
E-15	YOU THINK TWICE YOUR DATA OR SCALE COMMAND.	図表示に異常がある。 データおよびスケールの与え方を点検せよ。	PLOT
E-16	COMBINATION OF CATEGORY IS GREATER THAN 3000.	項目名ごとのカテゴリーの組合せの総数が3000をこえた。	BDOWN3
E-17	THIS COMMAND WAS MISUSED. ERROR CODE={A}	指定文に誤りがある。 項目名Aを点検せよ。	DFIND
E-18	COMMAND CHECK	指定文の解読が不可能である。	READK
E-19	YOUR GIVEN MATRIX IS MISSING. YOU MUST CHECK YOUR DATA.	与えられた行列に誤りがある。	AMAIN
E-20	IT IS IMPOSSIBLE TO COMPUTE THE ASSOCIATION MEASURE FOR THE FOLLOWING PAIR. {(A, B), (C, D),} (項目名の対のリスト)	関連係数の算出が不可能である。 その項目名の対は、(A, B), (C, D),である。	CROPRI
E-21	OVERLAPPED POINTS ARE MORE THAN 50.	図中の点の重複が50個をこえた。	PICT

[一般メッセージ]

コード	出力・メッセージ	意 味	関連のあるサブルーチン名
M-1	NO OMIT DATA.	不適切データ全くない。 すべて正常データである。	RANGE
M-2	ALL SAMPLES ARE CONSISTENT.	全データが適切データであった。	CONSI, CONIF
M-3	CATEGORY OF {A} IS SINGLETON. (項目名)	項目名のカテゴリーは1個しかない。 (1値データである) それは項目Aにおいて発生した。	CCONT2
M-4	MESSAGE (1) SYMBOL (*) MEANS NO APPEARED CODE. (2) OVERLAPPING-CODES APPEAR IN {A,B,}.	変換指定コードの中に重複コードがある。 その項目名はA, B,である。	RECODE
M-5	PROFILE PATTERN FOR FACE CATEGORY OF ITEM {A,B} (項目名)	このプロフィールパターンは項目名A, Bに関する図である。	PICT

3. 4 MINERVA 命令文一覧表

(その1)

命令語の種類	命 令 語	指 定 文
ジ ョ ブ 制 御 用	{READ} READ READ FORMAT ITEM {INPUT DEVICE} {OUTPUT DEVICE} {OPTION} END	{TAPE nn } CARD DISK nn () A, B, C, D,/ { nn } { nn } {PRINT= n / PUNCH/ STEPWISE/ COMBINATORIAL/ COEFFICIENT/ MARGINAL/ ENTRY/ }
デー タ の 加 工 ・ 検 証 用	RANGE CONSISTENCY PATTERN FIND RECODE	A= $n_1 \sim n_2$, B= $m_1 \sim m_2$ / IF (P)/ IF (P), THEN (Q)/ (P) IFF (Q)/ A, B, C, D,/ $n_1, n_2, m_1 \sim m_2$ / A($n_1, n_2, \dots = l_1; m_1, m_2, \dots = l_2$),/
統 計 分 析 用	ASSOCIATION BREAK DOWN MULTIWAY TABLES QCHART CROSS TABLES BMDP FINISH/	A, B, C, D,/ X WITH A, B, C,/ A, B, C, D,/ A, B, C, D,/ HEAD=A, B, C,, SIDE =X, Y, Z,/ SIDE =A, B, C,, HEAD=X, Y, Z,/ MULTIPLE=A, B, C,, BY (X, Y), (U, V),/ P1D P2D P4D P5D P6D P1F P2F P3F (BMDP連結用命令文)

(その2)

命令語の種類	命 令 語	主 な 制 限 事 項				そ の 他 の 事 項	関係ページ 連シ
		命令文カードの最大枚数	項目名の最大数	命令文カード終了のスラッシュ	作業ファイルの使用の可否		
ジョブ制御用	READ	1	—	なし	—	●標準値は磁気テープ(省略可能)。 ●磁気ディスクからの入力機番指定は不可欠。	10
	FORMAT	10	—	〃	—	●実数型(F型)入力に限る。 ●ITEM文指定の項目数と個数が一致すること。	11
	ITEM	制限なし	50	必要	—	●項目名は英文字で始まること。 ●[READ], FORMATと組み合わせて使用(独立使用できない)。	12
	[INPUT DEVICE] [OUTPUT DEVICE]	1 1	— —	なし 〃	— —	機番nnとして11以下の使用は避けること(とくに10, 11)。	14
	[OPTION]	} 2	—	必要	—		31 39 47
	END		1	—	なし	—	●入力データがカードのとき、この文の後に置く。
データの加工・検証用	RANGE	制限なし	50	必要	INPUT OUTPUT	●数値の正符号(+)は使えない。 ●複数項目のRANGE指定時の論理処理は共通部分をとる。	17
	CONSISTENCY	〃	50	〃	INPUT OUTPUT	●項目名、演算子、数値などが継続カードにまたがってはいけない。 ●左右括弧の数が一致すること。	20
	PATTERN	〃	50	〃	INPUT	●パターンの総数は1024通りまで。	26
	FIND	〃	—	〃	不 可	●参照サンプル数は最大500個まで。 ●参照番号を区間指定するとき"〜"記号の前後が継続カードにまたがらぬこと ●MINERVAファイルに対してのみ有効(機番11または10)。	23
	RECODE	〃	20	〃	INPUT OUTPUT	●1項目当りの変換指定は10組まで。 ●1項目当り、括弧は1組だけ許される。 ●変換指定の等号の省略は出来ない。	27
統計分析用	ASSOCIATION	〃	50	〃	INPUT	項目数が2個のとき係数行列の出力(印刷, パンチ)は不可。	30
	BREAK DOWN	1	9	〃	〃	●項目名Xは1個のみ指定 ●ブレイク・キー項目名A, B, C, ...は9個まで。 ●ブレイク・キーのカテゴリーの組み合わせ総数は最大3000まで。	35
	MULTIWAY TABLES	1	7	〃	〃	●5項目以上は逐次選択方式が標準値である。 ●モデル選択方式の重複指定は不可。	39
	QCHART	1	10	〃	〃	●サンプル数が500個までQ-chartに出力。 ●散布図は項目が2個のときのみ出力。	37
	CROSS TABLES	制限なし	15~20	〃	〃	●ブレイク・ダウンつきのクロス表のとき、()を忘れてはならない。	44
	BMDP	1	—	なし	BMDPジョブ内で指定	●BMDP用のプログラムを編集しなくてはならない。 ●作業用ファイルを介してデータを受け渡す。	50
	FINISH/	1	—	—	—	FINISHカードはBMDPと対で指定すること。	

3.5 例題集

MINERVA の処理機能の理解を容易にするために、また、利用者にテスト・データの便を図るために、簡単な MINERVA ジョブの例を4つあげておく。それぞれに簡単な説明を付与し、また、出力表示の形式・内容を部分的にあげた。

〔例題 1〕

例題1のように命令文カードを与えると、流れ図〔EX-1〕の処理が行われる。これを命令文カードの左横に付与した番号（あるいは流れ図の中の箱番号）に沿って簡単に説明する。

①：磁気テープ上にある原データから必要個所のデータを読みとり、項目名を付与するとともに、MINERVA ファイルを作成する。

②～⑤：それぞれ、MINERVA ファイル内のデータに対して、consistency check を行う。たとえば、④は、「年令 (AGE 2) がコード5 (50才代) であるとき、その職業 (OCCUPATN) はコード7 (学生) であってはならない」という論理関係の検証に相当する。この検証結果が、〔表示1-1〕のように出力される。このケースでは、条件からはずれたサンプルが丁度1サンプルあり、それが、AGE 2 = 5, OCCUPATN = 7 をとることがわかる。

また、論理関係式を満たすサンプルの構成が要約表として出力される。

⑤も類似のケースで、「収入 (INCOME) のコードが7 (600万円台) 以下または10 (800万円台) 以上である」かまたは「年令 (AGE 2) はコード2 (20代) 以上である」を検証する。つまり、年令が10代であって、所得がかなり多いと回答したサンプルを抽出してみようというのである。この結果、〔表示1-2〕を得る。すなわち、9サンプル不適切データがあり、INCOME, AGE 2 がそれぞれどんなコードであったか表示される。

⑥、⑦：与えたサンプル番号のデータ内容を参照する。とくに、⑦は、⑤の検証で表われたデータの内容を参照する命令文となっている (→〔表示1-3〕)。

⑧：データの選出を RANGE 文により指示する。これにより年令 (AGE 2) がコード1 (10代) のサンプルだけを抽出し、ファイル12に格納する。

⑨：CONSISTENCY 文を用いて条件式を満たすデータを選出し、ファイル13の上に格納する。

⑩：⑨で生成したデータを使って、クロス表を作成する。このとき、SIDE 中の AREA、および HEAD 中の AREA は同一項目であるから、この項目を組み合わせたクロス表は出力しない。

⑪：⑧で生成したデータを使って、BREAK DOWN を行う。層化の対象は所得 (INCOME) で、ブレイク・キーは、調査回数 (NO. SURVY)、性 (SEX)、年令 (AGE 2) である。この結果が〔表示1-4〕である。

MINERVA ジョブの最終情報として、ジョブの履歴表を出力する (→3.2.3 の例2参照)。

これをみると、MINERVA ファイル上のデータは8145サンプルであり、⑧で生成したデータは、255サンプル、同様に⑨で生成したデータが2527サンプルあることがわかる。このとき同時

に、MINERVA ファイル、一時作業ファイルに対して、どのような処理を行ったか、その履歴が表示されている。

〔例 題 2〕

例題2のように命令文カードを与えた場合を考えよう。このときの処理の流れは、図〔EX-2〕のとうりである。ここでは主として、データ選出と、対数線形モデル解析の反復処理となっている。

- ①：磁気ディスク上にある原データから MINERVA ファイルを作る。
- ②：RECODE 文により年令 (AGE) のコード変換を行う。
変換結果を作業ファイルに出力 (機番10) する。
- ③：②のコード変換データを使って、学歴 (EDUC 2) がコード1～3にあるデータを選別し、これを作業ファイル12に格納する。
- ④：さらに、ファイル12から「質問 Q 2 に1～2、かつ、国籍 (NATION) がコード1である」ようなサンプルを選出して、ファイル13の上に格納する。
- ⑤：ファイル13上のデータに対して、対数線形モデル解析を行う。モデル選択方式は、総当り法である。オプションとして、出力条件を指定する。
- ⑥：同じく、ファイル13のデータに対して、クロス表 (性×年令) を作成する。オプションとして、比率検定と、印刷出力条件を指定する。
- ⑦：ファイル12のデータを再び呼び出して、質問 Q 15 にコード1、2と回答しているサンプルだけを選出し、それをファイル17に格納する。
- ⑧：そのデータを使って、5元の対数線形モデル解析を行う。クロス表の次元が5元であるから、自動的に、逐次選択方式によりモデルの評価を行う。
- ⑨：ここでは、ファイル12のデータに対して、質問 Q 17 が1～3と回答した者を選出し、ファイル14に格納する。
- ⑩：そのデータに対して、4元の対数線形モデル解析を行う。このとき、オプションにより、モデル選択方式として逐次方式を、印刷条件として条件2を指定する。

上の処理のうち、とくに⑤および⑩の出力の一部をあげておこう (→〔表示2-1〕～〔表示2-4〕)。

⑤の指示に従って分析を行うと、まず、取り上げた項目名リストを表示し、次に総当り法であることを表示して対数線形モデルのうちの完全独立モデルの計算に入る (CONFIG の情報が (0, 0, 0) である)。そして、性 (SEX), 年令 (AGE), 質問 (Q 2) の3元クロス表を算出し、完全独立モデルに対する度数のあてはめおよび各種の統計量を出力する (〔表示2-1〕)。

以下3元クロス表で考えられる19通りのすべてのモデルについて同様の計算を行う。たとえば、〔表示2-2〕は、性、年令、Q 2 の主効果の項に加えて、性×Q 2、年令×Q 2 の交互作用項までを取り入れたモデルの出力結果である。

そして、モデルをすべて総当りした後に、AIC 統計量のいみで、もっとも適合のよいモデルか

ら順に、19通りのモデルの項目の組み合わせ情報を印刷する。同時に、 χ^2 値、確率、AIC などを一覧表として出力する（〔表示2-3〕）。

次に、⑩の場合を説明しよう。この例は、モデル選択方式として逐次選択方式の指定がある。⑤の場合と同様の出力表示の他に、項目名および交互作用項それぞれの取りうる AIC 項別情報の表、モデル選択に利用したモデルの組み合わせと逐次選択によって選ばれたモデルの種々の情報を出力する（〔表示2-4〕〔1〕～〔3〕）。とくに、〔表示2-4〕の〔3〕から、AIC の最も小さい、17番目のモデルが、最適モデルとして選出されたことを知ることができる。

〔例題 3〕

この例は、カード・データを入力し、これに対して分析処理を行う場合である。

- ①：入力データがカードであることを指示し、カード・データを MINERVA ファイルとして定義する。
 - ②：年齢（AGE）に無回答（コードはブランク「 \square 」であるが数値データとして入力するので「0」と与える）のサンプルを排除して、ファイル12上にこのデータを生成する。
 - ③：上のデータを利用して、AGE、PERCNT をコード変換する。とくに、出力ファイルの指定がないので、標準値である機番10がセットされ、このファイル上にコード変換済みのデータが格納される。以下、④～⑨の処理にあたって、とくに入力データの機番指定がないので、このファイル10が MINERVA ファイルとして、呼び出される（つまり、RECODE により、データの“永久変換”を行ったことになる）。なお〔表示3-1〕が、コード変換情報である。
 - ④：クロス表を作成し、同時にオプションで、比率検定と、印刷条件とを指示する（〔表示3-2〕がその出力の一部である）。
 - ⑤：次に、ブレイク・ダウン機能を伴った、クロス表を作成する。
 - ⑥：続いて、TSCORE のデータを V1、V2、V3 のカテゴリーでブレイク・ダウンを行う。
 - ⑦：4つの項目名 AGE、V2、V3、V1 について、この並びで、出現回答パターンとその度数を確認する（〔表示3-3〕）。
 - ⑧：2項目を指定した場合の、関連係数の算出処理である。2元クロス表と全関連係数を印刷する（〔表示3-4〕）。
 - ⑨：指定した6つの項目間の関連係数行列を算出する。オプションにより3種の係数（SQUARE、CONTINGENCY、MCNEMER）を指定した（〔表示3-5〕）。
- このとき、 2×2 次のクロス表は、項目間の組み合わせの中に1個も含まれていなかったのもので、MCNEMER の出力にそのことが表示される。

〔例題 4〕

この例も、〔例題 3〕と同様に、カード・データ入力の場合である。

- ①：カード・データを読み込み、MINERVA ファイルを作成する。
- ②：5項目を使って、多変量管理図を作成する。
- ③：データを選出し、ファイル12に格納する。
- ④：そのデータを使って、2つの項目について、2変量管理図を作成する。〔表示4-1〕が、その出力の一部である。Q管理図の他に、管理楕円、3シグマ限界線を記入した、散布図を出力する。この例では、2つのサンプル（17, 21）が管理はずれとなったことがわかる。
- ⑤：4つの項目の関連係数行列を出力する。係数の種類はすべてを算出し、かつ、カード上にパンチすることをオプションで指定する。
- ⑥：ここで、MINERVA ファイルに対して、3つの項目、SCORE4, X4, ID についてコード変換を行う。その結果を新たに MINERVA ファイルとして準備する（出力ファイルの指定がないので標準機番10に変換データが格納され、かつ、以後の処理は、これを MINERVA ファイルとする）。コード変換の内容を、〔表示4-2〕のように印刷する。
- ⑦：コード変換データに対して、クロス表を作成する。同時に、比率検定を行い、印刷条件を指定する。
- ⑧：同じく、項目 ID がコード4であるようなデータを選出してファイル14に格納する。
- ⑨：指定したサンプル番号のサンプルを参照する。

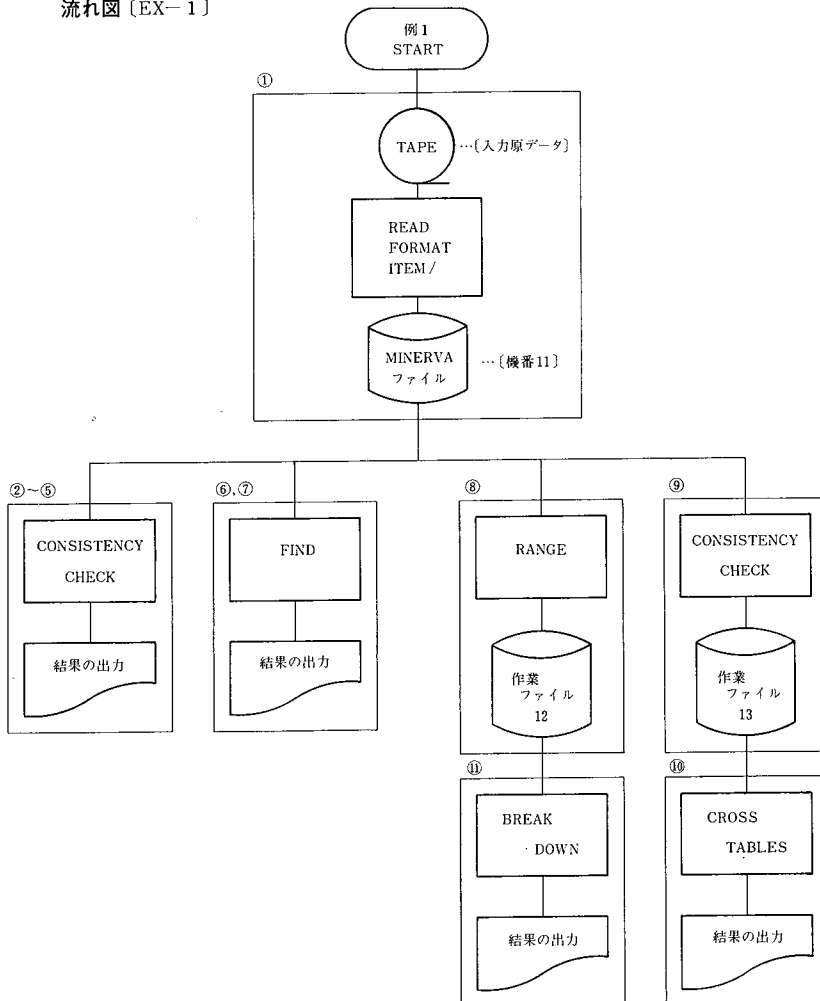
〔テスト・データのリスト〕

上の例題のうち、例題3, 4のテストに使用したデータ・カードのリストを最後にあげておく。
〔例題3〕の場合は、カード1枚（80カラム）内に1サンプルの観測データベクトルをパンチしてある。また、〔例題4〕のデータは、FORMAT文にあるようにカード3枚で1サンプルに対応している。したがって、リストの3行（カード3枚）で、1サンプルのデータに相当する。

【例題 1】の命令文カード

①	READ FORMAT ITEM	TAPE (F2.0,F4.0,6F2.0,X,F2.0,2X,F2.0,30X,3F2.0,10X,F2.0) NO,SURVY,NO,SAMPL,NO,CARD,AREA,PREFEC,CITYSCL,SEX,AGE1, AGE2,OCCUPATN,INCOME,CAREER,TIMES,INDA'S,SIGHT,TI SCORE /
②	[CONSIST-CHECK	IF(AGE2 LE 2),THE(AGE2 GT 2 OR INCOME GE 7) /
③	[CONSIST	IF(SEX EQ 2 AND AGE2 GE 6),THEN(CAREER LE 5) /
④	[CONSIST TEST	IF(AGE2 GE 5),THE(OCCUPATN NE 7) /
⑤	[CONSISTENCY	IF(AGE2 GT 2 OR INCOME LE 7 OR INCOME GE 10) /
⑥	[FIND DATA	1,4,9,14,20,32,38,59,70,84,108,1580,2123,2917,3823,3535 /
⑦	[FIND DATA	245,445,2062,2740,4473,5730,6869,7170,7214 /
⑧	[RANGE [OUTPUT DEVICE	AGE2 1 / 12
⑨	[CONSISTENCY	IF((NO,SURVY GE 2 AND NO,SURVY LT 6) AND SEX EQ 1) /
⑩	[OUTPUT DEVICE	13
⑪	[INPUT DEVICE	SIDE=AGE2,AREA HEAD=INCOME,CAREER,AREA /
⑫	[CROSS TABLES	
⑬	[INPUT DEVICE [BREAK DOWN	12 INCOME WITH NO,SURVY,SEX,AGE2 /
⑭	[END	

流れ図 [EX-1]



〔表示 1－1〕 論理検証の出力例

```

*****
I CONSISTENCY CHECK I
*****

IF THEN
-----

----- INCONSISTENT DATA ----- ( A --> B )

NO. DATA      AGE2      DATA X --
1  3539         5.0      OCCUPATH
                        7.0

SUMMARY TABLE
-----

TOTAL NUMBER OF SAMPLES  TN = 8145
NUMBER OF SAMPLES       N = 8145
IF ( A ) , THEN ( B )
A --- TRUE = 1905 23.39 % ( 1/N )
      FALSE = 6240 76.61 % ( 1/N )
A --- TRUE , B --- TRUE = 1904 99.95 % 23.38 % ( 1/N )
      FALSE = 1 0.05 % 0.01 % ( 1/N )

```

〔表示 1－2〕 論理検証の出力例

```

*****
I CONSISTENCY CHECK I
*****

IF THEN
-----

----- INCONSISTENT DATA ----- ( A )

NO. DATA      AGE2      DATA X --
1  245          2.0      INCOME
2  445          2.0      INCOME
3  2062         1.0      INCOME
4  2740         2.0      INCOME
5  4473         2.0      INCOME
6  5730         2.0      INCOME
7  6968         2.0      INCOME
8  7170         2.0      INCOME
9  7214         2.0      INCOME

SUMMARY TABLE
-----

TOTAL NUMBER OF SAMPLES  TN = 8145
NUMBER OF SAMPLES       N = 8145
IF ( A )
A --- TRUE = 8136 99.89 % ( 1/N )
      FALSE = 9 0.11 % ( 1/N )

```

〔表示 1－3〕 データ参照の例

```

*****
I FIND DATA I
*****

*****
ITEM      FIND DATA NUMBER
*****
NO. SURVY  245  445  2062  2740  4473  5730  6869  7170  7214
NO. SAMPL  804.0 3603.0 575.0 1253.0 38.0 1295.0 282.0 583.0 627.0
NO. CARD   1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0  1.0
AREA       24.0 24.0 25.0 24.0 11.0 25.0 21.0 21.0 21.0
PREPEC     98.0 98.0 98.0 98.0 1.0 21.0 14.0 12.0 8.0
CITYSCL    11.0 21.0 50.0 11.0 32.0 42.0 42.0 32.0 50.0
SEX        1.0  1.0  1.0  2.0  2.0  2.0  2.0  1.0  1.0
AGE1       98.0 98.0 98.0 98.0 27.0 20.0 98.0 98.0 98.0
AGE2       2.0  2.0  1.0  2.0  2.0  2.0  5.0  2.0  2.0
OCCUPATH    3.0  2.0  2.0  5.0  2.0  7.0  9.0  3.0  1.0
INCOME      8.0  8.0  8.0  9.0  9.0  8.0 10.0 9.0 8.0
CAREER      3.0  4.0  3.0  2.0  3.0  4.0  3.0  3.0  2.0
TIMES       1.0  1.0  2.0  2.0  1.0  2.0  2.0  1.0  1.0
NOANSW     98.0 98.0 1.0  1.0  1.0  1.0  1.0  1.0  1.0
SIGHT      98.0 98.0 98.0 98.0 1.0 -1.0 -1.0 1.0 -1.0
TLCORE     98.0 98.0 98.0 98.0 4.0 -1.0 0.0 2.0 92.0
*****

```

〔例題 2〕の命令文カード

```

① READ      DISK 01
   FORMAT   (35F1.0)
   ITEM      02+04+05+06+07+08+09+011+012+014+015+016+017+018+019+1+019+2+
   019+3+019+4+021+022+023+024+026+027+030+033+039+040+042+
   SEX+AGE+EDUC2+AGE2+NATION /

② RECODE    AGE(1,2=1;3,4=2;5,6=3;7,8=4;9=5) /
   OUTPUT DEVICE 10
   INPUT DEVICE 10
③ RANGE     EDUC2 1-3 /
   OUTPUT DEVICE 12
   INPUT DEVICE 12
④ RANGE     02 1-2 ,NATION =1 /
   OUTPUT DEVICE 13
   INPUT DEVICE 13
⑤ MULTIWAY TABLES SEX AGE 02 /
   OPTION    PRINT=4 /

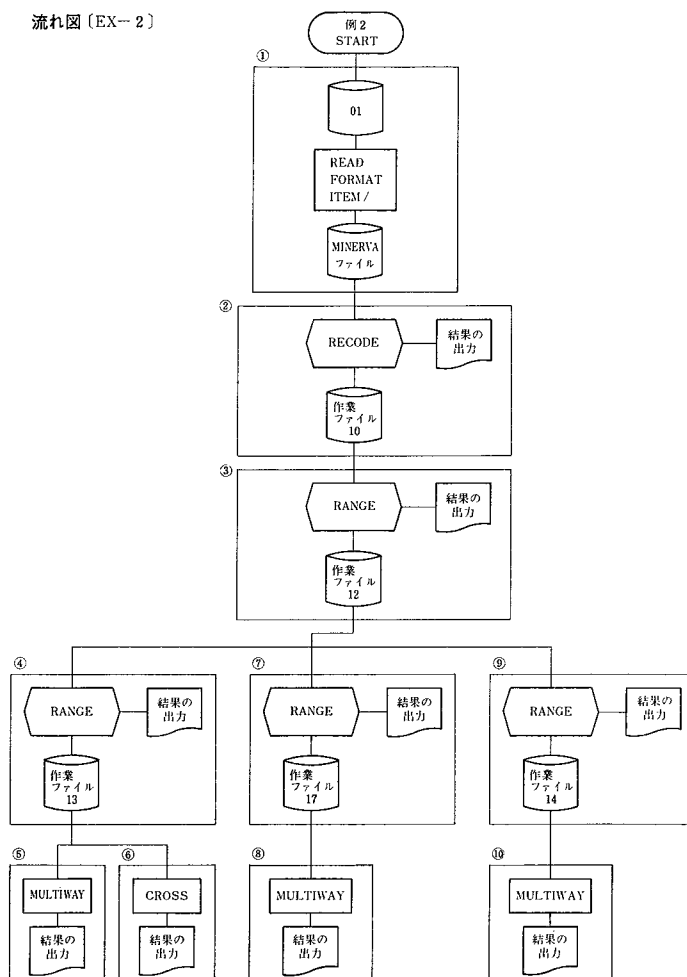
⑥ INPUT DEVICE 13
   CROSSTAR  HEAD = SEX SIDE = AGE /
   OPTION    MARGINAL, PRINT=1 /

⑦ INPUT DEVICE 12
   RANGE     015 1-2 /
   OUTPUT DEVICE 17
   INPUT DEVICE 17
⑧ MULTIWAY TABLES SEX AGE EDUC2 NATION 015 /

⑨ INPUT DEVICE 12
   RANGE     017 1-3 /
   OUTPUT DEVICE 14
   INPUT DEVICE 14
⑩ MULTIWAY TABLES SEX AGE EDUC2 017 /
   OPTION    STEPWISE, PRINT=2 /

⑪ END
  
```

流れ図〔EX-2〕



[表示 2-1] 対数線形モデルの出力の一部 (完全独立モデルのとき)

```

MULTIDIMENSIONAL
CONTINGENCY
TABLE

```

----- LIST OF ITEMS -----

```

ITEM      CODE
-----
SEX        1
AGE        2
Q2         3

```

ORDINARY METHOD

```

** CONFIG **
SEQ NUM    1 0 0 0

```

```

ITEM      1<SEX  >
ITEM 3 <Q2  > 1      2
2
<AGE  >
1      122.0000  47.0000  169.0000
2      171.0000  37.0000  208.0000
3      166.0000  26.0000  192.0000
4      103.0000  27.0000  130.0000
5      100.0000  42.0000  142.0000
        662.0000  179.0000  841.0000

```

```

ITEM      1<SEX  >
ITEM 3 <Q2  > 1      2
2
<AGE  >
1      147.0000  91.0000  238.0000
2      174.0000  65.0000  239.0000
3      183.0000  55.0000  238.0000
4      123.0000  43.0000  166.0000
5      84.0000   63.0000  147.0000
        711.0000  317.0000  1028.0000

```

** CONTINGENCY TABLES **

```

ITEM      1<SEX  >
ITEM 3 <Q2  > 1      2
2
<AGE  >
1      122.0000  47.0000  169.0000
        93.4500  93.4500  186.9000
2      171.0000  37.0000  208.0000
        93.4500  93.4500  186.9000
3      166.0000  26.0000  192.0000
        93.4500  93.4500  186.9000
4      103.0000  27.0000  130.0000
        93.4500  93.4500  186.9000
5      100.0000  42.0000  142.0000
        93.4500  93.4500  186.9000
        662.0000  179.0000  841.0000
        467.2500  467.2500  934.5000

```

```

ITEM      1<SEX  >
ITEM 3 <Q2  > 1      2
2
<AGE  >
1      147.0000  91.0000  238.0000
        93.4500  93.4500  186.9000
2      174.0000  65.0000  239.0000
        93.4500  93.4500  186.9000
3      183.0000  55.0000  238.0000
        93.4500  93.4500  186.9000
4      123.0000  43.0000  166.0000
        93.4500  93.4500  186.9000
5      84.0000   63.0000  147.0000
        93.4500  93.4500  186.9000
        711.0000  317.0000  1028.0000
        467.2500  467.2500  934.5000

```

** TABLE OF MARG(,) **

1869.0

CONVERGENCE ... NUMBER OF ITERATIONS = 1

** THE MAXIMUM OBSERVED DIFFERENCE ENCOUNTERED IN ITERATION CYCLE () BETWEEN AN OBSERVED AND FITTED MARGINAL TOTAL **
ITERATION 1
0.00000

** INDICATOR OF MARG **
1

```

PEARSON'S CHISQUARE ( GOODNESS OF FIT ) = 570.218834
LIKELIHOOD RATIO CHISQUARE = 580.516642
PROBABILITY OF PEARSON'S CHISQUARE = 0.0
PROBABILITY OF LIKELIHOOD RATIO CHISQUARE = 0.0
DEGREES OF FREEDOM = 19
AIC BASED ON GOODNESS OF FIT = 532.218834
AIC BASED ON LIKELIHOOD RATIO STATISTIC = 542.516642

```

〔表示2-2〕対数線形モデルの出力の一部（交互作用項も含めたとき）

```

---- MULTINOMIAL TABLES SEX AGE Q2
*****

## CONFIG ##
SEQ NUM 1 1 0 0 SEX
          2 1 0 0 AGE
          3 1 0 0 Q2
          4 1 1 0 SEX
          5 1 2 0 AGE Q2

## CONTINGENCY TABLES ##

ITEM 1<SEX 1
ITEM 3 <Q2 1 2
2<AGE 1
1 122.0000 47.0000 169.0000
  129.6999 49.0024 179.5023
2 171.0000 37.0000 208.0000
  166.3438 36.0105 203.1543
3 166.0000 26.0000 192.0000
  168.2724 29.2319 197.5043
4 103.0000 27.0000 130.0000
  108.9672 25.2621 134.2293
5 100.0000 42.0000 142.0000
  88.7167 37.8931 126.6098
662.0000 179.0000 841.0000
662.0000 179.0000 841.0000

ITEM 1<SEX 2
ITEM 3 <Q2 1 2
2<AGE 1
1 147.0000 91.0000 238.0000
  139.3001 88.1976 227.4977
2 174.0000 65.0000 239.0000
  178.6562 65.1895 243.8457
3 183.0000 55.0000 238.0000
  180.7276 51.7681 232.4957
4 123.0000 43.0000 166.0000
  117.0328 44.7379 161.7707
5 84.0000 63.0000 147.0000
  95.2833 67.1069 162.3902
711.0000 317.0000 1028.0000
711.0000 317.0000 1028.0000

## TABLE OF MARG(.) ##
841.0 1028.0 407.0 447.0 430.0 296.0 289.0 1373.0 496.0 662.0
711.0 179.0 317.0 269.0 345.0 349.0 226.0 184.0 138.0 102.0
81.0 78.0 105.0

CONVERGENCE ... NUMBER OF ITERATIONS = 2
## THE MAXIMUM OBSERVED DIFFERENCE ENCOUNTERED IN ITERATION CYCLE ( . ) BETWEEN AN OBSERVED AND FITTED MARGINAL TOTAL ##
ITERATION 1 2
          438.50000 0.00000

## INDICATOR OF MARG ##
1 3 8 10 14
PEARSON'S CHISQUARE ( GOODNESS OF FIT ) = 6.286777
LIKELIHOOD RATIO CHISQUARE = 6.288571
PROBABILITY OF PEARSON'S CHISQUARE = 0.015143
PROBABILITY OF LIKELIHOOD RATIO CHISQUARE = 0.014942
DEGREES OF FREEDOM = 8
AIC BASED ON GOODNESS OF FIT = -9.713223
AIC BASED ON LIKELIHOOD RATIO STATISTIC = -9.711429

```

〔表示2-3〕 対数線形モデルのコンフィギュレーションの表と統計量一覧表

---- MULTIWAY TABLES SEX AGE Q2							
MODEL	NUMBER	CONFIGURATION	OF MODELS				
1	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	1 3 0	SEX				
	5	2 3 0	AGE				
2	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	1 2 0	SEX				
	5	1 3 0	SEX				
	6	2 3 0	AGE				
3	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	1 2 0	SEX				
	5	1 3 0	SEX				
	6	2 3 0	AGE				
	7	1 2 3	SEX				
4	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	2 3 0	AGE				
5	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	1 2 0	SEX				
	5	2 3 0	AGE				
6	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	1 3 0	SEX				
7	1	2 0 0	AGE				
	2	3 0 0	Q2				
	3	2 3 0	AGE				
8	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	1 2 0	SEX				
	5	1 3 0	SEX				
9	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
10	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	3 0 0	Q2				
	4	1 2 0	SEX				
11	1	2 0 0	AGE				
	2	3 0 0	Q2				
12	1	1 0 0	SEX				
	2	3 0 0	Q2				
	3	1 3 0	SEX				
13	1	1 0 0	SEX				
	2	3 0 0	Q2				
14	1	3 0 0	Q2				
15	1	1 0 0	SEX				
	2	2 0 0	AGE				
16	1	1 0 0	SEX				
	2	2 0 0	AGE				
	3	1 2 0	SEX				
17	1	2 0 0	AGE				
18	1	1 0 0	SEX				
19	1	0 0 0					

---- MULTIWAY TABLES SEX AGE Q2							
MODEL	DEGREES OF	PEARSON'S GOODNESS OF FIT	LIKELIHOOD RATIO	AIC BASED ON	AIC BASED ON		
SEQ NUM	FREEDOM	CHISQUARE (PROBABILITY)	CHISQUARE (PROBABILITY)	GOODNESS OF FIT	LIKELIHOOD RATIO STATISTIC		
1	8	6.286777	0.6151	6.286571	0.6149	-9.713223	-9.711429
2	4	1.057205	0.9010	1.051501	0.9019	-6.942795	-6.948499
3	0	---	---	---	---	---	---
4	9	27.862843	0.0010	28.202847	0.0009	9.862843	10.202847
5	5	23.315261	0.0003	23.631996	0.0003	13.315261	13.631996
6	12	49.089671	0.0000	48.992995	0.0000	25.089671	24.992995
7	10	46.293922	0.0000	46.944195	0.0000	26.293922	26.944195
8	8	44.643448	0.0000	44.422144	0.0000	28.643448	28.422144
9	13	70.412093	0.0000	70.907271	0.0000	44.412093	44.907271
10	9	66.141684	0.0000	66.336420	0.0000	48.141684	48.336420
11	14	90.064318	0.0000	89.648619	0.0000	62.064318	61.648619
12	16	109.645489	0.0000	111.728308	0.0000	77.645489	79.728308
13	17	132.470073	0.0000	133.642584	0.0000	98.470073	99.642584
14	18	153.016070	0.0000	152.383932	0.0000	117.016070	116.383932
15	14	468.778672	0.0	499.039981	0.0	440.778672	471.039981
16	10	463.097511	0.0	494.469130	0.0	443.097511	474.469130
17	15	475.753861	0.0	517.781329	0.0	445.753861	487.781329
18	18	562.775294	0.0	561.775294	0.0	526.578860	525.775294
19	19	570.218834	0.0	580.516642	0.0	532.218834	542.516642

〔表示2-4〕 逐次選択方式の出力表示の一部

〔1〕 項別情報量の表

---- MULTIWAY TABLES SEX AGE EDUC2 Q17 ----				
EFFECT OF ADDITIONAL TERM BY AIC BASED ON LIKELIHOOD RATIO STATISTIC				
ADDITIONAL TERM	DEGREES OF FREEDOM	EFFECT OF AIC		
0	89	1601.4563		
1	1	-11.3337	SEX	
2	4	-54.3632	AGE	
3	2	-141.4304	EDUC2	
4	2	-923.8646	Q17	
5	4	-4.0606	SEX	AGE
12	2	-48.5673	SEX	EDUC2
13	4	-4.8975	SEX	Q17
14	2	-324.0716	AGE	EDUC2
23	8	-30.5425	AGE	Q17
24	8	-102.2022	EDUC2	Q17
34	4	2.7234	SEX	AGE
123	8	8.0001	SEX	AGE
124	4	7.2996	SEX	EDUC2
134	16	8.0446	AGE	EDUC2
234	16	17.7215	SEX	AGE
1234				Q17

〔2〕 逐次選択に用いるモデルの一覧表

---- MULTIWAY TABLES SEX AGE EDUC2 Q17 ----				
NUMBER OF SEQUENTIAL MODELS	CONFIGURATION OF MODELS			
1	1 0 0 0 0			
2	1 1 0 0 0	SEX		
3	1 2 0 0 0	AGE		
4	1 3 0 0 0	EDUC2		
5	1 4 0 0 0	Q17		
6	1 1 0 0 0	SEX		
2	2 0 0 0 0	AGE		
3	1 2 0 0 0	SEX	AGE	
7	1 1 0 0 0	SEX		
2	3 0 0 0 0	EDUC2		
3	1 3 0 0 0	SEX	EDUC2	
8	1 1 0 0 0	SEX		
2	4 0 0 0 0	Q17		
3	1 4 0 0 0	SEX	Q17	
9	1 2 0 0 0	AGE		
2	3 0 0 0 0	EDUC2		
3	2 3 0 0 0	AGE	EDUC2	
10	1 2 0 0 0	AGE		
2	4 0 0 0 0	Q17		
3	2 4 0 0 0	AGE	Q17	
11	1 3 0 0 0	EDUC2		
2	4 0 0 0 0	Q17		
3	3 4 0 0 0	EDUC2	Q17	
12	1 1 0 0 0	SEX		
2	2 0 0 0 0	AGE		
3	3 0 0 0 0	EDUC2		
4	1 2 0 0 0	SEX	AGE	
5	1 3 0 0 0	SEX	EDUC2	
6	2 3 0 0 0	AGE	EDUC2	
7	1 2 3 0 0	SEX	AGE	EDUC2
13	1 1 0 0 0	SEX		
2	2 0 0 0 0	AGE		
3	4 0 0 0 0	Q17		
4	1 2 0 0 0	SEX	AGE	
5	1 4 0 0 0	SEX	Q17	
6	2 4 0 0 0	AGE	Q17	
7	1 2 4 0 0	SEX	AGE	Q17
14	1 1 0 0 0	SEX		
2	2 0 0 0 0	AGE		
3	3 0 0 0 0	EDUC2		
4	4 0 0 0 0	Q17		
5	1 2 0 0 0	SEX	AGE	
6	1 3 0 0 0	SEX	EDUC2	
7	1 4 0 0 0	SEX	Q17	
8	2 3 0 0 0	AGE	EDUC2	
9	2 4 0 0 0	AGE	Q17	
10	3 4 0 0 0	EDUC2	Q17	
15	1 2 0 0 0	AGE		
2	3 0 0 0 0	EDUC2		
3	4 0 0 0 0	Q17		
4	1 3 0 0 0	SEX	EDUC2	Q17
5	1 4 0 0 0	SEX	Q17	
6	3 4 0 0 0	AGE	EDUC2	Q17
7	2 3 4 0 0	AGE	EDUC2	Q17
16	1 1 0 0 0	SEX		
2	2 0 0 0 0	AGE		
3	3 0 0 0 0	EDUC2		
4	4 0 0 0 0	Q17		
5	1 2 0 0 0	SEX	AGE	
6	1 3 0 0 0	SEX	EDUC2	
7	1 4 0 0 0	SEX	Q17	
8	2 3 0 0 0	AGE	EDUC2	
9	2 4 0 0 0	AGE	Q17	
10	3 4 0 0 0	EDUC2	Q17	
17	1 1 0 0 0	SEX		
2	2 0 0 0 0	AGE		
3	3 0 0 0 0	EDUC2		
4	4 0 0 0 0	Q17		
5	1 2 0 0 0	SEX	AGE	
6	1 3 0 0 0	SEX	EDUC2	
7	1 4 0 0 0	SEX	Q17	
8	2 3 0 0 0	AGE	EDUC2	
9	2 4 0 0 0	AGE	Q17	
10	3 4 0 0 0	EDUC2	Q17	

〔3〕 モデルの一覧と各種統計量（逐次選択の結果）

---- MULTIWAY TABLES SEX AGE EDUC2 Q17 ----				
MODEL SEQ NUM	DEGREES OF FREEDOM	PEARSON'S GOODNESS OF FIT CHISQUARE (PROBABILITY)	LIKELIHOOD RATIO CHISQUARE (PROBABILITY)	AIC BASED ON GOODNESS OF FIT
1	89	1854.682625	0.0	1676.682625
2	88	1824.525202	0.0	1648.525202
3	85	1718.408488	0.0	1548.408488
4	87	1725.557567	0.0	1551.557567
5	87	880.678553	0.0	706.678553
6	80	1677.916691	0.0	1517.916691
7	84	1645.533473	0.0	1477.533473
8	84	834.013864	0.0	666.013864
9	75	1193.370054	0.0	1081.370054
10	75	720.618420	0.0	570.618420
11	81	607.283919	0.0	433.959138
12	60	1087.533200	0.0	967.533200
13	60	656.076136	0.0	536.076136
14	72	523.316938	0.0	379.316938
15	45	120.477636	0.0000	33.026452
16	0	---	---	---
17	52	79.398788	0.0085	-22.340979

〔例題 3〕の命令文カード

①	READ FORMAT ITEM LIST	CARD (F3.0,1X,3F2.0,14X,F2.0,2X,3F2.0,F3.0) SEQUENCE,V1,V2,V3,TSCORE,PERCNT,AGE,CLASS,NSAMPLE /
②	CONSISTENCY OUTPUT RECORD	IF(AGE NE 0) / 12
③	INPUT TRANSF RECODE	12 AGE(10~20=1,21~30=2,31~40=3, OTHER=4) PERCNT(-1~5=1,5~30=2,30~60=3,60~100=4) /
④	CROSS TABLES OPTION	HEAD= CLASS, AGE, SIDE = V1,V3,V2,PERCNT / PRINT=2, MARGINAL /
⑤	CROSS BREAK	MULTIPLE = CLASS BY(V1,V2), (V3,AGE), (PERCNT,AGE) /
⑥	BREAK TSCORE	TSCORE WITH V1,V2,V3 /
⑦	PATTERN	AGE,V2,V3,V1 /
⑧	ASSOCIATION	CLASS,V3 /
⑨	ASSOCIATION OPTIONS	CLASS,V3,V2,AGE,V1,PERCNT / COEFFICIENTS = SQUARE, CONTINGENCY, MCNEMER /
⑩	END	

データ・カード

〔表示 3-1〕コード変換の結果の要約

1 RECODE 1

SAMPLES = 67

AGE

RECODE = 67

NDN-RECODE = 0

PERCNT

RECODE = 67

NDN-RECODE = 0

ITEM	PRE-CODE	AFTER-CODE
AGE	10~20	1
	21~30	2
	31~40	3
	OTHER	4
PERCNT	-1~5	1
	5~30	2
	30~60	3
	60~100	4

--- MESSAGE ---

(1) SYMBOL (R) MEANS NO APPEARED CODE.

(2) OVERLAPPING-CODE APPEAR IN (PERCNT)

CROSS CLASSIFIED TABLES
AND
PERCENTILE TEST

(CLASS)										(CLASS)									
1.	2.	3.	4.	5.	8.	9.	1.	2.	3.	4.	5.	8.	9.	TOTAL					
(V1)							(V1)												
1.	0	0	0	0	0	1	1.	0.0	0.0	0.0	0.0	0.0	55.5	66.7+100.0					
2.	1	0	0	0	0	1	2.	8.3	0.0	0.0	0.0	8.3	25.5	58.3+100.0					
3.	1	0	0	0	0	1	3.	0.0	58.8	0.0	33.3	7.7	25.5	13.1+100.0					
4.	0	1	1	6	3	3	4.	19.2	3.8	42.3	23.1	11.5	0.0	0.0+100.0					
5.	1	3	4	3	0	0	5.	9.1	27.3	36.4	27.3	0.0	0.0	0.0+100.0					
99.	0	0	0	1	0	1	99.	0.0	0.0	0.0	50.0	0.0	50.0*	0.0+100.0					
TOTAL	7	8	15	14	5	6	TOTAL	10.4	11.9	22.4	20.9	7.5	9.0	17.9+100.0					
(V3)							(V3)												
1.	0	0	0	0	0	1	1.	0.0	0.0	0.0	0.0	0.0	0.0	100.0+100.0					
2.	0	0	0	0	0	1	2.	0.0	0.0	0.0	0.0	0.0	0.0	100.0+100.0					
3.	0	0	1	0	2	3	3.	15.4	0.0	7.7	0.0	15.4	0.0	58.3+100.0					
4.	2	2	0	0	0	4	4.	14.3	21.4	14.3	0.0	14.3	14.3	100.0+100.0					
5.	2	7	8	5	2	0	5.	8.3	29.2	33.3	20.8	8.3	0.0	0.0+100.0					
99.	1	0	0	3	4	0	99.	11.1	0.0	33.3	44.4	11.1	0.0	0.0+100.0					
TOTAL	7	8	15	14	5	6	TOTAL	10.4	11.9	22.4	20.9	7.5	9.0	17.9+100.0					
(V2)							(V2)												
1.	0	0	0	0	0	1	1.	0.0	0.0	0.0	0.0	0.0	0.0	100.0+100.0					
2.	0	1	0	0	1	2	2.	0.0	5.6	0.0	5.6	11.1	22.2	55.6+100.0					
3.	5	2	5	3	10	2	3.	22.7	1.1	13.5	45.5	0.0	0.0	0.0+100.0					
4.	2	5	9	2	2	0	4.	10.0	25.0	45.0	10.0	10.0	0.0	0.0+100.0					
5.	0	0	3	0	1	0	5.	0.0	0.0	75.0*	0.0	25.0	0.0	0.0+100.0					
99.	0	0	0	1	0	1	99.	0.0	0.0	0.0	50.0	0.0	50.0*	0.0+100.0					
TOTAL	7	8	15	14	5	6	TOTAL	10.4	11.9	22.4	20.9	7.5	9.0	17.9+100.0					
(PERCENT)							(PERCENT)												
1.	1	2	3	3	1	0	1.	9.1	18.2	27.3	27.3	9.1	9.1	10.0+100.0					
2.	1	3	5	6	8	3	2.	8.6	14.5	17.1	21.9	8.6	11.4	17.1+100.0					
3.	3	1	3	2	1	0	3.	23.1	7.7	23.1	15.4	7.7	0.0	23.1+100.0					
4.	3	5	9	3	3	0	4.	18.2	3.7	37.3	20.8	18.2	17.5	10.0+100.0					
TOTAL	7	8	15	14	5	6	TOTAL	10.4	11.9	22.4	20.9	7.5	9.0	17.9+100.0					

```

+-----+
| PATTERN CHECK |
+-----+

```

.....
 N = 67

〔表示3-4〕 関連係数（2元クロスするとき）の出力表示例

```

+-----+
| ASSOCIATION MEASURES |
|         FOR          |
| CONTINGENCY TABLES |
+-----+

```

V3 = CLASS

Y/X	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
0.	0.	0.	0.	0.	0.	0.	1.	1.	1.	1.
1.	0.	0.	0.	0.	0.	0.	4.	1.	4.	4.
2.	2.	0.	1.	0.	2.	3.	5.	1.	13.	13.
3.	2.	1.	3.	4.	0.	2.	2.	1.	14.	14.
4.	2.	7.	8.	5.	2.	0.	0.	1.	24.	24.
5.	1.	0.	3.	4.	1.	0.	0.	1.	9.	9.
99.	0.	0.	0.	1.	0.	1.	0.	1.	2.	2.
	1.	7.	8.	15.	14.	5.	6.	12.	1.	67.

PEARSON'S CHI-SQUARE

X2 = 64.582458

D(I,J) =

```

-0.346143 -0.371009 -0.541140 -0.517835 -0.286124 -0.315992 2.157029
-0.704483 -0.759479 -1.107748 -1.060041 -0.585715 -0.640856 4.415580
0.648206 -1.478872 -1.415895 -2.064135 1.210699 1.986248 2.152532
0.527853 -0.622410 -0.006834 0.759286 -1.194482 0.785354 -0.397688
-0.422719 3.246676 1.005694 -0.009354 0.202603 -1.917871 -2.856392
0.069920 -1.187307 0.846640 1.867620 0.447656 -1.011242 -1.506099
0.449821 -0.528706 -0.771151 1.027846 -0.407741 2.063695 -0.670664

```

V(I,J) =

```

0.882156 0.867454 0.764536 0.779238 0.911562 0.896859 0.808643
0.842058 0.828024 0.729784 0.743818 0.870127 0.856093 0.771887
0.721764 0.709735 0.625529 0.637559 0.745823 0.733794 0.661617
0.708398 0.696592 0.613945 0.625752 0.732012 0.720205 0.649365
0.574738 0.565159 0.498106 0.507685 0.593896 0.584517 0.526643
0.775228 0.762308 0.671865 0.684785 0.801069 0.788149 0.710626
0.868790 0.854311 0.752952 0.767432 0.897750 0.883276 0.796391

```

EE(I,J) =

```

-0.323230 -0.345557 -0.473160 -0.457116 -0.273179 -0.299253 1.939700
-0.646460 -0.691095 -0.946320 -0.914232 -0.546358 -0.598506 3.879400
0.550694 -1.245888 -1.119838 -1.648155 1.045573 1.701455 1.750867
0.444275 -0.519476 -0.075874 0.628300 -1.022142 0.660480 -0.320469
-0.320469 2.442260 1.133245 -0.006665 0.156135 -1.466033 -2.073283
0.001568 -1.036641 0.693969 1.545488 0.400663 -0.897758 -1.269621
-0.457116 -0.488678 -0.669150 0.900426 -0.386334 1.939700 -0.598506

```

E(I,J) =

```

0.104478 0.119403 0.223881 0.208955 0.074627 0.089552 0.179104
0.417910 0.477612 0.895522 0.835821 0.298507 0.338209 0.716418
1.358209 1.552238 2.910447 2.716417 0.970149 1.164179 2.328358
1.462687 1.671641 3.134328 2.925373 1.044776 1.253731 2.507403
2.507403 2.885071 5.373134 5.014925 1.791044 2.149253 4.298507
0.940298 1.074626 2.014925 1.880596 0.671642 0.805970 1.611939
0.208955 0.238806 0.447761 0.417910 0.149254 0.179104 0.358209

```

COEFFICIENT OF CONTINGENCY (CONSISTENCY)

P = 0.700581

F = 0.963917

KENDALL AND STUART'S

T = 0.106653

CRAMER'S

C = 0.400815

S(Y,X) = 145.030

S(X,X) = 490.567

S(Y,Y) = 17871.8

R = 0.948981

B(Y,X) = 0.295637

VB(Y) = 0.543744

KENDALL'S TAU (T)

TAU-B = -0.334929

TAU-C = -0.312913

GOODMAN AND KRUSKAL'S GAMMA

GAMMA = -0.404027

SOMERS'

DX = -0.321067

DXY = -0.349391

GOODMAN AND KRUSKAL'S

LAMBDA-A = 0.180046

LAMBDA-B = 0.230769

LAMBDA = 0.210526

〔表示 3－5〕 関連係数行列の表示例

```

+-----+
| ASSOCIATION MEASURES |
|           FOR         |
| CONTINGENCY TABLES |
+-----+

```

.....<< COEFFICIENT OF CONTINGENCY (PHI SQUARE) >>.....(LIST 1).....

	# CLASS	# V3	# V2	# AGE	# V1
V3	0.963917				
V2	1.126329	1.951356			
AGE	0.417842	0.165770	0.198861		
V1	0.996446	1.772968	2.111032	0.198197	
PERCNT	0.191536	0.285690	0.192874	0.084923	0.262533

.....<< COEFFICIENT OF CONTINGENCY (P) >>.....(LIST 1).....

	# CLASS	# V3	# V2	# AGE	# V1
V3	0.708581				
V2	0.727809	0.813125			
AGE	0.542865	0.356686	0.349184		
V1	0.706832	0.799610	0.823749	0.369580	
PERCNT	0.400933	0.471389	0.402105	0.279170	0.456006

.....<< MCNEMER'S CHI SQUARE >>.....(LIST 1).....

--- ALL TABLES ARE NOT (2*2) DIMENSION. ---

〔例題 4〕の命令文カード

```

① READ DATA      CARD
  FORMAT           /F3.0,4F5.1/8(8X,F2.0)/2F6.0,13X,F4.0/
  ITEM             NSFG,SCORE1,SCORE2,SCORE3,SCORE4,
                  X1,X2,X3,X4,X5,X6,X7,X8,RATE,EVALT,10 /

② QCHART          SCORE2,RATE,SCORE3,EVALT,SCORE1 /

③ CONSISTENCY     IF(10 LE 5000 AND X5 NE 1) /
  OUTPUT           12

④ INPUT DATA     12
  QCHART           SCORE2, SCORE3/

⑤ ASSOCIATION     X1,X2,X8,X7 /
  OPTIONS          COEFF = ALL, PUNCH /

⑥ RECODE          SCORE4(35-40=1,40-42=2,42-44=3, OTHER=9),
                  X4(0-2=1,3-4=2,5=3,6-10=4)
                  10(1001-1010=1,2001-2010=2,3001-4010=3, OTHER=4) /

⑦ CROSS CLASSIFY SIDE=10, HEAD=X2,X8,X4 /
  OPTION           MARGINAL TEST,PRINT=1 /

⑧ CONSIST CHECK   IF(10 EQ 4) /
  OUTPUT           14

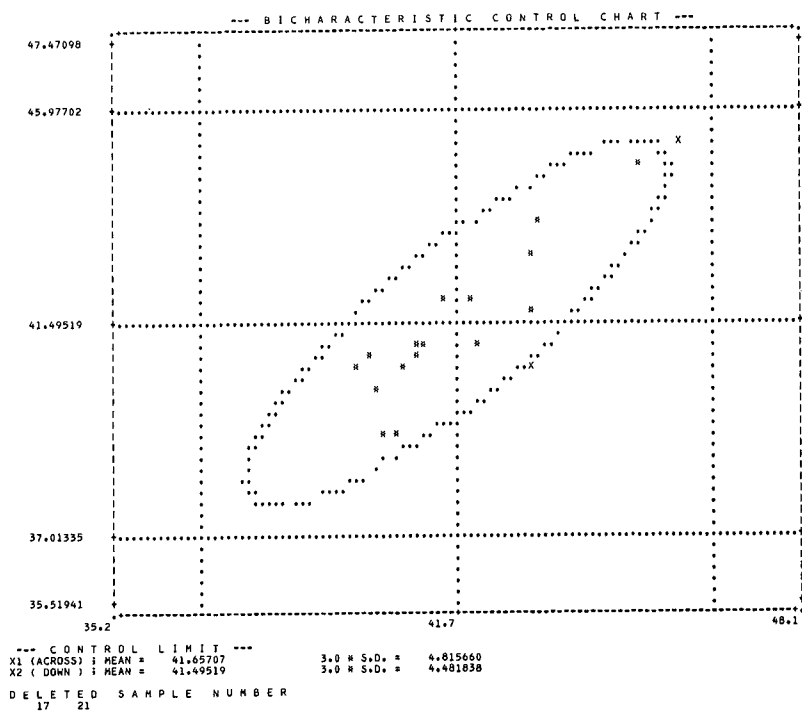
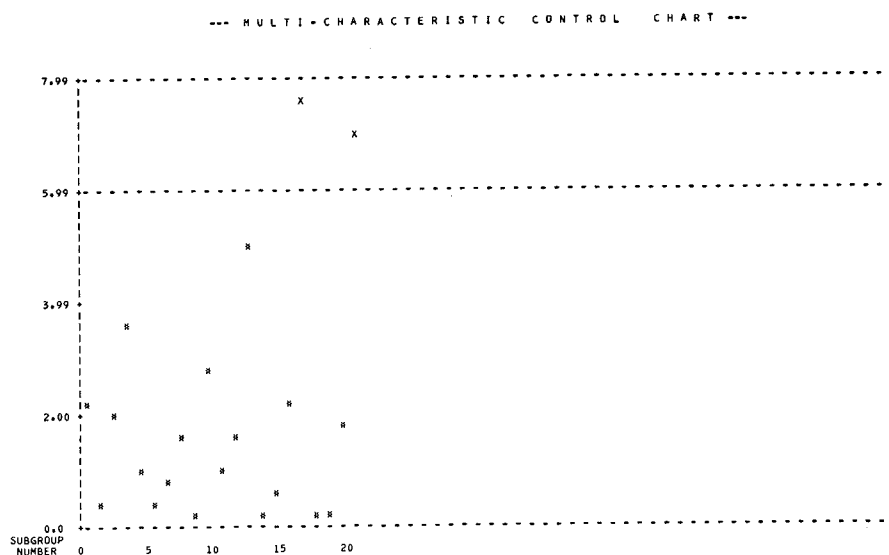
⑨ FIND FILE       5-10, 47, 15 /

⑩ END

```

データ・カード

〔表示 4-1〕 2 変量管理図の出力の一部



〔表示 4-2〕 コード変換の出力表示

```
*****
| RECODE |
*****
```

SAMPLES = 47

SCORE4

RECODE = 47
NON-RECODE = 0

X4

RECODE = 47
NON-RECODE = 0

ID

RECODE = 47
NON-RECODE = 0

ITEM	PRE-CODE	AFTER-CODE
SCORE4	35~40 40~42 42~44 OTHER	1 2 3 9
X4	0~2 3~4 5 6~10	1 2 3 4
ID	1001~1010 2001~2010 3001~4010 OTHER	1 2 3 4

--- MESSAGE ---

- (1) SYMBOL (8) MEANS NO APPEARED CODE.
(2) OVERLAPPING-CODE APPEAR IN (SCORE4)

3. 6 プログラム・リスト

***** MINERVA-SUBPROGRAM LIST *****

NO.	NAME	SEQ.	NO.	NAME	SEQ.
1	ACHECK	00898700	51	INVMAT	00512100
2	ADJUST	00285300	52	INVPRI	00906800
3	AMAIN	00695000	53	IODEV	00070800
4	ASSOCI	00636600	54	ITEM	00067800
5	BCHECK	00291700	55	ITQ	00127500
6	BDOWN1	00577600	56	LOGLIN	00323100
7	BDOWN2	00584100	57	MAIN	00020000
8	BDOWN3	00593300	58	MATOUT	00508700
9	BMUPP	00089700	59	MAXMAR	00340000
10	BRKDUL	00569700	60	MCCNCH	00548100
11	CALGFM	00530600	61	MCONTB	00340900
12	CARDXX	00786700	62	MEAN1	00504300
13	CATGS1	00739100	63	MINERV	00006300
14	CATGS2	00749200	64	MODEL	00370700
15	CATGYA	00803700	65	MSORT	00817400
16	CCCNT2	00273100	66	MSTEPW	00480100
17	CHANGE	00163300	67	MULTAB	00260500
18	CHIMC	00792800	68	NEWTAB	00240900
19	CHISQ	00295400	69	NORMAL	00415600
20	CLEAR	00563100	70	ONEWAY	00822000
21	COEFF	00670100	71	OPTIN	00075300
22	CULLAP	00299900	72	PATTRN	00243700
23	CUNBI	00456300	73	PERMUT	00416900
24	CONIF	00653500	74	PICT	00883300
25	CONS	00166400	75	PITEM	00129200
26	CON SIS	00189300	76	PLOT	00532500
27	CON TAB	00229800	77	PRIXX	00795900
28	CON TCH	00522600	78	PRTTAB	00419300
29	CORRLA	00722500	79	PTFILE	00639700
30	CORRLB	00734500	80	QCHART	00497100
31	COVMAT	00506000	81	RANGE	00800000
32	CPRINT	00754500	82	READK	00662100
33	CROPRI	00770900	83	RECERR	00941900
34	CROSSA	00809600	84	RECIK	00946700
35	CROSSB	00846900	85	RECOD	00909700
36	CRUSTB	00305500	86	RECODE	00924000
37	DATAPT	00099600	87	SBRANG	00133400
38	DATOUT	00564200	88	SIGINF	00880800
39	DECODE	00111700	89	STEPWS	00464600
40	DFIND	00621300	90	SUMTAB	00153800
41	DGFREE	00314300	91	TCROSS	00718100
42	EFFEC1	00438100	92	TWOWAY	00830400
43	EFFEC2	00449500	93	WRKARA	00559800
44	EFFEC3	00433000			
45	FINDOT	00630000			
46	FMA	00161800			
47	FORM	00130900			
48	IF THEN	00174500			
49	INITIA	00320400			
50	INPUT	00557400			

```

C
C
C ..... 00000100
C ..... 00000200
C ..... 00000300
C ..... 00000400
C ..... 00000500
C M M I I I I N N E E E E E E R R R R R R V V A 00000600
C M M M M I N N N E R R V V A A 00000700
C M M M M I N N N E R R V V A A 00000800
C M M M I N N N E E E E R R R R R R V V A A 00000900
C M M I N N N E R R V V A A A A A A 00001000
C M M I N N N E R R V A A 00001100
C M M I I I I N N E E E E E E R R V A A 00001200
C ..... 00001300
C ..... 00001400
C ..... 00001500
C ..... 00001600
C ..... 00001700
C ..... 00001800
C ..... 00001900
C ..... 00002000
C ..... 00002100
C ..... 00002200
C ..... 00002300
C ..... 00002400
C ..... 00002500
C ..... 00002600
C ..... 00002700
C ..... 00002800
C ..... 00002900
C ..... 00003000
C ..... 00003100
C ..... 00003200
C ..... 00003300
C ..... 00003400
C ..... 00003500
C ..... 00003600
C ..... 00003700
C ..... 00003800
C ..... 00003900
C THIS PACKAGE IS THE FIRST VERSION OF MINERVA. 00004000
C MINERVA IS AN ABBREVIATION TO " MINI-PACKAGE FOR EVALUATING AND 00004100
C RATING THE VALIDITY OF SURVEY DATA ", WHICH HAS BEEN MAINLY 00004200
C DEVELOPED BY THE STAFFS OF THE SIXTH DIVISION OF THE INSTITUTE OF 00004300
C STATISTICAL MATHEMATICS SINCE 1978. 00004400
C THE NAME MINERVA IS LIKEN TO THAT OF WELL-KNOWN GODDESS OF WISDOM. 00004500
C ALL OF SOURCE STATEMENTS ARE WRITTEN BY THE STANDARD FORTRAN. 00004600
C A PROTOTYPE OF MINERVA HAS ALREADY CONSTRUCTED AT DECEMBER IN 1978. 00004700
C AND THIS VERSION -01- HAS BEEN FORMALLY NOW PUBLISHED AND RELEASED 00004800
C AS A RESULT OF FURTHER IMPROVEMENT OF IT. SEVERAL STATISTICAL 00004900
C PROCEDURES INCLUDED IN MINERVA ARE USEFUL FOR SUMMARIZING THE 00005000
C INFORMATION ABOUT THE ASSESSMENT OF THE QUALITY OF DATA. 00005100
C IN PARTICULAR, BY USING THE MINERVA COMMAND LANGUAGE, IT IS 00005200
C AVAILABLE TO CARRY OUT THE CONSISTENCY CHECKING OF DATA-SET, THE 00005300
C VERIFICATION OR DETECTION OF THE WILD-CODED VALUES, AND THE ANALYSIS 00005400
C OF MULTIDIMENSIONAL CONTINGENCY TABLES BY THE LOG-LINEAR MODEL, THE 00005500
C CALCULATION OF VARIOUS MEASURES OF ASSOCIATION, AND SO ON. 00005600
C PROGRAMS IN MINERVA ARE DESIGNED AND CODED BY MINERVA WORKING-GROUP 00005700
C THAT CONSISTS OF N.OHSUMI, K.KATSURA, Y.YANAGISAWA, M.OHUCHI AND 00005800
C A.MURAISHI. 00005900
C ..... 00006000
C ..... 00006100
C ..... 00006200

```

	PROGRAM MINERVA	00006300
	COMMON / OUTIN / INPT, IOUT	00006400
	DIMENSION I(NNNNN), A(MMMMM)	00006500
C		00006600
	INPT=5	00006700
	IOUT=6	00006800
C		00006900
	WRITE(IOUT, 9001)	00007000
	WRITE(IOUT, 9200)	00007100
	WRITE(IOUT, 9300)	00007200
	WRITE(IOUT, 9400)	00007300
	WRITE(IOUT, 9000)	00007400
	WRITE(IOUT, 9400)	00007500
	WRITE(IOUT, 9300)	00007600
	WRITE(IOUT, 9200)	00007700
	WRITE(IOUT, 9100)	00007800
	WRITE(IOUT, 9200)	00007900
	WRITE(IOUT, 9150)	00008000
	WRITE(IOUT, 9200)	00008100
	WRITE(IOUT, 9500)	00008200
C		00008300
C	ALLOCATE THE WORK AREA	00008400
C		00008500
	LIMIT1=65000	00008600
	LIMITA=87000	00008700
	NN=250	00008800
	MA=50	00008900
	MX=50	00009000
	MY=50	00009100
	IA1=2	00009200
	IA2=650	00009300
	MB=MA*8	00009400
	KA1=20	00009500
	KA2=11	00009600
	NZ1=1030	00009700
	NNK=500	00009800
C		00009900
C	THE NEXT TWO VARIABLES ARE USED FOR MULTIWAY, Q-CHART, BREAK-DOWN	00010000
C		00010100
	LIMIT1=52000	00010200
	LIMIT2=40000	00010300
C		00010400
C	ESTABLISH INITIAL ALLOCATION BY USING THE ADJUSTABLE ARRAYS.	00010500
C		00010600
	I2=1	00010700
	I3=I2+IA2	00010800
	I4=I3+MY	00010900
	I5=I4+NN	00011000
	I6=I5+MB	00011100
	I7=I6+MA	00011200
	I8=I7+MA	00011300
	I9=I8+MX	00011400
	I10=I9+MX	00011500
	I11=I10+NN	00011600
	I12=I11+NN	00011700
	I13=I12+MX	00011800
	I14=I13+NZ1	00011900
	I18=I14+MY	00012000
	I19=I18+MY	00012100
	I20=I19+MY	00012200
	I21=I20+MY	00012300
	I22=I21+MY	00012400
	I23=I22+MY	00012500
	I24=I23+MY	00012600
	I25=I24+MB	00012700
	I26=I25+MB	00012800
	I27=I26+MB	00012900
	I28=I27+NNK	00013000
	I29=I28+NNK	00013100
	I30=I29+IA1*IA2	00013200
	I31=I30+MX	00013300

```

I32=I31+KA1                                00013400
I33=I32+KA1*KA2                            00013500
I38=I33+KA1*KA2                            00013600
IEND=I38+LIMIT2-1                          00013700
C                                             00013800
C                                             00013900
N1=1                                         00014000
N2=N1+MA                                    00014100
N3=N2+MX                                    00014200
N4=N3+MX                                    00014300
N8=N4+MY                                    00014400
N9=N8+MY                                    00014500
N10=N9+NN*MA                               00014600
N11=N10+KA1*KA2                           00014700
N12=N11+KA1*KA2                           00014800
N14=N12+KA1                                00014900
NEND=N14+LIMIT1-1                          00015000
C                                             00015100
WRITE(IOUT,600) IEND,NEND                  00015200
IF(IEND.GT.LIMIT1.OR.NEND.GT.LIMITA) GO TO 10 00015300
C                                             00015400
C BEGINNING OF MAIN JOB ( MINERVA JOB )     00015500
C                                             00015600
CALL MAIN(NN,MA,MX,MY,MB,IA1,IA2,LIMIT1,LIMIT2,KA1,KA2,NZ1,NNK,
*      I(12),I(13),I(14),I(15),I(16),I(17),I(18),I(19),I(110), 00015800
*      I(111),I(112),I(113),I(114),I(118),I(119),I(120),      00015900
*      I(121),I(122),I(123),I(124),I(125),I(126),I(127),I(128),00016000
*      I(129),I(130),I(131),I(132),I(133),I(138),            00016100
*      A(N1),A(N2),A(N3),A(N4),A(N8),A(N9),A(N10),A(N11),      00016200
*      A(N12),A(N14) )                                          00016300
C                                             00016400
STOP                                         00016500
10 WRITE(IOUT,610)                               00016600
STOP                                         00016700
C                                             00016800
600 FORMAT( 1H1,///1X,                                00016900
*      'USED WORKING AREA'/1X,17('-' )///                    00017000
*      5X,'INTEGER I =',I7//5X,'REAL A =',I7)                00017100
610 FORMAT(1H0,5X,'--- WORKING AREA OVER ---')              00017200
9000 FORMAT(5X,'M M I I I I N N EEEEEEE RRRRRR ',          00017300
*      ' V V A ',/00017400
*      5X,'MM MM I NN N E ' V R V R ',/00017500
*      5X,'M M M M I N N N E ' V R V R ',/00017600
*      5X,'M M M I N N N EEEEE RRRRRR ',/00017700
*      5X,'M M M I N N N EEEEE RRRRRR ',/00017800
*      5X,'M M I N N N E ' V R V A A ',/00017900
*      5X,'M M I N N N E ' V V AAAAAA ',/00018000
*      5X,'M M I N NN E ' R R A ',/00018100
*      5X,'M M I N NN E ' R R A ',/00018200
*      5X,'M M I I I I N N EEEEEEE R R ',/00018300
*      ' V A A ')00018400
9001 FORMAT( 1H1 )                                00018500
9100 FORMAT( 1H , 4X, 'MINI-PACKAGE FOR EVALUATING ',        00018600
*      'AND RATING THE VALIDITY OF SURVEY DATA' )           00018700
9150 FORMAT( 1H ,26X, ' ( VERSION - V01 ) ' ,///            00018800
*      29X, 'FEBRUARY 1980 ' )                               00018900
9180 FORMAT( /// )                                    00019000
9200 FORMAT( /// )                                    00019100
9300 FORMAT( 1H , 2X, 69( ':' ) )                    00019200
9400 FORMAT( // )                                    00019300
9500 FORMAT(1H ,11X,'--- THE INSTITUTE OF STATISTICAL MATHEMATICS ---',
*      //25X,'( THE SIXTH DIVISION )' )                    00019400
C                                             00019500
END                                             00019600
C                                             00019700
C                                             00019800
C                                             00019900

```

```

SUBROUTINE MAIN(NN,MA,MX,MY,MB,IA1,IA2,LIMIT1,LIMIT2,KA1,KA2,NZ1, 00020000
*      NNK,FMT,LIT,MCON,Q,MM,MP,MQ,NO,NOMIT,ICOL,MS,NS, 00020100
*      MG7,MQ8,RELAT7,RELAT8,LOGIC7,LOGIC8, 00020200
*      LOGICS,IFTAB7,IFTAB8,IFTABS,INCOA,INCOB,ITEMP, 00020300
*      MQH,MM4,MMM,MM3,IWORK, 00020400
*      XX,A,B,CONST7,CONST8,X,AM1,AM2,COFF,WORK ) 00020500
COMMON / OUTIN / INPT, IOUT 00020600
COMMON NUMDT(20) 00020700
INTEGER AA(100,80),ABC(26),Q(MB),RELAT7(MY),RELAT8(MY),FMT(IA2),00020800
*      DFMT(4),DITEM(4),DRANG(4),DCONS(4),DINPU(4), 00020900
*      DOUTP(4),COMMA,BLANK,SLASH,OUTDEV,DEND(3),DPATT(4), 00021000
*      DMULT(4),DOPTI(4),OPT(10),DQCHA(4),DBMDP(4),DASSO(4), 00021100
*      FINI(4),BREA(4),FIND(4),DREAD(4),DRECO(4),DCROS(4) 00021200
DIMENSION NUM(10),NUMB(10),MM(MA),MP(MA),MQ(MX),NO(MX),NOMIT(NN), 00021300
*      ICOL(NN),MS(MX),NS(NZ1),XX(MA),A(MX),B(MX), 00021400
*      MQ7(MY),LOGIC7(MY),IFTAB7(MB),INCOA(NNK),LOGICS(MY), 00021500
*      MQ8(MY),LOGIC8(MY),IFTAB8(MB),INCOB(NNK),IFTABS(MB), 00021600
*      ITEMP(IA1,IA2),MCON(NN),LIT(MY),COFF(KA1),MQH(MX), 00021700
*      CONST7(MY),CONST8(MY),X(NN,MA),NOF(20),NFILE(20,15), 00021800
*      AM1(KA1,KA2),AM2(KA1,KA2),MM3(KA1,KA2),MM4(KA1), 00021900
*      MMM(KA1,KA2),WORK(LIMIT1),IWORK(LIMIT2) 00022000
DATA DFMT/'F','D','R','M'//,DITEM/'I','T','E','M'//, 00022100
*      DRANG/'R','A','N','G'//,COMMA/' ',' ','BLANK/' ','SLASH/' '/'/, 00022200
*      DCONS/'C','O','N','S'//,DINPU/'I','N','P','U'//, 00022300
*      DOUTP/'O','U','T','P'//,DEND/'E','N','D'//, 00022400
*      DPATT/'P','A','T','T'//,DMULT/'M','U','L','T'//, 00022500
*      DOPTI/'O','P','T','I'//,DQCHA/'Q','C','H','A'//, 00022600
*      DBMDP/'B','M','D','P'//,FINI/'F','I','N','I'//, 00022700
*      BREA/'B','R','E','A'//,FIND/'F','I','N','D'// 00022800
DATA DREAD/'R','E','A','D'//,DRECO/'R','E','C','O'// 00022900
DATA DASSO/'A','S','S','O'//,DCROS/'C','R','O','S'// 00023000
DATA NUM/'0','1','2','3','4','5','6','7','8','9'//, 00023100
*      ABC/'A','B','C','D','E','F','G','H','I','J','K','L','M','N', 00023200
*      'O','P','Q','R','S','T','U','V','W','X','Y','Z'// 00023300
C 00023400
INDEV=11 00023500
OUTDEV=0 00023600
ID=4 00023700
IX=16 00023800
IY=80 00023900
KN=10 00024000
KA=26 00024100
J=0 00024200
IJK=1 00024300
IFI=0 00024400
KNEW=9 00024500
REC=0 00024600
C 00024700
DO 28 I=1,20 00024800
NUMDT(I)=0 00024900
NOF(I)=0 00025000
DO 28 J=1,15 00025100
NFILE(I,J)=0 00025200
28 CONTINUE 00025300
C 00025400
DO 18 I=1,IA2 00025500
18 FMT(I)=BLANK 00025600
WRITE(IOUT,600) 00025700
C 00025800
K=0 00025900
99 K=K+1 00026000
READ(INPT,500) (AA(K,I),I=1,80) 00026100
WRITE(IOUT,610) (AA(K,I),I=1,80) 00026200
DO 100 I=1,3 00026300
IF(AA(K,I).NE.DEND(I)) GO TO 99 00026400
100 CONTINUE 00026500
IF(AA(K,4).EQ.SLASH) GO TO 99 00026600

```

C		00026700
C		00026800
	DO 200 II=1,K	00026900
C		00027000
	IPUNCH=0	00027100
	IERG=0	00027200
	DO 52 I=1,80	00027300
	IF(AA(II,I).NE.BLANK) GO TO 53	00027400
52	CONTINUE	00027500
	GO TO 200	00027600
53	CONTINUE	00027700
C		00027800
	IF(IJK.EQ.8) GO TO 8	00027900
	DO 21 I=1,15	00028000
	IF(AA(II,I).NE.BLANK) GO TO 22	00028100
21	CONTINUE	00028200
	LD=II-1	00028300
	DO 15 I=1,4	00028400
	IF(AA(LD,I).NE.DOPTI(I)) GO TO 131	00028500
15	CONTINUE	00028600
	GO TO 200	00028700
131	GO TO (1,2,3,4,5,6,7,8,9,10,11,124,130,140),IJK	00028800
C		00028900
	22 DO 17 I=1,3	00029000
	IF(DEND(I).NE.AA(II,I)) GO TO 16	00029100
17	CONTINUE	00029200
	GO TO 111	00029300
C		00029400
	16 DO 19 I=1,ID	00029500
	IF(DFMT(I).NE.AA(II,I)) GO TO 20	00029600
19	CONTINUE	00029700
C		00029800
	1 DO 30 I=IX,IY	00029900
	IF(AA(II,I).EQ.BLANK) GO TO 30	00030000
	J=J+1	00030100
	FMT(J)=AA(II,I)	00030200
30	CONTINUE	00030300
	IJK=1	00030400
	GO TO 200	00030500
C		00030600
	20 DO 40 I=1,ID	00030700
	IF(DITEM(I).NE.AA(II,I)) GO TO 50	00030800
40	CONTINUE	00030900
C		00031000
	M=0	00031100
	M1=0	00031200
C		00031300
	2 CONTINUE	00031400
	CALL ITEM(AA,Q,MM,IY,MB,MA,M,M1,IX,COMMA,BLANK,SLASH,II)	00031500
	IJK=2	00031600
	GO TO 77	00031700
C		00031800
	50 CONTINUE	00031900
	DO 90 I=1,ID	00032000
	IF(DINPU(I).NE.AA(II,I)) GO TO 95	00032100
90	CONTINUE	00032200
	CALL IODEV(AA,NUM,KN,II,INDEV,BLANK,NUMB)	00032300
C		00032400
	GO TO 200	00032500
C		00032600
	95 DO 60 I=1,ID	00032700
	IF(DRANG(I).NE.AA(II,I)) GO TO 75	00032800
60	CONTINUE	00032900
C		00033000
	MQ1=0	00033100
	ITWO=0	00033200
	ISIDE=0	00033300
	IBY=0	00033400
	MQ2=0	00033500
C		00033600
	3 CONTINUE	00033700

	CALL DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE,IBY,	00033800
	* MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA,BLANK,SLASH)	00033900
	IJK=3	00034000
	GO TO 77	00034100
C		00034200
	75 CONTINUE	00034300
	DO 80 I=1,ID	00034400
	IF(DCONS(I).NE.AA(II,I)) GO TO 85	00034500
	80 CONTINUE	00034600
C		00034700
	IFFSW=0	00034800
	NCON=0	00034900
	IT=0	00035000
	IS=0	00035100
	MO7=0	00035200
	MO8=0	00035300
	MP7=0	00035400
	MP8=0	00035500
	LL1=0	00035600
	LL2=0	00035700
C		00035800
	KUNT=0	00035900
	4 CONTINUE	00036000
	IX=16	00036100
	CALL CONS(AA,Q,MM,ABC,NUM,MQ7,RELAT7,CONST7,LOGIC7,IFTAB7,	00036200
	* II,NUMB,MQ8,RELAT8,CONST8,LOGIC8,IFTAB8,IY,MB,MA,KA,KN,	00036300
	* MY,M1,IX,NCON,IT,IS,MU7,MO8,KUNT,MP7,MP8,LL1,LL2,IFFSW)	00036400
	IJK=4	00036500
	IX=16	00036600
	GO TO 77	00036700
C		00036800
	85 CONTINUE	00036900
	DO 24 I=1,ID	00037000
	IF(DPATT(I).NE.AA(II,I)) GO TO 31	00037100
	24 CONTINUE	00037200
C		00037300
	MQ1=0	00037400
	ITWO=0	00037500
	ISIDE=0	00037600
	IBY=0	00037700
	MQ2=0	00037800
	5 CONTINUE	00037900
	CALL DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE,IBY,	00038000
	* MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA,BLANK,SLASH)	00038100
	IJK=5	00038200
	GO TO 77	00038300
C		00038400
	31 CONTINUE	00038500
	DO 36 I=1,ID	00038600
	IF(DMULT(I).NE.AA(II,I)) GO TO 32	00038700
	36 CONTINUE	00038800
C		00038900
	MQ1=0	00039000
	ITWO=0	00039100
	ISIDE=0	00039200
	IBY=0	00039300
	MQ2=0	00039400
	6 CONTINUE	00039500
	CALL DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE,IBY,	00039600
	* MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA,BLANK,SLASH)	00039700
	IJK=6	00039800
	GO TO 77	00039900
C		00040000
	32 CONTINUE	00040100
	DO 37 I=1,ID	00040200
	IF(DQCHA(I).NE.AA(II,I)) GO TO 33	00040300
	37 CONTINUE	00040400
C		00040500
	MQ1=0	00040600
	ITWO=0	00040700
	ISIDE=0	00040800

	IBY=0	00040900
	MQ2=0	00041000
	7 CONTINUE	00041100
	CALL DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE,IBY,	00041200
	* MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA,BLANK,SLASH)	00041300
	IJK=7	00041400
	GO TO 77	00041500
C		00041600
	33 CONTINUE	00041700
	DO 46 I=1,ID	00041800
	IF(DBMDP(I).NE.AA(II,I)) GO TO 48	00041900
	46 CONTINUE	00042000
	IJK=8	00042100
	IBM=11	00042200
	8 CONTINUE	00042300
	DO 47 I=1,ID	00042400
	IF(FINI(I).NE.AA(II,I)) GO TO 200	00042500
	47 CONTINUE	00042600
	CALL BMDPP(AA,IBM,II,BLANK)	00042700
	IJK=1	00042800
	GO TO 200	00042900
C		00043000
	48 CONTINUE	00043100
	DO 49 I=1,ID	00043200
	IF(BREA(I).NE.AA(II,I)) GO TO 121	00043300
	49 CONTINUE	00043400
C		00043500
	MQ1=0	00043600
	ITWO=0	00043700
	ISIDE=0	00043800
	IBY=0	00043900
	MQ2=0	00044000
	9 CONTINUE	00044100
	CALL DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE,IBY,	00044200
	* MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA,BLANK,SLASH)	00044300
	IJK=9	00044400
	GO TO 77	00044500
	121 CONTINUE	00044600
	DO 51 I=1,ID	00044700
	IF(FIND(I).NE.AA(II,I)) GO TO 122	00044800
	51 CONTINUE	00044900
C		00045000
	MQ1=0	00045100
	10 CONTINUE	00045200
	CALL DFIND(AA,II,MCON,MQ1,NN,NUM,NUMB,COMMA,BLANK,SLASH)	00045300
	IJK=10	00045400
	GO TO 77	00045500
C		00045600
	122 CONTINUE	00045700
	DO 54 I=1,ID	00045800
	IF(DREAD(I).NE.AA(II,I)) GO TO 126	00045900
	54 CONTINUE	00046000
C		00046100
	CALL READK(AA,II,NUM,KN,KNEW)	00046200
	GO TO 77	00046300
C		00046400
	126 CONTINUE	00046500
	DO 55 I=1,ID	00046600
	IF(DRECO(I).NE.AA(II,I)) GO TO 127	00046700
	55 CONTINUE	00046800
C		00046900
	MQ1=0	00047000
	11 CONTINUE	00047100
C		00047200
	CALL RECOD(AA,II,BLANK,COMMA,SLASH,NUM,Q,MQ1,MQ,MM,M1,KN,KA1,KA2,	00047300
	* MB,MX,MA,ABC,KA,AM1,AM2,MM3,MM4,NUMB)	00047400
C		00047500
	IJK=11	00047600
	GO TO 77	00047700
C		00047800
	127 CONTINUE	00047900

	DO 56 I=1,1D	00048000
	IF(DASSO(I).NE.AA(II,I)) GO TO 129	00048100
	56 CONTINUE	00048200
C		00048300
	MQ1=0	00048400
	ITWO=0	00048500
	ISIDE=0	00048600
	IBY=0	00048700
	MQ2=0	00048800
	130 CONTINUE	00048900
	CALL DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE,IBY,	00049000
	* MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA,BLANK,SLASH)	00049100
	IJK=13	00049200
	GO TO 77	00049300
C		00049400
	129 CONTINUE	00049500
	DO 61 I=1,1D	00049600
	IF(DCROS(I).NE.AA(II,I)) GO TO 62	00049700
	61 CONTINUE	00049800
C		00049900
	MQ1=0	00050000
	ITWO=0	00050100
	ISIDE=0	00050200
	IBY=0	00050300
	MQ2=0	00050400
	140 CONTINUE	00050500
	CALL DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE,IBY,	00050600
	* MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA,BLANK,SLASH)	00050700
	IJK=14	00050800
	GO TO 77	00050900
C		00051000
	62 IERO=1	00051100
C		00051200
	77 CONTINUE	00051300
C		00051400
	IF(IJK.LE.2) GO TO 410	00051500
	IF(IJK.EQ.4.AND.IT.LE.0) GO TO 123	00051600
	IF(IJK.NE.4.AND.MQ1.LE.0) GO TO 123	00051700
	410 CONTINUE	00051800
	DO 23 I=1,80	00051900
	IF(AA(II,I).EQ.SLASH) GO TO 88	00052000
	23 CONTINUE	00052100
	GO TO 200	00052200
C		00052300
	88 CONTINUE	00052400
	DO 133 I=1,4	00052500
	IF(AA(II,I).NE.DOPTI(I)) GO TO 132	00052600
	133 CONTINUE	00052700
	GO TO 200	00052800
C		00052900
	132 CONTINUE	00053000
C		00053100
	IF(IJK.LE.2.OR.IJK.EQ.10) GO TO 29	00053200
	IF(IJK.GT.14) GO TO 29	00053300
	NOF(INDEV)=NOF(INDEV)+1	00053400
	NF=NOF(INDEV)	00053500
	IF(NF.GT.15.OR.INDEV.GT.20) GO TO 29	00053600
	NFILE(INDEV,NF)=IJK	00053700
	29 CONTINUE	00053800
C		00053900
	IF(IJK.EQ.10) IFI=1	00054000
C		00054100
	IOP=0	00054200
	DO 91 I=1,15	00054300
	COFF(I)=0	00054400
	IF(I.LE.10) OPT(I)=0	00054500
	91 CONTINUE	00054600
C		00054700
	J1=II+1	00054800
	J2=II+2	00054900
	MC=1	00055000

	COFF(1)=6	00055100
	KJI=0	00055200
	KMZW=0	00055300
	NN1=1	00055400
	MM1=0	00055500
	MM2=0	00055600
C		00055700
	DO 300 JJ=J1,J2	00055800
	DO 96 I=1,ID	00055900
	IF(DOUTP(I).NE.AA(JJ,I)) GO TO 97	00056000
96	CONTINUE	00056100
	IERO=0	00056200
	CALL IODEV(AA,NUM,KN,JJ,OUTDEV,BLANK,NUMB)	00056300
	GO TO 300	00056400
97	CONTINUE	00056500
	IF(KJI.EQ.1) GO TO 14	00056600
C		00056700
C		00056800
	DO 13 I=1,ID	00056900
	IF(DOPTI(I).NE.AA(JJ,I)) GO TO 300	00057000
13	CONTINUE	00057100
	MC=0	00057200
	IERO=0	00057300
14	CALL OPTIN (AA,NUM,NUMB,ABC,OPT,IOP,JJ,BLANK,CUMMA,COFF,MC,IERO,	00057400
	* KJI,IPUNCH,KMZW,NN1,MM1,MM2,KA1)	00057500
300	CONTINUE	00057600
C		00057700
	IF(IERO.EQ.1) GO TO 123	00057800
C		00057900
C		00058000
C		00058100
	GO TO (111,222,333,444,555,666,777,111,999,1010,1020,111,1030,	00058200
	* 1040),IJK	00058300
	GO TO 111	00058400
C		00058500
222	CONTINUE	00058600
	CALL DATAPT(XX,Q,MM,MP,ITEMP,FMT,KNEW,MA,MB,IA1,IA2,M1,N,BLANK)	00058700
	IF(KNEW.NE.99) GO TO 200	00058800
	RETURN	00058900
C		00059000
333	CONTINUE	00059100
	CALL RANGE(XX,Q,MM,MQ,NO,NOMIT,ICOL,MS,NS,A,B,NUM,MA,MB,MX,NN,	00059200
	* KN,N,M1,MQ1,BLANK,INDEV,OUTDEV,WORK,IWORK,LIMIT1,	00059300
	* LIMIT2)	00059400
	GO TO 199	00059500
C		00059600
444	CONTINUE	00059700
	CALL CONSIS(XX,Q,MM,ITEMP,NUM,MA,MB,MY,NNK,KN,IA1,IA2,	00059800
	* MQ7,RELAT7,CONST7,LOGIC7,IFTAB7,INCDA,LOGICS,	00059900
	* MQ8,RELAT8,CONST8,LOGIC8,IFTAB8,INCDB,IFTABS,	00060000
	* N,M1,IT,IS,M07,M08,MP7,MP8,IFFSW,BLANK,LIT,KUNT,	00060100
	* INDEV,OUTDEV)	00060200
	GO TO 199	00060300
C		00060400
555	CONTINUE	00060500
	KWORK=1024	00060600
	CALL PATTRN(XX,MQ,WORK(1),NS,Q,MM,ITEMP,NUM,N,KWORK,NZ1,MA,MB,	00060700
	* IA1,IA2,MQ1,M1,INDEV,BLANK)	00060800
	GO TO 199	00060900
C		00061000
666	CONTINUE	00061100
C		00061200
C		00061300
	CALL MULTAB(WORK,IWORK,LIMIT1,LIMIT2,OPT,Q,MM,MB,XX,MQ,AA,II,MQ1,	00061400
	* M1,MA,MX,INDEV)	00061500
	GO TO 199	00061600
C		00061700
777	CONTINUE	00061800
	CALL QCHART(WORK,IWORK,XX,MQ,MQ1,M1,MA,MX,INDEV,LIMIT1,LIMIT2)	00061900
	GO TO 199	00062000

C	999	CONTINUE	00062100
		CALL BRKDUL(WORK,IWORK,LIMIT1,LIMIT2,AA,XX,MQ,N,MQ1,M1,MA,MX,	00062200
	*	INDEV,II)	00062300
		GO TO 199	00062400
C	1010	CONTINUE	00062500
		CALL FINDDT(XX, Q, MM, X, MCON, MQ1, M1, MA, MB, NN, INDEV)	00062600
		GO TO 199	00062700
C	1020	CONTINUE	00062800
		CALL RECODE(XX,MQ,INDEV,OUTDEV,AM1,AM2,MM3,MM4,MQ1,N,MA,M1,	00062900
	*	MX,KA1,KA2,MM,Q,MB,NUM,KN,MMM,REC)	00063000
		GO TO 199	00063100
C	1030	CONTINUE	00063200
		CALL ASSOCI(XX,MA,MQ,MX,M1,INDEV,BLANK,Q,MM,MB,WORK,LIMIT1,	00063300
	*	NUM,KN,COFF,MC,MQ1,IPUNCH,KA1)	00063400
		GO TO 199	00063500
C	1040	CONTINUE	00063600
		CALL CROSSA(XX,MA,MQ,MQ1,MQH,MQ2,MM,Q,MX,MB,WORK,IWORK,LIMIT1,	00063700
	*	LIMIT2,INDEV,M1,ITWO,KMZ,MM1,MM2,NN1,NUM,KN)	00063800
C	199	CONTINUE	00063900
		IF(OUTDEV.NE.0) REWIND OUTDEV	00064000
C		IF(REC.EQ.1) GO TO 98	00064100
		INDEV=11	00064200
		OUTDEV=0	00064300
		GO TO 200	00064400
C	98	INDEV=10	00064500
		OUTDEV=0	00064600
		GO TO 200	00064700
C	123	CONTINUE	00064800
		WRITE(IOUT,699)	00064900
	124	WRITE(IOUT,700) (AA(II,I),I=1,80)	00065000
		IJK=12	00065100
C	200	CONTINUE	00065200
			00065300
C			00065400
			00065500
C	111	CONTINUE	00065600
		CALL PTFILE(NOF,NFILE,DRANG,DCONS,DPATT,FIND,IFI,DMULT,DQCHA,	00065700
	*	BREA,BLANK,DRECO,DASSO,DCROS)	00065800
		RETURN	00065900
C	500	FORMAT(80A1)	00066000
	600	FORMAT(1H1,///1X,'INPUT INFORMATION BY CARD IMAGE'/1X,31('-')//)	00066100
	610	FORMAT(1H,5X,80A1)	00066200
	699	FORMAT(1H1,///1X,'--- INPUT COMMAND CARD IS INVALID ---')	00066300
	700	FORMAT(1H,5X,80A1)	00066400
		END	00066500
			00066600
			00066700
			00066800
			00066900
			00067000
			00067100
			00067200
			00067300
			00067400
			00067500
			00067600
			00067700

	SUBROUTINE ITEM(AA,Q,MM,IY,MB,MA,M,M1,IX,COMMA,BLANK,SLASH,II)	00067800
	INTEGER AA(100,80),Q(MB),COMMA,BLANK,SLASH	00067900
	DIMENSION MM(MA)	00068000
C		00068100
C	THIS SUBROUTINE OBSERVES AND DECODES THE ITEM COMMAND .	00068200
C		00068300
	IBC=0	00068400
	K=0	00068500
	DO 10 I=IX,IY	00068600
	IF(AA(II,I).EQ. SLASH) GO TO 40	00068700
	IF(AA(II,I).EQ.COMMA.OR.AA(II,I).EQ.BLANK) GO TO 20	00068800
	K=K+1	00068900
	GO TO 10	00069000
40	IBC=1	00069100
	IB=I-1	00069200
	IF(AA(II,IB).EQ. BLANK .OR. AA(II,IB).EQ. COMMA) RETURN	00069300
20	IF(K.EQ.0) GO TO 10	00069400
	K1=I-K	00069500
	K2=I-1	00069600
	DO 30 J=K1,K2	00069700
	M=M+1	00069800
30	Q(M)=AA(II,J)	00069900
	M1=M1+1	00070000
	MM(M1)=K	00070100
	K=0	00070200
	IF(IBC .EQ. 1) RETURN	00070300
10	CONTINUE	00070400
C		00070500
	RETURN	00070600
	END	00070700
	SUBROUTINE IODEV(AA,NUM,KN,II,IO,BLANK,NUMB)	00070800
	COMMON / OUTIN / INPT, IOUT	00070900
	INTEGER AA(100,80),BLANK	00071000
	DIMENSION NUM(KN),NUMB(KN)	00071100
C		00071200
	K=0	00071300
	DO 10 I=16,80	00071400
	IF(AA(II,I).EQ.BLANK) GO TO 25	00071500
C		00071600
	DO 20 J=1,10	00071700
	IF(AA(II,I).EQ.NUM(J)) GO TO 22	00071800
20	CONTINUE	00071900
	GO TO 70	00072000
22	K=K+1	00072100
	GO TO 10	00072200
25	IF(K.NE.0) GO TO 26	00072300
C		00072400
10	CONTINUE	00072500
C		00072600
26	CONTINUE	00072700
	K1=I-K	00072800
	K2=I-1	00072900
C		00073000
	IA=0	00073100
	DO 30 J=K1,K2	00073200
	IA=IA+1	00073300
	DO 40 JN=1,10	00073400
	IF(AA(II,J).EQ.NUM(JN)) GO TO 50	00073500
40	CONTINUE	00073600
50	JN=JN+1	00073700
	NUMB(IA)=JN	00073800
30	CONTINUE	00073900
C		00074000
	JA=0	00074100
	DO 60 JJ=1,IA	00074200
	J1=IA-JJ	00074300
60	JA=JA+NUMB(JJ)*10**J1	00074400
	IO=JA	00074500
	RETURN	00074600
C		00074700
70	WRITE(IOUT,600) (AA(II,J),J=1,80)	00074800
600	FORMAT(1H1,///1X,'--- INPUT OR OUTPUT DEVICE CODE WAS MISUSED, ',	00074900
	* 'THUS CHECK AND SPECIFY AGAIN' / 5X,80A1)	00075000
	RETURN	00075100
	END	00075200

```

SUBROUTINE OPTIN(AA,NUM,NUMB,ABC,OPT,IOP,II,BLANK,COMMA,COFF,MC, 00075300
* IERO,KJI,IPUNCH,KMZW,NN1,MM1,MM2,KA1 ) 00075400
C 00075500
C THIS ROUTINE A DECODER OF OPTION COMMAND. 00075600
C 00075700
COMMON / OUTIN / INPT, IOUT 00075800
INTEGER AA(100,80),ABC(26),OPT(10),COMB(4),DUAL(4),COEF(4), 00075900
* PRIN(4),STEP(4),BLANK,CUMMA,EQUAL,PUNC(4),LETT(4), 00076000
* DMARG(4),ENTR(4),SLASH 00076100
* DIMENSION NUM(10),NUMB(10),COFF(KA1) 00076200
DATA PRIN/'P','R','I','N'//, STEP/'S','T','E','P'//, EQUAL/'='//, 00076300
* COMB/'C','O','M','B'//, DUAL/'D','U','A','L'//, SLASH/'/'//, 00076400
* COEF/'C','O','E','F'//, LETT/'L','E','T','T'//, 00076500
* DMARG/'M','A','R','G'//, ENTR/'E','N','T','R'//, 00076600
* PUNC/'P','U','N','C'// 00076700
C 00076800
K=0 00076900
C 00077000
IF(KJI.EQ.1) GO TO 99 00077100
C 00077200
DO 10 I=16,80 00077300
C 00077400
IF(AA(II,I).EQ.BLANK.OR.AA(II,I).EQ.EQUAL 00077500
* .OR.AA(II,I).EQ.SLASH 00077600
* .OR.AA(II,I).EQ.COMMA) GO TO 20 00077700
K=K+1 00077800
GO TO 10 00077900
C 00078000
20 IF(K.EQ.0) GO TO 10 00078100
K1=I-K 00078200
K2=I-1 00078300
C 00078400
DO 25 J=1,26 00078500
IF(AA(II,K1).EQ.ABC(J)) GO TO 35 00078600
25 CONTINUE 00078700
C 00078800
IA=0 00078900
DO 30 J=K1,K2 00079000
IA=IA+1 00079100
DO 40 JN=1,10 00079200
IF(AA(II,J).EQ.NUM(JN)) GO TO 50 00079300
40 CONTINUE 00079400
50 JN=JN-1 00079500
NUMB(IA)=JN 00079600
30 CONTINUE 00079700
C 00079800
JA=0 00079900
DO 60 JJ=1,IA 00080000
J1=IA-JJ 00080100
60 JA=JA+NUMB(JJ)*10**J1 00080200
OPT(IOP)=JA 00080300
K=0 00080400
KMZW=JA 00080500
GO TO 10 00080600
C 00080700
35 L=K1 00080800
DO 45 J=1,4 00080900
IF(PRIN(J).NE.AA(II,L)) GO TO 55 00081000
45 L=L+1 00081100
IOP=1 00081200
K=0 00081300
GO TO 10 00081400
C 00081500
55 L=K1 00081600
DO 65 J=1,4 00081700
IF(STEP(J).NE.AA(II,L)) GO TO 70 00081800
65 L=L+1 00081900

```

	IOP=2	00082000
	OPT(IOP)=4	00082100
	K=0	00082200
	GO TO 10	00082300
C		00082400
70	L=K1	00082500
	DO 75 J=1,4	00082600
	IF(COMB(J).NE.AA(II,L)) GO TO 80	00082700
75	L=L+1	00082800
	IOP=2	00082900
	OPT(IOP)=3	00083000
	K=0	00083100
	GO TO 10	00083200
C		00083300
80	L=K1	00083400
	DO 31 J=1,4	00083500
	IF(LETT(J).NE.AA(II,L)) GO TO 32	00083600
31	L=L+1	00083700
	NN1=0	00083800
	K=0	00083900
	GO TO 10	00084000
C		00084100
32	L=K1	00084200
	DO 33 J=1,4	00084300
	IF(DMARG(J).NE.AA(II,L)) GO TO 34	00084400
33	L=L+1	00084500
	MM1=1	00084600
	K=0	00084700
	GO TO 10	00084800
C		00084900
34	L=K1	00085000
	DO 36 J=1,4	00085100
	IF(ENTR(J).NE.AA(II,L)) GO TO 37	00085200
36	L=L+1	00085300
	MM2=1	00085400
	K=0	00085500
	GO TO 10	00085600
C		00085700
37	L=K1	00085800
	DO 85 J=1,4	00085900
	IF(DUAL(J).NE.AA(II,L)) GO TO 90	00086000
85	L=L+1	00086100
	IOP=11	00086200
	RETURN	00086300
C		00086400
90	L=K1	00086500
	DO 91 J=1,4	00086600
	IF(PUNC(J).NE.AA(II,L)) GO TO 92	00086700
91	L=L+1	00086800
	IPUNCH=1	00086900
	K=0	00087000
	GO TO 10	00087100
C		00087200
92	L=K1	00087300
	DO 96 J=1,4	00087400
	IF(COEF(J).NE.AA(II,L)) GO TO 95	00087500
96	L=L+1	00087600
C		00087700
99	CONTINUE	00087800
	KK=K2+1	00087900
	IF(KJ1.EQ.1) KK=16	00088000
	CALL COEFF (AA,BLANK,COMMA,II,COFF,IERO,MC,ABC,KK,KJ1,IPUNCH,	00088100
	* KA1)	00088200
C		00088300
	IF(IERO.EQ.1) GO TO 95	00088400
	RETURN	00088500
C		00088600
10	CONTINUE	00088700
C		00088800
	RETURN	00088900
C		00089000
95	CONTINUE	00089100
	WRITE(IOUT,600) (AA(II,J),J=1,80)	00089200
600	FORMAT(IH1,///IX,'--- THIS COMMAND WAS MISUSED ---' / 5X,80A1)	00089300
C		00089400
	RETURN	00089500
	END	00089600

	SUBROUTINE BMDPP(AA,IBM,I1,BLANK)	00089700
C		00089800
C	THIS SUBROUTINE DECODES THE BMDP COMMAND.	00089900
C		00090000
	COMMON / OUTIN / INPT, IOUT	00090100
	INTEGER AA(100,80),BLANK,PROG,	00090200
	* P1D(3), P2D(3), P4D(3), P5D(3), P6D(3),	00090300
	* P1F(3), P2F(3), P3F(3)	00090400
	DATA P1D/'P','1','D'/, P2D/'P','2','D'/, P4D/'P','4','D'/,	00090500
	* P5D/'P','5','D'/, P6D/'P','6','D'/, P1F/'P','1','F'/,	00090600
	* P2F/'P','2','F'/, P3F/'P','3','F'/'	00090700
C		00090800
	K=0	00090900
C		00091000
	DO 10 I=1,80	00091100
	IF(AA(IBM,I).EQ.BLANK) GO TO 11	00091200
	K=K+1	00091300
	GO TO 10	00091400
	11 IF(K.EQ.0) GO TO 10	00091500
	K1=I-K	00091600
	K2=I-1	00091700
	GO TO 12	00091800
	10 CONTINUE	00091900
C		00092000
	12 CONTINUE	00092100
	IF(K2-K1.EQ.2) GO TO 13	00092200
	WRITE(IOUT,600) (AA(IBM,I),I=1,80)	00092300
	RETURN	00092400
C		00092500
	13 L=1	00092600
	DO 20 I=K1,K2	00092700
	IF(P1D(L).NE.AA(IBM,I)) GO TO 21	00092800
	L=L+1	00092900
	PROG=21	00093000
	GO TO 40	00093100
C		00093200
	21 L=1	00093300
	DO 22 I=K1,K2	00093400
	IF(P2D(L).NE.AA(IBM,I)) GO TO 23	00093500
	22 L=L+1	00093600
	PROG=22	00093700
	GO TO 40	00093800
C		00093900
	23 L=1	00094000
	DO 24 I=K1,K2	00094100
	IF(P4D(L).NE.AA(IBM,I)) GO TO 25	00094200
	24 L=L+1	00094300
	PROG=23	00094400
	GO TO 40	00094500
C		00094600
	25 L=1	00094700
	DO 26 I=K1,K2	00094800
	IF(P5D(L).NE.AA(IBM,I)) GO TO 27	00094900
	26 L=L+1	00095000
	PROG=24	00095100
	GO TO 40	00095200
C		00095300
	27 L=1	00095400
	DO 28 I=K1,K2	00095500
	IF(P6D(L).NE.AA(IBM,I)) GO TO 29	00095600
	28 L=L+1	00095700
	PROG=25	00095800
	GO TO 40	00095900
C		00096000
	29 L=1	00096100
	DO 30 I=K1,K2	00096200
	IF(P1F(L).NE.AA(IBM,I)) GO TO 31	00096300
	30 L=L+1	00096400
	PROG=26	00096500
	GO TO 40	00096600

C		00096700
	31 L=1	00096800
	DO 32 I=K1,K2	00096900
	IF(P2F(L).NE.AA(IBM,I)) GO TO 33	00097000
	32 L=L+1	00097100
	PROG=27	00097200
	GO TO 40	00097300
C		00097400
	33 L=1	00097500
	DO 34 I=K1,K2	00097600
	IF(P3F(L).NE.AA(IBM,I)) GO TO 35	00097700
	34 L=L+1	00097800
	PROG=28	00097900
	GO TO 40	00098000
C		00098100
	35 WRITE(IOUT,600) (AA(IBM,I),I=1,80)	00098200
	RETURN	00098300
C		00098400
	40 CONTINUE	00098500
	JJ=IBM+1	00098600
	DO 50 I=JJ,II	00098700
	WRITE(PROG,610) (AA(I,J),J=1,80)	00098800
	50 CONTINUE	00098900
C		00099000
	600 FORMAT(1H1,///1X,'--- THIS COMMAND WAS MISUSED ---' / 5X,80A1)	00099100
	610 FORMAT(80A1)	00099200
C		00099300
	RETURN	00099400
	END	00099500

	SUBROUTINE DATAPT(XX,Q,MM,MP,ITEMP,FMT,KNEW,MA,MB,IA1,IA2,	00099600
	* M1,N,BLANK)	00099700
C		00099800
C	THIS SUBROUTINE EXECUTES INPUT-DATA PRINTING.	00099900
C		00100000
	COMMON / OUTIN / INPT, IOUT	00100100
	COMMON NUMDT(20)	00100200
	INTEGER Q(MB),FMT(IA2),BLANK	00100300
	DIMENSION XX(MA),MM(MA),MP(MA),ITEMP(IA1,IA2),YY(50,40)	00100400
C		00100500
	IF(KNEW.EQ.99) GO TO 999	00100600
	KOLD=11	00100700
C		00100800
	WRITE(IOUT,605)	00100900
C		00101000
	DO 70 I=1,M1	00101100
70	MP(I)=I	00101200
C		00101300
	M3=M1	00101400
	IF(M3.GT.20) M3=20	00101500
	IC0=0	00101600
	M2=1	00101700
77	IC0=IC0+1	00101800
	IF(IC0.GE.2) WRITE(IOUT,630)	00101900
	WRITE(IOUT,620) IC0	00102000
	DO 31 I=1,2	00102100
	DO 31 J=1,400	00102200
31	ITEMP(I,J)=BLANK	00102300
C		00102400
	I=0	00102500
	IR=1	00102600
	L1=1	00102700
	DO 32 J=M2,M3,2	00102800
	CALL ITQ(Q,MM,MP,ITEMP,MB,MA,IA1,IA2,I,IR,L1,J,M1)	00102900
32	CONTINUE	00103000
	I=6	00103100
	IR=IR+1	00103200
	L1=1	00103300
	M4=M2+1	00103400
	DO 34 J=M4,M3,2	00103500
	CALL ITQ(Q,MM,MP,ITEMP,MB,MA,IA1,IA2,I,IR,L1,J,M1)	00103600
34	CONTINUE	00103700
C		00103800
	KKK=120	00103900
	DO 35 I=1,IR	00104000
	WRITE(IOUT,690) (ITEMP(I,J),J=1,KKK)	00104100
35	KKK=KKK+5	00104200
C		00104300
	IF(M2.LE.20) GO TO 40	00104400
	LL=1	00104500
	IA=121	00104600
	IB=240	00104700
100	CONTINUE	00104800
	ITEMP(2,IA)=BLANK	00104900
	M5=M3-20	00105000
	M6=M5	00105100
	IF(M5.GT.(LL+20-1)) M6=LL+20-1	00105200
	DO 50 I=1,50	00105300
	WRITE(IOUT,601) I,(YY(I,K),K=LL,M6)	00105400
	IF(I.EQ.N) GO TO 110	00105500
50	CONTINUE	00105600
110	CONTINUE	00105700
	IF(M5.LE.M6) GO TO 30	00105800
	IC0=IC0+1	00105900
	WRITE(IOUT,630)	00106000
	WRITE(IOUT,620) IC0	00106100
	LL=M6+1	00106200

DO 33 I=1,IR	00106300
WRITE(IOUT,690) (ITEMP(I,J),J=IA,IB)	00106400
33 IB=IB+5	00106500
IA=IB-4	00106600
IB=IB+120	00106700
GO TO 100	00106800
C 40 CONTINUE	00106900
C	00107000
N=0	00107100
1 N=N+1	00107200
READ(KNEW,FMT,END=11,ERR=20) (XX(K),K=1,M1)	00107300
WRITE(KOLD) N,(XX(K),K=1,M1)	00107400
IF(N.GT.50) GO TO 1	00107500
WRITE(IOUT,601) N,(XX(K),K=1,M3)	00107600
IF(M1.LE.20) GO TO 1	00107700
J=0	00107800
M6=M3+1	00107900
DO 60 I=M6,M1	00108000
J=J+1	00108100
60 YY(N,J)=XX(I)	00108200
GO TO 1	00108300
C	00108400
11 N=N-1	00108500
IF(M1.LE.20) GO TO 30	00108600
M2=M3+1	00108700
M3=M1	00108800
GO TO 77	00108900
C	00109000
30 IF(N.GT.50) WRITE(IOUT,610) N	00109100
NUMDT(KOLD)=N	00109200
REWIND KNEW	00109300
RETURN	00109400
C	00109500
20 WRITE(IOUT,900)	00109600
900 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00109700
* 'SPECIFIED BY COMMAND ---')	00109800
RETURN	00109900
C	00110000
999 WRITE(IOUT,600)	00110100
600 FORMAT(1H1,///1X,'--- COMMAND WAS MISUSED ---'//5X,'READ TAPE'//	00110200
* 10X,'CARD'//10X,'DISK XX'//12X,'< XX NUMBER OF DISK',	00110300
* ' (FILE) >')	00110400
C	00110500
RETURN	00110600
C	00110700
605 FORMAT(1H1,///1X,'INPUT DATA'/1X,10(' '-'))	00110800
690 FORMAT(1H ,5X,126A1)	00110900
601 FORMAT(1H ,14,20F6.1)	00111000
610 FORMAT(///5X,'< FROM 51 TO',I6,' >')	00111100
620 FORMAT(1H ,95X,'(PAGE',I3,')'//)	00111200
630 FORMAT(1H1,///)	00111300
C	00111400
END	00111500
	00111600

```

SUBROUTINE DECODE(AA,Q,MM,MQ,A,B,ABC,NUM,NUMB,II,MQH,ITWO,ISIDE, 00111700
*      IBY,MQ2,IY,MB,MA,MX,KA,KN,M1,IX,MQ1,COMMA, 00111800
*      BLANK,SLASH ) 00111900
C
C THIS ROUTINE IS A PART OF DECODER. 00112000
C 00112100
COMMON / OUTIN / INPT, IOUT 00112200
INTEGER AA(100,80),Q(MB),ABC(KA),COMMA,BLANK,SLASH,POINT, 00112300
* WITH(4),SIDE(4),HEAD(4),MULT(4),BY(2),OPC,CRC,EQUAL, 00112400
* CORON,SCORON 00112500
* DIMENSION MM(MA),MQ(MX),A(MX),B(MX),NUM(KN),NUMB(KN),MQH(MX) 00112600
DATA POINT/'.'/, MINUS/'-'/, NAMI/'~'/, OPC/'('/, CRC/'('/, 00112700
* WITH/'W','I','T','H'/, SIDE/'S','I','D','E'/, 00112800
* HEAD/'H','E','A','D'/, MULT/'M','U','L','T'/, 00112900
* BY/'B','Y'/, EQUAL/'='/, CORON/';', SCORON/'.'/ 00113000
C 00113100
L1=1 00113200
K=0 00113300
IC=0 00113400
IERO = 0 00113500
IAB=0 00113600
C 00113700
DO 10 I=IX,IY 00113800
IF(AA(II,I).EQ.SLASH) GO TO 99 00113900
IF(AA(II,I).EQ.COMMA.OR.AA(II,I).EQ.BLANK 00114000
* .OR.AA(II,I).EQ.CORON 00114100
* .OR.AA(II,I).EQ.NAMI ) GO TO 20 00114200
IF(AA(II,I).EQ.OPC.OR.AA(II,I).EQ.CRC 00114300
* .OR.AA(II,I).EQ.SCORON 00114400
* .OR.AA(II,I).EQ.EQUAL) GO TO 20 00114500
K=K+1 00114600
GO TO 10 00114700
99 IAB=1 00114800
IIA=I-1 00114900
IF(AA(II,IIA).EQ.BLANK) GO TO 97 00115000
20 IF(K.EQ.0) GO TO 10 00115100
K1=I-K 00115200
K2=I-1 00115300
DO 30 J=1,KA 00115400
IF(AA(II,K1).EQ.ABC(J)) GO TO 40 00115500
30 CONTINUE 00115600
C 00115700
IP=0 00115800
IA=0 00115900
FG=1 00116000
DO 50 J=K1,K2 00116100
IF(AA(II,J).NE.MINUS) GO TO 60 00116200
FG=-1 00116300
GO TO 50 00116400
60 IF(AA(II,J).NE.POINT) GO TO 70 00116500
IP=IA 00116600
GO TO 50 00116700
70 IA=IA+1 00116800
DO 80 JN=1,KN 00116900
IF(AA(II,J).EQ.NUM(JN)) GO TO 85 00117000
80 CONTINUE 00117100
85 JN=JN-1 00117200
NUMB(IA)=JN 00117300
50 CONTINUE 00117400
C 00117500
JA=0 00117600
DO 15 JJ=1,IA 00117700
J1=IA-JJ 00117800
15 JA=JA+NUMB(JJ)*10**J1 00117900
IF(IC.EQ.0) GO TO 25 00118000
IF(IP.EQ.0.OR.IP.EQ.1A) GO TO 16 00118100
B(MQ1)=FG*FLUAT(JA)/10.**((IA-IP) 00118200
00118300

```

	GO TO 17	00118400
16	B(MQ1)=FG#FLOAT(JA)	00118500
17	IC=0	00118600
	K=0	00118700
	GO TO 98	00118800
25	IF(IP.EQ.0.OR.IP.EQ.IA) GO TO 26	00118900
	A(MQ1)=FG#FLOAT(JA)/10.##(IA-IP)	00119000
	GO TO 27	00119100
26	A(MQ1)=FG#FLOAT(JA)	00119200
27	IC=1	00119300
	K=0	00119400
	B(MQ1)=A(MQ1)	00119500
	GO TO 98	00119600
C		00119700
40	CONTINUE	00119800
	IC=0	00119900
	JW=K1	00120000
	DO 41 LW=1,4	00120100
	IF(WITH(LW).NE.AA(II,JW)) GO TO 42	00120200
41	JW=JW+1	00120300
	K=0	00120400
	GO TO 98	00120500
C		00120600
42	JW=K1	00120700
	DO 31 J=1,4	00120800
	IF(SIDE(J).NE.AA(II,JW)) GO TO 32	00120900
31	JW=JW+1	00121000
	ITWO=1	00121100
	ISIDE=1	00121200
	K=0	00121300
	GO TO 98	00121400
C		00121500
32	JW=K1	00121600
	DO 33 J=1,4	00121700
	IF(HEAD(J).NE.AA(II,JW)) GO TO 34	00121800
33	JW=JW+1	00121900
	ITWO=1	00122000
	ISIDE=2	00122100
	K=0	00122200
	GO TO 98	00122300
C		00122400
34	JW=K1	00122500
	DO 35 J=1,4	00122600
	IF(MULT(J).NE.AA(II,JW)) GO TO 36	00122700
35	JW=JW+1	00122800
	ITWO=2	00122900
	K=0	00123000
	GO TO 98	00123100
C		00123200
36	JW=K1	00123300
	DO 37 J=1,2	00123400
	IF(BY(J).NE.AA(II,JW)) GO TO 38	00123500
37	JW=JW+1	00123600
	IBY=1	00123700
	K=0	00123800
	GO TO 98	00123900
38	CONTINUE	00124000
	DO 90 L=1,M1	00124100
	M9=MM(L)	00124200
	L2=L1+M9-1	00124300
	IF(K.EQ.M9) GO TO 45	00124400
46	L1=L2+1	00124500
90	CONTINUE	00124600
	K=0	00124700
	IF(IERO.EQ.0) WRITE(IOUT,620)	00124800
	WRITE(IOUT,600) (AA(II,J),J=K1,K2)	00124900
600	FORMAT(1H,5X,'ERROR ITEM = ',8A1)	00125000
620	FORMAT(1H1,///1X,'--- THIS ITEM UNDEFINED ---')	00125100
	IERO = 1	00125200
	GO TO 98	00125300
C		00125400
45	J=K1	00125500
	DO 55 LL=L1,L2	00125600
	IF(Q(LL).NE.AA(II,J)) GO TO 46	00125700
55	J=J+1	00125800
	GO TO (75,76),ISIDE	00125900
	IF(IBY.EQ.1) GO TO 76	00126000

75	MQ1=MQ1+1	00126100
	MQ(MQ1)=L	00126200
	K=0	00126300
	L1=1	00126400
	GO TO 98	00126500
76	MQ2=MQ2+1	00126600
	MQH(MQ2)=L	00126700
	K=0	00126800
	L1=1	00126900
98	IF(IAB.EQ.1) GO TO 97	00127000
10	CONTINUE	00127100
97	IF(IERO.EQ.1) MQ1=0	00127200
	RETURN	00127300
	END	00127400
	SUBROUTINE ITQ(Q,MM,MQ,ITEMP,MB,MA,IA1,IA2,I,IR,L1,J,M1)	00127500
C		00127600
C		00127700
C		00127800
	INTEGER Q(MB)	00127900
	DIMENSION MM(MA),MQ(MA),ITEMP(IA1,IA2)	00128000
C		00128100
	L5=L1	00128200
	K=MQ(J)	00128300
	CALL PITEM(MM,MA,M1,K,L5,L2,KP)	00128400
	L9=12-(L2-L5+1)	00128500
	DO 33 L=L5,L2	00128600
	I=I+1	00128700
33	ITEMP(IR,I)=Q(L)	00128800
	I=I+L9	00128900
	RETURN	00129000
	END	00129100
	SUBROUTINE PITEM(MM,MA,M1,IQ,L1,L2,KP)	00129200
C		00129300
C		00129400
C		00129500
	DIMENSION MM(MA)	00129600
C		00129700
	DO 10 J=1,M1	00129800
	M9=MM(J)	00129900
	L2=L1+M9-1	00130000
	IF(IQ.EQ.J) GO TO 20	00130100
	L1=L2+1	00130200
10	CONTINUE	00130300
	KP=0	00130400
	RETURN	00130500
20	KP=1	00130600
	RETURN	00130700
	END	00130800
	SUBROUTINE FORM(NUM,KN,FMT,IA,L1,L2,K)	00130900
C		00131000
C	THIS IS A DECODER OF FORMAT-COMMAND.	00131100
C		00131200
	INTEGER FMT(K),BLANK	00131300
	DIMENSION NUM(KN)	00131400
	DATA BLANK/' '/	00131500
C		00131600
	IB=IA/10	00131700
	IC=IA-IB*10	00131800
C		00131900
	DO 10 JJ=1,10	00132000
	J=JJ-1	00132100
	IF(IB.EQ.J) N8=JJ	00132200
	IF(IC.EQ.J) N9=JJ	00132300
10	CONTINUE	00132400
C		00132500
	IF(IB.EQ.0) GO TO 20	00132600
	FMT(L1)=NUM(N8)	00132700
	FMT(L2)=NUM(N9)	00132800
	RETURN	00132900
20	FMT(L1)=BLANK	00133000

	FMT(L2)=NUM(N9)	00133100
	RETURN	00133200
	END	00133300
	SUBROUTINE SBRANG(XX,Q,MM,MQ,NQ,NOMIT,ICOL,MS,NS,IN,IZ,IS,A,B,	00133400
	* Y,Z,ZS,FT,MA,MB,MX,NN,KN,NM1,J1,BLANK,INDEV,	00133500
	* OUTDEV)	00133600
C		00133700
C	THIS SUBROUTINE CARRIES OUT THE RANGE CHECKING PROCEDURE.	00133800
		00133900
	COMMON / OUTIN / INPT, IOUT	00134000
	COMMON NUMDT(20)	00134100
	INTEGER Q(MB),FT(KN),FMT1(14),FMT3(12),BLANK,OUTDEV	00134200
	DIMENSION XX(MA),MM(MA),MQ(MX),NQ(MX),NOMIT(NN),ICOL(NN),MS(MX),	00134300
	* NS(NN),IN(MX,NN),IZ(MX,NN),IS(MX,NN),A(MX),B(MX),	00134400
	* Y(MX,NN),Z(MX,NN),ZS(MX,NN)	00134500
	DATA FMT1/'(1H ',' ,',' ,',' ,',' ,',' ,',' ,',' ,',' ,',' ,',' ,',' ,'	00134600
	* '5.1','4H' ~ '4H',F5,4H.1,'4H)')'/	00134700
	DATA FMT3/'(1H ',' ,',' ,',' ,',' ,',' ,',' ,',' ,',' ,',' ,'	00134800
	* '5.1','4H')',' ,')'/	00134900
		00135000
	WRITE(IOUT,800)	00135100
800	FORMAT(1H1,///1X,'+-----+' /1X,' RANGE CHECK ',	00135200
	* /1X,'+-----+')	00135300
		00135400
	KNUM=NUMDT(INDEV)	00135500
	IF(KNUM.EQ.0) GO TO 1000	00135600
	DO 5 I=1,J1	00135700
5	NO(I)=0	00135800
		00135900
	JX=0	00136000
	ID=0	00136100
	NOS=0	00136200
	REWIND INDEV	00136300
	DO 20 I=1,N	00136400
	LND=0	00136500
	READ(INDEV,END=99,ERR=777) KK,(XX(J),J=1,M1)	00136600
	NOS=NOS+1	00136700
	DO 21 J=1,J1	00136800
	K=MQ(J)	00136900
	IF(XX(K).GE.A(J).AND. XX(K).LE.B(J)) GO TO 21	00137000
	LND=LND+1	00137100
	NO(J)=NO(J)+1	00137200
	NORA=NO(J)	00137300
	IF(NORA.GE.201) GO TO 21	00137400
	IN(J,NORA)=KK	00137500
	Y(J,NORA)=XX(K)	00137600
21	CONTINUE	00137700
	IF(LND.GT.0) GO TO 20	00137800
	ID=ID+1	00137900
	IF(OUTDEV.LE.0) GO TO 20	00138000
	WRITE(OUTDEV) KK,(XX(J),J=1,M1)	00138100
20	CONTINUE	00138200
		00138300
		00138400
99	CONTINUE	00138500
	IF(OUTDEV.LE.0) GO TO 98	00138600
	IF(ID.LE.0) WRITE(IOUT,690)	00138700
	NUMDT(OUTDEV)=ID	00138800
		00138900
98	CONTINUE	00139000
	L1=1	00139100
	DO 10 J=1,J1	00139200
	IF(J.GE.2) GO TO 1	00139300
	WRITE(IOUT,620) N,NOS	00139400
	GO TO 2	00139500
1	WRITE(IOUT,625)	00139600
2	CONTINUE	

	K=MQ(J)	00139700
	CALL PITEM(MM,MA,M1,K,L1,L2,KP)	00139800
	IF(KP.EQ.0) GO TO 22	00139900
C		00140000
	CALL FMA(L1,L2,L7,N7,N8,N9)	00140100
	FMT1(4)=FT(N9)	00140200
	FMT3(4)=FT(N9)	00140300
	IF(L7.LT.0) GO TO 3	00140400
	FMT1(6)=FT(2)	00140500
	FMT3(6)=FT(2)	00140600
	FMT1(7)=FT(N7)	00140700
	FMT3(7)=FT(N7)	00140800
	GO TO 4	00140900
3	FMT1(6)=BLANK	00141000
	FMT3(6)=BLANK	00141100
	FMT1(7)=FT(N8)	00141200
	FMT3(7)=FT(N8)	00141300
4	IF(A(J).NE.B(J)) GO TO 18	00141400
	WRITE(IOUT,FMT3) (Q(L),L=L1,L2),A(J)	00141500
	GO TO 19	00141600
18	WRITE(IOUT,FMT1) (Q(L),L=L1,L2),A(J),B(J)	00141700
19	WRITE(IOUT,650)	00141800
	L1=1	00141900
22	CUNTINUE	00142000
C		00142100
	II=NO(J)	00142200
	IF(II.NE.0) GO TO 24	00142300
	WRITE(IOUT,630)	00142400
	GO TO 10	00142500
24	DO 25 I=1,II	00142600
	IF(I.GE.201) GO TO 26	00142700
	WRITE(IOUT,610) I,IN(J,I),Y(J,I)	00142800
25	CUNTINUE	00142900
	GO TO 10	00143000
26	WRITE(IOUT,700) II	00143100
	JX=1	00143200
10	CUNTINUE	00143300
	IF(JX.EQ.1) J1=1	00143400
	IF(J1.EQ.1) GO TO 112	00143500
	KPA=0	00143600
	K1=1	00143700
	DO 30 I=1,J1	00143800
	L=NO(I)	00143900
	IF(L.EQ.0) GO TO 30	00144000
	IF(KPA.NE.0) GO TO 33	00144100
	NOMIT(K1)=IN(I,1)	00144200
	KPA=1	00144300
33	CUNTINUE	00144400
	DO 30 J=1,L	00144500
	DO 40 K=1,K1	00144600
	IF(IN(I,J).EQ.NOMIT(K)) GO TO 30	00144700
40	CUNTINUE	00144800
	K1=K1+1	00144900
	NOMIT(K1)=IN(I,J)	00145000
	CALL CHANGE(NOMIT,NOMIT,NN,K1,1)	00145100
30	CUNTINUE	00145200
	IF(KPA.NE.0) GO TO 45	00145300
	WRITE(IOUT,640)	00145400
	WRITE(IOUT,630)	00145500
	RETURN	00145600
C		00145700
45	DO 55 I=1,J1	00145800
	DO 55 J=1,K1	00145900
	Z(I,J)=0	00146000
55	IZ(I,J)=0	00146100
	DO 50 K=1,K1	00146200
	L1=0	00146300
	DO 60 I=1,J1	00146400
	L=NO(I)	00146500
	IF(L.EQ.0) GO TO 60	00146600
	DO 70 J=1,L	00146700
	IF(IN(I,J).EQ.NOMIT(K)) GO TO 80	00146800
70	CUNTINUE	00146900
	GO TO 60	00147000
80	Z(I,K)=Y(I,J)	00147100

	L1=L1+1	00147200
	IZ(I,K)=1	00147300
60	CONTINUE	00147400
	ICOL(K)=L1	00147500
50	CONTINUE	00147600
	WRITE(IOUT,640)	00147700
	WRITE(IOUT,608)	00147800
C	CALL SUMTAB(Q,MM,MQ,NOMIT,IZ,Z,FT,MB,MA,MX,NN,KN,K1,J1,M1,BLANK)	00147900
C		00148000
	DO 8 I=1,J1	00148100
	MS(I)=MQ(I)	00148200
	DO 8 J=1,K1	00148300
	NS(J)=NOMIT(J)	00148400
	IS(I,J)=IZ(I,J)	00148500
8	ZS(I,J)=Z(I,J)	00148600
C		00148700
	CALL CHANGE(NO,MS,MX,J1,2)	00148800
	CALL CHANGE(ICOL,NS,NN,K1,2)	00148900
C		00149000
	DO 14 I=1,J1	00149100
	DO 14 J=1,K1	00149200
	II=MS(I)	00149300
	JJ=NS(J)	00149400
	DO 15 I5=1,J1	00149500
	IF(MQ(I5).EQ.II) IL=I5	00149600
15	CONTINUE	00149700
	DO 16 J5=1,K1	00149800
	IF(NOMIT(J5).EQ.JJ) JL=J5	00149900
16	CONTINUE	00150000
	IZ(I,J)=IS(IL,JL)	00150100
14	Z(I,J)=ZS(IL,JL)	00150200
	WRITE(IOUT,685)	00150300
	WRITE(IOUT,609)	00150400
	CALL SUMTAB(Q,MM,MS,NS,IZ,Z,FT,MB,MA,MX,NN,KN,K1,J1,M1,BLANK)	00150500
C		00150600
112	CONTINUE	00150700
	WRITE(IOUT,680) ID	00150800
	RETURN	00150900
C		00151000
777	WRITE(IOUT,900)	00151100
	RETURN	00151200
C		00151300
1000	WRITE(IOUT,6001) INDEV	00151400
	RETURN	00151500
C		00151600
620	FORMAT(1H0,7X,'TOTAL NUMBER OF SAMPLES TN =',I6//	00151700
*	8X,'NUMBER OF SAMPLES N =',I6)	00151800
625	FORMAT(1H1)	00151900
650	FORMAT(1H0,10X,'NUMBER',2X,'OMIT NUMBER',3X,'DATA X')	00152000
610	FORMAT(1H ,10X,I4,4X,I5,9X,F5.1)	00152100
630	FORMAT(1H0,10X,'--- NO OMIT DATA ---')	00152200
640	FORMAT(1H1,///1X,'SUMMARY TABLE'/1X,I3('---'))	00152300
608	FORMAT(1H ,7X,'< BEFORE ORDERING >')	00152400
685	FORMAT(///)	00152500
670	FORMAT(1H ,18X,20I5)	00152600
609	FORMAT(1H ,7X,'< AFTER ORDERING >')	00152700
680	FORMAT(///10X,'NUMBER OF VALID DATA =',I6)	00152800
690	FORMAT(1H0,'--- ERROR FOR THE EXECUTION ABOUT WRITE TO DISK ---')	00152900
700	FORMAT(1H0,10X,'< FROM 201 TO ',I5,' >')	00153000
900	FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00153100
*	'SPECIFIED BY COMMAND ---')	00153200
6001	FORMAT(///1H ,'---- FILE NUMBER IS',I3,'. THIS FILE DOES NOT',	00153300
*	' INCLUDE DATA ---')	00153400
C		00153500
END		00153600
		00153700

	SUBROUTINE SUMTAB(Q,MM,MQ,NOMIT,IZ,Z,FT,MB,MA,MX,NN,KN,K1,J1,M1,	00153800
*	BLANK)	00153900
	COMMON / OUTIN / INPT, IOUT	00154000
	INTEGER Q(MB),FT(KN),FMT2(8),FMT4(7),FMT5(7),POINT(30),	00154100
*	BLANK,BAR	00154200
	DIMENSION MM(MA),MQ(MX),NOMIT(NN),IZ(MX,NN),Z(MX,NN)	00154300
	DATA FMT2/'(1H','5X',' ','A1',' ','4HX','2H')/'	00154400
	DATA FMT4/'(1H','16X',' ',' ','X,F5','1')/'	00154500
	DATA FMT5/'(1H','16X',' ',' ','X,A5','')/'	00154600
	DATA POINT/30*4H..../BAR/' -- '/'	00154700
C		00154800
C		00154900
	KS=K1/20	00155000
	KA=K1-KS*20	00155100
	KE=KS+1	00155200
C		00155300
	K2=1	00155400
	DO 10 KK=1,KE	00155500
C		00155600
	IF(KK.NE.KE) GO TO 1	00155700
	IF(KA.EQ.0) RETURN	00155800
C		00155900
	1 IF(KK.EQ.KE) GO TO 20	00156000
	IF(KK+1.NE.KE) GO TO 30	00156100
	IF(KA.NE.0) GO TO 30	00156200
20	K3=K1	00156300
	GO TO 40	00156400
30	K3=KK*20	00156500
40	IF(KK.EQ.KE) GO TO 50	00156600
	KP=26	00156700
	GO TO 60	00156800
50	KP=KA*5/4+1	00156900
60	WRITE(IOUT,620)	00157000
	WRITE(IOUT,630) (NOMIT(I),I=K2,K3,2)	00157100
	K4=K2+1	00157200
	IF(KA.LE.1.AND.KK.EQ.KE) GO TO 9	00157300
	WRITE(IOUT,640) (NOMIT(I),I=K4,K3,2)	00157400
9	WRITE(IOUT,660) (POINT(I),I=1,KP)	00157500
C		00157600
	L1=1	00157700
	DO 12 I=1,J1	00157800
	IQ=MQ(I)	00157900
	CALL PITEM(MM,MA,M1,IQ,L1,L2,KP)	00158000
	IF(KP.EQ.0) GO TO 12	00158100
C		00158200
	CALL FMA(L1,L2,L7,N7,N8,N9)	00158300
	FMT2(3)=FT(N9)	00158400
	IF(L7.LT.0) GO TO 5	00158500
	FMT2(5)=FT(2)	00158600
	FMT2(6)=FT(N7)	00158700
	GO TO 6	00158800
5	FMT2(5)=BLANK	00158900
	FMT2(6)=FT(N8)	00159000
6	WRITE(IOUT,FMT2) (Q(L),L=L1,L2)	00159100
C		00159200
	IA=2	00159300
	DO 90 K=K2,K3	00159400
	IF(IZ(I,K).EQ.1) GO TO 91	00159500
	CALL FORM(FT,KN,FMT5,IA,4,5,7)	00159600
	WRITE(IOUT,FMT5) BAR	00159700
	GO TO 90	00159800
91	CALL FORM(FT,KN,FMT4,IA,4,5,7)	00159900
	WRITE(IOUT,FMT4) Z(I,K)	00160000
90	IA=IA+5	00160100
C		00160200
	L1=1	00160300
12	CONTINUE	00160400
C		00160500
	WRITE(IOUT,680)	00160600
680	FORMAT(//)	00160700
C		00160800
	10 K2=K3+1	00160900
C		00161000
620	FORMAT(1H , 5X, 4X ,6X,'.',5X,'O M I T T E D D A T A ')	00161100
630	FORMAT(1H ,5X,'ITEM',6X,'.',2X,10(15,5X))	00161200
640	FORMAT(1H ,15X,'.',2X,10(5X,15))	00161300
660	FORMAT(1H ,5X,12('.',),28A4)	00161400
C		00161500
	RETURN	00161600
	END	00161700

C	SUBROUTINE FMA(L1,L2,L7,N7,N8,N9)	00161800
	L9=L2-L1+1	00161900
	L8=10-L9	00162000
	L7=L8-10	00162100
	DO 10 I=1,10	00162200
	K=I-1	00162300
	IF(L9.EQ.K) N9=I	00162400
	IF(L7.LT.0) GO TO 20	00162500
	IF(L7.EQ.K) N7=I	00162600
	GO TO 10	00162700
20	IF(L8.EQ.K) N8=I	00162800
10	CONTINUE	00162900
	RETURN	00163000
	END	00163100
		00163200

	SUBROUTINE CHANGE(IR,MM,M1,K,LOGC)	00163300
C		00163400
C		00163500
C		00163600
	DIMENSION IR(M1),MM(M1)	00163700
	IF(LOGC.EQ.1) GO TO 20	00163800
	K1=K-1	00163900
	DO 10 I=1,K1	00164000
	J1=I+1	00164100
	DO 10 J=J1,K	00164200
	IF(IR(I).GE. IR(J)) GO TO 10	00164300
	II=IR(I)	00164400
	IR(I)=IR(J)	00164500
	IR(J)=II	00164600
	NN=MM(I)	00164700
	MM(I)=MM(J)	00164800
	MM(J)=NN	00164900
10	CONTINUE	00165000
	RETURN	00165100
C		00165200
20	K1=K-1	00165300
	DO 30 I=1,K1	00165400
	J1=I+1	00165500
	DO 30 J=J1,K	00165600
	IF(IR(I).LE. IR(J)) GO TO 30	00165700
	NN=MM(I)	00165800
	MM(I)=MM(J)	00165900
	MM(J)=NN	00166000
30	CONTINUE	00166100
	RETURN	00166200
	END	00166300

```

SUBROUTINE CONS(AA,Q,MM,ABC,NUM,MQ7,RELAT7,CONST7,LOGIC7,IFTAB7, 00166400
*      II,NUMB,MQ8,RELAT8,CONST8,LOGIC8,IFTAB8,IY,MB,MA, 00166500
*      KA,KN,MY,M1,IX,NCON,IT,IS,MO7,MO8,KUNT,MP7,MP8, 00166600
*      LL1,LL2,IFFSW ) 00166700
      INTEGER AA(100,80),Q(MB),ABC(KA),RELAT7(MY),RELAT8(MY), 00166800
      DIF(2),DTHEN(4),DIFF(3),BLANK,RIGHT,COMMA,SLASH 00166900
      DIMENSION MM(MA),NUM(KN),MQ7(MY),CONST7(MY),LOGIC7(MY),IFTAB7(MB), 00167000
      NUMB(KN),MQ8(MY),CONST8(MY),LOGIC8(MY),IFTAB8(MB) 00167100
      DATA DIF/'I','F'//,DTHEN/'T','H','E','N'//,DIFF/'I','F','F'//, 00167200
      BLANK/' ','/','LEFT'/'('//,RIGHT/')'//,COMMA/',','/','SLASH'/'/'// 00167300
C      K=0 00167400
      CALL ACHECK(AA,II) 00167500
C      IF(NCON=1) 3,15,25 00167600
3      IERO=0 00167700
      DO 1 I=IX,IY 00167800
C      IF(AA(II,I).EQ.BLANK) GO TO 1 00167900
      IF(AA(II,I).NE.LEFT) GO TO 2 00168000
      NCON=1 00168100
      IFFSW=1 00168200
      GO TO 15 00168300
1      CONTINUE 00168400
C      10 CONTINUE 00168500
2      IF(AA(II,IX).EQ.BLANK.OR.AA(II,IX).EQ.COMMA.OR. 00168600
      *      AA(II,IX).EQ.SLASH.OR. 00168700
      *      AA(II,IX).EQ.LEFT.OR.AA(II,IX).EQ.RIGHT) GO TO 11 00168800
      K=K+1 00168900
      GO TO 20 00169000
11      IF(K.NE.0) GO TO 29 00169100
20      IX=IX+1 00169200
      IF(IX.GT.IY) RETURN 00169300
      GO TO 10 00169400
C      29 K1=IX-K 00169500
      J=K1 00170000
      K9=K 00170100
      K=0 00170200
      GO TO (20,45,65,55),K9 00170300
      GO TO 20 00170400
C      45 DO 30 I=1,2 00170500
      IF(AA(II,J).NE.DIF(I)) GO TO 20 00170600
30      J=J+1 00170700
      NCON=1 00170800
C      15 CALL IFTHEN(AA,Q,MM,ABC,NUM,NUMB,MQ7,RELAT7,CONST7,LOGIC7, 00170900
      *      IFTAB7,IY,MB,MA,KA,KN,MY,M1,IX,KCON,IT,II,IERO, 00171000
      *      MO7,MP7,LL1,LL2,BLANK,LEFT,RIGHT,DTHEN,DIFF) 00171100
C      IF(IERO.EQ.1) IT=0 00171200
      IF(KCON.EQ.1) RETURN 00171300
      NCON=2 00171400
      GO TO 20 00171500
C      55 DO 40 I=1,4 00171600
      IF(AA(II,J).NE.DTHEN(I)) GO TO 20 00171700
40      J=J+1 00171800
      NCON=2 00171900
      KUNT=1 00172000
      GO TO 25 00172100
C      65 DO 50 I=1,3 00172200
      IF(AA(II,J).NE.DIFF(I)) GO TO 20 00172300
50      J=J+1 00172400
      NCON=3 00172500
      IFFSW=1 00172600
      KUNT=1 00172700
C      25 CALL IFTHEN(AA,Q,MM,ABC,NUM,NUMB,MQ8,RELAT8,CONST8,LOGIC8, 00172800
      *      IFTAB8,IY,MB,MA,KA,KN,MY,M1,IX,KCON,IS,II,IERO, 00172900
      *      MO8,MP8,LL1,LL2,BLANK,LEFT,RIGHT,DTHEN,DIFF) 00173000
99      IF(IERO.EQ.1) GO TO 70 00173100
      RETURN 00173200
70      IT=0 00173300
C      RETURN 00173400
C      END 00173500
      00173600
      00173700
      00173800
      00173900
      00174000
      00174100
      00174200
      00174300
      00174400

```

	SUBROUTINE IFTHEN(AA,Q,MM,ABC,NUM,NUMB,MQ,RELAT,CONST,LOGIC,	00174500
*	IFTAB,IY,MB,MA,KA,KN,MY,M1,IX,KCON,IT,II,IERO,	00174600
*	MQ1,MP1,LL1,LL2,BLANK,LEFT,RIGHT,DTHEN,DIFF)	00174700
	COMMON / OUTIN / INPT, IOUT	00174800
	INTEGER AA(100,80),Q(MB),ABC(KA),RELAT(MY),BLANK,RIGHT,POINT,	00174900
*	DTHEN(4),DIFF(3)	00175000
*	DIMENSION MM(MA),NUM(KN),NUMB(KN),MQ(MY),CONST(MY),LOGIC(MY),	00175100
*	IFTAB(MB),IREL(12),ILOG(5)	00175200
	DATA IREL/'L','T','L','E','E','Q','N','E','G','T','G','E'/'	00175300
	DATA ILOG/'O','R','A','N','D'/'	00175400
	DATA POINT/'.'/,MINUS/'-'/'	00175500
C		00175600
	K=0	00175700
	IRI=0	00175800
C		00175900
	DO 10 I=IX,IY	00176000
C		00176100
	15 IF(AA(II,I).EQ.BLANK) GO TO 20	00176200
	IF(AA(II,I).NE.LEFT) GO TO 25	00176300
	LL1=LL1+1	00176400
	IT=IT+1	00176500
	IFTAB(IT)=6	00176600
	GO TO 20	00176700
	25 IF(AA(II,I).NE.RIGHT) GO TO 35	00176800
	LL2=LL2+1	00176900
	IRI=1	00177000
	GO TO 20	00177100
	35 K=K+1	00177200
	GO TO 10	00177300
C		00177400
	20 IF(K.EQ.0) GO TO 65	00177500
	K1=I-K	00177600
	K2=I-1	00177700
C		00177800
	DO 30 J=1,KA	00177900
	IF(AA(II,K1).EQ.ABC(J)) GO TO 46	00178000
	30 CONTINUE	00178100
	GO TO 45	00178200
C		00178300
	46 L1=1	00178400
	DO 40 L=1,M1	00178500
	M9=MM(L)	00178600
	L2=L1+M9-1	00178700
	IF(K.EQ.M9) GO TO 55	00178800
	GO TO 56	00178900
	55 J=K1	00179000
	DO 50 LL=L1,L2	00179100
	IF(Q(LL).NE.AA(II,J)) GO TO 56	00179200
	50 J=J+1	00179300
	MQ1=MQ1+1	00179400
	MQ(MQ1)=L	00179500
	GO TO 65	00179600
	56 L1=L2+1	00179700
	40 CONTINUE	00179800
C		00179900
	J1=1	00180000
	IF(K.NE.2) GO TO 75	00180100
	82 J2=J1+1	00180200
	J=K1	00180300
	DO 60 JJ=J1,J2	00180400
	IF(IREL(JJ).NE.AA(II,J)) GO TO 81	00180500
	60 J=J+1	00180600
	J3=J2/2	00180700
	RELAT(MQ1)=J3	00180800
	GO TO 65	00180900
	81 J1=J1+2	00181000
	IF(J1.LE.11) GO TO 82	00181100

C	J=K1	00181200
	DO 70 JJ=1,2	00181300
	IF(ILOG(JJ).NE.AA(II,J)) GO TO 75	00181400
70	J=J+1	00181500
	MP1=MP1+1	00181600
	LOGIC(MP1)=1	00181700
	GO TO 86	00181800
75	J=K1	00181900
	DO 80 JJ=3,5	00182000
	IF(ILOG(JJ).NE.AA(II,J)) GO TO 85	00182100
80	J=J+1	00182200
	MP1=MP1+1	00182300
	LOGIC(MP1)=2	00182400
86	IT=IT+1	00182500
	IFTAB(IT)=8	00182600
	GO TO 65	00182700
85	J=K1	00182800
	DO 87 JJ=1,4	00182900
	IF(DTHEN(JJ).NE.AA(II,J)) GO TO 88	00183000
87	J=J+1	00183100
	GO TO 65	00183200
88	J=K1	00183300
	DO 89 JJ=1,3	00183400
	IF(DIFF(JJ).NE.AA(II,J)) GO TO 83	00183500
89	J=J+1	00183600
	GO TO 65	00183700
83	IF(IERO .EQ. 0) WRITE(IOUT,620)	00183800
	WRITE(IOUT,600) (AA(II,J9),J9=K1,K2)	00183900
600	FORMAT(1H ,5X,'ERROR = ',8A1)	00184000
620	FORMAT(1H1,///1X,'--- LOGICAL EXPRESSIONS ARE ERROR, THEN CHECK '	00184100
*	'AGAIN EACH ITEM AND LOGIC ---'	00184200
	IERO = 1	00184300
	GO TO 65	00184400
C		00184500
45	IP=0	00184600
	IA=0	00184700
	FG=1	00184800
	DO 90 J=K1,K2	00184900
	IF(AA(II,J).NE.MINUS) GO TO 91	00185000
	FG=-1	00185100
	GO TO 90	00185200
91	IF(AA(II,J).NE.POINT) GO TO 92	00185300
	IP=IA	00185400
	GO TO 90	00185500
92	IA=IA+1	00185600
	DO 95 JN=1,10	00185700
	IF(AA(II,J).EQ.NUM(JN)) GO TO 93	00185800
95	CONTINUE	00185900
93	JN=JN-1	00186000
	NUMB(IA)=JN	00186100
90	CONTINUE	00186200
	JA=0	00186300
	DO 96 JJ=1,IA	00186400
	J1=IA-JJ	00186500
96	JA=JA+NUMB(JJ)*10**J1	00186600
	IF(IP.EQ.0.OR.IP.EQ.1A) GO TO 97	00186700
	CONST(MQ1)=FG*FLOAT(JA)/10.** (IA-IP)	00186800
	GO TO 98	00186900
97	CONST(MQ1)=FG*FLOAT(JA)	00187000
98	IT=IT+1	00187100
	IFTAB(IT)=7	00187200
65	K=0	00187300
	IF(IRI.NE.1) GO TO 10	00187400
	IT=IT+1	00187500
	IFTAB(IT)=9	00187600
	IRI=0	00187700
C		00187800
	IF(LL1.NE.LL2) GO TO 10	00187900
	LL1=0	00188000
	LL2=0	00188100
	KCON=0	00188200
	IX=1	00188300
	RETURN	00188400
C		00188500
10	CONTINUE	00188600
	KCON=1	00188700
	IF(IERO.EQ.1) IT=0	00188800
C		00188900
	RETURN	00189000
	END	00189100
		00189200

[illegible]

C		00195900
	66 NOA=0	00196000
	NC=0	00196100
	NOS=0	00196200
C		00196300
	REWIND IOLD	00196400
	REWIND INDEV	00196500
	DO 10 I=1,N	00196600
C		00196700
	DO 11 J=1,MP7	00196800
	11 LOGICS(J)=LOGIC7(J)	00196900
	DO 12 J=1,IT	00197000
	12 IFTABS(J)=IFTAB7(J)	00197100
C		00197200
	READ(INDEV,END=888,ERR=777) KK,(XX(J),J=1,M1)	00197300
	NOS=NOS+1	00197400
C		00197500
	CALL CONTAB(XX,MQ7,RELAT7,CONST7,LOGICS,IFTABS,LIT,MA,MY,MB,MD7,	00197600
	* MP7,IT,MTRU)	00197700
C		00197800
	IF(MTRU.EQ.1) GO TO 15	00197900
	NOA=NOA+1	00198000
	GO TO 10	00198100
	15 NC=NC+1	00198200
C		00198300
	WRITE(IOLD) KK,(XX(J),J=1,M1)	00198400
C		00198500
	10 CONTINUE	00198600
C		00198700
	888 CONTINUE	00198800
	FN=NOS	00198900
	IF(NC.NE.0) GO TO 25	00199000
	WRITE(IOUT,640)	00199100
	640 FORMAT(1H0,3X,'--- ALL DATA ARE INSUFFICIENT, THEREFORE ',	00199200
	* 'IT IS IMPOSSIBLE TO STORE ON DISK . ---')	00199300
	RETURN	00199400
C		00199500
	25 CONTINUE	00199600
	ATRU=FLOAT(NC)/FN*100.	00199700
	AFAU=FLOAT(NOA)/FN*100.	00199800
C		00199900
	WRITE(IOUT,650)	00200000
	650 FORMAT(1H0, 1X,'----- INCONSISTENT DATA ----- (A ---- B)',	00200100
	WRITE(IOUT,651)	00200200
	651 FORMAT(1H+,42X,'>')	00200300
	WRITE(IOUT,652)	00200400
	652 FORMAT(1H0,' NO.',2X,'DATA',15X,'-- DATA X --')	00200500
C		00200600
	WRITE(IOUT,690) (ITEMP(1,J),J=1,110)	00200700
	690 FORMAT(1H 15X,110A1)	00200800
C		00200900
	INA=0	00201000
	MC=0	00201100
C		00201200
	REWIND IOLD	00201300
	DO 30 I=1,NC	00201400
C		00201500
	READ(IOLD,ERR=777) K,(XX(J),J=1,M1)	00201600
C		00201700
	DO 31 J=1,MP8	00201800
	31 LOGICS(J)=LOGIC8(J)	00201900
	DO 32 J=1,IS	00202000
	32 IFTABS(J)=IFTAB8(J)	00202100
C		00202200
	CALL CONTAB(XX,MQ8,RELAT8,CONST8,LOGICS,IFTABS,LIT,	00202300
	* MA,MY,MB,MD8,MP8,IS,MTRU)	00202400
C		00202500
	IF(MTRU.EQ.1) GO TO 35	00202600
	INA=INA+1	00202700
	INCUA(INA)=K	00202800
	IF(INA.GE.201) GO TO 27	00202900
	WRITE(IOUT,620) INA,K	00203000

620	FORMAT(1H ,I3,2X,I5)	00203100
	IA=1	00203200
	DO 70 J=1,M07	00203300
	IF(J.GT.10) GO TO 30	00203400
	IQ=MQ7(J)	00203500
	CALL FORM(NUM,KN,FMT6,IA,4,5,7)	00203600
	WRITE(IOUT,FMT6) XX(IQ)	00203700
70	IA=IA+12	00203800
	DO 75 J=1,M08	00203900
	IF(J.M07.GT.10) GO TO 30	00204000
	IQ=MQ8(J)	00204100
	CALL FORM(NUM,KN,FMT6,IA,4,5,7)	00204200
	WRITE(IOUT,FMT6) XX(IQ)	00204300
75	IA=IA+12	00204400
	GO TO 30	00204500
C		00204600
35	MC=MC+1	00204700
	IF(OUTDEV.LE.0) GO TO 30	00204800
	WRITE(OUTDEV) K,(XX(J),J=1,M1)	00204900
	GO TO 30	00205000
27	CALL INVPRI (NUM,KN,LF,INA,K,0)	00205100
C		00205200
30	CONTINUE	00205300
	IF(LF.NE.0) CALL INVPRI(NUM,KN,LF,INA,K,1)	00205400
	IF(OUTDEV.GE.1) NUMDT(OUTDEV)=MC	00205500
C		00205600
	IF(INA.GE.1) GO TO 34	00205700
	WRITE(IOUT,691)	00205800
691	FORMAT(1H0,5X,' --- ALL SAMPLES ARE CONSISTENT --- ')	00205900
C		00206000
34	BTRA=FLOAT(MC)/FLOAT(NC)*100.	00206100
	BTRN=FLOAT(MC)/FN*100.	00206200
	BFAA=FLOAT(INA)/FLOAT(NC)*100.	00206300
	BFAN=FLOAT(INA)/FN*100.	00206400
C		00206500
	IF(IFFSW.EQ.1) GO TO 90	00206600
C		00206700
	WRITE(IOUT,700) N,NOS	00206800
700	FORMAT(1H1,///1X,'SUMMARY TABLE'/1X,13('-'),	00206900
	* ///11X,'TOTAL NUMBER OF SAMPLES TN =',I6,	00207000
	* ///11X,'NUMBER OF SAMPLES N =',I6)	00207100
	WRITE(IOUT,705)	00207200
705	FORMAT(1H0,12X,'IF (A) , THEN (B)')	00207300
	WRITE(IOUT,710) NC,ATRU,NOA,AFAU	00207400
710	FORMAT(1H0,15X,'A --- TRUE =',I7,2X,F6.2,' % (1/N)'/	00207500
	* 22X,'FALSE = ', I5,2X,F6.2,' % (1/N)')	00207600
	WRITE(IOUT,720) MC,BTRA,BTRN,INA,BFAA,BFAN	00207700
720	FORMAT(1H0,15X,'A --- TRUE , B --- TRUE =',	00207800
	* I7,2X,F6.2,' %',3X,F6.2,' % (1/N)'/ 35X,'FALSE = ',	00207900
	* I5,2X,F6.2,' %',3X,F6.2,' % (1/N)')	00208000
	RETURN	00208100
C		00208200
C		00208300
90	CONTINUE	00208400
	NOB=0	00208500
	IC=0	00208600
C		00208700
	REWIND IOLD	00208800
	REWIND INDEV	00208900
	DO 40 I=1,N	00209000
C		00209100
	DO 41 J=1,MP8	00209200
41	LOGICS(J)=LOGIC8(J)	00209300
	DO 42 J=1,IS	00209400
42	IFTABS(J)=IFTAB8(J)	00209500
C		00209600
	READ(INDEV,END=999,ERR=777) KK,(XX(J),J=1,M1)	00209700
C		00209800
	CALL CONTAB(XX,MQ8,RELAT8,CONST8,LOGICS,IFTABS,LIT,MA,MY,MB,M08,	00209900
	* MP8,IS,MTRU)	00210000

C	IF(MTRU.EQ.1) GO TO 45	00210100
	NOB=NOB+1	00210200
	GO TO 40	00210300
45	IC=IC+1	00210400
C	WRITE(IOLD) KK,(XX(J),J=1,M1)	00210500
C		00210600
		00210700
C	40 CONTINUE	00210800
		00210900
C		00211000
999	CONTINUE	00211100
	IF(IC.NE.0) GO TO 46	00211200
	WRITE(IOUT,640)	00211300
	RETURN	00211400
C		00211500
46	CONTINUE	00211600
	BTRU=FLOAT(IC)/FN*100.	00211700
	BFAU=FLOAT(NOB)/FN*100.	00211800
C		00211900
	WRITE(IOUT,660)	00212000
660	FORMAT(1H1,///2X,'----- INCONSISTENT DATA ----- (B ---- A)')	00212100
	WRITE(IOUT,651)	00212200
	WRITE(IOUT,652)	00212300
	WRITE(IOUT,690) (ITEMP(1,J),J=1,110)	00212400
C		00212500
	INB=0	00212600
	JC=0	00212700
C		00212800
	REWIND IOLD	00212900
	DO 50 I=1,IC	00213000
C		00213100
	READ(IOLD,ERR=777) K,(XX(J),J=1,M1)	00213200
C		00213300
	DO 51 J=1,MP7	00213400
51	LOGICS(J)=LOGIC7(J)	00213500
	DO 52 J=1,IT	00213600
52	IFTABS(J)=IFTAB7(J)	00213700
C		00213800
	CALL CONTAB(XX,MQ7,RELAT7,CONAT7,LOGICS,IFTABS,LIT,MA,MY,MB,MO7,	00213900
	* MP7,IT,MTRU)	00214000
C		00214100
	IF(MTRU.EQ.1) GO TO 55	00214200
	INB=INB+1	00214300
	INCOB(INB)=K	00214400
	IF(INB.GE.201) GO TO 48	00214500
	WRITE(IOUT,620) INB,K	00214600
	IA=1	00214700
	DO 80 J=1,MO7	00214800
	IF(J.GT.10) GO TO 50	00214900
	IQ=MQ7(J)	00215000
	CALL FORM(NUM,KN,FMT6,IA,4,5,7)	00215100
	WRITE(IOUT,FMT6) XX(IQ)	00215200
80	IA=IA+12	00215300
	DO 85 J=1,MO8	00215400
	IF(J*MO7.GT.10) GO TO 50	00215500
	IQ=MQ8(J)	00215600
	CALL FORM(NUM,KN,FMT6,IA,4,5,7)	00215700
	WRITE(IOUT,FMT6) XX(IQ)	00215800
85	IA=IA+12	00215900
	GO TO 50	00216000
55	JC=JC+1	00216100
	GO TO 50	00216200
C		00216300
48	CALL INVPRI(NUM,KN,LG,INB,K,0)	00216400
50	CONTINUE	00216500
C		00216600
	IF(LG.NE.0) CALL INVPRI (NUM,KN,LG,INB,K,1)	00216700
	IF(INB.GE.1) GO TO 36	00216800
	WRITE(IOUT,691)	00216900
36	INAB=INA+INB	00217000
	IF(INA.GT.200.OR.INB.GT.200) GO TO 6	00217100
	IF(INAB.LE.0) GO TO 28	00217200

J=INA	00217300
DO 95 I=1,INB	00217400
J=J+1	00217500
95 INCOA(J)=INCOB(I)	00217600
I1=J-1	00217700
DO 96 I=1,I1	00217800
K1=I+1	00217900
DO 96 K=K1,J	00218000
IF(INCOA(I).LT.INCOA(K)) GO TO 96	00218100
I1=INCOA(I)	00218200
INCOA(I)=INCOA(K)	00218300
INCOA(K)=I1	00218400
96 CONTINUE	00218500
C	00218600
28 WRITE(IOUT,670)	00218700
670 FORMAT(1H1,///2X,'----- INCONSISTENT DATA NUMBER -----',	00218800
* 5X,'(A --- B , B --- A)')	00218900
WRITE(IOUT,671)	00219000
671 FORMAT(1H+,48X,'>',9X,'>')/)	00219100
IF(INAB.GT.1) GO TO 29	00219200
WRITE(IOUT,691)	00219300
GO TO 6	00219400
29 CONTINUE	00219500
C	00219600
JS=J/15	00219700
JA=J-JS*15	00219800
JE=JS+1	00219900
C	00220000
J1=1	00220100
DO 91 JJ=1,JE	00220200
C	00220300
IF(JJ.NE.JE) GO TO 5	00220400
IF(JA.EQ.0) GO TO 6	00220500
C	00220600
5 IF(JJ.EQ.JE) GO TO 92	00220700
IF(JJ+1.NE.JE) GO TO 93	00220800
IF(JA.NE.0) GO TO 93	00220900
92 J2=J	00221000
GO TO 94	00221100
93 J2=JJ*15	00221200
94 IF(JJ.EQ.JE) GO TO 97	00221300
KP=23	00221400
GO TO 98	00221500
97 KP=JA*6/4+1	00221600
C	00221700
98 WRITE(IOUT,680) (POINT(I),I=1,KP)	00221800
680 FORMAT(1H ,5X,14(' '),23A4)	00221900
WRITE(IOUT,681) (I,I=J1,J2)	00222000
681 FORMAT(1H ,9X,'LOGIC',5X,15I6)	00222100
WRITE(IOUT,680) (POINT(I),I=1,KP)	00222200
WRITE(IOUT,682)	00222300
682 FORMAT(1H ,6X,'(A --- B)')	00222400
WRITE(IOUT,683)	00222500
683 FORMAT(1H+,12X,'>')	00222600
C	00222700
IA=2	00222800
DO 81 I=J1,J2	00222900
DO 82 K=1,INB	00223000
IF(INCOA(I).EQ.INCOB(K)) GO TO 81	00223100
82 CONTINUE	00223200
CALL FORM(NUM,KN,FMT7,IA,4,5,7)	00223300
WRITE(IOUT,FMT7).INCOA(I)	00223400
81 IA=IA+6	00223500
C	00223600
WRITE(IOUT,684)	00223700
684 FORMAT(1H ,6X,'(B --- A)')	00223800
WRITE(IOUT,683)	00223900
IA=2	00224000
DO 83 I=J1,J2	00224100
DO 84 K=1,INB	00224200
IF(INCOA(I).EQ.INCOB(K)) GO TO 86	00224300

84	CONTINUE	00224400
	GO TO 83	00224500
86	CALL FORM(NUM,KN,FMT7,IA,4,5,7)	00224600
	WRITE(IOUT,FMT7) INCOA(I)	00224700
83	IA=IA+6	00224800
C		00224900
	WRITE(IOUT,685)	00225000
685	FORMAT(//)	00225100
C		00225200
91	J1=J2+1	00225300
C		00225400
6	CONTINUE	00225500
	ATRB=FLOAT(JC)/FLOAT(IC)*100.	00225600
	ATRN=FLOAT(JC)/FN*100.	00225700
	AFAB=FLOAT(INB)/FLOAT(IC)*100.	00225800
	AFAN=FLOAT(INB)/FN*100.	00225900
	IOR=NC+IC-MC	00226000
	IAND=MC	00226100
	INOT=IOR-IAND	00226200
	AORN=FLOAT(IOR)/FN*100.	00226300
	ANDA=FLOAT(IAND)/FLUAT(IOR)*100.	00226400
	ANDN=FLOAT(IAND)/FN*100.	00226500
	ANOA=FLOAT(INOT)/FLOAT(IOR)*100.	00226600
	ANON=FLOAT(INOT)/FN*100.	00226700
C		00226800
	WRITE(IOUT,700) N,NOS	00226900
	WRITE(IOUT,706)	00227000
706	FORMAT(IH0,12X,'(A) IFF (B)')/	00227100
	WRITE(IOUT,710) NC,ATRU,NOA,AFAU	00227200
	WRITE(IOUT,720) MC,BTRA,BTRN,INA,BFAA,BFAN	00227300
	WRITE(IOUT,730) IC,BTRU,NOB,BFAU	00227400
730	FORMAT(IH0,15X,'B --- TRUE =' ,I7,2X,F6.2,' % (1/N)'/	00227500
	* 22X,'FALSE = ' , I5,2X,F6.2,' % (1/N)')	00227600
	WRITE(IOUT,740) JC,ATRB,ATRN,INB,AFAB,AFAN	00227700
740	FORMAT(IH0,15X,'B --- TRUE , A --- TRUE =' ,	00227800
	* I7,2X,F6.2,' % ',3X,F6.2,' % (1/N)'/ 35X,'FALSE = ' ,	00227900
	* I5,2X,F6.2,' % ',3X,F6.2,' % (1/N)')	00228000
	WRITE(IOUT,750) IOR,AORN	00228100
750	FORMAT(IH0,15X,'NUMBERS OF (A OR B) =' ,	00228200
	* I5,2X,F6.2,' % (1/N)')	00228300
	WRITE(IOUT,760) IAND,ANDA,ANDN	00228400
760	FORMAT(IH0,15X,'NUMBERS OF (A AND B) =' ,	00228500
	* I5,2X,F6.2,' % ',3X,F6.2,' % (1/N)')	00228600
	WRITE(IOUT,770) INOT,ANOA,ANON	00228700
770	FORMAT(IH0,15X,'NUMBERS OF < NOT (A AND B) IN (A OR B) > =' ,	00228800
	* I5,2X,F6.2,' % ',3X,F6.2,' % (1/N)')	00228900
C		00229000
	RETURN	00229100
C		00229200
777	WRITE(IOUT,900)	00229300
900	FORMAT(IH1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ' ,	00229400
	* 'SPECIFIED BY COMMAND ---')	00229500
	RETURN	00229600
	END	00229700

	SUBROUTINE CONTAB(XX,MQ,RELAT,CONST,LOGIC,IFTAB,LIT,MA,MY,MB,MQ1,	00229800
	* MPI,IT,MTRU)	00229900
C		00230000
C	TO EXECUTE LOGICAL CHECKING FOR CONSISTENCY.	00230100
C		00230200
	COMMON / OUTIN / INPT, IOUT	00230300
	INTEGER RELAT(MY)	00230400
	DIMENSION XX(MA),MQ(MY),CONST(MY),LOGIC(MY),IFTAB(MB),LIT(MY)	00230500
C		00230600
C		00230700
	DO 10 J=1,MQ1	00230800
	K=MQ(J)	00230900
	C=CONST(J)	00231000
	JR=RELAT(J)	00231100
	GO TO (1,2,3,4,5,6),JR	00231200
	WRITE(IOUT,600) J,RELAT(J)	00231300
	600 FORMAT(1H0,'--- RELATIONAL OPERATOR ERROR ---'//	00231400
	* 5X,'RELAT('',12,'')=',13)	00231500
	RETURN	00231600
C		00231700
	1 IF(XX(K),LT,C) GO TO 12	00231800
	ITRU=0	00231900
	GO TO 20	00232000
	12 ITRU=1	00232100
	GO TO 20	00232200
C		00232300
	2 IF(XX(K),LE,C) GO TO 13	00232400
	ITRU=0	00232500
	GO TO 20	00232600
	13 ITRU=1	00232700
	GO TO 20	00232800
C		00232900
	3 IF(XX(K),EQ,C) GO TO 14	00233000
	ITRU=0	00233100
	GO TO 20	00233200
	14 ITRU=1	00233300
	GO TO 20	00233400
C		00233500
	4 IF(XX(K),NE,C) GO TO 15	00233600
	ITRU=0	00233700
	GO TO 20	00233800
	15 ITRU=1	00233900
	GO TO 20	00234000
C		00234100
	5 IF(XX(K),GT,C) GO TO 16	00234200
	ITRU=0	00234300
	GO TO 20	00234400
	16 ITRU=1	00234500
	GO TO 20	00234600
C		00234700
	6 IF(XX(K),GE,C) GO TO 17	00234800
	ITRU=0	00234900
	GO TO 20	00235000
	17 ITRU=1	00235100
C		00235200
	20 CONTINUE	00235300
	DO 30 L=1,IT	00235400
	IF(IFTAB(L),EQ,7) GO TO 35	00235500
	30 CONTINUE	00235600
	GO TO 10	00235700
	35 IFTAB(L)=ITRU	00235800
	10 CONTINUE	00235900

C		00236000
	IF(MP1.GT.0) GO TO 33	00236100
	MTRU=ITRU	00236200
	RETURN	00236300
33	MT=IT	00236400
	MP2=MP1	00236500
C		00236600
	99 CONTINUE	00236700
C		00236800
	LO=0	00236900
	LN=0	00237000
	DO 40 J=1,MT	00237100
	IF(IFTAB(J).NE.6) GO TO 42	00237200
	LN=0	00237300
	I1=J	00237400
	GO TO 40	00237500
42	IF(IFTAB(J).NE.9) GO TO 44	00237600
	I2=J	00237700
	GO TO 50	00237800
44	IF(IFTAB(J).NE.8) GO TO 40	00237900
	LO=LO+1	00238000
	LN=LN+1	00238100
	LIT(LO)=J	00238200
40	CONTINUE	00238300
C		00238400
	50 IF(LN.EQ.1) GO TO 70	00238500
	LP=LO-LN+1	00238600
	DO 60 L=LP,LO	00238700
	IF(LOGIC(L).EQ.2) GO TO 62	00238800
60	CONTINUE	00238900
	LJ=LP	00239000
	GO TO 64	00239100
62	LJ=L	00239200
64	L1=LIT(LJ)-1	00239300
	L2=LIT(LJ)+1	00239400
C		00239500
	CALL NEWTAB(LOGIC,IFTAB,MY,MB,L1,L2,L1,L2,MP2,MT,LJ,MEND)	00239600
C		00239700
	GO TO 80	00239800
C		00239900
	70 L1=LIT(LO)-1	00240000
	L2=LIT(LO)+1	00240100
C		00240200
	CALL NEWTAB(LOGIC,IFTAB,MY,MB,L1,L2,I1,I2,MP2,MT,LO,MEND)	00240300
C		00240400
	80 IF(MEND.NE.1) GO TO 99	00240500
	MTRU=IFTAB(1)	00240600
	RETURN	00240700
	END	00240800
	SUBROUTINE NEWTAB(LOGIC,IFTAB,MY,MB,L1,L2,I1,I2,MP1,IT,LJ,MEND)	00240900
	DIMENSION LOGIC(MY),IFTAB(MB)	00241000
C		00241100
	MEND=0	00241200
	IF(LOGIC(LJ).EQ.1) GO TO 10	00241300
	LL=IFTAB(L1)*IFTAB(L2)	00241400
	GO TO 20	00241500
10	LL=IFTAB(L1)+IFTAB(L2)	00241600
	IF(LL.GT.1) LL=1	00241700
C		00241800
	20 IFTAB(I1)=LL	00241900
	IF(I1.NE.1) GO TO 30	00242000
	MEND=1	00242100
	RETURN	00242200
C		00242300
	30 JT=I1	00242400
	J1=I2+1	00242500
	DO 40 J=J1,IT	00242600
	JT=JT+1	00242700
40	IFTAB(JT)=IFTAB(J)	00242800
C		00242900
	IT=JT	00243000
	MP1=MP1-1	00243100
	DO 50 I=1,MP1	00243200
	IF(I.GE.LJ) LOGIC(I)=LOGIC(I+1)	00243300
50	CONTINUE	00243400
	RETURN	00243500
	END	00243600

```

      SUBROUTINE PATTRN(XX,MQ,PP,NS,Q,MM,ITEMP,NUM,N,KWORK,NN,MA,MB,IA1,00243700
      *      IA2,MPA,M1,INDEV,BLANK ) 00243800
C 00243900
C THIS SUBROUTINE CARRIES OUT A PATTERN PROCEDURE. 00244000
C 00244100
      COMMON / OUTIN / INPT, IOUT 00244200
      COMMON NUMDT(20) 00244300
      INTEGER FMP(11),FMQ(18),FMR(9),FMO(9),FMT(7),FMU(8), 00244400
      * Q(MB),BLANK 00244500
      DIMENSION XX(MA),MQ(MA),PP(KWORK,MPA),NS(NN),NUM(10), 00244600
      * MM(MA),ITEMP(IA1,IA2) 00244700
      DATA FMP/'(1H','3X','4H'PAT,'TERN','4H','I4','2X',' ',' ', 00244800
      * 'F5.0','8X','15)'/, 00244900
      * FMQ/'(///','4H4X','PATT','ERN','4HNO',' ','6X','4H'---, 00245000
      * 'PAT','TERN',' ---','4H','10,4HX','-', '-- P','AGE','4H','I1, 00245100
      * 4H',' -','4H'---','2H ) /, 00245200
      * FMR/'(1H','17X',' ',' ',' ','X,4X',' ','10('4H'.')1H)/, 00245300
      * FMO/'(1H','17X',' ',' ',' ','X,5X','4H','N','4H='I','5)'/, 00245400
      * FMT/'(36X'1H',' ',' ','4HX','T','OTAL','2H') /, 00245500
      * FMU/'(1H+' ','60X','4H','-(','CON','TINU','ED )','4H'---','2H ) / 00245600
C 00245700
      N1=0 00245800
      NSAM=0 00245900
      DO 10 I=1,NN 00246000
10 NS(I)=1 00246100
      DO 1 I=1,2 00246200
      DO 1 J=1,600 00246300
1 ITEMP(I,J)=BLANK 00246400
C 00246500
      WRITE(IOUT,800) 00246600
800 FORMAT(1H1,///1X,'+-----+ '/ 00246700
      * 1X,'| PATTERN CHECK |' / 00246800
      * 1X,'+-----+ ') 00246900
C 00247000
      KNUM=NUMDT(INDEV) 00247100
      IF(KNUM.EQ.0) GO TO 1000 00247200
C 00247300
      I=0 00247400
      IR=1 00247500
      L1=1 00247600
      DO 90 J=1,MPA,2 00247700
      CALL ITQ(Q,MM,MQ,ITEMP,MB,MA,IA1,IA2,I,IR,L1,J,M1) 00247800
      I=I-2 00247900
90 CONTINUE 00248000
      I=5 00248100
      IR=IR+1 00248200
      L1=1 00248300
      DO 91 J=2,MPA,2 00248400
      CALL ITQ(Q,MM,MQ,ITEMP,MB,MA,IA1,IA2,I,IR,L1,J,M1) 00248500
      I=I-2 00248600
91 CONTINUE 00248700
C 00248800
      REWIND INDEV 00248900
      DO 20 I=1,N 00249000
C 00249100
      READ(INDEV,END=99,ERR=777) KK,(XX(J),J=1,M1) 00249200
      NSAM=NSAM+1 00249300
      IF(I.EQ.1) GO TO 25 00249400
C 00249500
      DO 30 L=1,N1 00249600
      DO 40 J=1,MPA 00249700
      K=MQ(J) 00249800
      IF(PP(L,J).NE.XX(K)) GO TO 30 00249900
40 CONTINUE 00250000
      NS(L)=NS(L)+1 00250100
      GO TO 20 00250200
30 CONTINUE 00250300
C 00250400
      25 N1=N1+1 00250500
      IF(N1.GT.1024) GO TO 89 00250600
      DO 50 J=1,MPA 00250700
      K=MQ(J) 00250800
      50 PP(N1,J)=XX(K) 00250900
C 00251000
      20 CONTINUE 00251100

```

C		00251200
	99 I1=N1-1	00251300
	DO 60 I=1,I1	00251400
	J1=I+1	00251500
	DO 60 J=J1,N1	00251600
	IF(NS(I).GE.NS(J)) GO TO 60	00251700
	DO 70 L=1,MPA	00251800
	P=PP(I,L)	00251900
	PP(I,L)=PP(J,L)	00252000
	70 PP(J,L)=P	00252100
	NSP=NS(I)	00252200
	NS(I)=NS(J)	00252300
	NS(J)=NSP	00252400
	60 CONTINUE	00252500
C		00252600
	JJJ=1	00252700
	KKK=1	00252800
	IJK=MPA	00252900
	ICO=0	00253000
	100 CUNTINUE	00253100
C		00253200
	ICO=ICO+1	00253300
	KJI=IJK-KKK+1	00253400
	MPB=(KJI-2)*5	00253500
	IF(KJI.GT.18) MPB=80	00253600
	IF(MPB.LE.0) MPB=5	00253700
	CALL FORM(NUM,10,FMT,MPB,3,4,7)	00253800
	WRITE(IOUT,FMQ) ICO	00253900
	WRITE(IOUT,FMT)	00254000
C		00254100
	IF((IJK-KKK+1) .GT. 18) IJK=KKK+17	00254200
	III=10*((IJK+1)/2)	00254300
C		00254400
	DO 92 I=1,IR	00254500
	WRITE(IOUT,610) (ITEMP(I,J),J=JJJ,III)	00254600
610	FORMAT(1H ,18X,110A1)	00254700
	III=III+5	00254800
	92 CONTINUE	00254900
C		00255000
	IABC=IJK-KKK+1	00255100
	CALL FORM(NUM,10,FMP,IABC,7,8,11)	00255200
	DO 80 I=1,N1	00255300
	WRITE(IOUT,FMP) I,(PP(I,J),J=KKK,IJK),NS(I)	00255400
	IF(I.EQ.N1) GO TO 80	00255500
	IF(I.NE.(I/50*50)) GO TO 80	00255600
C		00255700
	600 FORMAT(1H1/)	00255800
	WRITE(IOUT,600)	00255900
	WRITE(IOUT,FMQ) ICO	00256000
	WRITE(IOUT,FMU)	00256100
	WRITE(IOUT,FMT)	00256200
	III=III-10	00256300
	DO 93 II=1,IR	00256400
	WRITE(IOUT,610) (ITEMP(II,J),J=JJJ,III)	00256500
	III=III+5	00256600
	93 CONTINUE	00256700
	80 CONTINUE	00256800

C	MPC=(IJK-KKK+1)*5	00256900
	CALL FORM(NUM,10,FMR,MPC,4,5,9)	00257000
	CALL FORM(NUM,10,FMO,MPC,4,5,9)	00257100
	WRITE(IOUT,FMR)	00257200
	WRITE(IOUT,FMO) NSAM	00257300
	WRITE(IOUT,FMR)	00257400
		00257500
C	IF(IJK.GE.MPA) GO TO 110	00257600
	WRITE(IOUT,600)	00257700
	JJJ=III-9	00257800
	KKK=IJK+1	00257900
	IJK=MPA	00258000
	ITEMP(2,III)=BLANK	00258100
	ITEMP(2,III+1)=BLANK	00258200
	ITEMP(2,III+2)=BLANK	00258300
	GO TO 100	00258400
110	CONTINUE	00258500
C	RETURN	00258600
		00258700
C	89 WRITE(IOUT,6000)	00258800
	6000 FORMAT(///10X,'--- TOTAL NUMBER OF PATTERNS IS MORE THAN 1024.',	00258900
	* ' ---')	00259000
	RETURN	00259100
C	1000 WRITE(IOUT,6001) INDE	00259200
	RETURN	00259300
	6001 FORMAT(///1H,'--- FILE NUMBER IS',I3,'. THIS FILE DOES NOT',	00259400
	* ' INCLUDE DATA ---')	00259500
C	777 WRITE(IOUT,900)	00259600
	900 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00259700
	* 'SPECIFIED BY COMMAND ---')	00259800
	RETURN	00259900
	END	00260000
		00260100
		00260200
		00260300
		00260400

```

      SUBROUTINE MULTAB(WORK,IWORK,LIMIT1,LIMIT2,OPT,Q,MM,MB,XX,MQ,AA, 00260500
      *      II,MQ1,M1,MA,MX,INDEV ) 00260600
C 00260700
C THIS SUBROUTINE IMPLEMENTS THE LOG-LINEAR MODEL BY ITERATIVE 00260800
C PROPORTIONAL FITTING PROCEDURE. AND A NEW PROCEDURE OF MODEL 00260900
C SELECTION BASED ON AIC(AKAIKE'S INFORMATION CRITERION) IS INTRODUCED. 00261000
C BY USING THIS PROCEDURE, REASONABLE MODELS ARE SELECTED ALMOST 00261100
C AUTOMATICALLY AND EASILY. 00261200
C THREE SUBROUTINES IN MULTAB(ADJUST, COLLAP, LOGLIN) ARE DERIVED FROM 00261300
C ALGORITHM AS 51 APPL. STATIST. (1972),VOL.21,NO.2 BY HABERMAN. 00261400
C ANOTHER SUBROUTINES (SPECIALLY THE METHOD OF MODEL-SELECTION BY 00261500
C STEPWISE-PROCEDURE) ARE WRITTEN BY Y.YANAGISAWA AND N.OHSUMI. 00261600
C 00261700
      REAL*8      DEVMAX 00261800
      COMMON      NUMDT(20) 00261900
      COMMON / OPTION /      IOPT 00262000
      COMMON / METHOD /      METH 00262100
      COMMON / OUTIN /      LIN, LOUT 00262200
      INTEGER      SLASH,BLANK,COMMA 00262300
      INTEGER      AA(100,80),OPT(10) 00262400
      DIMENSION     WORK(LIMIT1), IWORK(LIMIT2), 00262500
      *      XX(MA),MQ(MX),IMT(10,8),Q(MB),MM(MA) 00262600
      DATA SLASH/' '/' ,BLANK/' ' ,COMMA/' ' , 00262700
C 00262800
      NSAMP=NUMDT(INDEV) 00262900
      NVAR=MQ1 00263000
      IOPT=OPT(1) 00263100
      METH=OPT(2) 00263200
      MAXIT=OPT(3) 00263300
      DEVMAX=OPT(4) 00263400
C 00263500
      DO 10 I = 1, LIMIT1 00263600
      WORK(I) = 0.0 00263700
10 CONTINUE 00263800
      DO 20 I = 1, LIMIT2 00263900
      IWORK(I) = 0 00264000
20 CONTINUE 00264100
C 00264200
C SET THE DEFAULT VALUES 00264300
C 00264400
      IF( ( NVAR .EQ. 2 .OR. NVAR .EQ. 3 ) .AND. ( METH .LE. 0 ) ) 00264500
      *      METH = 2 00264600
      IF( ( NVAR .EQ. 4 ) .AND. ( METH .LE. 0 ) ) 00264700
      *      METH = 2 00264800
      IF( METH .LE. 1 )      METH = 4 00264900
      IF( METH .GE. 4 )      METH = 4 00265000
      IF( ( METH .NE. 1 ) .AND. ( NVAR .GE. 5 ) ) 00265100
      *      METH = 4 00265200
      IF( NTAPE .LE. 0 )      NTAPE = LIN 00265300
      IF( MAXIT .LE. 0 )      MAXIT = 20 00265400
      IF( DEVMAX .LE. 0.000 ) DEVMAX = 0.000100 00265500
C 00265600
      NTITLE = 80 00265700
      NUM1 = 1 00265800
      DO 50 I=NUM1,NTITLE 00265900
      IF(AA(II,I).EQ.SLASH) GO TO 60 00266000
      IWORK(I) = AA(II,I) 00266100
      GO TO 50 00266200
60 IWORK(I) = BLANK 00266300
50 CONTINUE 00266400
C 00266500
      WRITE( LOUT, 6000) 00266600
      WRITE(LOUT, 6100) 00266700
      WRITE(LOUT, 6200) 00266800
      WRITE(LOUT, 6300) 00266900
      WRITE(LOUT,6200) 00267000

```

C	KNUM=NUMDT(INDEV)	00267100
	IF(KNUM.EQ.0) GO TO 1000	00267200
	K = 0	00267300
	KCODE = 1	00267400
	DO 70 I = 16, 80	00267500
	IF(AA(I,I) .EQ. BLANK .OR. AA(I,I) .EQ. COMMA	00267600
	* .OR. AA(I,I) .EQ. SLASH) GO TO 75	00267700
	K = K + 1	00267800
	GO TO 70	00267900
75	IF(K .EQ. 0) GO TO 70	00268000
	K1 = I - K	00268100
	K2 = I - 1	00268200
	DO 80 J = 1, 8	00268300
80	IMT(KCODE,J)=BLANK	00268400
	J = 1	00268500
	DO 85 L = K1, K2	00268600
	IMT(KCODE,J)=AA(I,L)	00268700
85	J = J + 1	00268800
	WRITE(LOUT,6400) (IMT(KCODE,J),J=1,8),KCODE	00268900
	KCODE = KCODE + 1	00269000
	K = 0	00269100
70	CONTINUE	00269200
C	WRITE(LOUT, 6200)	00269300
	GO TO (100, 140, 130, 120), METH	00269400
120	WRITE(LOUT, 6520)	00269500
	GO TO 100	00269600
130	WRITE(LOUT, 6530)	00269700
	GO TO 100	00269800
140	WRITE(LOUT, 6540)	00269900
100	CONTINUE	00270000
C	NUM2 = NUM1 + NTITLE	00270100
	LIMIT3 = LIMIT2 - NUM2 + 1	00270200
	CALL CCONT2(NSAMP,NVAR,MAXIT,DEVMAX,Q,MM,MB,LIMIT1,LIMIT3,NTITLE,	00270300
	* NUM2,IWORK(1),IWORK(NUM2),WORK(1),XX,MQ,M1,MA,MX,	00270400
	* INDEV,IMT)	00270500
C	RETURN	00270600
C	1000 WRITE(LOUT,6001) INDEV	00270700
	RETURN	00270800
C	6000 FORMAT(1H1,///1X,'+-----+')	00270900
	* 1X,' MULTIDIMENSIONAL '	00271000
	* 1X,' CONTINGENCY '	00271100
	* 1X,' TABLE '	00271200
	* 1X,'+-----+')	00271300
	6100 FORMAT(1H0,3X,'----- LIST OF ITEMS -----')	00271400
	6200 FORMAT(1H ,6X,19('.'))	00271500
	6300 FORMAT(1H ,8X,'ITEM',7X,'CODE')	00271600
	6400 FORMAT(1H ,8X,8A1,5X,11)	00271700
	6520 FORMAT(///1X,'STEP-WISE METHOD'/1X,16('-'//)	00271800
	6530 FORMAT(///1X,'COMBINATORIAL METHOD'/1X,20('-'//)	00271900
	6540 FORMAT(///1X,'ORDINARY METHOD'/1X,15('-'//)	00272000
	6001 FORMAT(///1H ,'--- FILE NUMBER IS',I3,'. THIS FILE DOES NOT',	00272100
	* ' INCLUDE ---')	00272200
C	END	00272300

```

SUBROUTINE CCONT2(NSAMP,NVAR,MAXIT,DEVMAX,Q,MM,MB,LIMIT1,LIMIT3, 00273100
*      NTITLE,JNUM2,TITLE,IWORK,WORK,XX,MQ,M1,MA,MX, 00273200
*      INDEV,ITM ) 00273300
REAL*8 00273400
COMMON / METHOD / METH 00273500
COMMON / OUTIN / LIN,LOUT 00273600
COMMON / OPTION / IOPT 00273700
INTEGER TITLE,Q,QQ 00273800
DIMENSION TITLE(NTITLE),XX(MA),MQ(MX),Q(MB),MM(MA),IWORK(LIMIT3), 00273900
*      FMTI(80),QQ(8),ITM(10,8) 00274000
DATA QQ/ 8*' ' / 00274100
C 00274200
CALL CROSTB(NSAMP,NVAR,NTAB,INDEV,ZZZ,XX,MQ,MA,MX,M1,WORK(1), 00274300
*      WORK(NVAR+1),WORK(2*NVAR+1),WORK(3*NVAR+1),IWORK(1), 00274400
*      KK ) 00274500
C 00274600
IF(ZZZ.EQ.0) GO TO 5 00274700
L1=1 00274800
CALL PITEM(MM,MA,M1,KK,L1,L2,KP) 00274900
DO 15 L=L1,L2 00275000
QQ(L)=Q(L) 00275100
15 CONTINUE 00275200
WRITE(LOUT,6300) (QQ(L),L=1,8) 00275300
RETURN 00275400
C 00275500
5 NVAR3=3*NVAR 00275600
DO 10 I = 1, NTAB 00275700
WORK(I) = WORK(I + NVAR3) 00275800
10 CONTINUE 00275900
NTAB1 = NTAB + 1 00276000
NTAB2 = NTAB + NVAR3 00276100
C 00276200
DO 20 I = NTAB1, NTAB2 00276300
WORK(I) = 0.000 00276400
20 CONTINUE 00276500
C 00276600
C 00276700
C 00276800
ICHECK = 0 00276900
DO 60 I = 1, NTAB 00277000
IF( WORK(I) .LE. 0.000 ) ICHECK = 1 00277100
60 CONTINUE 00277200
ERROR = 0.000 00277300
IF( ICHECK .EQ. 1 ) ERROR = 0.500 00277400
DO 70 I = 1, NTAB 00277500
WORK(I) = WORK(I) + ERROR 00277600
70 CONTINUE 00277700
C 00277800
CALL MAXMAR(NVAR, IWORK(1), NMAR ) 00277900
NU = NMAR 00278000
NCON = 2 * NVAR - 1 00278100
NVAR1 = NVAR - 1 00278200
IF( NVAR1 .LE. 0 ) NVAR1 = 1 00278300
KDIM = IWORK(NVAR1) 00278400
IF( METH .EQ. 1 ) MAXMOD = 100 00278500
IF( ( METH .EQ. 2 ) .AND. ( NVAR .EQ. 2 ) ) MAXMOD = 5 00278600
IF( ( METH .EQ. 2 ) .AND. ( NVAR .EQ. 3 ) ) MAXMOD = 19 00278700
IF( ( METH .EQ. 2 ) .AND. ( NVAR .EQ. 4 ) ) MAXMOD = 167 00278800
IF( METH .GE. 3 ) MAXMOD = MAX0( 100, 3 * NCON ) 00278900
ITERAT = 6 00279000
ITER1 = NVAR * NCON + 2 00279100
KNCON1 = NCON - NVAR 00279200
IF( KNCON1 .LE. 0 ) KNCON1 = 1 00279300
KNCON2 = 2 * KNCON1 - 1 00279400
IF( METH .NE. 4 ) GO TO 80 00279500
KNCON1 = 1 00279600
KNCON2 = NCON 00279700

```

	80 CONTINUE	00279800
C	NUM1 = 1	00279900
	NUM2 = NUM1 + NTAB	00280000
	NUM3 = NUM2 + NTAB	00280100
	NUM4 = NUM3 + NMAR	00280200
	NUM5 = NUM4 + NU	00280300
	NUM6 = NUM5 + MAXIT	00280400
	NUM7 = NUM6 + KDIM	00280500
	NUM8 = NUM7 + KDIM	00280600
	NUM9 = NUM8 + NCON	00280700
	NUM10 = NUM9 + MAXMOD * ITERAT	00280800
	MLIMIT = 2 * NUM10 - 1	00280900
C	INUM1 = 1	00281000
	INUM2 = INUM1 + NVAR	00281100
	INUM3 = INUM2 + NVAR	00281200
	INUM4 = INUM3 + NVAR	00281300
	INUM5 = INUM4 + NVAR * NCON	00281400
	INUM6 = INUM5 + NCON	00281500
	INUM7 = INUM6 + NVAR * NCON	00281600
	INUM8 = INUM7 + NCON	00281700
	INUM9 = INUM8 + NCON	00281800
	INUM10 = INUM9 + NCON	00281900
	INUM11 = INUM10 + NCON	00282000
	INUM12 = INUM11 + KNCON1 * KNCON2	00282100
	IF(METH .EQ. 4) INUM12 = INUM11 + 1	00282200
	INUM13 = INUM12 + KNCON2	00282300
	INUM14 = INUM13 + ITER1 * MAXMOD + JNUM2	00282400
	LIMIT4 = LIMIT1 - 2 * NUM10	00282500
	LIMIT5 = LIMIT3 - INUM14	00282600
	IF((LIMIT4 .LT. 0) .OR. (LIMIT5 .LT. 0)) GO TO 90	00282700
C	CALL MCONTB(NVAR, NCON, NMAR, NU, MAXIT, DEVMAX, NTAB,	00282800
	* NTITLE, ITERAT, ITER1, MAXMOD, NVAR1, KDIM,	00282900
	* TITLE, FMT1, KNCON1, KNCON2,	00283000
	* WORK(NUM1), WORK(NUM2), WORK(NUM3), WORK(NUM4),	00283100
	* WORK(NUM5), WORK(NUM6), WORK(NUM7), WORK(NUM8),	00283200
	* WORK(NUM9),	00283300
	* IWORK(INUM1), IWORK(INUM2), IWORK(INUM3),	00283400
	* IWORK(INUM4), IWORK(INUM5), IWORK(INUM6),	00283500
	* IWORK(INUM7), IWORK(INUM8), IWORK(INUM9),	00283600
	* IWORK(INUM10), IWORK(INUM11), IWORK(INUM12),	00283700
	* IWORK(INUM13), ITM)	00283800
	RETURN	00283900
C	90 CONTINUE	00284000
	WRITE(LOUT, 6100) MLIMIT, INUM14	00284100
	RETURN	00284200
C	6100 FORMAT(1H , ' USED WORKING AREA WORK(1) ~ WORK(1, I6, '))' /	00284300
	* 1H , ' USED WORKING AREA IWORK(1) ~ IWORK(1, I6, '))')	00284400
	6300 FORMAT(///1H , '--- CATEGORY OF ',8A1, ' IS SINGLETON ---')	00284500
C	END	00284600
		00284700
		00284800
		00284900
		00285000
		00285100
		00285200

	SUBROUTINE ADJUST(NVAR,X,Y,Z,LOCX,LOCY,LOCZ,NX,NY,NZ,DIM,CONFIG,	00285300
	* D)	00285400
C	REAL * 8 X(NX), Y(NY), Z(NZ), D, E	00285500
	INTEGER SIZE(8),DIM(NVAR),CONFIG(NVAR),COORD(7)	00285600
C		00285700
C	INITIALIZE SIZE ARRAY	00285800
C		00285900
	SIZE(1)=1	00286000
	DO 10 K=1,NVAR	00286100
	L=CONFIG(K)	00286200
	IF(L.EQ.0) GO TO 20	00286300
	SIZE(K+1)=SIZE(K)*DIM(L)	00286400
10	CONTINUE	00286500
		00286600
C		00286700
C	FIND NUMBER OF VARIABLES IN CONFIGURATION	00286800
C		00286900
	K=NVAR+1	00287000
20	N=K-1	00287100
C		00287200
C	TEST SIZE OF DEVIATION	00287300
C		00287400
	IF(LOCZ .LE. 0) LOCZ = 1	00287500
	L=SIZE(K)	00287600
	J=LOCY	00287700
	K=LOCZ	00287800
	DO 30 I=1,L	00287900
	E=ABS(Z(K)-Y(J))	00288000
	IF(E.GT.D)D=E	00288100
	J=J+1	00288200
	K=K+1	00288300
30	CONTINUE	00288400
C		00288500
C	INITIALIZE COORDINATES	00288600
C		00288700
	DO 40 K=1,NVAR	00288800
40	COORD(K)=0	00288900
	I=LOCX	00289000
C		00289100
C	PERFORM ADJUSTMENT	00289200
C		00289300
50	J=0	00289400
	IF(N .LE. 0) GO TO 65	00289500
	DO 60 K=1,N	00289600
	L=CONFIG(K)	00289700
	J=J+COORD(L)*SIZE(K)	00289800
60	CONTINUE	00289900
65	CONTINUE	00290000
	K=J+LOCZ	00290100
	J=J+LOCY	00290200
C		00290300
	IF(Y(J) .LE. 0.0D0) X(I) = 0.0D0	00290400
	IF(Y(J) .GT. 0.0D0) X(I) = X(I) * Z(K) / Y(J)	00290500
C		00290600
C	UPDATE COORDINATES	00290700
C		00290800
	I=I+1	00290900
	DO 70 K=1,NVAR	00291000
	COORD(K)=COORD(K)+1	00291100
	IF(COORD(K).LT.DIM(K))GOTO50	00291200
	COORD(K)=0	00291300
70	CONTINUE	00291400
	RETURN	00291500
	END	00291600

```

      SUBROUTINE BCHECK(NTAB,NVAR,NCON,MDCON,DIM,ICONF,IDFREE,JDFREE ) 00291700
C      THIS SUBROUTINE CHECKS THE ADAPTABILITY OF MODEL SPECIFIED BY USER. 00291800
C      COMMON / OUTIN / LIN, LOUT 00291900
C      COMMON / OPTION / IOPT 00292000
      INTEGER DIM 00292100
      DIMENSION DIM(NVAR), ICONF(NVAR, NCON) 00292200
C      K1 = 0 00292300
      DO 20 J = 1, MDCON 00292400
      K2 = 1 00292500
      DO 10 I = 1, NVAR 00292600
      K3 = ICONF(I, J) 00292700
      IF( K3 .LE. 0 ) GO TO 10 00292800
      K3 = DIM(K3) - 1 00292900
      IF( K3 .LE. 0 ) K3 = 1 00293000
      K2 = K2 * K3 00293100
10 CONTINUE 00293200
      K1 = K1 + K2 00293300
20 CONTINUE 00293400
C      JDFREE = NTAB - K1 - 1 00293500
      IF( MDCON .EQ. 1 ) .AND. (ICONF(1, 1) .LE. 0) ) 00293600
      * JDFREE = NTAB - 1 00293700
      IF( IDFREE .EQ. JDFREE ) RETURN 00293800
      IOPT = 1 00293900
      WRITE(LOUT, 6100) 00294000
      WRITE(LOUT, 6200) IDFREE, JDFREE 00294100
C      6100 FORMAT(1H0, 37( 1H* ) / 1H , 00294200
      * ' MISUSE OF A MODEL ( CONFIG ) . #' / 1H , 00294300
      * 37( 1H* ) / ) 00294400
      6200 FORMAT(1H , ' DEGREES OF FREEDOM EQUAL TO ', I5, ' BY ONE METHOD.', 00294500
      * ' BUT EQUAL TO ', I5, ' BY ANOTHER METHOD .' ) 00294600
      RETURN 00294700
      END 00294800
      SUBROUTINE CHISQ( X, N, Q ) 00294900
      IMPLICIT REAL * 8 ( A - H , O - Z ) 00295000
C      THIS SUBROUTINE CALCULATES THE TAILED PROBABILITY TO A GIVEN 00295100
C      CHI-SQUARE VALUE. 00295200
C      THIS PROGRAM WAS MADE BY K.MIZUNO. 00295300
C      IF SPECIFIED CONDITION IS EXP( -X/2 ) < 1.0E-60 , THEN X EQUAL 00295400
C      ALMOST 260. 00295500
C      IF( X .GT. 260.000 ) GO TO 30 00295600
      IA = N - N / 2 * 2 00295700
      C = 0.000 00295800
      IF( N .EQ. 1 ) GO TO 40 00295900
      S = 1.000 00296000
      C = 1.000 00296100
      M = N - 2 00296200
      J = IA + 2 00296300
10 CONTINUE 00296400
      IF( J .GT. M ) GO TO 20 00296500
      RJ = J 00296600
      C = C * X / RJ 00296700
      IF( C .LT. 1.0D-60 ) C = 0.000 00296800
      IF( C .LE. 0.000 ) GO TO 20 00296900
      S = S + C 00297000
      J = J + 2 00297100
      GO TO 10 00297200
20 CONTINUE 00297300
      C = S * DEXP( - X / 2.000 ) 00297400
      GO TO 40 00297500
30 C = 0.000 00297600
40 CONTINUE 00297700
      IF( IA .EQ. 0 ) GO TO 70 00297800
      IF( X .LT. 1.0D-30 ) GO TO 50 00297900
      D = DSQRT( X ) 00298000
      GO TO 60 00298100
50 D = 0.000 00298200
60 CONTINUE 00298300
      C = C * D * 0.797884560802865D0 00298400
      CALL NORMAL( D, P, Q, Y ) 00298500
      C = 2.000 * ( 1.000 - P ) + C 00298600
70 CONTINUE 00298700
      Q = C 00298800
      P = 1.000 - C 00298900
      RETURN 00299000
      END 00299100

```

SUBROUTINE COLLAP(NVAR,X,Y,LOCX,LOCY,NX,NY,DIM,CONFIG,OPTION)	00299900
REAL*8 X(NX),Y(NY)	00300000
INTEGER SIZE(8),DIM(NVAR),CONFIG(NVAR),COORD(7)	00300100
LOGICAL OPTION	00300200
C	00300300
C INITIALISE ARRAYS	00300400
C	00300500
IF(LOCY .LE. 0) LOCY = 1	00300600
SIZE(1)=1	00300700
DO 10 K=1,NVAR	00300800
L=CONFIG(K)	00300900
IF(L.EQ.0)GOTO20	00301000
SIZE(K+1)=SIZE(K)*DIM(L)	00301100
10 CONTINUE	00301200
C	00301300
C FIND NUMBER OF VARIABLES IN CONFIGURATION	00301400
C	00301500
K=NVAR+1	00301600
20 N=K-1	00301700
C	00301800
C IF MARGINAL TABLE DESIRED, INITIALISE Y. FIRST CELL OF MARGINAL	00301900
C TABLE IS AT Y(LOCY) AND TABLE HAS SIZE(K) ELEMENTS	00302000
C	00302100
IF(.NOT.OPTION) GO TO 40	00302200
LOCU=LOCY+SIZE(K)-1	00302300
DO 30 J=LOCY,LOCU	00302400
30 Y(J)=0.0D0	00302500
C	00302600
C INITIALISE COORDINATES	00302700
C	00302800
40 DO 50 K=1,NVAR	00302900
50 COORD(K)=0	00303000
C	00303100
C FIND LOCATIONS IN TABLES	00303200
C	00303300
I=LOCX	00303400
60 J=LOCY	00303500
IF(N .LE. 0) GO TO 75	00303600
DO 70 K=1,N	00303700
L=CONFIG(K)	00303800
J=J+COORD(L)*SIZE(K)	00303900
70 CONTINUE	00304000
75 CONTINUE	00304100
IF(OPTION)Y(J)=Y(J)+X(I)	00304200
IF(.NOT.OPTION)X(I)=X(I)-Y(J)	00304300
C	00304400
C UPDATE COORDINATES	00304500
C	00304600
I=I+1	00304700
DO 80 K=1,NVAR	00304800
COORD(K)=COORD(K)+1	00304900
IF(COORD(K).LT.DIM(K)) GO TO 60	00305000
COORD(K)=0	00305100
80 CONTINUE	00305200
RETURN	00305300
END	00305400

	SUBROUTINE CROSTB(NSAMP,NVAR,NTAB,INDEV,ZZZ,XX,MQ,MA,MX,M1,RMAX,	00305500
	* RMIN,OBSERV,FREQ,IWORK,KK)	00305600
C		00305700
C	THIS SUBROUTINE GENERATES THE CONTINGENCY TABLES FOR ANALYZING THE	00305800
C	LOG-LINEAR MODEL.	00305900
C		00306000
	IMPLICIT REAL * 8 (A-H, O-Z)	00306100
	REAL * 4 OUTIN, XX	00306200
	COMMON / OUTIN / LIN, LOU	00306300
	DIMENSION RMAX(NVAR), RMIN(NVAR), OBSERV(NVAR),	00306400
	* FREQ(1), IWORK(1), XX(MA), MQ(MX)	00306500
C		00306600
C		00306700
	NSA = 0	00306800
	ZZZ=0	00306900
	REWIND INDEV	00307000
	READ(INDEV,END=99,ERR=777) KK,(XX(J),J=1,M1)	00307100
	NSA = NSA + 1	00307200
	DO 5 J = 1, NVAR	00307300
	L = MQ(J)	00307400
	5 OBSERV(J) = XX(L)	00307500
C		00307600
	DO 10 I = 1, NVAR	00307700
	RMAX(I) = OBSERV(I)	00307800
	RMIN(I) = OBSERV(I)	00307900
	10 CONTINUE	00308000
C		00308100
	DO 20 I1 = 2, NSAMP	00308200
	READ(INDEV,END=99,ERR=777) KK,(XX(J),J=1,M1)	00308300
	NSA = NSA + 1	00308400
	DO 15 J = 1, NVAR	00308500
	L = MQ(J)	00308600
	15 OBSERV(J) = XX(L)	00308700
	DO 20 I2 = 1, NVAR	00308800
	IF(OBSERV(I2) .LT. RMIN(I2)) RMIN(I2) = OBSERV(I2)	00308900
	IF(OBSERV(I2) .GT. RMAX(I2)) RMAX(I2) = OBSERV(I2)	00309000
	20 CONTINUE	00309100
	99 CONTINUE	00309200
C		00309300
	DO 45 K=1,NVAR	00309400
	IF(RMIN(K).EQ.RMAX(K)) GO TO 97	00309500
	45 CONTINUE	00309600
	GO TO 98	00309700
C		00309800
	97 ZZZ=1	00309900
	KK=MQ(K)	00310000
	RETURN	00310100
C		00310200
	98 CONTINUE	00310300
C		00310400
	NTAB = 1	00310500
	DO 30 I1 = 1, NVAR	00310600
	IWORK(I1) = RMAX(I1) - RMIN(I1) + 1.0	00310700
	NTAB = NTAB * IWORK(I1)	00310800
	30 CONTINUE	00310900
C		00311000
	IF(NTAB .LE. 0) GO TO 60	00311100
	REWIND INDEV	00311200
	DO 50 I1 = 1, NSAMP	00311300
	READ(INDEV,END=88,ERR=777) KK,(XX(J),J=1,M1)	00311400
	DO 35 J = 1, NVAR	00311500
	L = MQ(J)	00311600
	35 OBSERV(J) = XX(L)	00311700
	MULT = 1	00311800
	IADD = 0	00311900
	DO 40 I2 = 2, NVAR	00312000
	MULT = MULT * IWORK(I2 - 1)	00312100
	K = OBSERV(I2) - RMIN(I2) + 1.000	00312200
	IADD = IADD + (K - 1) * MULT	00312300
	40 CONTINUE	00312400
	K = OBSERV(1) - RMIN(1) + 1.000	00312500
	IADD = IADD + K	00312600
	FREQ(IADD) = FREQ(IADD) + 1.000	00312700
	50 CONTINUE	00312800
C		00312900
	88 CONTINUE	00313000
	KK=KK	00313100
	RETURN	00313200
	60 CONTINUE	00313300
	WRITE(LOUT,6100)	00313400
	6100 FORMAT(1H ,28H YOU MUST CHECK YOUR DATA .)	00313500
	RETURN	00313600
	777 WRITE(LOUT,900)	00313700
	900 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00313800
	* 'SPECIFIED BY COMMAND ---')	00313900
	RETURN	00314000
C		00314100
	END	00314200

	SUBROUTINE DGFREE(NTAB,NVAR,NCON,CONFIG,IDIM,ICONF,DFREE)	00314300
C		00314400
C	CALCULATE THE DEGREES OF FREEDOM.	00314500
C		00314600
	INTEGER CONFIG(NVAR,NCON),DFREE	00314700
	DIMENSION IDIM(NVAR),ICONF(NVAR)	00314800
C		00314900
	NVAR1 = NVAR - 1	00315000
	DFREE = NTAB - 1	00315100
C		00315200
	DO 10 I = 1, NVAR	00315300
	ICONF(I) = 0	00315400
10	CONTINUE	00315500
C		00315600
20	K = 1	00315700
30	IF(ICONF(K) .NE. K) GO TO 40	00315800
	IF(K .EQ. NVAR) RETURN	00315900
	K = K + 1	00316000
	GO TO 30	00316100
40	IF(K - 2) 50, 100, 70	00316200
50	KK = NVAR1	00316300
C		00316400
	DO 60 I = 1, NVAR1	00316500
	ICONF(KK + 1) = ICONF(KK)	00316600
	KK = KK - 1	00316700
60	CONTINUE	00316800
	GO TO 100	00316900
70	KK = 2	00317000
C		00317100
	DO 80 I = K, NVAR	00317200
	ICONF(KK) = ICONF(I)	00317300
	KK = KK + 1	00317400
80	CONTINUE	00317500
	L = K - 2	00317600
C		00317700
	DO 90 I = 1, L	00317800
	ICONF(KK) = 0	00317900
	KK = KK + 1	00318000
90	CONTINUE	00318100
100	ICONF(1) = K	00318200
C		00318300
	DO 160 L = 1, NCON	00318400
	K = 1	00318500
C		00318600
	DO 120 I = 1, NVAR	00318700
	IF(ICONF(K) .EQ. 0) GO TO 130	00318800
	IF(CONFIG(I, L) .EQ. 0) GO TO 160	00318900
	IF(CONFIG(I, L) - ICONF(K)) 120, 110, 160	00319000
110	K = K + 1	00319100
120	CONTINUE	00319200
130	KK = 1	00319300
C		00319400
	DO 140 I = 1, NVAR	00319500
	K = ICONF(I)	00319600
	IF(K .EQ. 0) GO TO 150	00319700
140	KK = KK * (IDIM(K) - 1)	00319800
150	DFREE = DFREE - KK	00319900
	GO TO 20	00320000
160	CONTINUE	00320100
	GO TO 20	00320200
	END	00320300
	SUBROUTINE INITIA(NTAB,NU,NMAR,MAXIT,NVAR,NCON,FIT,U,MARG,DEV,	00320400
	* ICONF)	00320500
C		00320600
C	CLEAR ARRAYS	00320700
C		00320800
	REAL*8 U(NU),MARG(NMAR),DEV(MAXIT),FIT(NTAB)	00320900
	DIMENSION ICONF(NVAR,NCON)	00321000
C		00321100
	DO 10 I = 1, NTAB	00321200
	FIT(I) = 1.0D0	00321300
10	CONTINUE	00321400
	DO 20 I = 1, NU	00321500
	U(I) = 0.0D0	00321600
20	CONTINUE	00321700
	DO 30 I = 1, NMAR	00321800
	MARG(I) = 0.0D0	00321900
30	CONTINUE	00322000
	DO 40 I = 1, MAXIT	00322100
	DEV(I) = 0.0D0	00322200
40	CONTINUE	00322300
	DO 50 I = 1, NVAR	00322400
	DO 50 J = 1, NCON	00322500
	ICONF(I, J) = 0	00322600
50	CONTINUE	00322700
C		00322800
	RETURN	00322900
	END	00323000

	SUBROUTINE LOGLIN(NVAR,DIM,NCON,CONFIG,NTAB,TABLE,FIT,LOCMAR,	00323100
*	NMAR,MARG,NU,MAXDEV,MAXIT,DEV,NLAST,IFAU,LT,U)	00323200
	REAL * 8 X, Y, MARG(NMAR), U(NU), TABLE(NTAB), FIT(NTAB),	00323300
*	DEV(MAXIT),MAXDEV,XMAX	00323400
	INTEGER DIM(NVAR),CONFIG(NVAR,NCON),LOCMAR(NCON),ICON(7),	00323500
*	CHECK(7),POINT,SIZE	00323600
C		00323700
	DATA MAXVAR/7/	00323800
C		00323900
C	NVAR:THE NUMBER OF VARIABLES D IN THE TABLE. (NVAR .LE. 7)	00324000
C	DIM:THE NUMBER OF CATEGORIES R(J) IN EACH VARIABLE OF THE TABLE	00324100
C	NCON:THE NUMBER S OF MARGINAL TOTALS TO BE FIT.	00324200
C	CONFIG:THE SETS C(K),K=1,...,S, INDICATING MARGINAL TOTALS TO	00324300
C	BE FIT.	00324400
C	NTAB:THE NUMBER OF ELEMENTS IN THE TABLE.	00324500
C	TABLE:THE TABLE TO BE FIT.	00324600
C	FIT:THE FITTED TABLE	00324700
C	LOCMAR:POINTERS TO THE TABLES IN MARG.	00324800
C	NMAR:THE DIMENSION OF MARG.	00324900
C	MARG:THE MARGINAL TABLES TO BE FIT.	00325000
C	NU:THE DIMENSION OF U.	00325100
C	U:A WORK AREA USED TO STORE FITTED MARGINAL TABLES.	00325200
C	MAXDEV:THE MAXIMUM PERMISSIBLE DIFFERENCE BETWEEN AN OBSERVED	00325300
C	AND FITTED MARGINAL TOTAL.	00325400
C	MAXIT:THE MAXIMUM PERMISSIBLE NUMBER OF ITERATIONS.	00325500
C	DEV:DEV(I) IS THE MAXIMUM OBSERVED DIFFERENCE ENCOUNTERED	00325600
C	IN ITERATION CYCLE I BETWEEN AN OBSERVED	00325700
C	AND FITTED MARGINAL TOTAL.	00325800
C	NLAST:THE NUMBER OF THE LAST ITERATION.	00325900
C	IFAU:AN ERROR INDICATOR.SEE FAILURE INDICATIONS BELOW.	00326000
C		00326100
C	CHECK VALIDITY OF NVAR, THE NUMBER OF VARIABLES,	00326200
C	AND OF MAXIT, THE MAXIMUM NUMBER OF ITERATIONS	00326300
C		00326400
	IFAU = 0	00326500
	IF(NVAR.GT.0.AND.NVAR.LE.MAXVAR.AND.MAXIT.GT.0) GO TO 10	00326600
5	IFAU=4	00326700
	RETURN	00326800
C		00326900
C	LOOK AT TABLE AND FIT CONSTANTS	00327000
C		00327100
10	SIZE=1	00327200
	DO 30 J=1,NVAR	00327300
	IF(DIM(J).LE.0) GO TO 5	00327400
20	SIZE=SIZE+DIM(J)	00327500
30	CONTINUE	00327600
	IF(SIZE.LE.NTAB) GO TO 40	00327700
35	IFAU=2	00327800
	RETURN	00327900
40	X=0.000	00328000
	Y=0.000	00328100
	DO 60 I=1,SIZE	00328200
	IF((TABLE(I).LT.0.000) .OR. (FIT(I).LT.0.000)) GO TO 5	00328300
	X=X+TABLE(I)	00328400
	Y=Y+FIT(I)	00328500
60	CONTINUE	00328600
C		00328700
C	MAKE A PRELIMINARY ADJUSTMENT TO OBTAIN THE FIT TO AN	00328800
C	EMPTY CONFIGURATION LIST	00328900
C		00329000
	IF(Y.EQ.0.000) GO TO 5	00329100
	X=X/Y	00329200
	DO 80 I=1,SIZE	00329300
80	FIT(I)=X*FIT(I)	00329400

C		00329500
C	ALLOCATE MARGINAL TABLES	00329600
C		00329700
	POINT=1	00329800
	DO 150 I=1,NCON	00329900
C		00330000
C	A ZERO BEGINNING A CONFIGURATION INDICATES THAT THE LIST IS	00330100
C	COMPLETED	00330200
C		00330300
	IF(CONFIG(1,I).EQ.0) GO TO 160	00330400
C		00330500
C	GET MARGINAL TABLE SIZE. WHILE DOING THIS TASK, SEE IF THE	00330600
C	CONFIGURATION LIST CONTAINS DUPLICATIONS OR ELEMENTS OUT OF	00330700
C	RANGE	00330800
C		00330900
	SIZE=1	00331000
	DO 90 J=1,NVAR	00331100
90	CHECK(J)=0	00331200
	DO 120 J=1,NVAR	00331300
	K=CONFIG(J,I)	00331400
C		00331500
C	A ZERO INDICATES THE END OF THE STRING	00331600
C		00331700
	IF(K.EQ.0) GO TO 130	00331800
C		00331900
C	SEE IF ELEMENT VALID	00332000
C		00332100
	IF(K.GE.0.AND.K.LE.NVAR) GO TO 100	00332200
95	IF(AULT=1	00332300
	RETURN	00332400
C		00332500
C	CHECK FOR DUPLICATION	00332600
C		00332700
	100 IF(CHECK(K).EQ.1) GO TO 95	00332800
	110 CHECK(K)=1	00332900
C		00333000
C	GET SIZE	00333100
C		00333200
	SIZE=SIZE*DIM(K)	00333300
120	CONTINUE	00333400
C		00333500
C	SINCE U IS USED TO STORE FITTED MARGINALS, SIZE MUST NOT	00333600
C	EXCEED NU	00333700
C		00333800
	130 IF(SIZE.GT.NU) GO TO 35	00333900
C		00334000
C	LOCMAR POINTS TO MARGINAL TABLES TO BE PLACED IN MARG	00334100
C		00334200
	140 LOCMAR(I)=POINT	00334300
	POINT=POINT+SIZE	00334400
150	CONTINUE	00334500
C		00334600
C	GET N, NUMBER OF VALID CONFIGURATIONS	00334700
C		00334800
	I=NCON+1	00334900
160	N=I-1	00335000
C		00335100
C	SEE IF MARG CAN HOLD ALL MARGINAL TABLES	00335200
C		00335300
	IF(POINT.GT.NMAR+1) GO TO 35	00335400
C		00335500
C	OBTAIN MARGINAL TABLES	00335600
C		00335700
	170 DO 190 I=1,N	00335800
	DO 180 J=1,NVAR	00335900
180	ICON(J)=CONFIG(J,I)	00336000
	CALL COLLAP(NVAR,TABLE,MARG,1,LOCMAR(I),NTAB,NMAR,DIM,ICON,.TRUE.)	00336100
190	CONTINUE	00336200

C		00336300
C	PERFORM ITERATIONS	00336400
C		00336500
C	DD 220 K=1,MAXIT	00336600
C		00336700
C	XMAX IS MAXIMUM DEVIATION OBSERVED BETWEEN FITTED AND TRUE	00336800
C	MARGINAL DURING A CYCLE	00336900
C		00337000
C	XMAX=0.000	00337100
C	DO 210 I=1,N	00337200
C	DO 200 J=1,NVAR	00337300
200	ICON(J)=CONFIG(J,I)	00337400
	CALL COLLAP(NVAR,FIT,U,1,1,NTAB,NU,DIM,ICON,.TRUE.)	00337500
	CALL ADJUST(NVAR,FIT,U,MARG,1,1,LOCMAR(I),NTAB,NU,NMAR,DIM,	00337600
	* ICON,XMAX)	00337700
210	CONTINUE	00337800
C		00337900
C	TEST CONVERGENCE	00338000
C		00338100
C	DEV(K)=XMAX	00338200
	IF(XMAX.LT.MAXDEV) GO TO 240	00338300
220	CONTINUE	00338400
	IF(MAXIT.GT.1) GO TO 230	00338500
	NLAST=1	00338600
	RETURN	00338700
C		00338800
C	NO CONVERGENCE	00338900
C		00339000
230	IFALT=3	00339100
	NLAST=MAXIT	00339200
	RETURN	00339300
C		00339400
C	NORMAL TERMINATION	00339500
C		00339600
240	NLAST=K	00339700
	RETURN	00339800
	END	00339900
	SUBROUTINE MAXMAR(NVAR, IDIM, ITERAT)	00340000
	DIMENSION IDIM(NVAR)	00340100
C		00340200
	ITERAT = IDIM(1)	00340300
	DO 10 I = 2, NVAR	00340400
	ITERAT = ITERAT * (1+ IDIM(I)) + IDIM(I)	00340500
10	CONTINUE	00340600
	RETURN	00340700
	END	00340800

```

SUBROUTINE MCONTB(NVAR,NCON,NMAR,NU,MAXIT,MAXDEV,NTAB,NTITLE, 00340900
* ITERAT,ITER1,MAXMOD,NVAR1,KDIM,TITLE,FMTI, 00341000
* KNCON1,KNCON2,TABLE,FIT,MARG,U,DEV,TSUM,FSUM, 00341100
* EFFEC,WORK,DIM,ICONF,MDIM,CONFIG,LOCMAR,JCON, 00341200
* MCON,NEFEC1,NEFEC2,NDFREE,KCONF,LCON,IWORK,ITM )00341300
C
C THIS SUBPROGRAM IS MAIN ROUTINE FOR MODEL-SELECTION 00341500
C OF LOG-LINER MODEL. 00341600
C 00341700
IMPLICIT REAL * 8 ( A - H , 0 - Z ) 00341800
REAL*8 MARG,MAXDEV 00341900
REAL*4 TITLE,FMTI,OUTIN,OPTION 00342000
INTEGER DIM,CHECK,CONFIG,POINT,SIZE,BLANK 00342100
COMMON / METHOD / METH 00342200
COMMON / OUTIN / LIN,LOUT 00342300
COMMON / OPTION / IOPT 00342400
COMMON / ICRIT / CHIG,CHIL,PROBG,PROBL,AICG,AICL, 00342500
* IDFREE,ICONBI 00342600
* DIMENSION MARG(NMAR),U(NU),TABLE(NTAB),FIT(NTAB),DEV(MAXIT), 00342700
* TITLE(NTITLE),FMTI(NTITLE),DIM(NVAR),MDIM(NVAR), 00342800
* ICONF(NVAR),CONFIG(NVAR,NCON),LOCMAR(NCON), 00342900
* TSUM(KDIM),FSUM(KDIM),EFFEC(NCON),NEFEC1(NCON), 00343000
* NEFEC2(NCON),KCONF(KNCON1,1), 00343100
* LCON(KNCON2),JCON(NVAR,NCON),MCON(NCON),WORK(1), 00343200
* IWORK(1),NDFREE(NCON),ITM(10,8),ITM1(10,8) 00343300
DATA BLANK/' '/ 00343400
C SET THE INITIAL PARAMETERS 00343500
C KTIME = 1 00343600
C IPIVOT = 6 00343700
C JPIVOT = ITER1 00343800
C ICALCU = 2 00343900
C ITER0 = ITER1 - 2 00344000
C 00344100
C IF( IOPT .LE. 2 ) CALL PRITAB( NVAR, NVAR1, NTAB, KDIM, DIM, 00344200
* MDIM,TABLE,FIT,TSUM,FSUM,0,ITM ) 00344300
C 00344400
10 CONTINUE 00344500
C INITIALIZE OF FIT, U, MARG, DEV 00344600
C CALL INITIA ( NTAB, NU, NMAR, MAXIT, NVAR, NCON, 00344700
* FIT, U, MARG, DEV, CONFIG ) 00344800
C 00344900
C CALL MODEL ( NVAR, NCON, MDCON, NTITLE, KTIME, ISTOP, 00345000
* ITERAT, ITER1, IPIVOT, JPIVOT, MAXMOD, KNCON1, 00345100
* KNCON2, CONFIG, JCON, MCON, NEFEC1, NEFEC2, 00345200
* NDFREE, KCONF, LCON, IWORK, ICALCU, TITLE, FMTI, 00345300
* EFFEC, WORK ) 00345400
C IF( ( ICALCU .EQ. 1 ) .OR. ( ICALCU .EQ. 3 ) ) 00345500
* GO TO 130 00345600
C IF( ISTOP .GT. 0 ) GO TO 200 00345700
C 00345800
C CALL DGFREE( NTAB, NVAR, MDCON, CONFIG, DIM, ICONF, IDFREE ) 00345900
C CALL BCHECK( NTAB, NVAR, NCON, MDCON, DIM, CONFIG, 00346000
* IDFREE, JDFREE ) 00346100
C 00346200
C IF( IOPT .LE. 0 ) GO TO 30 00346300
C WRITE(LOUT, 6020) 00346400
C DO 20 J = 1, MDCON 00346500
C WRITE(LOUT, 6030) J 00346600
C WRITE(LOUT, 6040) ( CONFIG(K, J) , K = 1, NVAR ) 00346700
C 00346800
C DO 21 I1=1,10 00346900
C DO 21 I2=1, 8 00347000
C 21 ITM1(I1,I2)=BLANK 00347100
C 00347200
C KK=0 00347300
C DO 22 I1=1,NVAR 00347400
C IF(CONFIG(I1,J),EQ.0) GO TO 22 00347500

```

	KK=KK+1	00347600
	JJ=CONFIG(I1,J)	00347700
	DO 23 I2=1,8	00347800
23	ITM1(KK,I2)=ITM(JJ,I2)	00347900
22	CONTINUE	00348000
	WRITE(LOUT,6001) ((ITM1(I,JJ),JJ=1,8),I=1,KK)	00348100
C		00348200
	20 CONTINUE	00348300
C		00348400
	IF(KTIME .NE. 1) GO TO 30	00348500
	IF(IOPT .LE. 3) GO TO 30	00348600
	CALL PRTTAB(NVAR, NVAR1, NTAB, KDIM, DIM, MDIM,	00348700
	* TABLE, FIT, TSUM, FSUM, 0, ITM)	00348800
30	CONTINUE	00348900
	IF(IDFREE .NE. JDFREE) RETURN	00349000
C		00349100
C	START THE FITTING ITERATION	00349200
C		00349300
	CALL LOGLIN(NVAR, DIM, MDCON, CONFIG, NTAB, TABLE, FIT,	00349400
	* LOCMAR, NMAR, MARG, NU, MAXDEV, MAXIT, DEV,	00349500
	* NLAST, IFAULT, U)	00349600
C		00349700
	CHIG = 0.000	00349800
	CHIL = 0.000	00349900
	DO 40 I = 1, NTAB	00350000
	ERROR = 0.000	00350100
	IF((TABLE(I) .LE. 0.000) .OR. (FIT(I) .LE. 0.000))	00350200
	* ERROR = 0.500	00350300
	CHIG = CHIG + (TABLE(I) - FIT(I)) / (FIT(I) + ERROR)	00350400
	* (TABLE(I) - FIT(I))	00350500
	CHIL = CHIL + (TABLE(I) + ERROR) *	00350600
	* (LOG(TABLE(I) + ERROR) - LOG(FIT(I) + ERROR))	00350700
40	CONTINUE	00350800
	CHIL = 2.000 * CHIL	00350900
	CALL CHISQ(CHIG, IDFREE, PROBG)	00351000
	CALL CHISQ(CHIL, IDFREE, PROBL)	00351100
	DFREE = IDFREE	00351200
	AICG = CHIG - 2.000 * DFREE	00351300
	AICL = CHIL - 2.000 * DFREE	00351400
C		00351500
	IF(IOPT .LE. 2) GO TO 50	00351600
C		00351700
C	OUTPUT ALL INFORMATION GENERATED	00351800
C		00351900
	WRITE(LOUT, 6100)	00352000
	CALL PRTTAB(NVAR, NVAR1, NTAB, KDIM, DIM, MDIM,	00352100
	* TABLE, FIT, TSUM, FSUM, 1, ITM)	00352200
50	CONTINUE	00352300
C		00352400
	IF(IOPT .LE. 1) GO TO 80	00352500
	WRITE(LOUT, 6200)	00352600
	DO 60 I = 1, NMAR	00352700
	IF(MARG(I) .LE. 0.000) GO TO 70	00352800
60	CONTINUE	00352900
	I = NMAR + 1	00353000
70	MAXMAR = I - 1	00353100
	IF(MAXMAR .LE. 0) MAXMAR = 1	00353200
	WRITE(6, 6000) (MARG(I), I = 1, MAXMAR)	00353300
80	CONTINUE	00353400
	IF(IOPT .LE. 0) GO TO 100	00353500
	IF(IFAULT .EQ. 0) GO TO 90	00353600
	WRITE(LOUT, 6300)	00353700
	GO TO 100	00353800
90	WRITE(LOUT, 6310) NLAST	00353900
100	CONTINUE	00354000
C		00354100
	IF(IOPT .LE. 1) GO TO 120	00354200
	WRITE(LOUT, 6400)	00354300
	KLAST = (NLAST - 1) / 10 + 1	00354400
	DO 110 I = 1, KLAST	00354500
	FIRST = (I - 1) * 10 + 1	00354600

	ILAST = IFIRST + 9	00354700
	IF(ILAST .GT. NLAST) ILAST = NLAST	00354800
	WRITE(LOUT, 6420) (J, J = IFIRST, ILAST)	00354900
	WRITE(LOUT, 6440) (DEV(J), J = IFIRST, ILAST)	00355000
110	CONTINUE	00355100
C		00355200
	WRITE(LOUT, 6500)	00355300
	WRITE(LOUT, 6600) (LOCMAR(I), I = 1, MDCON)	00355400
120	CONTINUE	00355500
	IF(IOPT .LE. 0) GO TO 130	00355600
	WRITE(LOUT, 6700) CHIG, CHIL	00355700
	WRITE(LOUT, 6720) PROBG, PROBL	00355800
	WRITE(LOUT, 6800) IDFREE	00355900
	WRITE(LOUT, 6820) AICG, AICL	00356000
	IF((IFault .NE. 0) .AND. (IFault .NE. 3))	00356100
	* WRITE(LOUT, 6840) IFault	00356200
130	CONTINUE	00356300
C		00356400
	IFIRST = (KTIME - 1) * ITERAT + 1	00356500
	WORK(IFIRST) = CHIG	00356600
	WORK(IFIRST + 1) = PROBG	00356700
	WORK(IFIRST + 2) = CHIL	00356800
	WORK(IFIRST + 3) = PROBL	00356900
	WORK(IFIRST + 4) = AICG	00357000
	WORK(IFIRST + 5) = AICL	00357100
C		00357200
	JFIRST = (KTIME - 1) * ITER1 + 1	00357300
	IWORK(JFIRST) = MDCON	00357400
	DO 140 K1 = 1, MDCON	00357500
	DO 140 K2 = 1, NVAR	00357600
	KFIRST = JFIRST + NVAR * (K1 - 1) + K2	00357700
	IWORK(KFIRST) = CONFIG(K2, K1)	00357800
140	CONTINUE	00357900
	JF = JFIRST + ITER0 + 1	00358000
	IWORK(JF) = IDFREE	00358100
	KTIME = KTIME + 1	00358200
C		00358300
	GO TO 10	00358400
200	CONTINUE	00358500
	KTIME = KTIME + 1	00358600
C		00358700
	IF(METH .GE. 3) CALL EFFEC3(NVAR, NCON, NTITLE, ITERAT,	00358800
	* IPIVOT, MAXMOD, JCON, MCON,	00358900
	* NDFREE, EFFEC, TITLE, WORK,	00359000
	* IWORK(ITER1), ITM)	00359100
C		00359200
C		00359300
	IF((METH .LT. 3) .AND. (KTIME .GT. 1))	00359400
	* CALL PERMUT(ITERAT, ITER1, KTIME, IPIVOT,	00359500
	* WORK, IWORK)	00359600
	WRITE(LOUT, 6005)	00359700
	WRITE(LOUT, 6010) TITLE	00359800
	WRITE(LOUT, 6900)	00359900
C		00360000
	DO 220 I1 = 1, KTIME	00360100
	IF(METH.EQ.2 .AND. IOPT.NE.1 .AND. I1.GE.26) GO TO 222	00360200
	JFIRST = (I1 - 1) * ITER1 + 1	00360300
	MDCON = IWORK(JFIRST)	00360400
	WRITE(LOUT, 6930)	00360500
	WRITE(LOUT, 6910) I1	00360600
	DO 210 I2 = 1, MDCON	00360700
	KFIRST = JFIRST + (I2 - 1) * NVAR + 1	00360800
	KLAST = KFIRST + NVAR - 1	00360900
	WRITE(LOUT, 6920) I2, (IWORK(I), I = KFIRST, KLAST)	00361000
C		00361100
	DO 204 I=1,10	00361200
	DO 204 JJ=1,8	00361300
204	ITM1(I,JJ)=BLANK	00361400
C		00361500
	KK=0	00361600
	DO 205 I=KFIRST,KLAST	00361700
	IF(IWORK(I).EQ.0) GO TO 205	00361800
	KK=KK+1	00361900
	J=IWORK(I)	00362000
	DO 206 JJ=1,8	00362100
206	ITM1(KK,JJ)=ITM(J,JJ)	00362200
205	CONTINUE	00362300
	WRITE(LOUT,6001) ((ITM1(I,JJ),JJ=1,8),I=1,KK)	00362400

C	WRITE(LOUT,6930)	00362500
	210 CONTINUE	00362600
	220 CONTINUE	00362700
C	222 CONTINUE	00362800
	WRITE(LOUT, 6005)	00362900
	WRITE(LOUT, 6010) TITLE	00363000
	WRITE(LOUT, 6940)	00363100
C		00363200
	DO 260 I1 = 1, KTIME	00363300
	IFIRST = (I1 - 1) * ITERAT + 1	00363400
	ILAST = IFIRST + ITERAT - 1	00363500
	JFIRST = (I1 - 1) * ITER1 + 1	00363600
	JF = JFIRST + ITER0 + 1	00363700
	KDFREE = IWORK(JF)	00363800
	IF(KDFREE.EQ. 0) GO TO 230	00363900
	IF((METH.GE. 3) .AND. (WORK(IFIRST) .LE. - 1.000))	00364000
	* GO TO 240	00364100
	WRITE(LOUT, 6950) (WORK(I), I = IFIRST, ILAST)	00364200
	* GO TO 250	00364300
	230 CONTINUE	00364400
	WRITE(LOUT, 6960)	00364500
	GO TO 250	00364600
	240 CONTINUE	00364700
	WRITE(LOUT, 6970) WORK(IFIRST + 2), WORK(IFIRST + 5)	00364800
	250 CONTINUE	00364900
	WRITE(LOUT, 6910) I1, KDFREE	00365000
	WRITE(LOUT, 6930)	00365100
	260 CONTINUE	00365200
C		00365300
C		00365400
	6000 FORMAT(/ (1H , 10(F10.1, 2X))	00365500
	6001 FORMAT(1H+,50X,9(BA1,' '))	00365600
	6005 FORMAT(1H1	00365700
	6010 FORMAT(1H , ' ---- ',80A1,' ---- ' /	00365800
	6020 FORMAT(1H , ' ** CONFIG ** ' / 1H , 'SEQ NUM'	00365900
	6030 FORMAT(1H , 17, ' ')	00366000
	6040 FORMAT(1H+, 10X, 24I5 / (1H , 10X, 24I5)	00366100
	6100 FORMAT(1H0, ' ** CONTINGENCY TABLES ** '	00366200
	6200 FORMAT(1H0, ' ** TABLE OF MARG(.) ** '	00366300
	6300 FORMAT(1H0, 33(1H*) /	00366400
	* 1H , '% TOO MANY CYCLE OF ITERATIONS. *' / 1H , 33(1H*)	00366500
	6310 FORMAT(1H0, ' CONVERGENCE ... NUMBER OF ITERATIONS = ', 15)	00366600
	6400 FORMAT(1H0, ' ** THE MAXIMUM OBSERVED DIFFERENCE ENCOUNTERED IN '	00366700
	* 'ITERATION CYCLE (.) BETWEEN AN OBSERVED AND FITTED'	00366800
	* ' MARGINAL TOTAL ** ')	00366900
	6420 FORMAT(1H , ' ITERATION', 6X, 10I10	00367000
	6440 FORMAT(1H , 16X, 10F10.5	00367100
	6500 FORMAT(1H0, ' ** INDICATOR OF MARG **'	00367200
	6600 FORMAT(1H , 24I5	00367300
	6700 FORMAT(1H0, 44HPEARSON'S CHISQUARE (GOODNESS OF FIT) = , F15.6	00367400
	* / 1H , 29HLIKELIHOOD RATIO CHISQUARE , 13X, 2H= , F15.6	00367500
	6720 FORMAT(1H , 44HPROBABILITY OF PEASON'S CHISQUARE = , F15.6	00367600
	* / 1H , 44HPROBABILITY OF LIKELIHOOD RATIO CHISQUARE = ,	00367700
	* F15.6)	00367800
	6800 FORMAT(1H , 18HDEGREES OF FREEDOM , 24X, 2H= , 18	00367900
	6820 FORMAT(1H , 28HAIC BASED ON GOODNESS OF FIT , 14X, 2H= , F15.6	00368000
	* / 1H , 44HAIC BASED ON LIKELIHOOD RATIO STATISTIC = ,	00368100
	* F15.6	00368200
	6840 FORMAT(1H0, ' FAULT VALUE = ', 15 /	00368300
	6900 FORMAT(1H , 9HNUMBER OF, 11H SEQUENTIAL, 2X,	00368400
	* 13HCONFIGURATION / 1H , 8H MODELS,	00368500
	* 3X, 6HNUMBER, 6X, 9HOF MODELS / 1H ,	00368600
	* 130(1H-)	00368700
	6910 FORMAT(1H+, 15, 5X, 16	00368800
	6920 FORMAT(1H+, 10X 17, 3X, 36I3 / (1H , 10X, 17, 3X, 36I3)	00368900
	6930 FORMAT(1H	00369000
	6940 FORMAT(1H , 6H MODEL, 3X, 10HDEGREES OF, 6X,	00369100
	* 25HPEARSON'S GOODNESS OF FIT, 10X,	00369200
	* 16HLIKELIHOOD RATIO, 2(10X, 12HAIC BASED ON) /	00369300
	* 1H , 7HSEQ NUM, 4X, 7HFREEDOM, 2X,	00369400
	* 2(7X, 9HCHISQUARE, 1X, 13H(PROBABILITY)),	00369500
	* 4X, 15HGOODNESS OF FIT, 2X,	00369600
	* 26HLIKELIHOOD RATIO STATISTIC / 1H , 130(1H-) /	00369700
	6950 FORMAT(1H , 22X, 2(F15.6, 5X, F7.4, 3X) , F15.6, 9X, F15.6	00369800
	6960 FORMAT(1H , 22X, 2(7X, 3H---, 11X, 3H---, 6X) ,	00369900
	* 7X, 3H---, 21X, 3H---)	00370000
	6970 FORMAT(1H , 29X, 3H---, 11X, 3H---, 6X, F15.6, 6X, 3H---,	00370100
	* 13X, 3H---, 14X, F15.6	00370200
	RETURN	00370300
	END	00370400
		00370500
		00370600

```

SUBROUTINE MODEL(NVAR,NCON,MDCON,NTITLE,KTIME,ISTOP,ITERAT,ITER1, 00370700
* IPIVOT,JPIVOT,MAXMOD,KNCON1,KNCON2,ICONF,JCON, 00370800
* MCON,NEFEC1,NEFEC2,NDFREE,KCONF,LCON,IWORK, 00370900
* ICALCU,TITLE,FMTI,EFFEC,WORK ) 00371000
C 00371100
C ARRANGE AND GENERATE THE COMBINATORIAL TERMS IN LOG-LINEAR MODELS. 00371200
C AND ALSO CALCULATE SEVERAL STATISTICS REQUIRED TO EVALUATE AND TO 00371300
C SELECT THE REASONABLE MODELS. 00371400
C MOREOVER,EXECUTE THE STEPWISE PROCEDURES. 00371500
C 00371600
REAL*8 WORK,CHIG,CHIL,PROBG,PROBL,AICG,AICL,EFFEC,DFREE 00371700
INTEGER*2 ICON,IC1,IC2,IC3 00371800
INTEGER TITLE 00371900
COMMON / METHOD / METH 00372000
COMMON / OPTION / IOPT 00372100
COMMON / OUTIN / LIN, LOUT 00372200
COMMON / ICRIT / CHIG, CHIL, PROBG, PROBL, AICG, AICL, 00372300
1 IDFREE, ICONBI 00372400
DIMENSION ICONF(NVAR,NCON),TITLE(NTITLE),FMTI(NTITLE), 00372500
* WORK(ITERAT,MAXMOD),IWORK(ITER1,MAXMOD),NEFEC1(NCON), 00372600
* NEFEC2(NCON),EFFEC(NCON),JCON(NVAR,NCON),MCON(NCON), 00372700
* NDFREE(NCON),KCONF(KNCON1,1),LCON(KNCON2),ICON(1650), 00372800
* IC1(459),IC2(527),IC3(244) 00372900
EQUIVALENCE ( ICON(420),IC1(1) ),( ICON(879),IC2(1) ), 00373000
* ( ICON(1406),IC3(1) ) 00373100
C 00373200
DATA ICON / 00373300
* 0,1,2,3,4, 00373400
* 1,2,1,3,1,4,2,3,2,4,3,4, 00373500
* 1,2,3,0,1,2,4,0,1,3,4,0,2,3,4,0,1,2,1,2,1,3,1,3,1,4,1,4,2,3,2,3, 00373600
* 2,4,2,4,3,4,3,4, 00373700
* 1,2,3,4,0,1,2,3,1,2,1,2,3,1,3,1,2,3,2,3,1,2,4,1,2,1,2,4,1,4,1,2, 00373800
* 4,2,4,1,3,4,1,3,1,3,4,1,4,1,3,4,3,4,2,3,4,2,3,2,3,4,2,4,2,3,4,3, 00373900
* 4, 00374000
* 1,2,3,1,2,1,3,1,2,3,1,2,2,3,1,2,3,1,3,2,3,1,2,4,1,2,1,4,1,2,4,1, 00374100
* 2,2,4,1,2,4,1,4,2,4,1,3,4,1,3,1,4,1,3,4,1,3,3,4,1,3,4,1,4,3,4,2, 00374200
* 3,4,2,3,2,4,2,3,4,2,3,3,4,2,3,4,2,4,3,4,1,2,3,4,0,1,2,1,2,3,4,0, 00374300
* 1,3,1,2,3,4,0,1,4,1,2,3,4,0,2,3,1,2,3,4,0,2,4,1,2,3,4,0,3,4, 00374400
* 1,2,3,1,2,1,3,2,3,1,2,4,1,2,1,4,2,4,1,3,4,1,3,1,4,3,4,2,3,4,2,3, 00374500
* 2,4,3,4,1,2,3,4,0,1,2,1,3,1,2,3,4,0,1,2,1,4,1,2,3,4,0,1,2,2,3, 00374600
* 2,3,4,0,1,2,2,4,1,2,3,4,0,1,2,3,4,1,2,3,4,0,1,3,1,4,1,2,3,4,0,1, 00374700
* 3,2,3,1,2,3,4,0,1,3,2,4,1,2,3,4,0,1,3,3,4,1,2,3,4,0,1,4,2,3,1,2, 00374800
* 3,4,0,1,4,2,4,1,2,3,4,0,1,4,3,4,1,2,3,4,0,2,3,2,4,1,2,3,4,0,2,3, 00374900
* 3,4,1,2,3,4,0,2,4,3,4, /00375000
C 00375100
DATA IC1 / 00375200
* 1,2,3,1,2,1,3,2,3,1,2,3,1,2,4,1,2,1,4,2,4,1,2,4,1,3,4,1,3,1,4,3, 00375300
* 4,1,3,4,2,3,4,2,3,2,4,3,4,2,3,4,1,2,3,4,0,1,2,1,3,1,4,0,1,2,3,4, 00375400
* 0,1,2,1,3,2,3,0,1,2,3,4,0,1,2,1,3,2,4,0,1,2,3,4,0,1,2,1,3,3,4,0, 00375500
* 1,2,3,4,0,1,2,1,4,2,3,0,1,2,3,4,0,1,2,1,4,2,4,0,1,2,3,4,0,1,2,1, 00375600
* 4,3,4,0,1,2,3,4,0,1,2,2,3,2,4,0,1,2,3,4,0,1,2,2,3,3,4,0,1,2,3,4, 00375700
* 0,1,2,2,4,3,4,0,1,2,3,4,0,1,3,1,4,2,3,0,1,2,3,4,0,1,3,1,4,2,4,0, 00375800
* 1,2,3,4,0,1,3,1,4,3,4,0,1,2,3,4,0,1,3,2,3,2,4,0,1,2,3,4,0,1,3,2, 00375900
* 3,3,4,0,1,2,3,4,0,1,3,2,4,3,4,0,1,2,3,4,0,1,4,2,3,2,4,0,1,2,3,4, 00376000
* 0,1,4,2,3,3,4,0,1,2,3,4,0,1,4,2,4,3,4,0,1,2,3,4,0,2,3,2,4,3,4,0, 00376100
* 1,2,1,3,2,3,1,2,3,1,2,1,4,2,4,1,2,4,1,3,1,4,3,4,1,3,4,2,3,2,4,3, 00376200
* 4,2,3,4,1,2,1,3,1,4,2,3,0,1,2,1,3,1,4,2,4,0,1,2,1,3,1,4,3,4,0,1, 00376300
* 2,1,3,2,3,2,4,0,1,2,1,3,2,3,3,4,0,1,2,1,3,2,4,3,4,0,1,2,1,4,2,3, 00376400
* 2,4,0,1,2,1,4,2,3,3,4,0,1,2,1,4,2,4,3,4,0,1,2,2,3,2,4,3,4,0,1,3, 00376500
* 1,4,2,3,2,4,0,1,3,1,4,2,3,3,4,0,1,3,1,4,2,4,3,4,0,1,3,2,3,2,4,3, 00376600
* 4,0,1,4,2,3,2,4,3,4,0, /00376700
C 00376800
DATA IC2 / 00376900
* 1,2,1,3,2,3,2,4,1,2,3,1,2,1,3,2,3,2,4,3,4,0,1,2,1,3,2,3,3,4,1,2, 00377000
* 3,1,2,1,3,1,4,2,3,2,4,0,1,2,1,3,1,4,2,3,3,4,0,1,2,1,3,3,4,2,3,1, 00377100
* 2,3,1,2,1,3,1,4,2,4,3,4,0,1,2,1,3,1,4,2,4,1,2,4,1,2,1,4,2,3,2,4, 00377200
* 3,4,0,1,2,1,4,2,3,2,4,1,2,4,1,2,1,4,2,4,3,4,1,2,4,1,2,2,3,2,4,3, 00377300

```

```

* 4,2,3,4,1,3,1,4,2,3,2,4,3,4,0,1,3,1,4,2,3,3,4,1,3,4,1,3,1,4,2,4, 00377400
* 3,4,1,3,4,1,3,2,3,2,4,3,4,2,3,4,1,4,2,3,2,4,3,4,2,3,4,1,2,1,3,1, 00377500
* 4,3,4,1,3,4, 00377600
* 1,2,1,3,1,4,2,3,2,4,3,4,0,1,2,1,3,1,4,2,3,2,4,1,2,3,1,2,1,3,1,4, 00377700
* 2,3,2,4,1,2,4,1,2,1,3,1,4,2,3,3,4,1,3,4,1,2,1,3,1,4,2,4,3,4,1,2, 00377800
* 4,1,2,1,3,2,3,2,4,3,4,1,2,3,1,2,1,4,2,3,2,4,3,4,1,2,4,1,2,1,4,2, 00377900
* 3,2,4,3,4,2,3,4,1,3,1,4,2,3,2,4,3,4,1,3,4,1,3,1,4,2,3,2,4,3,4,2, 00378000
* 3,4,1,2,1,3,1,4,2,3,3,4,1,2,3,1,2,1,3,1,4,2,4,3,4,1,3,4,1,2,1,3, 00378100
* 2,3,2,4,3,4,2,3,4, 00378200
* 1,2,1,3,1,4,2,3,2,4,1,2,3,1,2,4,1,2,1,3,1,4,2,3,3,4,1,2,3,1,3,4, 00378300
* 1,2,1,3,1,4,2,4,3,4,1,2,4,1,3,4,1,2,1,3,2,3,2,4,3,4,1,2,3,2,3,4, 00378400
* 1,2,1,4,2,3,2,4,3,4,1,2,4,2,3,4,1,3,1,4,2,3,2,4,3,4,1,3,4,2,3,4, 00378500
* 1,2,1,3,1,4,2,3,2,4,3,4,0,1,2,3,1,2,1,3,1,4,2,3,2,4,3,4,0,1,2,4, 00378600
* 1,2,1,3,1,4,2,3,2,4,3,4,0,1,3,4,1,2,1,3,1,4,2,3,2,4,3,4,0,2,3,4, /00378700
C DATA IC3 / 00378800
* 1,2,1,3,1,4,2,3,2,4,3,4,1,2,3,1,2,4,1,2,1,3,1,4,2,3,2,4,3,4,1,2, 00378900
* 3,1,3,4,1,2,1,3,1,4,2,3,2,4,3,4,1,2,3,2,3,4,1,2,1,3,1,4,2,3,2,4, 00379000
* 3,4,1,2,4,1,3,4,1,2,1,3,1,4,2,3,2,4,3,4,1,2,4,2,3,4,1,2,1,3,1,4, 00379100
* 2,3,2,4,3,4,1,3,4,2,3,4, 00379200
* 1,2,1,3,1,4,2,3,2,4,3,4,1,2,3,1,2,4,1,3,4,1,2,1,3,1,4,2,3,2,4,3, 00379300
* 4,1,2,3,1,2,4,2,3,4,1,2,1,3,1,4,2,3,2,4,3,4,1,2,3,1,3,4,2,3,4,1, 00379400
* 2,1,3,1,4,2,3,2,4,3,4,1,2,4,1,3,4,2,3,4, 00379500
* 1,2,1,3,1,4,2,3,2,4,3,4,1,2,3,1,2,4,1,3,4,2,3,4, 00379600
* 1,2,1,3,1,4,2,3,2,4,3,4,1,2,3,1,2,4,1,3,4,2,3,4, 00379700
* 1,2,1,3,1,4,2,3,2,4,3,4,1,2,3,1,2,4,1,3,4,2,3,4,1,2,3,4, /00379800
C 00379900
C SPECIFICATION OF A MODEL SELECTION PROCEDURE. 00380000
C 00380100
C ISTOP = 0 00380200
C GO TO ( 10, 1000, 2000, 2000 ), METH 00380300
C 00380400
C USER SPECIFIES ANY MODEL ( METHOD = 1 ) 00380500
C DATA-FORM IS CONTINGENCY TABLES. 00380600
C THIS OPTION DOES NOT WORK IN MINERVA. 00380700
C 00380800
C 10 CONTINUE 00380900
C READ(LIN, FMTI) MDCON, ICONF 00381000
C IF( MDCON .LE. 0 ) GO TO 80 00381100
C IF( ( IOPT .GE. 3 ) .AND. ( KTIME .GT. 1 ) ) WRITE(LOUT,6005) 00381200
C IF( IOPT .LE. 0 ) GO TO 20 00381300
C WRITE(LOUT, 6010) TITLE 00381400
C 20 CONTINUE 00381500
C DO 40 J = 2, NVAR 00381600
C DO 30 K = 1, MDCON 00381700
C IF( ICONF(J - 1, K) .GT. 0 ) GO TO 30 00381800
C ICONF(J, K) = 0 00381900
C 30 CONTINUE 00382000
C 40 CONTINUE 00382100
C 00382200
C DO 50 I = 1, MDCON 00382300
C IF( ICONF(1, I) .LE. 0 ) GO TO 60 00382400
C 50 CONTINUE 00382500
C GO TO 70 00382600
C 60 MDCON = I - 1 00382700
C 70 IF( MDCON .LE. 0 ) MDCON = 1 00382800
C RETURN 00382900
C 80 ISTOP = 1 00383000
C RETURN 00383100
C 00383200
C METHOD=2 ( DEFAULT VALUE ) 00383300
C THIS OPTION CALCULATES ALL POSSIBLE COMBINATIONS ( ORDINARY-METHOD ). 00383400
C 00383500
C 1000 IF( ( NVAR .LE. 1 ) .OR. ( NVAR .GT. 4 ) ) GO TO 80 00383600
C NVAR1 = NVAR - 1 00383700
C GO TO ( 1010, 1030, 1200 ), NVAR1 00383800
C 00383900
C 1010 IF( KTIME .GT. 5 ) GO TO 80 00384000
C IF( ( IOPT .GE. 3 ) .AND. ( KTIME .GT. 1 ) ) WRITE(LOUT,6005) 00384100
C IF( IOPT .LE. 0 ) GO TO 1020 00384200
C IF( KTIME .GT. 1 ) WRITE(LOUT, 6010) TITLE 00384300
C 1020 CONTINUE 00384400

```

MDCON = KTIME - 2		00384500
IF(MDCON .LE. 0)	MDCON = 1	00384600
IF(KTIME .LE. 1)	RETURN	00384700
IF((KTIME .EQ. 2) .OR. (KTIME .GE. 4))	ICONF(1, 1) = 1	00384800
IF(KTIME .EQ. 3)	ICONF(1, 1) = 2	00384900
IF(KTIME .GE. 4)	ICONF(1, 2) = 2	00385000
IF(KTIME .LT. 5)	RETURN	00385100
ICONF(1, 3) = 1		00385200
ICONF(2, 3) = 2		00385300
RETURN		00385400
C		00385500
1030 IF(KTIME .GT. 19)	GO TO 80	00385600
IF((IOPT .GE. 3) .AND. (KTIME .GT. 1))	WRITE(LOUT,6005)	00385700
IF(IOPT .LE. 0)	GO TO 1040	00385800
IF(KTIME .GT. 1)	WRITE(LOUT, 6010) TITLE	00385900
1040 CONTINUE		00386000
IF(KTIME .LE. 4)	MDCON = 1	00386100
IF((KTIME .GT. 4) .AND. (KTIME .LE. 7))	MDCON = 2	00386200
IF((KTIME .GT. 7) .AND. (KTIME .LE. 11))	MDCON = 3	00386300
IF((KTIME .GT. 11) .AND. (KTIME .LE. 14))	MDCON = 4	00386400
IF((KTIME .GT. 14) .AND. (KTIME .LE. 17))	MDCON = 5	00386500
C		00386600
IF((KTIME .EQ. 2) .OR. (KTIME .EQ. 5) .OR. (KTIME .EQ. 6)		00386700
*.OR.(KTIME .EQ. 8) .OR. (KTIME .EQ. 9))	ICONF(1, 1) = 1	00386800
IF(KTIME .LE. 10)	GO TO 1060	00386900
DO 1050 I = 1, 3		00387000
ICONF(1, I) = 1		00387100
1050 CONTINUE		00387200
C		00387300
1060 GO TO (1101, 1101, 1103, 1104, 1105, 1106, 1107, 1108, 1109,		00387400
* 1110, 1101, 1112, 1113, 1114, 1115, 1116, 1117, 1118,		00387500
* 1119), KTIME		00387600
1101 RETURN		00387700
1103 ICONF(1,1) = 2		00387800
RETURN		00387900
1104 ICONF(1,1) = 3		00388000
RETURN		00388100
1105 ICONF(1,2) = 2		00388200
RETURN		00388300
1106 ICONF(1,2) = 3		00388400
RETURN		00388500
1107 ICONF(1,1) = 2		00388600
ICONF(1,2) = 3		00388700
RETURN		00388800
1108 ICONF(1, 2) = 2		00388900
ICONF(1, 3) = 1		00389000
ICONF(2, 3) = 2		00389100
RETURN		00389200
1109 ICONF(1, 2) = 3		00389300
ICONF(1, 3) = 1		00389400
ICONF(2, 3) = 3		00389500
RETURN		00389600
1110 ICONF(1, 1) = 2		00389700
ICONF(1, 2) = 3		00389800
ICONF(1, 3) = 2		00389900
ICONF(2, 3) = 3		00390000
RETURN		00390100
1112 ICONF(1, 4) = 1		00390200
ICONF(2, 4) = 2		00390300
RETURN		00390400
1113 ICONF(1, 4) = 1		00390500
ICONF(2, 4) = 3		00390600
RETURN		00390700
1114 ICONF(1, 4) = 2		00390800
ICONF(2, 4) = 3		00390900
RETURN		00391000
1115 ICONF(1, 4) = 1		00391100
ICONF(2, 4) = 2		00391200
ICONF(1, 5) = 1		00391300
ICONF(2, 5) = 3		00391400
RETURN		00391500

1116	ICONF(1, 4) = 1	00391600
	ICONF(2, 4) = 2	00391700
	ICONF(1, 5) = 2	00391800
	ICONF(2, 5) = 3	00391900
	RETURN	00392000
1117	ICONF(1, 4) = 1	00392100
	ICONF(2, 4) = 3	00392200
	ICONF(1, 5) = 2	00392300
	ICONF(2, 5) = 3	00392400
	RETURN	00392500
1118	MDCON = 6	00392600
	ICONF(1, 4) = 1	00392700
	ICONF(2, 4) = 2	00392800
	ICONF(1, 5) = 1	00392900
	ICONF(2, 5) = 3	00393000
	ICONF(1, 6) = 2	00393100
	ICONF(2, 6) = 3	00393200
	RETURN	00393300
1119	MDCON = 7	00393400
	ICONF(1, 4) = 1	00393500
	ICONF(2, 4) = 2	00393600
	ICONF(1, 5) = 1	00393700
	ICONF(2, 5) = 3	00393800
	ICONF(1, 6) = 2	00393900
	ICONF(2, 6) = 3	00394000
	ICONF(1, 7) = 1	00394100
	ICONF(2, 7) = 2	00394200
	ICONF(3, 7) = 3	00394300
	RETURN	00394400
C		00394500
1200	IF(KTIME .GT. 167) GO TO 80	00394600
	IF((IOPT .GE. 3) .AND. (KTIME .GT. 1)) WRITE(LOUT,6005)	00394700
	IF(IOPT .LE. 0) GO TO 1201	00394800
	IF(KTIME .GT. 1) WRITE(LOUT, 6010) TITLE	00394900
1201	CONTINUE	00395000
	MVAR2 = 0	00395100
	MVAR3 = 0	00395200
C		00395300
	IF(KTIME .GT. 5) GO TO 1202	00395400
	MDCON = 1	00395500
	MVAR1 = 1	00395600
	IFIRST = KTIME	00395700
		00395800
	GO TO 1220	00395900
1202	IF(KTIME .GT. 11) GO TO 1203	00396000
	MDCON = 2	00396100
	MVAR1 = 2	00396200
	IFIRST = 2 * (KTIME - 5) + 4	00396300
	GO TO 1220	00396400
1203	IF(KTIME .GT. 21) GO TO 1204	00396500
	MDCON = 3	00396600
	MVAR1 = 2	00396700
	MVAR2 = 1	00396800
	IFIRST = 4 * (KTIME - 11) + 14	00396900
	GO TO 1220	00397000
1204	IF(KTIME .GT. 34) GO TO 1205	00397100
	MDCON = 4	00397200
	MVAR1 = 3	00397300
	MVAR2 = 1	00397400
	IFIRST = 5 * (KTIME - 21) + 53	00397500
	GO TO 1220	00397600
1205	IF(KTIME .GT. 52) GO TO 1206	00397700
	MDCON = 5	00397800
	MVAR1 = 3	00397900
	MVAR2 = 2	00398000
	IFIRST = 7 * (KTIME - 34) + 116	00398100
	GO TO 1220	00398200
1206	IF(KTIME .GT. 71) GO TO 1207	00398300
	MDCON = 6	00398400
	MVAR1 = 3	00398500
	MVAR2 = 3	00398600
	IFIRST = 9 * (KTIME - 52) + 240	00398700
	GO TO 1220	

1207 IF(KTIME .GT. 95)	GO TO 1208	00398800
MDCON = 7		00398900
MVAR1 = 3		00399000
MVAR2 = 3		00399100
MVAR3 = 1		00399200
IFIRST = 12 * (KTIME - 71) + 408		00399300
	GO TO 1220	00399400
1208 IF(KTIME .GT. 114)	GO TO 1209	00399500
MDCON = 8		00399600
MVAR2 = 3		00399700
MVAR3 = 1		00399800
IFIRST = 9 * (KTIME - 95) + 699		00399900
	GO TO 1220	00400000
1209 IF(KTIME .GT. 132)	GO TO 1210	00400100
MDCON = 9		00400200
MVAR2 = 4		00400300
MVAR3 = 1		00400400
IFIRST = 11 * (KTIME - 114) + 868		00400500
	GO TO 1220	00400600
1210 IF(KTIME .GT. 145)	GO TO 1211	00400700
MDCON = 10		00400800
MVAR2 = 5		00400900
MVAR3 = 1		00401000
IFIRST = 13 * (KTIME - 132) + 1064		00401100
	GO TO 1220	00401200
1211 IF(KTIME .GT. 155)	GO TO 1212	00401300
MDCON = 11		00401400
MVAR2 = 5		00401500
MVAR3 = 2		00401600
IFIRST = 16 * (KTIME - 145) + 1230		00401700
	GO TO 1220	00401800
1212 IF(KTIME .GT. 161)	GO TO 1213	00401900
MDCON = 12		00402000
MVAR2 = 6		00402100
MVAR3 = 2		00402200
IFIRST = 18 * (KTIME - 155) + 1388		00402300
	GO TO 1220	00402400
1213 IF(KTIME .GT. 165)	GO TO 1214	00402500
MDCON = 13		00402600
MVAR2 = 6		00402700
MVAR3 = 3		00402800
IFIRST = 21 * (KTIME - 161) + 1493		00402900
	GO TO 1220	00403000
1214 IF(KTIME .GT. 166)	GO TO 1215	00403100
MDCON = 14		00403200
MVAR2 = 6		00403300
MVAR3 = 4		00403400
IFIRST = 1598		00403500
	GO TO 1220	00403600
1215 IF(KTIME .GT. 167)	GO TO 80	00403700
MDCON = 15		00403800
MVAR2 = 6		00403900
MVAR3 = 4		00404000
IFIRST = 1622		00404100
1220 IF(MDCON .GE. 8)	GO TO 1240	00404200
DO 1230 I = 1, MVAR1		00404300
I1 = I + IFIRST - 1		00404400
ICONF(1, I) = ICON(I1)		00404500
1230 CONTINUE		00404600
IF(MVAR2 .LE. 0)	RETURN	00404700
JFIRST = MVAR1 + IFIRST		00404800
	GO TO 1260	00404900
1240 DO 1250 I = 1, 4		00405000
ICONF(1, I) = 1		00405100
1250 CONTINUE		00405200
MVAR1 = 4		00405300
JFIRST = IFIRST		00405400
1260 CONTINUE		00405500
DO 1270 I1 = 1, 2		00405600
DO 1270 I2 = 1, MVAR2		00405700

```

13 = 2 * I2 + JFIRST + I1 - 3
14 = I2 + MVAR1
ICONF(I1, I4) = ICON(I3)
1270 CONTINUE
IF( MVAR3 .LE. 0 ) RETURN
JFIRST = JFIRST + 2 * MVAR2
MVAR = MVAR1 + MVAR2
DO 1280 I1 = 1, 3
DO 1280 I2 = 1, MVAR3
I3 = 3 * I2 + JFIRST + I1 - 4
I4 = I2 + MVAR
ICONF(I1, I4) = ICON(I3)
1280 CONTINUE
IF( MVAR .LE. 0 ) RETURN
MVAR = MVAR + MVAR3
JFIRST = JFIRST + 3 * MVAR3
DO 1290 I1 = 1, 4
I2 = I1 + JFIRST - 1
ICONF(I1, I5) = ICON(I2)
1290 CONTINUE
RETURN

C
C METHOD = 3, 4
C
2000 IF( KTIME .GT. MAXMOD ) GO TO 80
IF( ICALCU .NE. 2 ) GO TO 2010
IF( ( ICALCU .EQ. 2 ) .AND. ( KTIME .GT. 2 + NCON ) ) GO TO 2030
*
C
IF( ( IOPT .GE. 3 ) .AND. ( KTIME .GT. 1 ) ) WRITE(LOUT, 6005)
IF( IOPT .LE. 0 ) GO TO 2010
IF( KTIME .GT. 1 ) WRITE(LOUT, 6010) TITLE
2010 CONTINUE
IF( KTIME .GT. NCON + 1 ) GO TO 2020
CALL EFFEC1 ( NVAR, NCON, MDCON, KTIME, ICONF, JCON, MCON )
RETURN
C
2020 CONTINUE
IF( KTIME .GT. NCON + 2 ) GO TO 2030
CALL EFFEC2( NCON, NVAR, IPIVOT, JPIVOT, ITERAT, ITER1, JCON,
* MCON, WORK, IWORK, EFFEC, NDFREE,
* NEFEC1, NEFEC2, MODECH )
ICALCU = 1
IF( MODECH .GT. 0 ) GO TO 2030
ICALCU = 2
MDCON = 1
2030 CONTINUE
IF( ICALCU .EQ. 2 ) ICALCU = 0
IF( ICALCU .EQ. 0 ) GO TO 80
IF( ICALCU .EQ. 3 ) GO TO 2050
IF( METH .EQ. 4 ) GO TO 2040
C
C METHOD = 3
C THIS OPTION WORKS COMBINATORIAL-MODE.
C
CALL MSTEPW( NCON, NVAR, KVAR, KCON, KNCON1, KNCON2,
* KTIME, ICONF, JCON, MCON, EFFEC, NDFREE,
* LCON, NEFEC1, NEFEC2, KCONF, MDCON,
* ICALCU, IWORK(JPIVOT,1), WORK(IPIVOT,1) )
IF( ICALCU .EQ. 0 ) GO TO 80
RETURN
C
2040 CONTINUE
C
C METHOD=4
C THIS OPTION WORKS STEPWISE-MODE.
C
CALL STEPWS( NCON, NVAR, KTIME, ICONF, JCON,
* MCON, EFFEC, NDFREE, NEFEC1, NEFEC2,
* LCON, MDCON, ICALCU, IWORK(JPIVOT,1),
* WORK(IPIVOT,1) )

```

```

00405800
00405900
00406000
00406100
00406200
00406300
00406400
00406500
00406600
00406700
00406800
00406900
00407000
00407100
00407200
00407300
00407400
00407500
00407600
00407700
00407800
00407900
00408000
00408100
00408200
00408300
00408400
00408500
00408600
00408700
00408800
00408900
00409000
00409100
00409200
00409300
00409400
00409500
00409600
00409700
00409800
00409900
00410000
00410100
00410200
00410300
00410400
00410500
00410600
00410700
00410800
00410900
00411000
00411100
00411200
00411300
00411400
00411500
00411600
00411700
00411800
00411900
00412000
00412100
00412200
00412300
00412400
00412500
00412600
00412700
00412800

```

	IF(IALCU .EQ. 0)	GO TO 80	00412900
	RETURN		00413000
C			00413100
2050	CONTINUE		00413200
	ICALCU = 2		00413300
	K = 0		00413400
	DO 2070 I1 = 1, NCON		00413500
	IF(NEFEC1(I1) .NE. 2)	GO TO 2070	00413600
	K = K + 1		00413700
	MD1 = MCON(I1)		00413800
	DO 2060 I2 = 1, MD1		00413900
	ICONF(I2, K) = JCON(I2, I1)		00414000
2060	CONTINUE		00414100
2070	CONTINUE		00414200
	MDCON = K		00414300
	IF(MDCON .GT. 0)	RETURN	00414400
	MDCON = 1		00414500
	ICONF(1, 1) = 0		00414600
	IDFREE = IWORK(JPIVOT, 1)		00414700
	DFREE = IDFREE		00414800
	AICL = WORK(IPIVOT, 1)		00414900
	CHIL = AICL + 2.000 * DFREE		00415000
	RETURN		00415100
C			00415200
6005	FORMAT(1H1)	00415300
6010	FORMAT(1H , ' ---- ', 80A1, ' ---- ' /)	00415400
	END		00415500
	SUBROUTINE NORMAL(Z, P, Q, Y)		00415600
	IMPLICIT REAL * 8 (A - H , O - Z)		00415700
C			00415800
C	GENERATE NORMAL DISTRIBUTION		00415900
C			00416000
	X = Z / 1.414213562373095000		00416100
	IF(Z .LT. 0.000) X = - X		00416200
	Y = DERF(X) / 2.000		00416300
	IF(Z .LT. 0.000) Y = - Y		00416400
	P = 0.500 + Y		00416500
	Q = 1.000 - P		00416600
	RETURN		00416700
	END		00416800
	SUBROUTINE PERMUT(IROW, JROW, ICOLM, IPIVOT, WORK, IWORK)		00416900
	REAL*8 WORK(IROW,ICOLM), PIVOT		00417000
	DIMENSION IWORK(JROW,ICOLM)		00417100
C			00417200
	ICOLM1 = ICOLM - 1		00417300
	DO 40 I1 = 1, ICOLM1		00417400
	ICOLM2 = I1 + 1		00417500
	DO 30 I2 = ICOLM2, ICOLM		00417600
	IF(WORK(IPIVOT, I1) .LT. WORK(IPIVOT, I2))	GO TO 30	00417700
C			00417800
	DO 10 I3 = 1, IROW		00417900
	PIVOT = WORK(I3, I1)		00418000
	WORK(I3, I1) = WORK(I3, I2)		00418100
	WORK(I3, I2) = PIVOT		00418200
10	CONTINUE		00418300
	DO 20 I3 = 1, JROW		00418400
	JPIVOT = IWORK(I3, I1)		00418500
	IWORK(I3, I1) = IWORK(I3, I2)		00418600
	IWORK(I3, I2) = JPIVOT		00418700
20	CONTINUE		00418800
30	CONTINUE		00418900
40	CONTINUE		00419000
	RETURN		00419100
	END		00419200


```

      SUBROUTINE PRRTAB(NVAR,NVAR1,NTAB,KDIM,NDIM,MDIM,TABLE,FIT,TSUM, 00419300
      *          FSUM,IOPT,IMT ) 00419400
C 00419500
C TO PRINT THE OUTPUT-INFORMATION AFTER COMPLETING THE MODEL SELECTION.00419600
C 00419700
      IMPLICIT REAL * 8 ( A - H ,   O - Z ) 00419800
      COMMON / OUTIN /   LIN, LOUT 00419900
      INTEGER      FMS( 8) 00420000
      DIMENSION    NDIM(NVAR), MDIM(NVAR), TABLE(NTAB), FIT(NTAB) 00420100
      DIMENSION    TSUM(KDIM), FSUM(KDIM) 00420200
      DIMENSION    SUM1(10), SUM2(10), IMT(10,8), NUM(9) 00420300
      DATA        NUM/'1','2','3','4','5','6','7','8','9'/' 00420400
      DATA        FMS/'(1H+',',14X',' ',',',4H(' <,4H',8A,4H1,'>', 00420500
      *          4H '))' / 00420600
C 00420700
      IF( NVAR .GE. 2 )      GO TO 20 00420800
      IF( NVAR .LE. 0 )      RETURN 00420900
      KTAB = NTAB / 10 + 1 00421000
      DO 10 I = 1, KTAB 00421100
      WRITE(LOUT, 6100) 00421200
      IFIRST = I * 10 - 9 00421300
      ILAST = IFIRST + 9 00421400
      IF( ILAST .GT. NTAB ) ILAST = NTAB 00421500
      WRITE(LOUT, 6120)      ( J ,          J = IFIRST, ILAST ) 00421600
      WRITE(LOUT, 6140)      I , ( TABLE(J), J = IFIRST, ILAST ) 00421700
10 CONTINUE 00421800
      RETURN 00421900
C 00422000
20 MVAR = NVAR - 2 00422100
      IF( MVAR .GE. 1 )      GO TO 30 00422200
      MVAR = 1 00422300
      MODUL1 = 1 00422400
      GO TO 50 00422500
30 MODUL1 = 1 00422600
      DO 40 I = 1, MVAR 00422700
      MODUL1 = MODUL1 * NDIM(I) 00422800
40 CONTINUE 00422900
50 MODULE = MODUL1 * NDIM( NVAR1 ) 00423000
      LDIM = NDIM( NVAR ) 00423100
C 00423200
C ( LDIM+1 ) = ( COLUMN TABLE+COLUMN TOTAL ) 00423300
C 00423400
      KTAB = LDIM / 10 + 1 00423500
      ICHECK = LDIM - LDIM / 10 * 10 00423600
C 00423700
      DO 170 I1 = 1, MODUL1 00423800
      IF( NVAR .LE. 2 )      GO TO 70 00423900
      J1 = 1 00424000
      DO 60 I2 = 1, MVAR 00424100
      J1 = J1 * NDIM(I2) 00424200
      MDIM(I2) = I1 - ( ( I1 - 1 ) / J1 ) * J1 00424300
      IF( I2 .EQ. 1 )      GO TO 60 00424400
      J2 = J1 / NDIM(I2) 00424500
      MDIM(I2) = ( MDIM(I2) - 1 ) / J2 + 1 00424600
60 CONTINUE 00424700
      WRITE(LOUT, 6100) 00424800
      WRITE(LOUT,6100) 00424900
      WRITE(LOUT, 6200)      ( J, J = 1, MVAR ) 00425000
      FMS(4)=NUM(MVAR) 00425100
      WRITE(LOUT,FMS)      ((IMT(I,J),J=1,8),I=1,MVAR) 00425200
      WRITE(LOUT, 6220)      ( MDIM(I), I = 1, MVAR ) 00425300
70 DO 170 I2 = 1, KTAB 00425400
      DO 80 I3 = 1, 10 00425500
      SUM1(I3) = 0.0D0 00425600
      SUM2(I3) = 0.0D0 00425700
80 CONTINUE 00425800

```

C	KFIRST = (I2 - 1) * 10 + 1	00425900
	KLAST = KFIRST + 9	00426000
	IF(KLAST .GT. LDIM) KLAST = LDIM	00426100
	WRITE(LOUT,6110) NVAR, (IMT(NVAR,J),J=1,8)	00426200
	IF((I2 .EQ. KTAB) .AND. (ICHECK .EQ. 0)) GO TO 90	00426300
	WRITE(LOUT, 6120) (J, J = KFIRST, KLAST)	00426400
90	CONTINUE	00426500
	WRITE(LOUT,6130) NVAR1, (IMT(NVAR1,J),J=1,8)	00426600
		00426700
		00426800
C	IFIRST = (I2 - 1) * 10 * MODULE + I1	00426900
	ILAST = I2 * 10 * MODULE + I1 - 1	00427000
	DO 160 I3 = 1, KDIM	00427100
	JFIRST = IFIRST + MODUL1 * (I3 - 1)	00427200
	JLAST = ILAST + MODUL1 * (I3 - 1)	00427300
	IF(JLAST .GT. NTAB) JLAST = NTAB	00427400
		00427500
C	IF(I2 .NE. 1) GO TO 100	00427600
	TSUM(I3) = 0.0D0	00427700
	FSUM(I3) = 0.0D0	00427800
100	CONTINUE	00427900
	I5 = 1	00428000
	IF((I2 .EQ. KTAB) .AND. (ICHECK .EQ. 0)) GO TO 120	00428100
	DO 110 I4 = JFIRST, JLAST, MODULE	00428200
	SUM1(I5) = SUM1(I5) + TABLE(I4)	00428300
	TSUM(I3) = TSUM(I3) + TABLE(I4)	00428400
	SUM2(I5) = SUM2(I5) + FIT(I4)	00428500
	FSUM(I3) = FSUM(I3) + FIT(I4)	00428600
	I5 = I5 + 1	00428700
110	CONTINUE	00428800
	IF(I5 .GT. 10) GO TO 130	00428900
120	CONTINUE	00429000
	SUM1(I5) = SUM1(I5) + TSUM(I3)	00429100
	SUM2(I5) = SUM2(I5) + FSUM(I3)	00429200
130	CONTINUE	00429300
		00429400
C	WRITE(LOUT,6150)	00429500
	IF(I2 .EQ. KTAB) GO TO 140	00429600
	WRITE(LOUT, 6140) I3, (TABLE(J), J = JFIRST, JLAST, MODULE)	00429700
	IF(IOPT .LE. 0) GO TO 160	00429800
	WRITE(LOUT, 6160) (FIT(J), J = JFIRST, JLAST, MODULE)	00429900
	GO TO 160	00430000
140	IF(ICHECK .EQ. 0) GO TO 150	00430100
	WRITE(LOUT, 6140) I3, (TABLE(J), J = JFIRST, JLAST, MODULE)	00430200
	* ,TSUM(I3)	00430300
	GO TO 160	00430400
	IF(IOPT .LE. 0) (FIT(J), J = JFIRST, JLAST, MODULE)	00430500
	* ,FSUM(I3)	00430600
	GO TO 160	00430700
150	WRITE(LOUT, 6140) I3, TSUM(I3)	00430800
	IF(IOPT .LE. 0) GO TO 160	00430900
	WRITE(LOUT, 6160) FSUM(I3)	00431000
160	CONTINUE	00431100
	WRITE(LOUT, 6150)	00431200
	IF(I5 .GT. 10) I5 = 10	00431300
	WRITE(LOUT, 6160) (SUM1(J), J = 1, I5)	00431400
	IF(IOPT .LE. 0) GO TO 170	00431500
	WRITE(LOUT, 6160) (SUM2(J), J = 1, I5)	00431600
170	CONTINUE	00431700
		00431800
C	6100 FORMAT(1H0)	00431900
	6110 FORMAT(1H0, 'ITEM ' / 1H, I6, ' <', 8A1, '>'	00432000
	6120 FORMAT(1H+, 7X, 10I12)	00432100
	6130 FORMAT(1H, I3, / 2X, '<', 8A1, '>'	00432200
	6140 FORMAT(1H, I3, 4X, 10F12.4)	00432300
	6150 FORMAT(1H)	00432400
	6160 FORMAT(1H, 7X, 10F12.4)	00432500
	6200 FORMAT(1H, ' ITEM ', I5,8I12)	00432600
	6220 FORMAT(1H, 10X, I5, 8I12)	00432700
	RETURN	00432800
	END	00432900

	SUBROUTINE EFFEC3(NVAR,NCON,NTITLE,ITERAT,PIPOT,MAXMOD,JCON,	00433000
	* MCON,NDFREE,EFFEC,TITLE,WORK,IDFREE,ITM)	00433100
C		00433200
C	CALCULATE THE EFFECT OF ADDITIONAL TERM BY AIC.	00433300
C		00433400
	REAL * 8 WORK(ITERAT, MAXMOD), EFFEC(NCON)	00433500
	COMMON / OUTIN / LIN, LOUT	00433600
	DIMENSION JCON(NVAR, NCON), MCON(NCON)	00433700
	INTEGER TITLE(NTITLE),BLANK	00433800
	DIMENSION NDFREE(NCON)	00433900
	DIMENSION ITM(10,8),ITM1(10,8)	00434000
	DATA BLANK/' '/	00434100
C		00434200
	WRITE(LOUT, 6005)	00434300
	WRITE(LOUT, 6010) TITLE	00434400
	WRITE(LOUT, 6020)	00434500
	WRITE(LOUT, 6030) IDFREE, WORK(PIPOT, 1)	00434600
	DO 10 I = 1, NCON	00434700
	J1 = MCON(I)	00434800
	WRITE(LOUT, 6050) (JCON(J, I), J = 1, J1)	00434900
C		00435000
	DO 20 I1=1,10	00435100
	DO 20 I2=1,8	00435200
	20 ITM1(I1,I2)=BLANK	00435300
C		00435400
	KK=0	00435500
	DO 30 I3=1,J1	00435600
	IF(JCON(I3,I),EQ.0) GO TO 30	00435700
	KK=KK+1	00435800
	JJ=JCON(I3,I)	00435900
	DO 40 I4=1,8	00436000
	40 ITM1(KK,I4)=ITM(JJ,I4)	00436100
	30 CONTINUE	00436200
	WRITE(LOUT,6001) ((ITM1(I1,I2),I2=1,8),I1=1,KK)	00436300
C		00436400
	WRITE(LOUT, 6060) NDFREE(I), EFFEC(I)	00436500
	10 CONTINUE	00436600
	RETURN	00436700
C		00436800
	6001 FORMAT(1H+,50X,9(8A1,' '))	00436900
	6005 FORMAT(1H1	00437000
	6010 FORMAT(1H , ' ---- ',80A1,' ---- ' /	00437100
	6020 FORMAT(1H , 43H EFFECT OF ADDITIONAL TERM BY AIC BASED ON ,	00437200
	* 27HLIKELIHOOD RATIO STATISTIC / /	00437300
	* 1H , 5X, 10H ADDITIONAL , 2X, 10H DEGREES , 5X,	00437400
	* 6HEFFECT / 1H , 8X, 4H TERM, 5X, 10H OF FREEDOM,	00437500
	* 5X, 6H OF AIC / 1H , 130(1H-))	00437600
	6030 FORMAT(1H , 5X, 2H0 , 10X, I6, 6X, F12.4	00437700
	6050 FORMAT(1H , 5X, 10I1	00437800
	6060 FORMAT(1H+, 17X, I6, 6X, F12.4	00437900
	END	00438000

	SUBROUTINE EFFEC1(NVAR,NCON,MCON,KTIME,ICONF,ICON,MCON)	00438100
	DIMENSION ICONF(NVAR, NCON), ICON(NVAR, NCON), MCON(NCON)	00438200
C	KTIME1 = KTIME - 1	00438300
	MCON = 1	00438400
	IF(KTIME .GT. 1) GO TO 150	00438500
C	DO 10 I = 1, NCON	00438600
	MCON(I) = 0	00438700
	DO 10 J = 1, NVAR	00438800
	ICON(J, I) = 0	00438900
10	CONTINUE	00439000
C	DO 30 I1 = 1, NCON	00439100
	I2 = I1	00439200
	K = 0	00439300
20	J1 = I2 / 2	00439400
	J2 = I2 - J1 * 2	00439500
	I2 = J1	00439600
	K = K + 1	00439700
	IF(J2 .GT. 0) ICON(K, I1) = K	00440000
	IF(J1 .GT. 0) GO TO 20	00440100
30	CONTINUE	00440200
C	DO 50 I = 1, NCON	00440300
	K = 0	00440400
	DO 40 J = 1, NVAR	00440500
	IF(ICON(J, I) .LE. 0) GO TO 40	00440600
	K = K + 1	00440700
	IF(K .GE. J) GO TO 40	00440800
	ICON(K, I) = ICON(J, I)	00440900
	ICON(J, I) = 0	00441000
40	CONTINUE	00441100
	MCON(I) = K	00441200
50	CONTINUE	00441300
C	NCON1 = NCON - 1	00441400
	DO 80 I1 = 1, NCON1	00441500
	IS = I1 + 1	00441600
	DO 70 I2 = IS, NCON	00441700
	IF(MCON(I1) .LE. MCON(I2)) GO TO 70	00441800
	MC = MCON(I1)	00441900
	MCON(I1) = MCON(I2)	00442000
	MCON(I2) = MC	00442100
	DO 60 I3 = 1, NVAR	00442200
	MC = ICON(I3, I1)	00442300
	ICON(I3, I1) = ICON(I3, I2)	00442400
	ICON(I3, I2) = MC	00442500
60	CONTINUE	00442600
70	CONTINUE	00442700
80	CONTINUE	00442800
C	NVAR1 = NVAR - 1	00442900
	NV0 = NVAR + 1	00443000
	NV2 = 1	00443100
	NV3 = 1	00443200
	NV4 = - 1	00443300
	DO 140 I1 = 1, NVAR1	00443400
	NV1 = NV0 - I1	00443500
	NV2 = NV2 * NV1 / I1	00443600
	NV4 = NV4 + NV2	00443700
	NV5 = NV4 + 1	00443800
	DO 130 I2 = NV3, NV4	00443900
	IS = I2 + 1	00444000
	DO 120 I3 = IS, NV5	00444100
	DO 90 I4 = 1, I1	00444200
	IF(ICON(I4, I2) - ICON(I4, I3)) 120, 90, 100	00444300
		00444400
		00444500
		00444600
		00444700

90	CONTINUE		00444800
		GO TO 120	00444900
100	CONTINUE		00445000
	DO 110 I5 = 1, NVAR		00445100
	MC = ICON(I5, I2)		00445200
	ICON(I5, I2) = ICON(I5, I3)		00445300
	ICON(I5, I3) = MC		00445400
110	CONTINUE		00445500
120	CONTINUE		00445600
130	CONTINUE		00445700
	NV3 = NV5 + 1		00445800
140	CONTINUE		00445900
	RETURN		00446000
C			00446100
150	IF(KTIME1 .GT. NVAR)	GO TO 160	00446200
	ICONF(1, 1) = ICON(1, KTIME1)		00446300
	RETURN		00446400
C			00446500
160	IF(KTIME1 .GT. NCON)	RETURN	00446600
	MDCON = MCON(KTIME1)		00446700
	K = 0		00446800
	KTIME2 = KTIME - 2		00446900
	IF(KTIME2 .LE. 1)	GO TO 210	00447000
C			00447100
	DO 200 I1 = 1, KTIME2		00447200
	MD = MCON(I1)		00447300
	DO 180 I2 = 1, MD		00447400
	DO 170 I3 = I2, MDCON		00447500
	IF(ICON(I2, I1) .EQ. ICON(I3, KTIME1))	GO TO 180	00447600
170	CONTINUE		00447700
		GO TO 200	00447800
180	CONTINUE		00447900
	K = K + 1		00448000
	DO 190 I2 = 1, MD		00448100
	ICONF(I2, K) = ICON(I2, I1)		00448200
			00448300
190	CONTINUE		00448400
200	CONTINUE		00448500
C			00448600
210	CONTINUE		00448700
	K = K + 1		00448800
	DO 220 I = 1, MDCON		00448900
	ICONF(I, K) = ICON(I, KTIME1)		00449000
220	CONTINUE		00449100
	MDCON = K		00449200
	RETURN		00449300
	END		00449400

```

SUBROUTINE EFFEC2(NCON,NVAR,IPIVOT,JPIVOT,ITERAT,ITER1,ICON,MCON, 00449500
*      WORK,IWORK,EFFEC,NDFREE,NEFEC1,NEFEC2,MODECH ) 00449600
REAL * 8      WORK, EFFEC 00449700
DIMENSION     EFFEC(NCON), ICON(NVAR, NCON) 00449800
DIMENSION     WORK(ITERAT, NCON), MCON(NCON) 00449900
DIMENSION     NEFEC1(NCON), NEFEC2(NCON) 00450000
DIMENSION     IWORK(ITER1, NCON), NDFREE(NCON) 00450100
C 00450200
DO 10 I1 = 1, NCON 00450300
EFFEC(I1) = WORK(IPIVOT, I1 + 1) - WORK(IPIVOT, 1) 00450400
NDFREE(I1) = IWORK(JPIVOT, 1) - IWORK(JPIVOT, I1 + 1) 00450500
10 CONTINUE 00450600
C 00450700
MODECH = 0 00450800
DO 90 J1 = 1, 2 00450900
IF( J1 .EQ. 1 ) GO TO 30 00451000
DO 20 I1 = 1, NCON 00451100
IF( EFFEC(I1) .GE. 0.000 ) GO TO 20 00451200
MODECH = 1 00451300
NEFEC1(I1) = 1 00451400
IF( I1 .LE. NVAR ) NEFEC2(I1) = 1 00451500
20 CONTINUE 00451600
IF( MODECH .EQ. 0 ) RETURN 00451700
30 CONTINUE 00451800
C 00451900
NV0 = NVAR + 1 00452000
NV2 = NVAR 00452100
NV3 = NV0 00452200
NV4 = NVAR 00452300
DO 90 I1 = 2, NVAR 00452400
NV1 = NV0 - I1 00452500
NV2 = NV2 * NV1 / I1 00452600
NV4 = NV4 + NV2 00452700
NV5 = NV3 - 1 00452800
DO 80 I2 = NV3, NV4 00452900
IF( ( NEFEC1(I2) .LE. 0 ) .AND. ( J1 .EQ. 2 ) ) GO TO 80 00453000
ICHECK = 0 00453100
JCHECK = 0 00453200
MD1 = MCON(I2) 00453300
DO 70 I3 = 1, NV5 00453400
MD2 = MCON(I3) 00453500
DO 50 I4 = 1, MD2 00453600
DO 40 I5 = 1, MD1 00453700
IF( ICON(I5, I2) .EQ. ICON(I4, I3) ) GO TO 50 00453800
40 CONTINUE 00453900
GO TO 70 00454000
50 CONTINUE 00454100
IF( J1 .EQ. 2 ) GO TO 60 00454200
EFFEC(I2) = EFFEC(I2) - EFFEC(I3) 00454300
NDFREE(I2) = NDFREE(I2) - NDFREE(I3) 00454400
GO TO 70 00454500
60 CONTINUE 00454600
ICHECK = ICHECK + 1 00454700
IF( NEFEC1(I3) .GT. 0 ) JCHECK = JCHECK + 1 00454800
70 CONTINUE 00454900
IF( ( J1 .EQ. 2 ) .AND. ( ICHECK .EQ. JCHECK ) ) 00455000
* NEFEC2(I2) = 1 00455100
80 CONTINUE 00455200
NV3 = NV4 + 1 00455300
90 CONTINUE 00455400
C 00455500
DO 100 I1 = 1, NCON 00455600
NEFEC1(I1) = NEFEC1(I1) + NEFEC2(I1) 00455700
IF( NEFEC1(I1) .NE. 1 ) GO TO 100 00455800
MODECH = 2 00455900
100 CONTINUE 00456000
RETURN 00456100
END 00456200

```

C	SUBROUTINE CONBI(NVAR,NCON,KVAR,KCON,ICON,MCON)	00456300
C	ARRANGE THE COMBINATORIAL PROCEDURE	00456400
C	DIMENSION ICON(NVAR, NCON), MCON(NCON)	00456500
C	KCON = 2 ** KVAR - 1	00456600
C	DO 10 I = 1, NCON	00456700
	MCON(I) = 0	00456800
	DO 10 J = 1, NVAR	00456900
	ICON(J, I) = 0	00457000
10	CONTINUE	00457100
C	DO 30 I1 = 1, KCON	00457200
	I2 = I1	00457300
	K = 0	00457400
20	J1 = I2 / 2	00457500
	J2 = I2 - J1 * 2	00457600
	I2 = J1	00457700
	K = K + 1	00457800
	IF(J2 .GT. 0)	00457900
	IF(J1 .GT. 0)	00458000
30	CONTINUE	00458100
	ICON(K,I1) = K	00458200
	GO TO 20	00458300
C	DO 50 I = 1, KCON	00458400
	K = 0	00458500
	DO 40 J = 1, KVAR	00458600
	IF(ICON(J, I) .LE. 0)	00458700
	K = K + 1	00458800
	IF(K .GE. J)	00458900
	ICON(K, I) = ICON(J, I)	00459000
	ICON(J, I) = 0	00459100
40	CONTINUE	00459200
	MCON(I) = K	00459300
50	CONTINUE	00459400
	IF(KCON .LE. 1)	00459500
	RETURN	00459600
C	KCON1 = KCON - 1	00459700
	DO 80 I1 = 1, KCON1	00459800
	IS = I1 + 1	00459900
	DO 70 I2 = IS, KCON	00460000
	IF(MCON(I1) .LE. MCON(I2))	00460100
	MC = MCON(I1)	00460200
	MCON(I1) = MCON(I2)	00460300
	MCON(I2) = MC	00460400
	DO 60 I3 = 1, KVAR	00460500
	MC = ICON(I3, I1)	00460600
	ICON(I3, I1) = ICON(I3, I2)	00460700
	ICON(I3, I2) = MC	00460800
60	CONTINUE	00460900
70	CONTINUE	00461000
80	CONTINUE	00461100
C	KVAR1 = KVAR - 1	00461200
	NV0 = KVAR + 1	00461300
	NV2 = 1	00461400
	NV3 = 1	00461500
	NV4 = - 1	00461600
	DO 140 I1 = 1, KVAR1	00461700
	NV1 = NV0 - I1	00461800
	NV2 = NV2 * NV1 / I1	00461900
	NV4 = NV4 + NV2	00462000
	NV5 = NV4 + 1	00462100
	DO 130 I2 = NV3, NV4	00462200
	IS = I2 + 1	00462300
	DO 120 I3 = IS, NV5	00462400
	DO 90 I4 = 1, I1	00462500
	IF(ICON(I4, I2) = ICON(I4, I3))	00462600
90	CONTINUE	00462700
	GO TO 120	00462800
100	CONTINUE	00462900
	DO 110 I5 = 1, KVAR	00463000
	MC = ICON(I5, I2)	00463100
	ICON(I5, I2) = ICON(I5, I3)	00463200
	ICON(I5, I3) = MC	00463300
110	CONTINUE	00463400
120	CONTINUE	00463500
130	CONTINUE	00463600
	NV3 = NV5 + 1	00463700
140	CONTINUE	00463800
	RETURN	00463900
	END	00464000
		00464100
		00464200
		00464300
		00464400
		00464500

```

      SUBROUTINE STEPWS(NCON,NVAR,KTIME,ICONF,ICON,MCON,EFFEC,NDFREE, 00464600
      * NEFEC1,NEFEC2,NEFEC3,MDCON,ICALCU,MDFREE,AICLM )00464700
C
C EXECUTE THE STEPWISE PROCEDURE 00464800
C 00464900
C 00465000
      IMPLICIT REAL * 8      ( A - H, O - Z ) 00465100
      COMMON / ICRIT /      CHIG, CHIL, PROBG, PROBL, AICG, AICL, 00465200
      * IDFREE, ICONBI 00465300
      DIMENSION ICONF(NVAR, NCON), ICON(NVAR, NCON), MCON(NCON) 00465400
      DIMENSION EFFEC(NCON), NDFREE(NCON) 00465500
      DIMENSION NEFEC1(NCON), NEFEC2(NCON), NEFEC3(NCON) 00465600
C 00465700
C SET THE INITIAL PARAMETER 00465800
C 00465900
      CHIG = - 1.000 00466000
      PROBG = 0.000 00466100
      PROBL = 0.000 00466200
      AICG = 0.000 00466300
      IEFEC = 0 00466400
C 00466500
      IF( KTIME .EQ. NCON + 2 ) GO TO 90 00466600
      DO 10 I = 1, NCON 00466700
      NEFEC2(I) = 0 00466800
      IF( NEFEC1(I) .EQ. 2 ) NEFEC2(I) = 1 00466900
      IF( IEFEC .NE. 0 ) GO TO 10 00467000
      IF( ( NEFEC1(I) .NE. 1 ) .OR. ( NEFEC3(I) .NE. 0 ) ) 00467100
      * GO TO 10 00467200
      NEFEC2(I) = 1 00467300
      IEFEC = I 00467400
10 CONTINUE 00467500
      IF( IEFEC .NE. 0 ) GO TO 20 00467600
      IALCU = 0 00467700
      RETURN 00467800
C 00467900
20 CONTINUE 00468000
      NV0 = NVAR + 1 00468100
      NV2 = NVAR 00468200
      NV3 = NV0 00468300
      NV4 = NVAR 00468400
      DO 80 I1 = 2, NVAR 00468500
      NV1 = NV0 - I1 00468600
      NV2 = NV2 * NV1 / I1 00468700
      NV4 = NV4 + NV2 00468800
      NV5 = NV3 - 1 00468900
      IF( ( IEFEC .LT. NV3 ) .OR. ( IEFEC .GT. NV4 ) ) 00469000
      * GO TO 70 00469100
      DO 60 I2 = NV3, NV4 00469200
      IF( I2 .NE. IEFEC ) GO TO 60 00469300
      MD1 = MCON(I2) 00469400
      DO 50 I3 = 1, NV5 00469500
      IF( NEFEC2(I3) .EQ. 1 ) GO TO 50 00469600
      MD2 = MCON(I3) 00469700
      DO 40 I4 = 1, MD2 00469800
      DO 30 I5 = 1, MD1 00469900
      IF( ICON(I5, I2) .EQ. ICON(I4, I3) ) GO TO 40 00470000
30 CONTINUE 00470100
      GO TO 50 00470200
40 CONTINUE 00470300
      NEFEC2(I3) = 1 00470400
50 CONTINUE 00470500
60 CONTINUE 00470600
70 CONTINUE 00470700
      NV3 = NV4 + 1 00470800
80 CONTINUE 00470900
C 00471000
90 CONTINUE 00471100
      K = 0 00471200
      IDFREE = MDFREE 00471300
      SUM = 0.000 00471400
      CHIL = 0.000 00471500
      AICL = AICLM 00471600

```


C			00471700
	DO 110 I1 = 1, NCON		00471800
	IF(NEFEC2(I1) .LE. 0)	GO TO 110	00471900
	IF(NEFEC1(I1) .NE. 2)	SUM = SUM + EFEC(I1)	00472000
	IDFREE = IDFREE - NDFREE(I1)		00472100
	AICL = AICL + EFEC(I1)		00472200
	K = K + 1		00472300
	DO 100 I2 = 1, NVAR		00472400
	ICONF(I2, K) = ICON(I2, I1)		00472500
100	CONTINUE		00472600
110	CONTINUE		00472700
	MDCON = K		00472800
	IF(MDCON .GT. 0)	GO TO 120	00472900
	MDCON = 1		00473000
	ICONF(1, 1) = 0		00473100
120	CONTINUE		00473200
	DFREE = IDFREE		00473300
	CHIL = AICL + 2.0D0 * DFREE		00473400
C			00473500
	IF(SUM .GE. 0.0D0)	GO TO 150	00473600
	ICALCU = 2		00473700
	DO 130 I1 = 1, NCON		00473800
	IF(NEFEC2(I1) .NE. 1)	GO TO 130	00473900
	NEFEC1(I1) = 2		00474000
	NEFEC3(I1) = 0		00474100
130	CONTINUE		00474200
	DO 140 I1 = 1, NCON		00474300
	IF((NEFEC1(I1) .EQ. 1) .AND. (NEFEC3(I1) .EQ. 0))		00474400
	* IICALCU = 1		00474500
140	CONTINUE		00474600
		GO TO 180	00474700
C			00474800
150	CONTINUE		00474900
	ICALCU = 3		00475000
	IF(KTIME .EQ. NCON + 2)	ICALCU = 2	00475100
	DO 170 I1 = 1, NCON		00475200
	IF((NEFEC1(I1) .EQ. 2) .OR. (NEFEC2(I1) .NE. 1))		00475300
	* GO TO 160		00475400
	NEFEC2(I1) = 0		00475500
	NEFEC3(I1) = 1		00475600
		GO TO 170	00475700
160	CONTINUE		00475800
	IF((NEFEC1(I1) .EQ. 1) .AND. (NEFEC3(I1) .EQ. 0))		00475900
	* IICALCU = 1		00476000
170	CONTINUE		00476100
C			00476200
180	CONTINUE		00476300
	IF(IICALCU .EQ. 1)	RETURN	00476400
	NV0 = NVAR + 1		00476500
	NV2 = NVAR		00476600
	NV3 = NV0		00476700
	NV4 = NVAR		00476800
	DO 230 I1 = 2, NVAR		00476900
	NV1 = NV0 - I1		00477000
	NV2 = NV2 * NV1 / I1		00477100
	NV4 = NV4 + NV2		00477200
	NV5 = NV3 - 1		00477300
	DO 220 I2 = NV3, NV4		00477400
	IF((NEFEC1(I2) .NE. 1) .OR. (NEFEC3(I2) .NE. 1))		00477500
	* GO TO 220		00477600
	ICHECK = 0		00477700
	JCHECK = 0		00477800
	MD1 = MCON(I2)		00477900
	DO 210 I3 = 1, NV5		00478000
	MD2 = MCON(I3)		00478100
	DO 200 I4 = 1, MD2		00478200
	DO 190 I5 = 1, MD1		00478300
	IF(ICON(I5, I2) .EQ. ICON(I4, I3))	GO TO 200	00478400
190	CONTINUE		00478500
		GO TO 210	00478600
200	CONTINUE		00478700
	ICHECK = ICHECK + 1		00478800
	IF(NEFEC3(I3) .EQ. 0)	JCHECK = JCHECK + 1	00478900
210	CONTINUE		00479000
	IF((ICHECK .EQ. 0) .OR. (ICHECK .NE. JCHECK))		00479100
	* GO TO 220		00479200
	NEFEC3(I2) = 0		00479300
	ICALCU = 1		00479400
	RETURN		00479500
220	CONTINUE		00479600
	NV3 = NV4 + 1		00479700
230	CONTINUE		00479800
	RETURN		00479900
	END		00480000

```

SUBROUTINE MSTEPW(NCON,NVAR,KVAR,KCON,KNCON1,KNCON2,KTIME,ICONF, 00480100
*      ICON,MCON,EFFEC,NDFREE,LCON,NEFEC1,NEFEC2, 00480200
*      KCONF,MDCON,ICALCU,MDFREE,AICLM ) 00480300
IMPLICIT REAL * 8 ( A - H, D - Z ) 00480400
COMMON / ICRIT / CHIG, CHIL, PROBG, PROBL, AICG, AICL, 00480500
*      IDFREE, ICONB1 00480600
DIMENSION ICONF(NVAR, NCON), ICON(NVAR, NCON), MCON(NCON) 00480700
DIMENSION EFFEC(NCON), NDFREE(NCON) 00480800
DIMENSION NEFEC1(NCON), NEFEC2(NCON) 00480900
DIMENSION KCONF(KNCON1, 1), LCON(KNCON2) 00481000
C 00481100
C SET THE INITIAL PARAMETER 00481200
C 00481300
CHIG = - 1.000 00481400
PROBG = 0.000 00481500
PROBL = 0.000 00481600
AICG = 0.000 00481700
C 00481800
DO 10 I = 1, NCON 00481900
NEFEC2(I) = 0 00482000
IF( NEFEC1(I) .EQ. 2 ) NEFEC2(I) = 1 00482100
10 CONTINUE 00482200
IF( KTIME .EQ. NCON + 2 ) GO TO 190 00482300
IF( ICONB1 .NE. 0 ) GO TO 120 00482400
C 00482500
CALL CONB1( KNCON1, KNCON2, KVAR, KCON, 00482600
*      KCONF, LCON ) 00482700
DO 30 I1 = 1, NCON 00482800
IF( NEFEC1(I1) .NE. 1 ) GO TO 30 00482900
K = K + 1 00483000
DO 20 I2 = 1, KVAR 00483100
DO 20 I3 = 1, KCON 00483200
IF( KCONF(I2, I3) .EQ. K ) KCONF(I2, I3) = - 11 00483300
20 CONTINUE 00483400
30 CONTINUE 00483500
DO 40 I1 = 1, KCON 00483600
DO 40 I2 = 1, KVAR 00483700
KCONF(I2, I1) = - KCONF(I2, I1) 00483800
40 CONTINUE 00483900
IF( KVAR .LE. 1 ) GO TO 90 00484000
C 00484100
DO 80 I1 = 2, KCON 00484200
L = LCON(I1) 00484300
IF( L .LE. 1 ) GO TO 80 00484400
J1 = KCONF(L, I1) 00484500
L1 = L - 1 00484600
MD1 = MCON(J1) 00484700
DO 70 I2 = 1, L1 00484800
J2 = KCONF(I2, I1) 00484900
MD2 = MCON(J2) 00485000
DO 60 I3 = 1, MD2 00485100
DO 50 I4 = 1, MD1 00485200
IF( ICON(I4, J1) .EQ. ICON(I3, J2) ) GO TO 60 00485300
50 CONTINUE 00485400
GO TO 70 00485500
60 CONTINUE 00485600
LCON(I1) = 0 00485700
GO TO 80 00485800
70 CONTINUE 00485900
80 CONTINUE 00486000
C 00486100
90 CONTINUE 00486200
K = 0 00486300
DO 110 I1 = 1, KCON 00486400
IF( LCON(I1 - K) .NE. 0 ) GO TO 110 00486500
J1 = I1 - K 00486600
K = K + 1 00486700

```

J2 = KCON - K		00486800
DO 100 I2 = J1, J2		00486900
LCON(I2) = LCON(I2 + 1)		00487000
DO 100 I3 = 1, KVAR		00487100
KCONF(I3, I2) = KCONF(I3, I2 + 1)		00487200
100 CONTINUE		00487300
110 CONTINUE		00487400
KCON = KCON - K		00487500
ICONBI = 0		00487600
C		00487700
120 CONTINUE		00487800
ICONBI = ICONBI + 1		00487900
K = 0		00488000
M1 = LCON(ICONBI)		00488100
DO 130 I1 = 1, M1		00488200
J1 = KCONF(I1, ICONBI)		00488300
NEFEC2(J1) = 1		00488400
130 CONTINUE		00488500
C		00488600
NV0 = NVAR + 1		00488700
NV2 = NVAR		00488800
NV3 = NV0		00488900
NV4 = NVAR		00489000
DO 180 I1 = 2, NVAR		00489100
NV1 = NV0 - I1		00489200
NV2 = NV2 * NV1 / I1		00489300
NV4 = NV4 + NV2		00489400
NV5 = NV3 - 1		00489500
DO 170 I2 = NV3, NV4		00489600
IF(NEFEC2(I2) .NE. 1)	GO TO 170	00489700
MD1 = MCON(I2)		00489800
DO 160 I3 = 1, NV5		00489900
IF(NEFEC2(I3) .EQ. 1)	GO TO 160	00490000
MD2 = MCON(I3)		00490100
DO 150 I4 = 1, MD2		00490200
DO 140 I5 = 1, MD1		00490300
IF(ICON(I5, I2) .EQ. ICON(I4, I3))	GO TO 150	00490400
140 CONTINUE	GO TO 160	00490500
		00490600
150 CONTINUE		00490700
NEFEC2(I3) = 1		00490800
160 CONTINUE		00490900
170 CONTINUE		00491000
NV3 = NV4 + 1		00491100
180 CONTINUE		00491200
C		00491300
190 CONTINUE		00491400
K = 0		00491500
IDFREE = MDFREE		00491600
SUM = 0.0D0		00491700
CHIL = 0.0D0		00491800
AICL = AICLM		00491900
C		00492000
DO 210 I1 = 1, NCON		00492100
IF(NEFEC2(I1) .LE. 0)	GO TO 210	00492200
IF(NEFEC1(I1) .NE. 2)	SUM = SUM + EFEC(I1)	00492300
IDFREE = IDFREE - NDFREE(I1)		00492400
AICL = AICL + EFEC(I1)		00492500
K = K + 1		00492600
DO 200 I2 = 1, NVAR		00492700
ICONF(I2, K) = ICON(I2, I1)		00492800
200 CONTINUE		00492900
210 CONTINUE		00493000
MDCON = K		00493100
IF(MDCON .GT. 0)	GO TO 220	00493200
MDCON = 1		00493300
ICONF(1, 1) = 0		00493400
220 CONTINUE		00493500
DFREE = IDFREE		00493600
CHIL = AICL + 2.0D0 * DFREE		00493700

C	IF((KTIME .NE. NCON + 2) .AND. (SUM .GE. 0.000))	00493800
	* GO TO 260	00493900
	ICALCU = 2	00494000
	KVAR = 0	00494100
	DO 240 I1 = 1, NCON	00494200
	IF(NEFEC2(I1) .NE. 1)	00494300
	NEFEC1(I1) = 2	00494400
	GO TO 230	00494500
	GO TO 240	00494600
230	CONTINUE	00494700
	IF(NEFEC1(I1) .EQ. 1)	00494800
240	CONTINUE	00494900
	IF(KVAR .EQ. 0)	00495000
	ICONBI = 0	00495100
	ICALCU = 1	00495200
	RETURN	00495300
C		00495400
250	CONTINUE	00495500
	ICALCU = 2	00495600
	RETURN	00495700
C		00495800
260	CONTINUE	00495900
	DO 270 I = 1, NCON	00496000
	IF(NEFEC1(I) .NE. 2)	00496100
	NEFEC2(I) = 0	00496200
270	CONTINUE	00496300
	IF(ICONBI .GE. KCON)	00496400
	ICALCU = 1	00496500
	RETURN	00496600
C		00496700
280	CONTINUE	00496800
	ICALCU = 3	00496900
	RETURN	00497000
	END	

```

SUBROUTINE QCHART(WORK,IWORK,XX,MQ,ICOLM,M1,MA,MX,INDEV,      00497100
*          LIMIT1,LIMIT2)                                     00497200
C                                                         00497300
C THIS SUBROUTINE EXECUTES THE MULTI-CHARACTERISTIC CONTROL CHART AND 00497400
C REPRESENTS GRAPHICALLY ELLIPTIC CHART IN THE CASE OF TWO-DIMENSIONAL 00497500
C DATA.                                                    00497600
C                                                         00497700
C      COMMON / OUTIN / LIN, LOUT                               00497800
C      COMMON   NUMDT(20)                                       00497900
C      COMMON / COSCL / SCALE, SCALEX, SCALEY, ORSCAL          00498000
C      COMMON / COOPT / ILIOPT, INROPT                          00498100
C      DIMENSION WORK( LIMIT1 ), IWORK( LIMIT2 )               00498200
C      DIMENSION XX(MA),MQ(MX)                                  00498300
C      DIMENSION CHI( 40 )                                     00498400
C      DATA CHI / 3.84146, 5.99146, 7.81473, 9.48773, 11.0705, 12.5916, 00498500
*      14.0671, 15.5073, 16.9190, 18.3070, 19.6751, 21.0261, 00498600
*      22.3620, 23.6848, 24.9958, 26.2962, 27.5871, 28.8693, 00498700
*      30.1435, 31.4104, 32.6706, 33.9244, 35.1725, 36.4150, 00498800
*      37.6525, 38.8851, 40.1133, 41.3371, 42.5570, 43.7730, 00498900
*      44.9853, 46.1943, 47.3999, 48.6020, 49.8018, 50.9985, 00499000
*      52.1923, 53.3835, 54.5722, 55.7585 / 00499100
C                                                         00499200
C      IROW=NUMDT(INDEV)                                       00499300
C      ILIOPT=0                                                00499400
C      INROPT=0                                                00499500
C      SCALE=0.0                                               00499600
C      ORSCAL=0.0                                              00499700
C                                                         00499800
C      SET THE DEFAULT VALUE OF COSCL ( COMMON OF SCALE )    00499900
C                                                         00500000
C      IF( SCALE .LE. 1.0E-10 ) SCALE = 4.0                  00500100
C      IF( ILIOPT .LE. 0 ) ILIOPT = 1                        00500200
C      IF( INROPT .LE. 0 ) INROPT = 0                        00500300
C      IF( ORSCAL .LE. 0.0 ) ORSCAL = 2.0                   00500400
C      N = ICOLM                                              00500500
C      IF( ICOLM .GT. 40 ) N = 40                            00500600
C      CHISQR = CHI ( N )                                    00500700
C                                                         00500800
C      WRITE(LOUT, 6200)                                       00500900
C      WRITE(LOUT, 6000)                                       00501000
C      WRITE(LOUT, 6100) LIN, LOUT, ILIOPT, INROPT, SCALE, ORSCAL, 00501100
*      CHISQR, IROW, ICOLM                                     00501200
C                                                         00501300
C      KNUM=NUMDT(INDEV)                                       00501400
C      IF(KNUM.EQ.0) GO TO 1000                                00501500
C                                                         00501600
C      CALL CLEAR(WORK, IWORK, LIMIT1, LIMIT2)                00501700
C      CALL WRKARA(IROW, ICOLM, CHISQR, XX, MQ, MA, MX,      00501800
*      WORK, IWORK, M1, INDEV, LIMIT1, LIMIT2 )              00501900
C      RETURN                                                  00502000
C                                                         00502100
C      1000 WRITE(LOUT,6001) INDEV                             00502200
C      RETURN                                                  00502300
C                                                         00502400
C      5100 FORMAT(4I5, 4F10.5)                                00502500
C      5150 FORMAT(F10.5)                                       00502600
C      5200 FORMAT(20A4)                                         00502700
C      6000 FORMAT(1H0, 33HI N P U T I N F O R M A T I O N ) 00502800
C      6100 FORMAT(1H0, 8HLIN =, I5, 12X, 8HLOUT =, I5, 12X, 00502900
*      8HILIOPT =, I5, 12X,                                     00503000
*      8HINROPT =, I5 / 1H, 8HSCALE =, F10.4, 7X,             00503100
*      8HORSCAL =, F10.4, 7X,                                   00503200
*      8HCHISQR =, F10.4 / 1H, 8HIROW = I5, 12X,              00503300
*      8HICOLM =, I5 )                                         00503400
C      6200 FORMAT(1H1,///1X,'+-----+/' 00503500
*      1X,'I MULTI-CHARACTERISTIC I'/' 00503600
*      1X,'I CONTROL CHART I'/' 00503700
*      1X,'I ( Q-CHART ) I'/' 00503800
*      1X,'+-----+/' ) 00503900
C      6001 FORMAT(///1H, '--- FILE NUMBER IS',I3,'. THIS FILE DOES NOT', 00504000
*      ' INCLUDE DATA ---' ) 00504100
C      END                                                    00504200

```

	SUBROUTINE MEAN1(IROW, ICOLM, X, XMEAN)	00504300
C		00504400
C	CALCULATION OF MEANS	00504500
C		00504600
	DIMENSION X(IROW, ICOLM), XMEAN(ICOLM)	00504700
C		00504800
	FROW = IROW	00504900
C		00505000
	DO 20 I = 1, ICOLM	00505100
	XMEAN(I) = 0.0	00505200
	DO 10 J = 1, IROW	00505300
	XMEAN(I) = XMEAN(I) + X(J, I)	00505400
10	CONTINUE	00505500
	XMEAN(I) = XMEAN(I) / FROW	00505600
20	CONTINUE	00505700
	RETURN	00505800
	END	00505900
	SUBROUTINE COVMAT(IROW, ICOLM, X, XMEAN, COV)	00506000
C		00506100
C	CALCULATION OF VARIANCE - COVARIANCE MATRIX	00506200
C		00506300
	DIMENSION X(IROW, ICOLM), XMEAN(ICOLM), COV(ICOLM, ICOLM)	00506400
C		00506500
	FROW = IROW	00506600
C		00506700
	DO 20 I1 = 1, ICOLM	00506800
	DO 20 I2 = 1, I1	00506900
	COV(I1, I2) = 0.0	00507000
	DO 10 I3 = 1, IROW	00507100
	F11 = X(I3, I1) - XMEAN(I1)	00507200
	F12 = X(I3, I2) - XMEAN(I2)	00507300
	COV(I1, I2) = COV(I1, I2) + F11 * F12	00507400
10	CONTINUE	00507500
	COV(I1, I2) = COV(I1, I2) / FROW	00507600
20	CONTINUE	00507700
C		00507800
	ICOLM1 = ICOLM - 1	00507900
	DO 30 I1 = 1, ICOLM1	00508000
	JCOLM = I1 + 1	00508100
	DO 30 I2 = JCOLM, ICOLM	00508200
	COV(I1, I2) = COV(I2, I1)	00508300
30	CONTINUE	00508400
	RETURN	00508500
	END	00508600
	SUBROUTINE MATOUT(IROW, ICOLM, X, SWETCH)	00508700
C		00508800
C	PRINT OF THE MATRIX X	00508900
C		00509000
	DIMENSION X(IROW, ICOLM)	00509100
	COMMON / OUTIN / LIN, LOUT	00509200
C		00509300
C	IF SWETCH IS LITTler THAN OR EQUAL TO ZERO THEN THIS PROGRAM WRITE	00509400
C	(IROW, ICOLM) MATRIX .	00509500
C	IF SWETCH IS GREATER THAN ZERO THEN THIS PROGRAM WRITES TRIANGULA	00509600
C		00509700
	I1ROW = 1	00509800
	NCOLM = (ICOLM - 1) / 5 + 1	00509900
C		00510000
	DO 20 I1 = 1, NCOLM	00510100
	IF(SWETCH .GT. 0.0) I1ROW = 5 * I1 - 4	00510200
	ILAST = 5 * I1	00510300
	IFIRST = ILAST - 4	00510400
	IF(ILAST .GT. ICOLM) ILAST = ICOLM	00510500
	WRITE(LOUT, 6200) (K, K = IFIRST, ILAST)	00510600
C		00510700
	ILAST1 = ILAST	00510800
	DO 10 I2 = I1ROW, IROW	00510900
	IF((SWETCH .GT. 0.0) .AND. (I2 .LE. ILAST)) ILAST1 = I2	00511000
	WRITE(LOUT, 6300) I2, (X(I2, J), J = IFIRST, ILAST1)	00511100
10	CONTINUE	00511200
	IF(IROW .NE. 1)	00511300
20	CONTINUE	00511400
C		00511500
6100	FORMAT(1H0, 120(1H-) /)	00511600
6200	FORMAT(1H , 12HROW / COLUMN, I8, 4(12X, I8) /)	00511700
6300	FORMAT(1H , 13, 9X, 5G20.7)	00511800
	RETURN	00511900
	END	00512000

	SUBROUTINE INVMAT(ICOLM, RMAT, IPIVOT, RMATIV, DET)	00512100
C		00512200
C	CALCULATION OF THE INVERSE MATRIX AND THE DETERMINANT	00512300
C		00512400
	COMMON / OUTIN / LIN, LOUT	00512500
	DIMENSION RMAT(ICOLM, ICOLM), RMATIV(ICOLM, ICOLM), IPIVOT(ICOLM)	00512600
C		00512700
	DET = 1.0	00512800
C		00512900
C	SET THE INITIAL VALUES FOR IPIVOT AND RMATIV	00513000
C	RMATIV SET TO IDENTITY MATRIX .	00513100
C		00513200
	DO 10 I = 1, ICOLM	00513300
	IPIVOT(I) = 1	00513400
	DO 10 J = 1, ICOLM	00513500
	RMATIV(I, J) = 0.0	00513600
	IF(I .EQ. J) RMATIV(I, J) = 1.0	00513700
10	CONTINUE	00513800
C		00513900
	DO 80 I = 1, ICOLM	00514000
	X = RMAT(I, 1)	00514100
	X = ABS(X)	00514200
	NCOLM = I	00514300
	NROW = I	00514400
C		00514500
C	SELECT MAXIMUM ELEMENT OF THE MATRIX	00514600
C		00514700
	DO 30 J = 1, ICOLM	00514800
	DO 30 K = 1, ICOLM	00514900
	Y = RMAT(J, K)	00515000
	Y = ABS(Y)	00515100
	IF(X - Y) 20, 30, 30	00515200
20	X = Y	00515300
	IF(X .LE. 0.0) GO TO 100	00515400
	NCOLM = J	00515500
	NROW = K	00515600
30	CONTINUE	00515700
C		00515800
C	CHANGING TO THE MATRIX OF THE I - TH COLUMN AND THE NROW - TH COLUMN	00515900
C		00516000
	II = IPIVOT(I)	00516100
	IPIVOT(I) = IPIVOT(NROW)	00516200
	IPIVOT(NROW) = II	00516300
	DO 40 J = 1, ICOLM	00516400
	X = RMAT(J, I)	00516500
	RMAT(J, I) = RMAT(J, NROW)	00516600
	RMAT(J, NROW) = X	00516700
40	CONTINUE	00516800
C		00516900
C	CHANGING TO THE MATRIX AND THE INVERSE MATRIX OF THE I - TH ROW	00517000
C	AND THE NCOLM - TH ROW .	00517100
C		00517200
	DO 50 J = 1, ICOLM	00517300
	X = RMAT(I, J)	00517400
	RMAT(I, J) = RMAT(NCOLM, J)	00517500
	RMAT(NCOLM, J) = X	00517600
	X = RMATIV(I, J)	00517700
	RMATIV(I, J) = RMATIV(NCOLM, J)	00517800
	RMATIV(NCOLM, J) = X	00517900
50	CONTINUE	00518000
C		00518100
C	RMAT(I, I) IS THE PIVOT .	00518200
C		00518300
	PIVOT = RMAT(I, I)	00518400
	DET = DET * PIVOT	00518500
	IF(DET .LE. 10.0E-20) GO TO 100	00518600

C	DO 60 J = 1, ICOLM	00518700
	RMAT(I, J) = RMAT(I, J) / PIVOT	00518800
	RMATIV(I, J) = RMATIV(I, J) / PIVOT	00518900
60	CONTINUE	00519000
	DO 70 J = 1, ICOLM	00519100
	PIV = RMAT(J, I)	00519200
	DO 70 K = 1, ICOLM	00519300
	IF(I .EQ. J) GO TO 70	00519400
	RMATIV(J, K) = RMATIV(J, K) - RMATIV(I, K) * PIV	00519500
	RMAT(J, K) = RMAT(J, K) - RMAT(I, K) * PIV	00519600
70	CONTINUE	00519700
80	CONTINUE	00519800
	ICOLM1 = ICOLM - 1	00519900
C		00520000
	DO 90 I = 1, ICOLM1	00520100
	II = I + 1	00520200
	DO 90 J = II, ICOLM	00520300
	IF(IPIVOT(I) .LE. IPIVOT(J)) GO TO 90	00520400
C		00520500
C	TRANSFORM I - TH COLUMN TO J - TH ROW AND J - TH COLUMN TO I - TH	00520600
C		00520700
	IPIV = IPIVOT(I)	00520800
	IPIVOT(I) = IPIVOT(J)	00520900
	IPIVOT(J) = IPIV	00521000
	DO 90 K = 1, ICOLM	00521100
	X = RMATIV(I, K)	00521200
	RMATIV(I, K) = RMATIV(J, K)	00521300
	RMATIV(J, K) = X	00521400
90	CONTINUE	00521500
	RETURN	00521600
100	I = I - 1	00521700
	WRITE(LOUT, 6100) I, (IPIVOT(J), J= 1, I)	00521800
	DET = 0.0	00521900
6100	FORMAT(1H , 47HDETERMINANT OF THE MATRIX IS ZERO THEN RANK OF ,	00522000
	* 16HMATRIX EQUAL TO , 13, 2H . / 1H , 13HUSING COLUMNS /	00522100
	* (1H , 2515))	00522200
	RETURN	00522300
	END	00522400
		00522500


```

SUBROUTINE CONTCH( IROW, QFORM, CHISQR, ICOUNT, IDELET, 00522600
* IFIRST, ILAST ) 00522700
C 00522800
C ILLUSTRATION OF THE MULTI - CHARACTERISTIC CONTROL CHART 00522900
C 00523000
DIMENSION QFORM(IROW), IDELET(IROW), CHART(41, 60) 00523100
COMMON / OUTIN / LIN, LOUT 00523200
DATA RLINE, SPACE, STSIGN, XSIGN / 1H-, 1H, 1H*, 1HX / 00523300
C 00523400
IF( IFIRST .GE. 301 ) GO TO 45 00523500
DO 40 I = 1, 41 00523600
IF( ( I .EQ. 1 ) .OR. ( I .EQ. 11 ) .OR. ( I .EQ. 41 ) ) 00523700
* GO TO 20 00523800
DO 10 J = 1, 60 00523900
CHART(I, J) = SPACE 00524000
10 CONTINUE 00524100
GO TO 40 00524200
20 DO 30 J = 1, 60 00524300
CHART(I, J) = RLINE 00524400
30 CONTINUE 00524500
40 CONTINUE 00524600
45 CONTINUE 00524700
C 00524800
DO 50 I = IFIRST, ILAST 00524900
C 00525000
IHIGHT = 30.0 * QFORM(I) / CHISQR + 0.5 00525100
IHIGHT = 41 - IHIGHT 00525200
IF( IHIGHT .LE. 0 ) IHIGHT = 1 00525300
IF( IHIGHT .GT. 41 ) IHIGHT = 41 00525400
C 00525500
ISIDE = 1 - IFIRST + 1 00525600
IF( ISIDE .LE. 0 ) ISIDE = 1 00525700
IF( ISIDE .GT. 60 ) ISIDE = 60 00525800
C 00525900
CHART(IHIGHT, ISIDE) = STSIGN 00526000
IF( QFORM(I) .LE. CHISQR ) GO TO 50 00526100
CHART(IHIGHT, ISIDE) = XSIGN 00526200
IDELET(ICOUNT) = 1 00526300
ICOUNT = ICOUNT + 1 00526400
50 CONTINUE 00526500
C 00526600
IF( IFIRST .GE. 301 ) RETURN 00526700
C 00526800
POINT0 = 0.0 00526900
POINT1 = CHISQR / 3.0 00527000
POINT2 = 2.0 * CHISQR / 3.0 00527100
POINT4 = 4.0 * CHISQR / 3.0 00527200
WRITE(LOUT, 6000) 00527300
C 00527400
DO 110 I = 1, 41 00527500
WRITE(LOUT, 6100) ( CHART(I, J) , J = 1, 60 ) 00527600
IF( I .EQ. 1 ) GO TO 60 00527700
IF( I .EQ. 11 ) GO TO 70 00527800
IF( I .EQ. 21 ) GO TO 80 00527900
IF( I .EQ. 31 ) GO TO 90 00528000
IF( I .EQ. 41 ) GO TO 100 00528100
WRITE(LOUT, 6200) 00528200
GO TO 110 00528300
60 WRITE(LOUT, 6300) POINT4 00528400
GO TO 110 00528500
70 WRITE(LOUT, 6300) CHISQR 00528600
GO TO 110 00528700
80 WRITE(LOUT, 6300) POINT2 00528800
GO TO 110 00528900
90 WRITE(LOUT, 6300) POINT1 00529000
GO TO 110 00529100
100 WRITE(LOUT, 6300) POINT0 00529200
110 CONTINUE 00529300
JFIRST = IFIRST - 1 00529400
WRITE(LOUT, 6400) ( J, J = JFIRST, ILAST, 5 ) 00529500
C 00529600
6000 FORMAT(1H1, /// 1H, 29X, 00529700
* 5H ---, 40H MULTI - CHARACTERISTIC , 00529800
* 4X, 33HC ONTROL CHART --- /// ) 00529900
6100 FORMAT(1H, 10X, 60( 1X, A1 ) ) 00530000
6200 FORMAT(1H+, 9X, 2H 1, 120X, 1H1 ) 00530100
6300 FORMAT(1H+, F9.2, 2H +, 120X, 1H+ ) 00530200
6400 FORMAT(1H, 8H SUBGROUP / 1H, 7H NUMBER, 1X, 13, 12( 7X, 13 ) ) 00530300
RETURN 00530400
END 00530500

```

	SUBROUTINE CALGFM(IROW, ICOLM, RMAT, X, XMEAN, WRKMAT, QFORM)	00530600
C		00530700
C	CALCULATION OF THE QUADRATIC FORM	00530800
C		00530900
	DIMENSION RMAT(ICOLM, ICOLM), X(IROW, ICOLM), XMEAN(ICOLM),	00531000
	* QFORM(IROW), WRKMAT(ICOLM)	00531100
C		00531200
	DO 20 I = 1, IROW	00531300
	QFORM(I) = 0.0	00531400
	DO 10 J = 1, ICOLM	00531500
	WRKMAT(J) = X(I, J) - XMEAN(J)	00531600
10	CONTINUE	00531700
C		00531800
	DO 20 J = 1, ICOLM	00531900
	DO 20 K = 1, ICOLM	00532000
	QFORM(I) = QFORM(I) + WRKMAT(J) * RMAT(J, K) * WRKMAT(K)	00532100
20	CONTINUE	00532200
	RETURN	00532300
	END	00532400
	SUBROUTINE PLOT(IROW, ICOLM, X, CHISQR, VAR1, VAR2, ICOUNT,	00532500
	* IDELET, XMEAN, WRKMAT, COVINV)	00532600
C		00532700
C	EXHIBIT BY BIVARIATES CONTROL CHART	00532800
C		00532900
	DIMENSION X(IROW, ICOLM), IDELET(ICOUNT), XMEAN(ICOLM),	00533000
	* CHART(51, 101), WRKMAT(ICOLM), POINT(2),	00533100
	* COVINV(ICOLM, ICOLM), SNUMB(31)	00533200
	COMMON / OUTIN / LIN, LOU	00533300
	COMMON / COSCL / SCALE, SCALEX, SCALEY, ORSCAL	00533400
	COMMON / COOPT / ILIOPT, INROPT	00533500
	DATA SPACE, DOT SIN, STSIGN, XSIGN / 1H, 1H., 1H*, 1HX /	00533600
	DATA SNUMB / 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9,	00533700
	* 1HA, 1HB, 1HC, 1HD, 1HE, 1HF, 1HG, 1HH,	00533800
	* 1HI, 1HJ, 1HK, 1HL, 1HM, 1HN, 1HO, 1HP,	00533900
	* 1HQ, 1HR, 1HS, 1HT, 1HU, 1HV, 1HW /	00534000
C		00534100
	SD1 = SQRT(VAR1)	00534200
	SD2 = SQRT(VAR2)	00534300
	SDSCL1 = SD1 * SCALE	00534400
	SDSCL2 = SD2 * SCALE	00534500
	POINT(1) = XMEAN(1) - SDSCL1	00534600
	XAXISF = POINT(1)	00534700
	XAXISL = XMEAN(1) + SDSCL1	00534800
	YAXISF = XMEAN(2) - SDSCL2	00534900
	POINT(2) = YAXISF	00535000
	YAXISL = XMEAN(2) + SDSCL2	00535100
C		00535200
C	153.0 = 51.0 * 3.0 (FOR THREE SIGMA)	00535300
C	78.0 = 26.0 * 3.0 (FOR THREE SIGMA)	00535400
C		00535500
	IFI = 51.5 - 153.0 * SD1 / SDSCL1	00535600
	ILA = 51.5 + 153.0 * SD1 / SDSCL1	00535700
	JFI = 26.5 - 78.0 * SD2 / SDSCL2	00535800
	JLA = 26.5 + 78.0 * SD2 / SDSCL2	00535900
	SCALEX = SDSCL1 / 100.0	00536000
	SCALEY = SDSCL2 / 50.0	00536100
	SCLERR = 0.1 * ORSCAL	00536200
C		00536300
	DO 20 I = 1, 101	00536400
	DO 10 J = 1, 51	00536500
	CHART(J, I) = SPACE	00536600
	IF((I.EQ. 51) .OR. (J.EQ. 26)) CHART(J, I) = DOT SIN	00536700
	IF(ILIOPT.EQ. 0) GO TO 10	00536800
	IF((I.EQ. IFI) .OR. (I.EQ. ILA) .OR. (J.EQ. JFI) .OR.	00536900
	* (J.EQ. JLA)) CHART(J, I) = DOT SIN	00537000
10	CONTINUE	00537100
20	CONTINUE	00537200

C	SCLERR = DRSCAL * SQRT((SCALEX) **2 + (SCALEY) ** 2) / 2.0	00537300
C		00537400
C		00537500
C		00537600
C	DETERMENANTE OF ELLIPSE POINT	00537700
C		00537800
	30 CALL CALQFM(1, ICOLM, COVINV, POINT, XMEAN, WRKMAT, QFORM)	00537900
	QF = ABS(QFORM - CHISQR)	00538000
	IF(QF .GT. SCLERR) GO TO 40	00538100
	IXAXIS = 51.0 * (POINT(1) - XMEAN(1)) / SDSCL1 + 51.5	00538200
	IYAXIS = 26.0 * (POINT(2) - XMEAN(2)) / SDSCL2 + 26.5	00538300
	IYAXIS = 52 - IYAXIS	00538400
	IF((IXAXIS .LE. 0) .OR. (IXAXIS .GT. 101) .OR.	00538500
	* (IYAXIS .LE. 0) .OR. (IYAXIS .GT. 51)) GO TO 40	00538600
	CHART(IYAXIS, IXAXIS) = DOTSSIN	00538700
	40 POINT(2) = POINT(2) + SCALEY	00538800
	IF(POINT(2) .LE. YAXISL) GO TO 30	00538900
	POINT(2) = YAXISF	00539000
	POINT(1) = POINT(1) + SCALEX	00539100
	IF(POINT(1) .LE. XAXISL) GO TO 30	00539200
C		00539300
	J = 1	00539400
C		00539500
	DO 50 I = 1, IROW	00539600
	IXAXIS = 51.0 * (X(I, 1) - XMEAN(1)) / SDSCL1 + 51.5	00539700
	IYAXIS = 26.0 * (X(I, 2) - XMEAN(2)) / SDSCL2 + 26.5	00539800
	IYAXIS = 52 - IYAXIS	00539900
	IF(IXAXIS .LE. 0) IXAXIS = 1	00540000
	IF(IXAXIS .GT. 101) IXAXIS = 101	00540100
	IF(IYAXIS .LE. 0) IYAXIS = 1	00540200
	IF(IYAXIS .GT. 51) IYAXIS = 51	00540300
	IF(CHART(IYAXIS, IXAXIS) .EQ. SPACE) GO TO 43	00540400
	IF(CHART(IYAXIS, IXAXIS) .EQ. XSIGN) GO TO 44	00540500
	IF(CHART(IYAXIS, IXAXIS) .EQ. STSIGN) GO TO 42	00540600
	DO 41 I2 = 1, 30	00540700
	IF(CHART(IYAXIS, IXAXIS) .NE. SNUMB(I2)) GO TO 41	00540800
	CHART(IYAXIS, IXAXIS) = SNUMB(I2+1)	00540900
	GO TO 50	00541000
	41 CONTINUE	00541100
	GO TO 50	00541200
	42 CONTINUE	00541300
	CHART(IYAXIS, IXAXIS) = SNUMB(1)	00541400
	GO TO 50	00541500
	43 CONTINUE	00541600
	CHART(IYAXIS, IXAXIS) = STSIGN	00541700
	IF(J .GT. ICOUNT) GO TO 50	00541800
	IF(I .NE. IDELET(J)) GO TO 50	00541900
	CHART(IYAXIS, IXAXIS) = XSIGN	00542000
	44 CONTINUE	00542100
	J = J + 1	00542200
	50 CONTINUE	00542300
C		00542400
	WRITE(LOUT, 6000)	00542500
	SD1 = 3.0 * SD1	00542600
	SD2 = 3.0 * SD2	00542700
	IF(ILIOPT .EQ. 0) GO TO 70	00542800
	YAXIS1 = XMEAN(2) - SD2	00542900
	YAXIS2 = XMEAN(2) + SD2	00543000
	WRITE(LOUT, 6100)	00543100

C	DO 60 I = 1, 51	00543200
	WRITE(LOUT, 6300) (CHART(I, J), J = 1, 101)	00543300
	IF(I .EQ. 1) WRITE(6, 6400) YAXISL	00543400
	IF(I .EQ. JF1) WRITE(6, 6400) YAXIS2	00543500
	IF(I .EQ. 26) WRITE(6, 6400) XMEAN(2)	00543600
	IF(I .EQ. JLA) WRITE(6, 6400) YAXIS1	00543700
	IF(I .EQ. 51) WRITE(6, 6400) YAXISF	00543800
	60 CONTINUE	00543900
C	GO TO 90	00544000
		00544100
C	70 WRITE(LOUT, 6100)	00544200
	DO 80 I = 1, 51	00544300
	WRITE(LOUT, 6300) (CHART(I, J), J = 1, 101)	00544400
	IF(I .EQ. 1) WRITE(LOUT, 6400) YAXISL	00544500
	IF(I .EQ. 26) WRITE(LOUT, 6400) XMEAN(2)	00544600
	IF(I .EQ. 51) WRITE(LOUT, 6400) YAXISF	00544700
	80 CONTINUE	00544800
		00544900
C	90 WRITE(LOUT, 6100)	00545000
	WRITE(LOUT, 6500) XAXISF, XMEAN(1), XAXISL	00545100
	WRITE(LOUT, 6200) XMEAN(1), SD1, XMEAN(2), SD2	00545200
C	JCOUNT = 0	00545300
	DO 100 I = 1, 51	00545400
	DO 100 J = 1, 101	00545500
	IF((CHART(I, J) .EQ. STSIGN) .OR. (CHART(I, J) .EQ. XSIGN))	00545600
	* JCOUNT = JCOUNT + 1	00545700
	100 CONTINUE	00545800
	JUDGE = IROW / 2	00545900
	IF(JUDGE .GT. JCOUNT) WRITE(LOUT, 6600)	00546000
C	6000 FORMAT(1H1, / 1H, 32X, 5H --- ,	00546100
	* 48H B I C H A R A C T E R I S T I C C O N T R O L	00546200
	* , 18H C H A R T ---)	00546300
	6100 FORMAT(1H, 15X, 2H+, 49(1H-), 1H+, 49(1H-), 2H+.)	00546400
	6200 FORMAT(/ 38H --- C O N T R O L L I M I T --- /	00546500
	* 1H, 21HX1 (ACROSS) ; MEAN = , G15.7, 10X,	00546600
	* 13H3.0 # S.D. = , G15.7 /	00546700
	* 1H, 21HX2 (DOWN) ; MEAN = , G15.7, 10X,	00546800
	* 13H3.0 # S.D. = , G15.7)	00546900
	6300 FORMAT(1H, 15X, 1H1, 101A1, 1H1)	00547000
	6400 FORMAT(1H+, G15.7, 1H-, 101X, 1H-)	00547100
	6500 FORMAT(1H, 9X, G10.3, 41X, G10.3, 41X, G10.3)	00547200
	6600 FORMAT(/ 1H, 41H--- YOU THINK TWICE YOUR DATA OR SCALE	00547300
	* , 13HCOMMAND ---)	00547400
	RETURN	00547500
	END	00547600
		00547700
		00547800
		00547900
		00548000

	SUBROUTINE MCCNCH(IROW, ICOLM, X, CHISQR, XMEAN, COV, COVINV,	00548100
	* IPIVOT, WRKMAT, QFORM, IDELET, XX, MQ, MA, MX,	00548200
	* M1, INDEV)	00548300
C		00548400
C	THIS SUBROUTINE CALCULATES AND PRINTS THE MULTI-CHARACTERISTIC	00548500
C		00548600
C	CONTROL CHART.	00548700
C		00548800
	COMMON / OUTIN / LIN, LOUT	00548900
	DIMENSION X(IROW, ICOLM), XMEAN(ICOLM), COV(ICOLM, ICOLM),	00549000
	* COVINV(ICOLM, ICOLM), IPIVOT(ICOLM), WRKMAT(ICOLM),	00549100
	* QFORM(IROW), IDELET(IROW)	00549200
	DIMENSION XX(MA), MQ(MX)	00549300
	COMMON / COOPT / ILIOPT, INROPT	00549400
C		00549500
	CALL INPUT(IROW, ICOLM, X, XX, MQ, MA, MX, M1, INDEV)	00549600
C		00549700
C	PRINT OF THE DATA	00549800
C		00549900
	CALL DATOUT(IROW, ICOLM, X)	00550000
C		00550100
C	CALCULATION OF MEANS	00550200
C		00550300
	CALL MEAN1(IROW, ICOLM, X, XMEAN)	00550400
	WRITE(LOUT, 6000)	00550500
	WRITE(LOUT, 6100)	00550600
	CALL MATOUT(1, ICOLM, XMEAN, 0.0)	00550700
C		00550800
C	CALCULATION OF THE VARIANCE-COVARIANCE MATRIX	00550900
C		00551000
	CALL COVMAT(IROW, ICOLM, X, XMEAN, COV)	00551100
	VAR1 = COV(1, 1)	00551200
	VAR2 = COV(2, 2)	00551300
	WRITE(LOUT, 6200)	00551400
	CALL MATOUT(ICOLM, ICOLM, COV, 1.0)	00551500
C		00551600
C	CALCULATION OF THE INVERSE MATRIX OF VARIANCE-COVARIANCE MATRIX	00551700
C		00551800
	CALL INVMA1(ICOLM, COV, IPIVOT, COVINV, DET)	00551900
	WRITE(LOUT, 6300)	00552000
	CALL MATOUT(ICOLM, ICOLM, COVINV, 0.0)	00552100
	IF(DET .EQ. 0.0) RETURN	00552200
C		00552300
C	CALCULATION OF THE QUADRATIC FORM	00552400
C		00552500
	CALL CALQFM(IROW, ICOLM, COVINV, X, XMEAN, WRKMAT, QFORM)	00552600
	IF(INROPT .EQ. 0) GO TO 10	00552700
C		00552800
C	PRINT THE QUADRATIC FORM	00552900
C		00553000
	MROW = IROW	00553100
	IF(IROW .GE. 300)	00553200
	WRITE(LOUT, 6400)	00553300
	CALL MATOUT(1, MROW, QFORM, 0.0)	00553400
	IF(IROW .GE. 300)	00553500
10	ICOUNT = 1	00553600
	MROW = (IROW - 1) / 60 + 1	00553700
	MROW1 = MROW	00553800
	IF(MROW .GE. 5)	00553900
C		00554000
	DO 20 I = 1, MROW	00554100
	ILAST = 60 * I	00554200
	IFIRST = ILAST - 59	00554300
	IF(ILAST .GT. IROW) ILAST = IROW	00554400
C		00554500
C	ILLUSTRATION OF THE MULTI-CHARACTERISTIC CONTROL CHART	00554600
C		00554700
	CALL CONTCH(IROW, QFORM, CHISQR, ICOUNT, IDELET, IFIRST, ILAST)	00554800
20	CONTINUE	00554900
	IF(MROW .NE. MROW1)	00555000
	ICOUNT = ICOUNT - 1	00555100
	IF(ICOLM .NE. 2) GO TO 30	00555200

```

C      ILLUSTRATION OF THE BICHARACTERISTIC CONTROL CHART      00555300
C      CALL PLOT(IROW, ICOLM, X, CHISQR, VAR1, VAR2, ICOUNT, IDELET, 00555400
C      *      XMEAN, WRKMAT, COVINV)      00555500
      30 WRITE(LOUT, 6500) ( IDELET(I), I = 1, ICOUNT )      00555600
C      6000 FORMAT(1H0, 120( 1H. ) / )      00555700
      6100 FORMAT(1H0, 21HS A M P L E - M E A N )      00555800
      6200 FORMAT(1H0, 51HS A M P L E - V A R I A N C E - C O V A R I A N C E      00555900
      *      , 14H M A T R I X )      00556000
      6300 FORMAT(1H0, 51H I N V E R S E M A T R I X O F V A R I A N C E      00556100
      *      , 36H - C O V A R I A N C E M A T R I X )      00556200
      6400 FORMAT(1H0, 51H S A M P L E O F T H E V A L U E O F T H      00556300
      *      , 31H E Q U A D R A T I C F O R M )      00556400
      6500 FORMAT(1H0, 41H D E L E T E D S A M P L E N U M B E R / ( 1H ,      00556500
      *      2016 ) )      00556600
      6600 FORMAT(1H0, 47H --- C O N T I N U E D M O R E T H A N ,      00556700
      *      16H3 0 0 . --- )      00556800
      RETURN      00556900
      END      00557000

      SUBROUTINE INPUT(IROW, ICOLM, X, XX, MQ, MA, MX, M1, INDEV )      00557100
      COMMON / OUTIN / LIN, LOU      00557200
      DIMENSION X(IROW, ICOLM)      00557300
      DIMENSION XX(MA), MQ(MX)      00557400
C      READ INPUT DATA      00557500
C      REWIND INDEV      00557600
      DO 10 I=1, IROW      00557700
C      READ(INDEV, END=99, ERR=777) KK, (XX(J), J=1, M1)      00557800
      DO 10 J=1, ICOLM      00557900
      K=MQ(J)      00558000
      X(I, J)=XX(K)      00558100
      10 CONTINUE      00558200
C      99 CONTINUE      00558300
      RETURN      00558400
      777 WRITE(LOUT, 900)      00558500
      900 FORMAT(1H1, ///1X, '--- IT IS IMPOSSIBLE TO FIND DATA ',      00558600
      *      'SPECIFIED BY COMMAND ---' )      00558700
      KK=KK      00558800
      RETURN      00558900
      END      00559000

      SUBROUTINE WRKARA(IROW, ICOLM, CHISQR, XX, MQ, MA, MX,      00559100
      *      WORK, IWORK, M1, INDEV, LIMIT1, LIMIT2 )      00559200
      COMMON / OUTIN / LIN, LOU      00559300
      DIMENSION WORK( LIMIT1 ), IWORK( LIMIT2 )      00559400
      DIMENSION XX(MA), MQ(MX)      00559500
C      M1 = IROW * ICOLM      00559600
      M2 = ICOLM * ICOLM      00559700
C      NUM1 = 1      00560000
      NUM2 = NUM1 + M1      00560100
      NUM3 = NUM2 + ICOLM      00560200
      NUM4 = NUM3 + M2      00560300
      NUM5 = NUM4 + M2      00560400
      NUM6 = NUM5 + ICOLM      00560500
      NUM7 = NUM6 + IROW - 1      00560600
C      NUMB1 = 1      00560700
      NUMB2 = NUM1 + ICOLM      00560800
      NUMB3 = NUMB2 + IROW - 1      00560900
C      IF( ( NUM7 .GT. LIMIT1 ) .OR. ( NUMB3 .GT. LIMIT2 ) ) GO TO 10      00561000
      CALL MCCNCH(IROW, ICOLM, WORK(NUM1), CHISQR, WORK(NUM2),      00561100
      *      WORK(NUM3), WORK(NUM4), IWORK(NUMB1), WORK(NUM5),      00561200
      *      WORK(NUM6), IWORK(NUMB2), XX, MQ, MA, MX,      00561300
      *      M1, INDEV )      00561400
      RETURN      00561500
C      10 WRITE(LOUT, 6000) NUM7, NUMB3      00561600
      6000 FORMAT( 1H0, 28HNEEDING WORK AREA SIZE = ( , 17, 2H ) /      00561700
      *      1H , 28HNEEDING IWORK AREA SIZE = ( , 17, 2H ) )      00561800
      RETURN      00561900
      END      00562000

```

	SUBROUTINE CLEAR(WORK, IWORK, LIMIT1, LIMIT2)	00563100
	DIMENSION WORK(LIMIT1), IWORK(LIMIT2)	00563200
C	DO 10 I = 1, LIMIT1	00563300
	WORK(I) = 0.0	00563400
10	CONTINUE	00563500
	DO 20 I = 1, LIMIT2	00563600
	IWORK(I) = 0	00563700
20	CONTINUE	00563800
	RETURN	00563900
	END	00564000
		00564100
	SUBROUTINE DATOUT(IROW, ICOLM, X)	00564200
C	PRINT OF THE DATA X	00564300
C		00564400
C	DIMENSION X(IROW, ICOLM)	00564500
	COMMON / OUTIN / LIN, LOUT	00564600
C		00564700
	IROW = 1	00564800
	NCOLM = (ICOLM - 1) / 5 + 1	00564900
	WRITE(LOUT, 6000)	00565000
C		00565100
	DO 20 I1 = 1, NCOLM	00565200
	ILAST = 5 * I1	00565300
	IFIRST = ILAST - 4	00565400
	IF(ILAST .GT. ICOLM) ILAST = ICOLM	00565500
	WRITE(LOUT, 6200) (K, K = IFIRST, ILAST)	00565600
	ICOUNT = ILAST - IFIRST + 1	00565700
	GO TO (1, 2, 3, 4, 5), ICOUNT	00565800
1	WRITE(LOUT, 6210)	00565900
	GO TO 6	00566000
2	WRITE(LOUT, 6220)	00566100
	GO TO 6	00566200
3	WRITE(LOUT, 6230)	00566300
	GO TO 6	00566400
4	WRITE(LOUT, 6240)	00566500
	GO TO 6	00566600
5	WRITE(LOUT, 6250)	00566700
C		00566800
	DO 10 I2 = IROW, IROW	00566900
	WRITE(LOUT, 6300) I2, (X(I2, J), J = IFIRST, ILAST)	00567000
	IF(I2 .GE. 50) GO TO 15	00567100
10	CONTINUE	00567200
15	WRITE(LOUT, 6100)	00567300
20	CONTINUE	00567400
	IF(IROW .GT. 50) GO TO 30	00567500
	WRITE(LOUT, 6400)	00567600
	RETURN	00567700
30	WRITE(LOUT, 6100)	00567800
	WRITE(LOUT, 6500)	00567900
C		00568000
	6000 FORMAT(1H1, / 1H, 3X, 29H--- DATA LIST --- / 1H,	00568100
	* 120(1H=))	00568200
6100	FORMAT(1H0, 120(1H=) /)	00568300
6200	FORMAT(1H0, 5(18X, 12))	00568400
6210	FORMAT(1H+, 17X, 2HX /)	00568500
6220	FORMAT(1H+, 7X, 2(10X, 2HX, 8X) /)	00568600
6230	FORMAT(1H+, 7X, 3(10X, 2HX, 8X) /)	00568700
6240	FORMAT(1H+, 7X, 4(10X, 2HX, 8X) /)	00568800
6250	FORMAT(1H+, 7X, 5(10X, 2HX, 8X) /)	00568900
6300	FORMAT(1H, 13, 3X, 5F20.7)	00569000
6400	FORMAT(1H, 3X, 37H--- DATA LIST END ---)	00569100
6500	FORMAT(1H, 3X, 46H(CONTINUED MORE THAN 50,	00569200
	* 3H))	00569300
	RETURN	00569400
	END	00569500
		00569600

	SUBROUTINE BRKDUL(WORK, IWORK, LIMIT1, LIMIT2, AA, XX, MQ,	00569700
	* N, MQ1, M1, MA, MX, INDEV, II)	00569800
C		00569900
C	EXECUTION OF THE BREAK-DOWN PRCDURE	00570000
C		00570100
	COMMON NUMDT(20)	00570200
	COMMON / TAPE / NTAPE	00570300
	COMMON / OUTIN / LIN, LOUT	00570400
	INTEGER AA(100,80)	00570500
	DIMENSION WORK(LIMIT1), IWORK(LIMIT2), XX(MA), MQ(MX)	00570600
C		00570700
	NTAPE = INDEV	00570800
	NVAR = MQ1 - 1	00570900
C		00571000
	WRITE(LOUT,6000)	00571100
	KNUM=NUMDT(INDEV)	00571200
	IF(KNUM.EQ. 0) GO TO 1000	00571300
C		00571400
	CALL CLEAR(WORK, IWORK, LIMIT1, LIMIT2)	00571500
	CALL BDOWN1(N, NVAR, NSAMP, WORK(1), WORK(NVAR+1), IWORK,	00571600
	* XX, MQ, M1, MA, MX)	00571700
	NSAMP = NSAMP	00571800
	MAXCTG = IWORK(1)	00571900
	IF(NVAR .GT. 1) GO TO 10	00572000
	ILAST = MAXCTG	00572100
	NTAB = MAXCTG	00572200
	GO TO 30	00572300
10	CONTINUE	00572400
	NTAB = 1	00572500
	DO 20 I = 1, NVAR	00572600
	J = NVAR - I + 1	00572700
	IF(IWORK(J) .GT. MAXCTG) MAXCTG = IWORK(J)	00572800
	NTAB = NTAB * IWORK(J) + 1	00572900
20	CONTINUE	00573000
	NTAB = NTAB - 1	00573100
	ILAST = NVAR * MAXCTG	00573200
30	CONTINUE	00573300
C		00573400
	IF(NTAB .LE. 0) GO TO 110	00573500
	NUM1 = 1	00573600
	NUM2 = NUM1 + NVAR	00573700
	NUM3 = NUM2 + ILAST	00573800
	NUM4 = NUM3 + NTAB	00573900
	NUM5 = NUM4 + NTAB	00574000
	NUM6 = NUM5 + NTAB	00574100
	NUM7 = NUM6 + NTAB	00574200
	NUM8 = NUM7 + NTAB	00574300
	INUM1 = 1	00574400
	INUM2 = INUM1 + NVAR	00574500
	INUM3 = INUM2 + NVAR	00574600
	INUM4 = INUM3 + NTAB	00574700
	IF((NUM8 .GT. LIMIT1) .OR. (INUM3 .GT. LIMIT2)) GO TO 100	00574800
	CALL BDOWN2(NSAMP, NVAR, NTAB, XX, MQ, M1, MA, MX,	00574900
	* WORK(NUM1), WORK(NUM2), WORK(NUM3), WORK(NUM4),	00575000
	* WORK(NUM5), WORK(NUM6), WORK(NUM7),	00575100
	* IWORK(INUM1), IWORK(INUM2), IWORK(INUM3))	00575200
	CALL BDOWN3(NVAR, NTAB, WORK(NUM2), WORK(NUM3), WORK(NUM4),	00575300
	* WORK(NUM5), WORK(NUM6), WORK(NUM7),	00575400
	* IWORK(INUM1), IWORK(INUM3), AA, II, NSAMP)	00575500
	RETURN	00575600
C		00575700
100	CONTINUE	00575800
	WRITE(LOUT, 6100) NUM8, INUM4	00575900
	RETURN	00576000
110	CONTINUE	00576100
	WRITE(LOUT, 6150)	00576200
	RETURN	00576300
1000	WRITE(LOUT,6001) INDEV	00576400
	RETURN	00576500
6001	FORMAT(///1H , '--- FILE NUMBER IS',I3,'. THIS FILE DOES NOT',	00576600
	* ' INCLUDE DATA ---')	00576700
6000	FORMAT(1H1,///1X,'+-----+/'	00576800
	* 1X,'I BREAKDOWN I'/'	00576900
	* 1X,'I PROCEDURE I'/'	00577000
	* 1X,'+-----+/')	00577100
6100	FORMAT(1H , 'USED WORKING AREA WORK(1) ~ WORK(' ,I7,') .'/'	00577200
	* 1H , 'USED WORKING AREA IWORK(1) ~ IWORK(' ,I7,') .')	00577300
6150	FORMAT(1H , 'YOU MUST CHECK YOUR DATA . (NTAB = 0) ')	00577400
	END	00577500

	SUBROUTINE BDOWN1(NSAMP, NVAR, NSAM, X, CATEGR, ISIZE,	00577600
	* XX, MQ, M1, MA, MX)	00577700
	COMMON / OUTIN / LIN, LOUT	00577800
	COMMON / TAPE / NTAPE	00577900
	DIMENSION X(NVAR), CATEGR(1), ISIZE(NVAR), XX(MA), MQ(MX)	00578000
C	NSAM = 0	00578100
	REWIND NTAPE	00578200
	READ(NTAPE,END=5,ERR=777) KK,(XX(J),J=1,M1)	00578300
	NSAM = NSAM + 1	00578400
	DO 4 J = 1, NVAR	00578500
	K = MQ(J+1)	00578600
	CATEGR(J) = XX(K)	00578700
	4 CONTINUE	00578800
	5 CONTINUE	00578900
		00579000
C	DO 10 I1=1, NVAR	00579100
	ISIZE(I1) = 1	00579200
10	CONTINUE	00579300
		00579400
C	DO 40 I1 = 2, NSAMP	00579500
	READ(NTAPE,END=45,ERR=777) KK,(XX(J),J=1,M1)	00579600
	NSAM = NSAM + 1	00579700
	DO 15 J = 1, NVAR	00579800
	K = MQ(J+1)	00579900
	X(J) = XX(K)	00580000
15	CONTINUE	00580100
	DO 30 I2=1, NVAR	00580200
	NSIZE = ISIZE(I2)	00580300
	DO 20 I3=1, NSIZE	00580400
	I4 = NVAR * (I3 - 1) + I2	00580500
	IF (X(I2) .NE. CATEGR(I4)) GO TO 20	00580600
	GO TO 30	00580700
20	CONTINUE	00580800
	N1 = NVAR * NSIZE + I2	00580900
	CATEGR(N1) = X(I2)	00581000
	ISIZE(I2) = ISIZE(I2) + 1	00581100
30	CONTINUE	00581200
40	CONTINUE	00581300
45	CONTINUE	00581400
		00581500
C	DO 70 I1 = 1, NVAR	00581600
	NSIZE = ISIZE(I1)	00581700
	IF(NSIZE .LE. 1) GO TO 70	00581800
	NSIZE1 = NSIZE - 1	00581900
	DO 60 I2 = 1, NSIZE1	00582000
	IS = I2 + 1	00582100
	J1 = NVAR * (I2 - 1) + I1	00582200
	DO 50 I3 = IS, NSIZE	00582300
	J2 = NVAR * (I3 - 1) + I1	00582400
	IF(CATEGR(J1) .LT. CATEGR(J2)) GO TO 50	00582500
	TRANS = CATEGR(J1)	00582600
	CATEGR(J1) = CATEGR(J2)	00582700
	CATEGR(J2) = TRANS	00582800
50	CONTINUE	00582900
60	CONTINUE	00583000
70	CONTINUE	00583100
	KK = KK	00583200
	RETURN	00583300
		00583400
C		00583500
	777 WRITE(LOUT, 900)	00583600
	900 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00583700
	* 'SPECIFIED BY COMMAND ---')	00583800
	RETURN	00583900
	END	00584000

	SUBROUTINE BDOWN2(NSAMP, NVAR, NTAB, XX, MQ, M1, MA, MX,	00584100
*	X, CATEGR, RMEAN, SD, CV, RMAX, RMIN,	00584200
*	KSIZE, IW, KSAMP	00584300
	COMMON /TAPE/ NTAPE	00584400
	COMMON /OUTIN/ LIN, LOUT	00584500
	DIMENSION X(NVAR), CATEGR(NVAR, 1)	00584600
	DIMENSION RMEAN(NTAB), SD(NTAB), CV(NTAB)	00584700
	DIMENSION RMAX(NTAB), RMIN(NTAB)	00584800
	DIMENSION KSIZE(NVAR), IW(NVAR)	00584900
	DIMENSION KSAMP(NTAB), XX(MA), MQ(MX)	00585000
C		00585100
	DO 100 I1 = 1, NVAR	00585200
	REWIND NTAPE	00585300
	DO 90 I2 = 1, NSAMP	00585400
	READ(NTAPE,END=95,ERR=777) KK,(XX(J),J=1,M1)	00585500
	K = MQ(1)	00585600
	Y = XX(K)	00585700
	DO 5 J = 1, NVAR	00585800
	K = MQ(J+1)	00585900
	X(J) = XX(K)	00586000
5	CONTINUE	00586100
	DO 20 I3 = 1, I1	00586200
	NSIZE = KSIZE(I3)	00586300
	DO 10 I4 = 1, NSIZE	00586400
	IF(X(I3) .NE. CATEGR(I3, I4))	00586500
	IW(I3) = I4	00586600
		00586700
	GO TO 20	00586800
10	CONTINUE	00586900
	WRITE(LOUT, 6100)	00587000
	RETURN	00587100
20	CONTINUE	00587200
	J1 = IW(1)	00587300
	IL = I1 - 1	00587400
	IF(IL .LE. 0)	00587500
	JTAB = 1	00587600
	DO 30 I3 = 1, IL	00587700
	J = IL - I3 + 1	00587800
	JTAB = JTAB * KSIZE(J) + 1	00587900
30	CONTINUE	00588000
	JTAB = JTAB - 1	00588100
	KCYCL = 1	00588200
40	CONTINUE	00588300
	KTAB = 1	00588400
	KLAST = I1 - KCYCL	00588500
	DO 50 I3 = 1, KLAST	00588600
	J = I1 - I3 + 1	00588700
	KTAB = KTAB * KSIZE(J)	00588800
50	CONTINUE	00588900
	KTAB = KTAB * (IW(KCYCL) - 1)	00589000
	JTAB = JTAB + KTAB	00589100
	KCYCL = KCYCL + 1	00589200
	IF(KCYCL .LT. I1)	00589300
	J1 = JTAB + IW(I1)	00589400
60	CONTINUE	00589500
	KSAMP(J1) = KSAMP(J1) + 1	00589600
	RMEAN(J1) = RMEAN(J1) + Y	00589700
	SD (J1) = SD (J1) + Y * # 2	00589800
	IF(KSAMP(J1) .LE. 1)	00589900
	IF(RMAX (J1) .LT. Y) RMAX(J1) = Y	00590000
	IF(RMIN (J1) .GT. Y) RMIN(J1) = Y	00590100
		00590200
70	CONTINUE	00590300
	RMAX(J1) = Y	00590400
	RMIN(J1) = Y	00590500
80	CONTINUE	00590600
90	CONTINUE	00590700
95	CONTINUE	00590800
100	CONTINUE	

C	DO 120 I1 = 1, NTAB	00590900
	N = KSAMP(I1)	00591000
	IF(N .LE. 0)	00591100
	FN = N	00591200
	RMEAN(I1) = RMEAN(I1)/ FN	00591300
	SD (I1) = SD (I1)/ FN	00591400
	SD (I1) = SD (I1) - RMEAN(I1) * * 2	00591500
	SD (I1) = SQRT(SD(I1))	00591600
	R = ABS(RMEAN(I1))	00591700
	IF(R .LE. 1.0E - 30)	00591800
	CV(I1) = SD(I1)/ RMEAN(I1)	00591900
	GO TO 110	00592000
	GO TO 120	00592100
110	CONTINUE	00592200
	CV(I1) = 1.0E30	00592300
120	CONTINUE	00592400
	KK = KK	00592500
	RETURN	00592600
777	WRITE(LOUT, 900)	00592700
	RETURN	00592800
900	FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00592900
*	'SPECIFIED BY COMMAND ---')	00593000
6100	FORMAT(1H , 'YOU MUST CHECK YOUR DATA.')	00593100
	END	00593200

```

      SUBROUTINE BDOWN3( NVAR, NTAB, CATEGR, RMEAN, SD, CV, RMAX,
*                      RMIN, KSIZE, KSAMP, AA, II, NSAMP )
C
C   ARRANGING AND EDITING THE STRATIFIED DATA.
C
      COMMON / OUTIN / LIN, LOUT
      DIMENSION CATEGR(NVAR, 1), RMEAN(NTAB), SD(NTAB), CV(NTAB),
*              KSIZE(NVAR), KSAMP(NTAB), RMAX(NTAB), RMIN(NTAB),
*              ITM(8), NUM(10)
      INTEGER AA(100,80), WITH(4), SLASH, BLANK, COMMA, FMAT(6)
      DATA WITH / 'W', 'I', 'T', 'H' /,
*            SLASH / '/' /, BLANK / ' ' /, COMMA / ',' /,
      DATA FMAT / '(1H ', '3X', ' ', ' ', ' ', 4H('-', ' '))' /,
*            NUM / '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' /
C
      K = 0
      KL = 0
      DO 14 I = 16, 80
      IF( KL .LE. 1 )          GO TO 11
      IF( AA(II,I) .NE. SLASH ) GO TO 14
      K3 = I - 1
      GO TO 16
11 IF( AA(II,I) .EQ. BLANK .OR. AA(II,I) .EQ. COMMA ) GO TO 12
      K = K + 1
      GO TO 14
12 IF( K .EQ. 0 )          GO TO 14
      K1 = I - K
      K2 = I - 1
      IF( KL .EQ. 0 ) WRITE( LOUT, 4100 ) ( AA(II,L), L = K1, K2 )
      KL = KL + 1
      K = 0
14 CONTINUE
C
16 CONTINUE
      WRITE( LOUT, 4200 ) ( AA(II,I), I = K1, K3 )
      K5 = K3 - K1 + 1
      IA = K5 + 23
      CALL FORM( NUM, 10, FMAT, IA, 3, 4, 6 )
      WRITE( LOUT, FMAT )
      WRITE( LOUT, 6001 )
C
      WRITE( LOUT, 5100 )
      WRITE( LOUT, 5200 )
      WRITE( LOUT, 5300 )
      WRITE( LOUT, 5200 )
C
      K = 0
      KCODE = 0
      DO 28 I = 16, 80
      IF( AA(II,I) .EQ. BLANK .OR. AA(II,I) .EQ. COMMA
*      .OR. AA(II,I) .EQ. SLASH ) GO TO 22
      K = K + 1
      GO TO 28
22 IF( K .EQ. 0 ) GO TO 28
      K1 = I - K
      K2 = I - 1
C
      J = 1
      DO 23 L = K1, K2
      IF( WITH(J) .NE. AA(II, L) ) GO TO 24
23 J = J + 1
      KCODE = 1
      K = 0
      GO TO 28
24 IF( KCODE .LE. 0 ) GO TO 27

```

C	DO 29 J = 1, 8	00599800
	29 ITM(J) = BLANK	00599900
	J = 1	00600000
	DO 26 L = K1, K2	00600100
	ITM(J) = AA(II,L)	00600200
	26 J = J + 1	00600300
	WRITE(LOUT, 5400) (ITM(J), J = 1, 8), KCODE	00600400
	KCODE = KCODE + 1	00600500
	27 K = 0	00600600
	28 CONTINUE	00600700
	WRITE(LOUT, 5200)	00600800
	WRITE(LOUT, 6001)	00600900
	WRITE(LOUT, 5500) NSAMP	00601000
	WRITE(LOUT, 6001)	00601100
	IL = II	00601200
C		00601300
	WRITE(LOUT, 6900)	00601400
	WRITE(LOUT, 6000)	00601500
	WRITE(LOUT, 6006) (I, I = 1, NVAR)	00601600
	IK = 0	00601700
	KSZ = KSIZE(1)	00601800
C		00601900
	DO 100 II = 1, NVAR	00602000
C		00602100
	WRITE(LOUT, 6900)	00602200
	GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), II	00602300
C		00602400
	1 KS1 = KSIZE(II)	00602500
	GO TO 15	00602600
	2 KS2 = KSIZE(II)	00602700
	GO TO 15	00602800
	3 KS3 = KSIZE(II)	00602900
	GO TO 15	00603000
	4 KS4 = KSIZE(II)	00603100
	GO TO 15	00603200
	5 KS5 = KSIZE(II)	00603300
	GO TO 15	00603400
	6 KS6 = KSIZE(II)	00603500
	GO TO 15	00603600
	7 KS7 = KSIZE(II)	00603700
	GO TO 15	00603800
	8 KS8 = KSIZE(II)	00603900
	GO TO 15	00604000
	9 KS9 = KSIZE(II)	00604100
	GO TO 15	00604200
	10 KS10 = KSIZE(II)	00604300
C		00604400
	15 CONTINUE	00604500
	IF(II .GE. 2) KSZ = KSZ * KSIZE(II)	00604600
	I1 = 1	00604700
	I2 = 1	00604800
	I3 = 1	00604900
	I4 = 1	00605000
	I5 = 1	00605100
	I6 = 1	00605200
	I7 = 1	00605300
	I8 = 1	00605400
	I9 = 1	00605500
	I10 = 1	00605600
	KN = 1	00605700
C		00605800
	IF(KSZ.GT.3000) GO TO 3000	00605900
C		00606000
	DO 90 JJ = 1, KSZ	00606100
C		00606200
	IK = IK + 1	00606300
C		00606400
	GO TO (65, 60, 55, 50, 45, 40, 35, 30, 25, 20), II	00606500
C		00606600
	20 CONTINUE	00606700
	IF(I10 .LE. KS10) GO TO 70	00606800
	I9 = I9 + 1	00606900
		00607000

	I10 = 1		00607100
25	CONTINUE		00607200
	IF(I9 .LE. KS9)	GO TO 70	00607300
	I8 = I8 + 1		00607400
	I9 = 1		00607500
30	CONTINUE		00607600
	IF(I8 .LE. KS8)	GO TO 70	00607700
	I7 = I7 + 1		00607800
	I8 = 1		00607900
35	CONTINUE		00608000
	IF(I7 .LE. KS7)	GO TO 70	00608100
	I6 = I6 + 1		00608200
	I7 = 1		00608300
40	CONTINUE		00608400
	IF(I6 .LE. KS6)	GO TO 70	00608500
	I5 = I5 + 1		00608600
	I6 = 1		00608700
45	CONTINUE		00608800
	IF(I5 .LE. KS5)	GO TO 70	00608900
	I4 = I4 + 1		00609000
	I5 = 1		00609100
50	CONTINUE		00609200
	IF(I4 .LE. KS4)	GO TO 70	00609300
	I3 = I3 + 1		00609400
	I4 = 1		00609500
55	CONTINUE		00609600
	IF(I3 .LE. KS3)	GO TO 70	00609700
	I2 = I2 + 1		00609800
	I3 = 1		00609900
60	CONTINUE		00610000
	IF(I2 .LE. KS2)	GO TO 70	00610100
	I1 = I1 + 1		00610200
	I2 = 1		00610300
65	CONTINUE		00610400
	IF(I1 .GT. KS1)	GO TO 100	00610500
C			00610600
70	CONTINUE		00610700
	LCTG = CATEGR(KN, I1)		00610800
	WRITE(LOUT, 6010)	LCTG	00610900
	IF(I1 .LE. 1)	I1 = I1 + 1	00611000
	KN = KN + 1		00611100
	IF(I1 .LE. 1)	GO TO 80	00611200
	LCTG = CATEGR(KN, I2)		00611300
	WRITE(LOUT, 6020)	LCTG	00611400
	IF(I1 .LE. 2)	I2 = I2 + 1	00611500
	KN = KN + 1		00611600
	IF(I1 .LE. 2)	GO TO 80	00611700
	LCTG = CATEGR(KN, I3)		00611800
	WRITE(LOUT, 6030)	LCTG	00611900
	IF(I1 .LE. 3)	I3 = I3 + 1	00612000
	KN = KN + 1		00612100
	IF(I1 .LE. 3)	GO TO 80	00612200
	LCTG = CATEGR(KN, I4)		00612300
	WRITE(LOUT, 6040)	LCTG	00612400
	IF(I1 .LE. 4)	I4 = I4 + 1	00612500
	KN = KN + 1		00612600
	IF(I1 .LE. 4)	GO TO 80	00612700
	LCTG = CATEGR(KN, I5)		00612800
	WRITE(LOUT, 6050)	LCTG	00612900
	IF(I1 .LE. 5)	I5 = I5 + 1	00613000
	KN = KN + 1		00613100
	IF(I1 .LE. 5)	GO TO 80	00613200
	LCTG = CATEGR(KN, I6)		00613300
	WRITE(LOUT, 6060)	LCTG	00613400
	IF(I1 .LE. 6)	I6 = I6 + 1	00613500
	KN = KN + 1		00613600
	IF(I1 .LE. 6)	GO TO 80	00613700
	LCTG = CATEGR(KN, I7)		00613800
	WRITE(LOUT, 6070)	LCTG	00613900
	IF(I1 .LE. 7)	I7 = I7 + 1	00614000
	KN = KN + 1		00614100

IF(II .LE. 7)	GO TO 80	00614200
LCTG = CATEGR(KN, I8)		00614300
WRITE(LOUT, 6080)	LCTG	00614400
IF(II .LE. 8)	I8 = I8 + 1	00614500
KN = KN + 1		00614600
IF(II .LE. 8)	GO TO 80	00614700
LCTG = CATEGR(KN, I9)		00614800
WRITE(LOUT, 6090)	LCTG	00614900
IF(II .LE. 9)	I9 = I9 + 1	00615000
KN = KN + 1		00615100
IF(II .LE. 9)	GO TO 80	00615200
LCTG = CATEGR(KN, I10)		00615300
WRITE(LOUT, 6100)	LCTG	00615400
IF(II .LE. 10)	I10 = I10 + 1	00615500
C		00615600
80 CONTINUE		00615700
IF(KN .GT. II)	KN = 1	00615800
IF(KSAMP(IK) .LE. 0)	GO TO 85	00615900
KSAP = KSAMP(IK)		00616000
PER = FLOAT(KSAP) / FLOAT(NSAMP) * 100.0		00616100
WRITE(LOUT, 6600)	KSAMP(IK), PER, RMEAN(IK),	00616200
*	SD(IK), CV(IK), RMAX(IK), RMIN(IK)	00616300
GO TO 90		00616400
85 CONTINUE		00616500
WRITE(LOUT, 6700)	KSAMP(IK)	00616600
C		00616700
90 CONTINUE		00616800
C		00616900
100 CONTINUE		00617000
C		00617100
WRITE(LOUT, 6900)		00617200
II = IL		00617300
C		00617400
RETURN		00617500
C		00617600
3000 WRITE(LOUT,6999)		00617700
II=IL		00617800
C		00617900
RETURN		00618000
C		00618100
4100 FORMAT(1H0, 3X, 8A1)		00618200
4200 FORMAT(1H+, 12X, 'IS BROKEN DOWN ', 60A1)		00618300
5100 FORMAT(1H0,3X,'----- LIST OF ITEMS -----'/)		00618400
5200 FORMAT(1H , 6X, 19('.'))		00618500
5300 FORMAT(1H , 8X, 'ITEM', 7X, 'CODE')		00618600
5400 FORMAT(1H , 8X, 8A1, 5X, I1)		00618700
5500 FORMAT(1H , 6X, 'NUMBER OF SAMPLES N =', I6)		00618800
6001 FORMAT(1H0)		00618900
6000 FORMAT(1H , 1X, 20HCOMBINATION OF ITEMS , 7X, 6HSAMPLE , 4X,		00619000
* 10HPERCENTILE , 21X, 9HSTANDARD , 2X, 14HCOEFFICIENT OF /		00619100
* 1H , 28X, 6H SIZE , 7X, 5H(%) , 11X,		00619200
* 4HMEAN , 8X, 9HDEVIATION, 2X, 14H VARIATION , 9X,		00619300
* 7HMAXIMUM , 9X, 7HMINIMUM)		00619400
6006 FORMAT(1H+, 1X, 10I3)		00619500
6010 FORMAT(1H , 1X, I3)		00619600
6020 FORMAT(1H+, 4X, I3)		00619700
6030 FORMAT(1H+, 7X, I3)		00619800
6040 FORMAT(1H+, 10X, I3)		00619900
6050 FORMAT(1H+, 13X, I3)		00620000
6060 FORMAT(1H+, 16X, I3)		00620100
6070 FORMAT(1H+, 19X, I3)		00620200
6080 FORMAT(1H+, 22X, I3)		00620300
6090 FORMAT(1H+, 25X, I3)		00620400
6100 FORMAT(1H+, 28X, I3)		00620500
6600 FORMAT(1H+, 29X, I5, 6X, F6.2, 5(2X, F14.4))		00620600
6700 FORMAT(1H+, 29X, I5, 8X, '----', 5(11X, 4(' - '), 1X))		00620700
6900 FORMAT(1H , 128('.'))		00620800
6999 FORMAT(///,5X,'--- COMBINATION OF CATEGORY IS GREATER',		00620900
*	' THAN 3000. ---',)	00621000
C		00621100
END		00621200

	SUBROUTINE DFIND(AA, II, MCON, MQ1, NN,	00621300
	* NUM, NUMB, COMMA, BLANK, SLASH)	00621400
C		00621500
C	THIS ROUTINE IS A DECODER FOR FIND DATA PROCEDURE.	00621600
C		00621700
	COMMON / OUTIN / INPT, IOUT	00621800
	INTEGER AA(100,80), COMMA, BLANK, SLASH	00621900
	DIMENSION MCON(NN), NUM(10), NUMB(10)	00622000
	DATA NAMI / '~' /	00622100
C		00622200
	K = 0	00622300
	NEX = 1	00622400
	IBC=0	00622500
C		00622600
	DO 60 I = 16, 80	00622700
C		00622800
	IF(AA(II,I).EQ. SLASH) GO TO 90	00622900
	IF(AA(II,I) .EQ. COMMA .OR. AA(II,I) .EQ. BLANK	00623000
	* .OR. AA(II,I) .EQ. NAMI) GO TO 10	00623100
	K = K + 1	00623200
	GO TO 60	00623300
90	IBC=1	00623400
	IB=I-1	00623500
	IF(AA(II,IB) .EQ. BLANK) GO TO 65	00623600
C		00623700
10	CONTINUE	00623800
	DO 12 J = I, 80	00623900
	IF(AA(II,J) .EQ. NAMI) GO TO 13	00624000
	IF(AA(II,J) .NE. BLANK) GO TO 15	00624100
12	CONTINUE	00624200
	GO TO 15	00624300
13	IF(NEX .NE. 3) NEX = 2	00624400
15	IF(K .EQ. 0) GO TO 60	00624500
	K1 = I - K	00624600
	K2 = I - 1	00624700
	K = 0	00624800
C		00624900
	IA = 0	00625000
	DO 30 J = K1, K2	00625100
C		00625200
	IA = IA + 1	00625300
	DO 20 JN = 1, 10	00625400
	IF(AA(II,J) .EQ. NUM(JN)) GO TO 25	00625500
20	CONTINUE	00625600
	WRITE(IOUT, 600) (AA(II,L), L = 1, 80),	00625700
	* (AA(II,L), L = K1, K2)	00625800
600	FORMAT(1H1 /// 1X, '--- THIS COMMAND WAS MISUSED ---' /	00625900
	* 5X, 80A1 / 7X, 'ERROR CODE = ', 80A1)	00626000
	MQ1 = 0	00626100
	RETURN	00626200
C		00626300
25	JN = JN - 1	00626400
	NUMB(IA) = JN	00626500
30	CONTINUE	00626600
C		00626700
	JA = 0	00626800
	DO 35 JJ = 1, IA	00626900
	J1 = IA - JJ	00627000
	JA = JA + NUMB(JJ) * 10 ** J1	00627100
35	CONTINUE	00627200
C		00627300
	IF(NEX - 2) 40, 45, 50	00627400
C		00627500
40	MQ1 = MQ1 + 1	00627600
	MCON(MQ1) = JA	00627700
	GO TO 58	00627800
C		00627900
45	JSTART = JA	00628000
	NEX = 3	00628100
	GO TO 58	00628200
C		00628300
50	JEND = JA	00628400
	DO 55 J = JSTART, JEND	00628500
	MQ1 = MQ1 + 1	00628600
	MCON(MQ1) = J	00628700
55	CONTINUE	00628800
	NEX = 1	00628900
58	IF(IBC .EQ. 1) GO TO 65	00629000
C		00629100
60	CONTINUE	00629200
	RETURN	00629300
C		00629400
65	CONTINUE	00629500
	CALL CHANGE(MCON, MCON, NN, MQ1, 1)	00629600
	RETURN	00629700
C		00629800
	END	00629900

	SUBROUTINE FINDDT(XX, Q, MM, X, MCON, MQ1, M1,	00630000
	MA, MB, NN, INDEV)	00630100
	COMMON NUMDT(20)	00630200
C		00630300
C	THIS IS A EXECUTING ROUTINE OF FIND DATA PROCEDURE.	00630400
C		00630500
	COMMON / OUTIN / INPT, IOUT	00630600
	INTEGER Q(MB), POINT(30), BLANK	00630700
	DIMENSION XX(MA), MM(MA), X(NN,MA), MCON(NN), IQ(10)	00630800
	DATA POINT / 30 * 4H.... /, BLANK / ' ' /	00630900
C		00631000
	WRITE(IOUT, 6000)	00631100
	I = 1	00631200
	REWIND INDEV	00631300
	KNEW=NUMDT(INDEV)	00631400
	IF(KNEW.EQ.0) GO TO 1000	00631500
C		00631600
	10 CONTINUE	00631700
	READ(INDEV,END=30,ERR=777) KK,(XX(J),J=1,M1)	00631800
	IF(KK .NE. MCON(I)) GO TO 10	00631900
	DO 20 J = 1, M1	00632000
	X(I,J) = XX(J)	00632100
	20 CONTINUE	00632200
	IF(I .GE. MQ1) GO TO 30	00632300
	I = I + 1	00632400
	GO TO 10	00632500
C		00632600
	30 CONTINUE	00632700
	KS = MQ1 / 15	00632800
	KA = MQ1 - KS * 15	00632900
	KE = KS + 1	00633000
	K2 = 1	00633100
C		00633200
	DO 80 KK = 1, KE	00633300
C		00633400
	IF(KK .NE. KE) GO TO 35	00633500
	IF(KA .EQ. 0) RETURN	00633600
C		00633700
	35 IF(KK .EQ. KE) GO TO 40	00633800
	IF(KK + 1 .NE. KE) GO TO 45	00633900
	IF(KA .NE. 0) GO TO 45	00634000
	40 K3 = MQ1	00634100
	GO TO 50	00634200
	45 K3 = KK * 15	00634300
	50 IF(KK .EQ. KE) GO TO 55	00634400
	KP = 27	00634500
	GO TO 60	00634600
	55 KP = KA * 7 / 4 + 1	00634700
	60 IF(KP .LT. 5) KP = 5	00634800
C		00634900
	WRITE(IOUT, 6400)	00635000
	WRITE(IOUT, 6100) (POINT(I), I = 1, KP)	00635100
	WRITE(IOUT, 6200) (MCON(I), I = K2, K3)	00635200
	WRITE(IOUT, 6100) (POINT(I), I = 1, KP)	00635300
C		00635400
	I2 = 0	00635500
	DO 75 I = 1, M1	00635600
	I1 = I2 + 1	00635700
	I2 = I2 + MM(I)	00635800
	DO 65 J = 1, 10	00635900
	IQ(J) = BLANK	00636000
	65 CONTINUE	00636100
	J = 0	00636200
	DO 70 IM = I1, I2	00636300
	J = J + 1	00636400
	IQ(J) = Q(IM)	00636500
	70 CONTINUE	00636600
	WRITE(IOUT, 6300) (IQ(J), J = 1, 10), (X(J,I), J = K2, K3)	00636700
	75 CONTINUE	00636800

C	WRITE(IOUT, 6100)	(POINT(I), I = 1, KP)	00636900
	IF(K3 .LT. MQ1)	WRITE(IOUT, 6500)	00637000
	K2 = K3 + 1		00637100
C	80 CONTINUE		00637200
	RETURN		00637300
C	1000 WRITE(IOUT,6600) INDEV		00637400
	RETURN		00637500
C	777 WRITE(IOUT, 900)		00637600
	RETURN		00637700
	900 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',		00637800
	* 'SPECIFIED BY COMMAND ---')		00637900
	6000 FORMAT(1H1 /// 1X, '-----' /		00638000
	* 1X, 'I FIND DATA I' /		00638100
	* 1X, '-----')		00638200
	6100 FORMAT(1H , 3X, 13('.'), 28A4)		00638300
	6200 FORMAT(1H , 5X, 'ITEM', 5X, ' ', 4X, 'FIND DATA NUMBER' /		00638400
	* 15X, ' ', 2X, 15I7)		00638500
	6300 FORMAT(1H , 4X, 10A1, ' ', 2X, 15F7.1)		00638600
	6400 FORMAT(///)		00638700
	6500 FORMAT(1H1)		00638800
	6600 FORMAT(/// 10X,'--- FILE NUMBER IS',I3,'. THIS FILE DOES NOT',		00638900
	* ' INCLUDE DATA ---')		00639000
C	END		00639100
			00639200
			00639300
			00639400
			00639500
			00639600

	SUBROUTINE PTFILE(NOF, NFILE, DRANG, DCONS, DPATT, FIND, IFI,	00639700
	* DMULT, DQCHA, BREA, BLANK, DRECO, DASSO,	00639800
	* DCROS)	00639900
C		00640000
C	THIS SUBROUTINE GENERATES THE HISTORY OF MINERVA JOB.	00640100
C		00640200
	COMMON NUMDT(20)	00640300
	COMMON / OUTIN / INPT, IOUT	00640400
	INTEGER DRANG(4),DCONS(4),DPATT(4),DMULT(4),DQCHA(4),DCROS(4),	00640500
	* BREA(4), POINT(25), BLANK, FIND(4), DRECO(4), DASSO(4)	00640600
	DIMENSION NOF(20), NFILE(20,15), NAME(100)	00640700
	DATA POINT / 25 * 4H.... /	00640800
C		00640900
	WRITE(IOUT, 6000)	00641000
	WRITE(IOUT, 6100)	00641100
C		00641200
	MAX = 1	00641300
	DO 10 I = 1, 20	00641400
	IF(NOF(I) .GT. MAX) MAX = NOF(I)	00641500
10	CONTINUE	00641600
	IP = MAX * 2	00641700
	IF(IFI .EQ. 1) IP=IP+2	00641800
	IF(IP .LT. 7) IP = 7	00641900
	IF(IP .GT. 20) IP = 20	00642000
	WRITE(IOUT, 6200) (POINT(I), I = 1, IP)	00642100
C		00642200
	DO 60 I = 1, 20	00642300
	DO 15 K = 1, 100	00642400
15	NAME(K) = BLANK	00642500
C		00642600
	IF(NUMDT(I) .LE. 0) GO TO 60	00642700
	NF = NOF(I)	00642800
	IF(NF .LE. 0) GO TO 55	00642900
C		00643000
	DO 50 J = 1, NF	00643100
	K = (J - 1) * 8 + 1	00643200
	M = NFILE(I,J)	00643300
	GO TO (50, 50, 20, 25, 30, 35, 40, 50, 45, 50, 48, 50, 51, 61),M	00643400
C		00643500
20	CONTINUE	00643600
	DO 21 L = 1, 4	00643700
	NAME(K) = DRANG(L)	00643800
	K = K + 1	00643900
21	CONTINUE	00644000
	GO TO 50	00644100
C		00644200
25	CONTINUE	00644300
	DO 26 L = 1, 4	00644400
	NAME(K) = DCONS(L)	00644500
	K = K + 1	00644600
26	CONTINUE	00644700
	GO TO 50	00644800
C		00644900
30	CONTINUE	00645000
	DO 31 L = 1, 4	00645100
	NAME(K) = DPATT(L)	00645200
	K = K + 1	00645300
31	CONTINUE	00645400
	GO TO 50	00645500
C		00645600
35	CONTINUE	00645700
	DO 36 L = 1, 4	00645800
	NAME(K) = DMULT(L)	00645900
	K = K + 1	00646000
36	CONTINUE	00646100
	GO TO 50	00646200
C		00646300

40	CONTINUE	00646400
	DO 41 L = 1, 4	00646500
	NAME(K) = DQCHA(L)	00646600
	K = K + 1	00646700
41	CONTINUE	00646800
	GO TO 50	00646900
C		00647000
45	CONTINUE	00647100
	DO 46 L = 1, 4	00647200
	NAME(K) = BREA(L)	00647300
	K = K + 1	00647400
46	CONTINUE	00647500
	GO TO 50	00647600
C		00647700
48	CONTINUE	00647800
	DO 49 L=1,4	00647900
	NAME(K)=DRECO(L)	00648000
	K=K+1	00648100
49	CONTINUE	00648200
	GO TO 50	00648300
C		00648400
51	CONTINUE	00648500
	DO 52 L=1,4	00648600
	NAME(K)=DASSO(L)	00648700
	K=K+1	00648800
52	CONTINUE	00648900
	GO TO 50	00649000
C		00649100
61	CONTINUE	00649200
	DO 62 L=1,4	00649300
	NAME(K)=DCROS(L)	00649400
	K=K+1	00649500
62	CONTINUE	00649600
C		00649700
50	CONTINUE	00649800
C		00649900
	IF(I .NE. 11) GO TO 55	00650000
	IF(IFI .NE. 1) GO TO 55	00650100
	K = K + 4	00650200
	DO 47 L = 1, 4	00650300
	NAME(K) = FIND(L)	00650400
	K = K + 1	00650500
47	CONTINUE	00650600
C		00650700
55	CONTINUE	00650800
	WRITE(IOUT, 6300) I, NUMDT(I), (NAME(K), K = 1, 80)	00650900
C		00651000
60	CONTINUE	00651100
	WRITE(IOUT, 6400)	00651200
	RETURN	00651300
C		00651400
6000	FORMAT(1H1 ///1X, 'JOB EXECUTING INFORMATION' / 1X, 24('-') //	00651500
6100	FORMAT(1H , 7X, 'SPECIFIED', 4X, 'SIZE OF DATA', 2X, '.' /	00651600
*	5X, 'NUMBER OF FILE', 3X, 'IN ITS FILE', 2X, '.',	00651700
*	6X, 'HISTORY OF PROCEDURES'	00651800
6200	FORMAT(1H , 4X, 31('.'), 20A4)	00651900
6300	FORMAT(1H , 8X, 15, 10X, 15, 6X, '.', 4X, 80A1)	00652000
6400	FORMAT(/// 1X, '--- LIST ---' //	00652100
*	5X, 'RANG = RANGE CHECK' //	00652200
*	5X, 'CONS = CONSISTENCY CHECK' //	00652300
*	5X, 'FIND = FIND DATA' //	00652400
*	5X, 'PATT = PATTERN CHECK' //	00652500
*	5X, 'MULT = MULTIWAY TABLES' //	00652600
*	5X, 'QCHA = Q-CHART' //	00652700
*	5X, 'BREA = BREAK DOWN' //	00652800
*	5X, 'RECO = RECODE' //	00652900
*	5X, 'ASSO = ASSOCIATION' //	00653000
*	5X, 'CROS = CROSS TABLES' //	00653100
*	5X, 'BMDP = P1D, P2D, P4D, P5D, P6D, P1F, P2F, P3F' //	00653200
*	5X, 'BLANK= NO EXECUTING PROCEDURE')	00653300
	END	00653400

```

SUBROUTINE CONIF( XX,MQ7,RELAT7,CONST7,LOGIC7,IFTAB7,LIT,MA,MY, 00653500
* MB,M07,IT,MTRU,N,ITEMP,IA1,IA2,NUM,KN,FMT6,IA, 00653600
* INCOA,NN,LOGICS,IFTABS,MP7,M1,INDEV,OUTDEV) 00653700
C 00653800
COMMON NUMDT(20) 00653900
COMMON / OUTIN / INPT,IOUT 00654000
INTEGER RELAT7(MY),FMT6(7),OUTDEV 00654100
DIMENSION XX(MA),ITEMP(IA1,IA2),NUM(KN),MQ7(MY),CONST7(MY), 00654200
* LOGIC7(MY),IFTAB7(MB),LIT(MY),INCOA(NN), 00654300
* LOGICS(MY),IFTABS(MB) 00654400
NOA=0 00654500
NC=0 00654600
NOS=0 00654700
INA=0 00654800
LF=0 00654900
WRITE(IOUT,650) 00655000
650 FORMAT(1H0,1X,'----- INCONSISTENT DATA ----- ( A ) ') 00655100
WRITE(IOUT,651) 00655200
651 FORMAT(1H0,'NO.',2X,'DATA',15X,'-- DATA X --') 00655300
WRITE(IOUT,690) (ITEMP(1,J),J=1,110) 00655400
690 FORMAT(1H 15X,110A1) 00655500
C 00655600
REWIND INDEV 00655700
DO 10 I=1,N 00655800
DO 11 J=1,MP7 00655900
11 LOGICS(J)=LOGIC7(J) 00656000
DO 12 J=1,IT 00656100
12 IFTABS(J)=IFTAB7(J) 00656200
READ(INDEV,END=888,ERR=777) KK,(XX(J),J=1,M1) 00656300
NOS=NOS+1 00656400
C 00656500
CALL CONTAB (XX,MQ7,RELAT7,CONST7,LOGICS,IFTABS,LIT, 00656600
* MA,MY,MB,M07,MP7,IT,MTRU) 00656700
IF(MTRU.EQ. 1) GO TO 40 00656800
INA=INA+1 00656900
NOA=NOA+1 00657000
INCOA(INA)=KK 00657100
IF(INA.GE. 201) GO TO 5 00657200
WRITE(IOUT,630) INA,KK 00657300
630 FORMAT(1H ,13,2X,15) 00657400
IA=1 00657500
DO 70 J=1,M07 00657600
IF(J.GT.10) GO TO 10 00657700
IQ=M07(J) 00657800
CALL FORM (NUM,KN,FMT6,IA,4,5,7) 00657900
WRITE(IOUT,FMT6) XX(IQ) 00658000
70 IA=IA+12 00658100
GO TO 10 00658200
C 00658300
5 CALL INVPRI (NUM,KN,LF,INA,KK,0 ) 00658400
GO TO 10 00658500
C 00658600
40 NC=NC+1 00658700
IF(OUTDEV.LE. 0) GO TO 10 00658800
WRITE(OUTDEV) KK,(XX(J),J=1,M1) 00658900
10 CONTINUE 00659000
C 00659100
IF(LF.NE.0) CALL INVPRI (NUM,KN,LF,INA,KK,1 ) 00659200
888 CONTINUE 00659300
FN=NOS 00659400
IF(NC.NE. 0) GO TO 25 00659500
WRITE(IOUT,640) 00659600
640 FORMAT(1H0,3X,'--- ALL DATA ARE INSUFFICIENT, THEREFORE ', 00659700
* ' IT IS IMPOSSIBLE TO STORE ON DISK. ---') 00659800
C 00659900
25 ATRU=FLOAT(NC)/FN*100. 00660000
AFAU=FLOAT(NOA)/FN*100. 00660100
IF(OUTDEV.GE. 1) NUMDT(OUTDEV)=NC 00660200
IF(INA.GE. 1) GO TO 34 00660300
WRITE(IOUT,691) 00660400
691 FORMAT(1H0,5X,' --- ALL SAMPLES ARE CONSISTENT --- ') 00660500
34 WRITE(IOUT,700) N,NOS 00660600
700 FORMAT(1H1,///1X,'SUMMARY TABLE'//1X,13(' '), 00660700
* ///11X,'TOTAL NUMBER OF SAMPLES TN =',16, 00660800
* ///11X,'NUMBER OF SAMPLES N =',16) 00660900
WRITE(IOUT,705) 00661000
705 FORMAT(1H0,12X,'IF ( A ) ') 00661100
WRITE(IOUT,710) NC,ATRU,NOA,AFAU 00661200
710 FORMAT(1H0,15X,'A --- TRUE =',17,2X,F6.2,' % ( 1/N ) '/ 00661300
* 22X,'FALSE = ', 15,2X,F6.2,' % ( 1/N ) ') 00661400
RETURN 00661500
777 WRITE( IOUT,900) 00661600
900 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ', 00661700
* 'SPECIFIED BY COMMAND ---' ) 00661800
RETURN 00661900
END 00662000

```

	SUBROUTINE READK(AA,II,NUM,KN,KNEW)	00662100
C		00662200
C	THIS SUBROUTINE IS A DECODER OF READ STATEMENT IN MINERVA.	00662300
C		00662400
	COMMON / OUTIN / INPT, IOUT	00662500
	INTEGER AA(100,80), DTAPE(4),DCARD(4), DDISK(4) ,	00662600
	* BLANK, EQUAL, SLASH, OPC, CRC	00662700
	DIMENSION NUM(KN), N(3)	00662800
	DATA DTAPE/'T','A','P','E'/, DCARD/'C','A','R','D'/,	00662900
	* DDISK/'D','I','S','K'/, BLANK/' ', EQUAL/'='/,	00663000
	* SLASH/'/'/, OPC/'('/, CRC/')'/, N/3*0/	00663100
C		00663200
	K1=0	00663300
	K3=0	00663400
	KS=0	00663500
	KNEW=9	00663600
C		00663700
	DO 10 I=1,80	00663800
	IF(AA(II,I).EQ.BLANK .OR. AA(II,I).EQ.EQUAL .OR. AA(II,I).EQ.SLASH	00663900
	* .OR. AA(II,I).EQ.OPC .OR. AA(II,I).EQ.CRC)	00664000
	* GO TO 20	00664100
	KS=KS+1	00664200
	GO TO 10	00664300
20	IF(KS.EQ.0) GO TO 10	00664400
	KK=I-KS	00664500
	KS=0	00664600
	DO 30 J=1,10	00664700
	IF(AA(II,KK).EQ.NUM(J)) GO TO 40	00664800
30	CONTINUE	00664900
	JJ=KK	00665000
	DO 31 J=1,4	00665100
	IF(DTAPE(J).NE.AA(II,JJ)) GO TO 32	00665200
31	JJ=JJ+1	00665300
	KNEW=9	00665400
	GO TO 10	00665500
C		00665600
	32 JJ=KK	00665700
	DO 33 J=1,4	00665800
	IF(DCARD(J).NE.AA(II,JJ)) GO TO 35	00665900
33	JJ=JJ+1	00666000
	KNEW=5	00666100
	GO TO 10	00666200
C		00666300
	35 JJ=KK	00666400
	DO 36 J=1,4	00666500
	IF(DDISK(J).NE.AA(II,JJ)) GO TO 37	00666600
36	JJ=JJ+1	00666700
	KNEW=99	00666800
	GO TO 10	00666900
C		00667000
	37 KNEW=99	00667100
	GO TO 10	00667200
C		00667300
	40 IA=KK	00667400
58	CONTINUE	00667500
	DO 61 K=1,10	00667600
	IF(AA(II,IA).EQ.NUM(K)) GO TO 57	00667700
61	CONTINUE	00667800
	GO TO 59	00667900
C		00668000
	57 K1=K1+1	00668100
	KKK=K-1	00668200
	N(K1)=KKK	00668300
	IA=IA+1	00668400
	GO TO 58	00668500
59	DO 62 KI=1,K1	00668600
	K2=K1-KI	00668700
	K3=K3+N(K1)*10**K2	00668800
62	CONTINUE	00668900
	KNEW=K3	00669000
10	CONTINUE	00669100
	IF(KNEW.EQ.99) GO TO 999	00669200
C		00669300
	RETURN	00669400
C		00669500
	999 WRITE(IOUT,998) (AA(II,I),I=1,80)	00669600
998	FORMAT(1H1///1H ,'---- COMMAND CHECK ---'//1H ,80A1)	00669700
	KNEW=99	00669800
	RETURN	00669900
	END	00670000

	SUBROUTINE	COEFF (AA,BLANK,COMMA,II,COFF,IERO,MC,ABC,KK,KJI,	00670100
	*	IPUNCH,KA1)	00670200
C			00670300
C	THIS SUBROUTINE	DECODES THE OPTION COMMAND IN ASSOCIATION PROCEDURE.	00670400
C			00670500
	INTEGER	AA(100,80),BLANK,COMMA,EQUAL,SLASH,	00670600
	*	ALL(3),PEAR(4),CONS(4),KEND(4),CRAM(4),	00670700
	*	CORR(4),TAU(3),GAMM(4),SOMM(4),GOOD(4),	00670800
	*	ABC(26),PUNC(4),YATE(4),MCNE(4)	00670900
	DIMENSION	COFF(KA1)	00671000
	DATA	EQUAL/'=',ALL/'A','L','L',PEAR/'P','E','A','R',	00671100
	*	CONS/'C','O','N','T',KEND/'S','Q','U','A',	00671200
	*	CRAM/'C','R','A','M',CORR/'C','O','R','R',	00671300
	*	TAU/'T','A','U',GAMM/'G','A','M','M',	00671400
	*	SOMM/'S','O','M','E',GOOD/'G','O','D','D',	00671500
	*	YATE/'Y','A','T','E',MCNE/'M','C','N','E',	00671600
	*	SLASH/'/',PUNC/'P','U','N','C',	00671700
C	III=0		00671800
	K=0		00671900
	DO 30 I=KK,80		00672000
	IF(AA(II,I).EQ.SLASH)	GO TO 13	00672100
	IF(AA(II,I).EQ.BLANK.OR.AA(II,I).EQ.EQUAL.OR.AA(II,I).EQ.COMMA)		00672200
	*	GO TO 20	00672300
	K=K+1		00672400
	GO TO 30		00672500
C			00672600
	13 III=1		00672700
	KJI=0		00672800
C			00672900
	20 IF(K.EQ.0)	GO TO 10	00673000
	K1=I-K		00673100
C			00673200
	DO 25 J=1,26		00673300
	IF(AA(II,K1).EQ.ABC(J))	GO TO 35	00673400
	25 CONTINUE		00673500
	IERO=1		00673600
	RETURN		00673700
C			00673800
	35 L=K1		00673900
	DO 45 J=1,3		00674000
	IF(ALL(J).NE.AA(II,L))	GO TO 55	00674100
	45 L=L+1		00674200
	DO 50 J=1,11		00674300
	50 COFF(J)=J		00674400
	MC=11		00674500
	K=0		00674600
	GO TO 10		00674700
C			00674800
	55 L=K1		00674900
	DO 60 J=1,4		00675000
	IF(PEAR(J).NE.AA(II,L))	GO TO 65	00675100
	60 L=L+1		00675200
	MC=MC+1		00675300
	COFF(MC)=1		00675400
	K=0		00675500
	GO TO 10		00675600
C			00675700
	65 L=K1		00675800
	DO 70 J=1,4		00675900
	IF(CONS(J).NE.AA(II,L))	GO TO 75	00676000
	70 L=L+1		00676100
	MC=MC+1		00676200
	COFF(MC)=4		00676300
	K=0		00676400
	GO TO 10		00676500
C			00676600
	75 L=K1		00676700
	DO 80 J=1,4		00676800
	IF(KEND(J).NE.AA(II,L))	GO TO 85	00676900
	80 L=L+1		00677000
	MC=MC+1		00677100
	COFF(MC)=5		00677200
	K=0		00677300
	GO TO 10		00677400
C			00677500
	85 L=K1		00677600
	DO 90 J=1,4		00677700
	IF(CRAM(J).NE.AA(II,L))	GO TO 95	00677800
	90 L=L+1		00677900
	MC=MC+1		00678000
	COFF(MC)=6		00678100
	K=0		00678200
	GO TO 10		00678300
			00678400

C	95 L=K1	00678500
	DO 99 J=1,4	00678600
	IF(CORR(J).NE.AA(II,L)) GO TO 88	00678700
	99 L=L+1	00678800
	MC=MC+1	00678900
	COFF(MC)=7	00679000
	K=0	00679100
	GO TO 10	00679200
		00679300
		00679400
C	88 L=K1	00679500
	DO 77 J=1,3	00679600
	IF(TAU(J).NE.AA(II,L)) GO TO 66	00679700
	77 L=L+1	00679800
	MC=MC+1	00679900
	COFF(MC)=8	00680000
	K=0	00680100
	GO TO 10	00680200
		00680300
C	66 L=K1	00680400
	DO 54 J=1,4	00680500
	IF(GAMM(J).NE.AA(II,L)) GO TO 44	00680600
	54 L=L+1	00680700
	MC=MC+1	00680800
	COFF(MC)=9	00680900
	K=0	00681000
	GO TO 10	00681100
		00681200
C	44 L=K1	00681300
	DO 33 J=1,4	00681400
	IF(SOMM(J).NE.AA(II,L)) GO TO 22	00681500
	33 L=L+1	00681600
	MC=MC+1	00681700
	COFF(MC)=10	00681800
	K=0	00681900
	GO TO 10	00682000
		00682100
C	22 L=K1	00682200
	DO 11 J=1,4	00682300
	IF(GOOD(J).NE.AA(II,L)) GO TO 12	00682400
	11 L=L+1	00682500
	MC=MC+1	00682600
	COFF(MC)=11	00682700
	K=0	00682800
	GO TO 10	00682900
		00683000
C	12 L=K1	00683100
	DO 14 J=1,4	00683200
	IF(PUNC(J).NE.AA(II,L)) GO TO 15	00683300
	14 L=L+1	00683400
	IPUNCH=1	00683500
	K=0	00683600
	GO TO 10	00683700
		00683800
C	15 L=K1	00683900
	DO 16 J=1,4	00684000
	IF(YATE(J).NE.AA(II,L)) GO TO 17	00684100
	16 L=L+1	00684200
	MC=MC+1	00684300
	COFF(MC)=2	00684400
	K=0	00684500
	GO TO 10	00684600
		00684700
C	17 L=K1	00684800
	DO 21 J=1,4	00684900
	IF(MCNE(J).NE.AA(II,L)) GO TO 23	00685000
	21 L=L+1	00685100
	MC=MC+1	00685200
	COFF(MC)=3	00685300
	K=0	00685400
	GO TO 10	00685500
		00685600
C	23 IERO=1	00685700
	RETURN	00685800
		00685900
C	10 IF(III.EQ.1) RETURN	00686000
	30 CONTINUE	00686100
	KJ1=1	00686200
		00686300
C	RETURN	00686400
	END	00686500


```

      SUBROUTINE ASSOCI(XX,MA,MQ,MX,M1,INDEV,BLANK,Q,MM,MB,      00686600
      *          WORK,LIMIT1,NUM,KN,COFF,MC,MQ1,IPUNCH,KA1 ) 00686700
C
C THIS IS A PRE-MAIN ROUTINE FOR ASSOCIATION PROCEDURE. 00686800
C 00686900
C 00687000
      COMMON NUMDT(20) 00687100
      COMMON / OUTIN / INPT,IOUT 00687200
      INTEGER Q(MB),BLANK 00687300
      DIMENSION XX(MA),MQ(MX),MM(MA),WORK(LIMIT1), 00687400
      *          NUM(KN),COFF(KA1) 00687500
C 00687600
      KNEW=NUMDT(INDEV) 00687700
C 00687800
      WRITE(IOUT,6000) 00687900
6000 FORMAT(1H1,///1H ,'+-----+','// 00688000
      * 1X,'| ASSOCIATION MEASURES '|,// 00688100
      * 1X,'| '|,// 00688200
      * 1X,'| FOR '|,// 00688300
      * 1X,'| '|,// 00688400
      * 1X,'| CONTINGENCY TABLES '|,// 00688500
      * 1X,'+-----+',' ) 00688600
C 00688700
      IF(KNEW.EQ.0) GO TO 99 00688800
      N=MQ1 00688900
      NN=N*N 00689000
      IA=20 00689100
      IA2=IA*IA 00689200
C 00689300
C PARTITION THE STORAGE ARRAY 00689400
C 00689500
      N1=1 00689600
      N2=N1+NN 00689700
      N3=N2+NN 00689800
      N4=N3+NN 00689900
      N5=N4+NN 00690000
      N6=N5+NN 00690100
      N7=N6+NN 00690200
      N8=N7+NN 00690300
      N9=N8+NN 00690400
      N10=N9+NN 00690500
      N11=N10+NN 00690600
      N12=N11+NN 00690700
      N13=N12+NN 00690800
      N14=N13+NN 00690900
      N15=N14+IA 00691000
      N16=N15+IA 00691100
      N17=N16+IA 00691200
      N18=N17+IA 00691300
      N19=N18+IA2 00691400
      N20=N19+IA2 00691500
      N21=N20+IA2 00691600
      N22=N21+IA2 00691700
      N23=N22+IA2 00691800
      N24=N23+IA*N 00691900
      N25=N24+N 00692000
      N26=N25+NN 00692100
      N27=N26+NN 00692200
      N28=N27+NN-1 00692300
C 00692400
C CHECK THE WORKING AREA 00692500
C 00692600
      LIMIT5=LIMIT1-N28 00692700
      IF(LIMIT5.LT.0) GO TO 90 00692800
C 00692900
      CALL AMAIN(XX,MA,MQ,MX,M1,INDEV,BLANK,Q,MM,MB,WORK(N1), 00693000
      * WORK(N2),WORK(N3),WORK(N4),WORK(N5),WORK(N6), 00693100
      * WORK(N7),WORK(N8),WORK(N9),WORK(N10),WORK(N11), 00693200
      * WORK(N12),WORK(N13),WORK(N14),WORK(N15),WORK(N16), 00693300
      * WORK(N17),WORK(N18),WORK(N19),WORK(N20),WORK(N21), 00693400
      * WORK(N22),WORK(N23),WORK(N24),WORK(N25),WORK(N26), 00693500
      * WORK(N27),N,IA,NUM,KN,COFF,MC,MQ1,IPUNCH,KA1 ) 00693600
C 00693700
      RETURN 00693800
C 00693900
      90 WRITE(IOUT,900) N28 00694000
      900 FORMAT(1H , ' USED WORKING AREA WORK(1) ~ WORK(' ,I6,')' ) 00694100
      RETURN 00694200
C 00694300
      99 WRITE(IOUT,910) INDEV 00694400
      910 FORMAT(///1H , '--- FILE NUMBER IS',I3, ' . THIS FILE DOES NOT ' , 00694500
      * 'INCLUDE DATA ---' ) 00694600
      RETURN 00694700
C 00694800
      END 00694900

```

```

SUBROUTINE AMAIN(XX,MA,MQ,MX,M1,INDEV,BLANK,G,MM,MB, 00695000
* X2,P,F,T,C,R,TB,TC,RG,DTX,XA,XB,XN,X,Y,COLUM,ROW, 00695100
* CROSS,D,V,EE,E,XY,YY,DXY,YATE,MCNE,N,IA,NUM,KN, 00695200
* COFF,MC,MQ1,IPUNCH,KAL ) 00695300
C 00695400
C THIS IS A MAIN SUBPROGRAM OF ASSOCIATION PROCEDURE. 00695500
C 00695600
COMMON / OUTIN / INPT, IOUT 00695700
REAL MCNE,MXM 00695800
INTEGER Q(MB),BLANK,FMT1(8),FMT2(9),FMT3(11),FMT4(10) 00695900
DIMENSION X2(N,N),P(N,N),F(N,N),T(N,N),C(N,N),R(N,N),TB(N,N), 00696000
* TC(N,N),RG(N,N),DTX(N,N),XA(N,N),XB(N,N),XN(N,N), 00696100
* X(IA),Y(IA),COLUM(IA),ROW(IA),CROSS(IA,IA),D(IA,IA), 00696200
* V(IA,IA),EE(IA,IA),E(IA,IA),XY(N,IA),YY(N), 00696300
* YATE(N,N),MCNE(N,N),COFF(KAL), 00696400
* XX(MA),MQ(MX),MM(MA),NUM(KN),DXY(N,N) 00696500
DIMENSION JK(2),IK(2),POINT(50) 00696600
DATA JK/2# ' /, IK/2# ' /, POINT/ 50# 2H-- / 00696700
DATA FMT1/'(/','10X',' ', '4H1','4H * ',' ',' ', 'A1)'/, 00696800
* FMT2/'(10X','4H','Y','4H ' , '4H ' , ' ', ' ', 'F6.0', 00696900
* 4H, ' ,4H ' ) /, 00697000
* FMT3/'(8X','F6.0','4H','I ,4H ' , ' ', ' ', 'F6.0', 00697100
* 4H, ' ,4H ' , 'F6.0',' )' /, 00697200
* FMT4/'(14X','4H','I ,4H ' , ' ', ' ', 'F6.0','4H, ' , 00697300
* 4H ' , 'F6.0',' )' / 00697400
C 00697500
IERO=0 00697600
ICOUNT=0 00697700
C 00697800
DETECT CATEGORIES AND THEIR NUMBERS 00697900
C 00698000
CALL CATGYA (XX,INDEV,MA,XY,YY,MQ,MX,M1,N,N,IA,IERO ) 00698100
C 00698200
KKK=MQ1-1 00698300
DO 70 IS=1,KKK 00698400
LLL=IS+1 00698500
DO 80 JS=LLL,MQ1 00698600
C 00698700
DO 9 KK=1,IA 00698800
X(KK)=XY(IS,KK) 00698900
Y(KK)=XY(JS,KK) 00699000
9 CONTINUE 00699100
J=YY(IS) 00699200
I=YY(JS) 00699300
C 00699400
C COMPLETE THE CROSS-CLASSIFIED TABLES 00699500
C 00699600
CALL TCROSS(XX,MA,MQ,HX,IS,JS,I,J,X,Y,CROSS,M1,INDEV,IA,IERO) 00699700
IF(IERO.EQ. 1) RETURN 00699800
C 00699900
TOTAL1=0 00700000
TOTAL2=0 00700100
C 00700200
DO 30 K=1,20 00700300
COLUM(K)=0 00700400
ROW(K)=0 00700500
30 CONTINUE 00700600
C 00700700
C CALCULATE THE SUMS OF ROWS OR COLUMNS 00700800
C 00700900
DO 10 II=1,1 00701000
DO 20 JJ=1,J 00701100
COLUM(II)=COLUM(II)+CROSS(II,JJ) 00701200
ROW(JJ)=ROW(JJ)+CROSS(II,JJ) 00701300
20 CONTINUE 00701400
10 CONTINUE 00701500

```

C	DD 50 K=1,I	00701600
	TOTAL1=TOTAL1+COLUM(K)	00701700
	50 CONTINUE	00701800
C	DD 60 KK=1,J	00701900
	TOTAL2=TOTAL2+ROW(KK)	00702000
	60 CONTINUE	00702100
	IF(TOTAL1.NE.TOTAL2) GO TO 999	00702200
	TOTAL=TOTAL1	00702300
C	IF(MQ1.NE.2) GO TO 90	00702400
C		00702500
C	GENERATE THE TWO-WAY CROSS TABLES	00702600
C		00702700
	N1=2	00702800
	M2=2	00702900
	JJJ=J	00703000
	11 J1=JJJ/10	00703100
	IF(J1.EQ.0) GO TO 12	00703200
	J2=JJJ-J1*10	00703300
	JK(N1)=J2	00703400
	JJJ=J1	00703500
	N1=N1-1	00703600
	GO TO 11	00703700
	12 JK(N1)=JJJ	00703800
	III=I	00703900
	13 I1=III/10	00704000
	IF(I1.EQ.0) GO TO 14	00704100
	I2=III-I1*10	00704200
	IK(M2)=I2	00704300
	III=I1	00704400
	M2=M2-1	00704500
	GO TO 13	00704600
	14 IK(M2)=III	00704700
	N2=JK(1)	00704800
	N3=JK(2)	00704900
	FMT2(6)=NUM(N3+1)	00705000
	FMT3(6)=NUM(N3+1)	00705100
	FMT4(5)=NUM(N3+1)	00705200
	IF(JK(1).EQ.BLANK) GO TO 15	00705300
	FMT2(5)=NUM(N2+1)	00705400
	FMT3(5)=NUM(N2+1)	00705500
	FMT4(4)=NUM(N2+1)	00705600
	15 CONTINUE	00705700
C		00705800
	KK=(38+6*(J-3))/2	00705900
C		00706000
	K1=MQ(1)	00706100
	L1=1	00706200
	CALL PITEM (MM,MA,M1,K1,L1,L2,KP)	00706300
	MM1=L2-L1+1	00706400
	FMT1(7)=NUM(MM1+1)	00706500
	K2=MQ(2)	00706600
	L3=1	00706700
	CALL PITEM (MM,MA,M1,K2,L3,L4,KI)	00706800
	MM2=L4-L3+1	00706900
	FMT1(3)=NUM(MM2+1)	00707000
C		00707100
	WRITE(IOUT,FMT1) (Q(L),L=L3,L4),(Q(LL),LL=L1,L2)	00707200
C		00707300
	WRITE(IOUT,6600)	00707400
	6600 FORMAT(1H ,////)	00707500
	WRITE(IOUT,FMT2) (X(II),II=1,J)	00707600
	WRITE(IOUT,6200) (POINT(K),K=1,KK)	00707700
	DO 40 II=1,I	00707800
	WRITE(IOUT,FMT3) Y(II),(CROSS(II,JJ),JJ=1,J),COLUM(II)	00707900
	40 CONTINUE	00708000
	WRITE(IOUT,6200) (POINT(K),K=1,KK)	00708100
	6200 FORMAT(9X,50A2)	00708200
	WRITE(IOUT,FMT4) (ROW(II),II=1,J),TOTAL	00708300
	WRITE(IOUT,6100)	00708400
	6100 FORMAT(///)	00708500
		00708600
		00708700
		00708800

C		00708900
	90 CONTINUE	00709000
C		00709100
C	CALCULATE THE MEASURES OF ASSOCIATION	00709200
C		00709300
	CALL CORRLA (CROSS,COLUM,ROW,I,J,D,EE,E,V,TOTAL,IK,JK,	00709400
*	NUM,KN,IA,MQ1,XXX,PP,BB,TT,CC,YXY,MXM,ICOUNT)	00709500
	CALL CORRLB (CROSS,COLUM,ROW,I,J,X,Y,TOTAL,IA,MQ1,SR)	00709600
	CALL CATGS1 (CROSS,TOTAL,I,J,IA,MQ1,TBK,TCK,RK,SD,SDX)	00709700
	CALL CATGS2 (CROSS,COLUM,ROW,I,J,TOTAL,IA,MQ1,GA,GB,GX)	00709800
		00709900
C	IF(MQ1.EQ. 2) RETURN	00710000
C		00710100
	X2(JS,IS)=XXX	00710200
	P(JS,IS)=PP	00710300
	F(JS,IS)=BB	00710400
	T(JS,IS)=TT	00710500
	C(JS,IS)=CC	00710600
	R(JS,IS)=SR	00710700
	TB(JS,IS)=TBK	00710800
	TC(JS,IS)=TCK	00710900
	RG(JS,IS)=RK	00711000
	DTX(JS,IS)=SD	00711100
	DXY(JS,IS)=SDX	00711200
	XA(JS,IS)=GA	00711300
	XB(JS,IS)=GB	00711400
	XN(JS,IS)=GX	00711500
	YATE(JS,IS)=YXY	00711600
	MCNE(JS,IS)=MXM	00711700
C		00711800
	X2(IS,JS)=XXX	00711900
	P(IS,JS)=PP	00712000
	F(IS,JS)=BB	00712100
	T(IS,JS)=TT	00712200
	C(IS,JS)=CC	00712300
	R(IS,JS)=SR	00712400
	TB(IS,JS)=TBK	00712500
	TC(IS,JS)=TCK	00712600
	RG(IS,JS)=RK	00712700
	DTX(IS,JS)=SD	00712800
	DXY(IS,JS)=SDX	00712900
	XA(IS,JS)=GA	00713000
	XB(IS,JS)=GB	00713100
	XN(IS,JS)=GX	00713200
	YATE(IS,JS)=YXY	00713300
	MCNE(IS,JS)=MXM	00713400
C		00713500
	DO 85 IZ=1,IA	00713600
	DO 95 IW=1,IA	00713700
	CROSS(IZ,IW)=0	00713800
95	CONTINUE	00713900
	COLUM(IZ)=0	00714000
	ROW(IZ)=0	00714100
	X(IZ)=0	00714200
	Y(IZ)=0	00714300
85	CONTINUE	00714400
C		00714500
	80 CONTINUE	00714600
70	CONTINUE	00714700
C		00714800
	DO 75 I=1,N	00714900
	X2(I,I)=0	00715000
	P(I,I)=0	00715100
	F(I,I)=0	00715200
	T(I,I)=0	00715300
	C(I,I)=0	00715400
	R(I,I)=0	00715500
	TB(I,I)=0	00715600
	TC(I,I)=0	00715700
	RG(I,I)=0	00715800
	DTX(I,I)=0	00715900
	DXY(I,I)=0	00716000
	XA(I,I)=0	00716100
	XB(I,I)=0	00716200
	XN(I,I)=0	00716300
	YATE(I,I)=0	00716400
	MCNE(I,I)=0	00716500
75	CONTINUE	00716600
C		00716700
	CALL CPRINT (X2,P,F,T,C,R,TB,TC,RG,DTX,DXY,XA,XB,XN,YATE,MCNE,	00716800
*	N,COFF,MQ,Q,MM,MX,MB,MA,MQ1,	00716900
*	MC,NUM,KN,M1,IPUNCH,ICOUNT,KA1)	00717000
C		00717100
	RETURN	00717200
C		00717300
	999 WRITE(IOUT,6000)	00717400
6000	FORMAT(///10X,' --- YOUR GIVEN MATRIX IS MISSING.' /25X,	00717500
*	'YOU MUST CHECK YOUR DATA --- ')	00717600
	RETURN	00717700
C		00717800
	END	00717900

C	SUBROUTINE	TCROSS(XX,MA,MQ,MX,I1,J1,I1,JJ,X,Y,CROSS,M1,	00718000
	*	INDEV,IA,IERO)	00718100
C			00718200
C	GENERATE THE CROSS TABLES FOR ASSOCIATION PROCEDURE.		00718300
C	COMMON	/ OUTIN / INPT,IOUT	00718400
	DIMENSION	XX(MA),MQ(MX),X(IA),Y(IA),CROSS(IA,IA)	00718500
C	I2=MQ(I1)		00718600
	J2=MQ(J1)		00718700
C			00718800
	DO 10 I=1,I1		00718900
	DO 10 J=1,JJ		00719000
	CROSS(I,J)=0		00719100
10	CONTINUE		00719200
C			00719300
	REWIND INDEV		00719400
C			00719500
20	CONTINUE		00719600
	READ(INDEV,END=99,ERR=999) KK,(XX(J),J=1,M1)		00719700
C			00719800
	DO 30 K1=1,JJ		00719900
	IF(X(K1).EQ.XX(I2)) GO TO 35		00720000
30	CONTINUE		00720100
	GO TO 20		00720200
35	CONTINUE		00720300
	DO 40 K2=1,I1		00720400
	IF(Y(K2).EQ.XX(J2)) GO TO 45		00720500
40	CONTINUE		00720600
	GO TO 20		00720700
45	CROSS(K2,K1)=CROSS(K2,K1)+1.		00720800
	GO TO 20		00720900
C			00721000
99	CONTINUE		00721100
C			00721200
	RETURN		00721300
C			00721400
999	WRITE(IOUT,900)		00721500
	IERO=1		00721600
	RETURN		00721700
C			00721800
900	FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',		00721900
	* 'SPECIFIED BY COMMAND ---')		00722000
	END		00722100
			00722200
			00722300
			00722400

```

SUBROUTINE CORRLA (CROSS,COLUM,ROW,II,JJ,DD,EE,E,V,TOTAL, 00722500
* IK,JK,NUM,KN,IA,M1,XX,PP,BB,TT, 00722600
* CC,XY,XM,ICOUNT ) 00722700
COMMON / OUTIN / INPT, IOUT 00722800
C 00722900
INTEGER BLANK,FMT(25),FMT1(26) 00723000
DIMENSION IK(2),JK(2) 00723100
DIMENSION CROSS(IA,IA),COLUM(IA),ROW(IA),EE(IA,IA),DD(IA,IA), 00723200
* E(IA,IA),V(IA,IA),NUM(KN) 00723300
DATA BLANK/' '/ 00723400
DATA FMT/'(','13X','4H'EE('I,J'),'4H = '/,' ' ' '(19X',' ' ', 00723500
* ' ' ' ',F11,' '6//',' 00723600
* '13X','4H' E('I,J'),'4H = '/,' ' ' '(19X',' ' ', 00723700
* ' 'F11,' '6//)' / 00723800
DATA FMT1/'(','13X','4H' D('I,J'),'4H = '/,' ' ' '(19X',' ' ', 00723900
* ' ' ' ',F11,' '6//',' 00724000
* '13X','4H' V('I,J'),'4H = '/,' ' ' '(19X',' ' ', 00724100
* ' 'F11,' '6//',' ' ) / 00724200
C 00724300
XX=0 00724400
R=II 00724500
C=JJ 00724600
C 00724700
DO 10 I=1,II 00724800
DO 20 J=1,JJ 00724900
C 00725000
C THE CHI-SQUARE VALUE 00725100
C 00725200
E(I,J)=COLUM(I)*ROW(J)/TOTAL 00725300
IF(E(I,J).EQ.0.0) GO TO 40 00725400
XX=XX+(CROSS(I,J)-E(I,J))**2/E(I,J) 00725500
C 00725600
C STANDARDIZED RESIDUALS E(I,J) 00725700
C 00725800
EE(I,J)=(CROSS(I,J)-E(I,J))/SQRT(E(I,J)) 00725900
GO TO 45. 00726000
40 EE(I,J)=0. 00726100
45 CONTINUE 00726200
C 00726300
C ESTIMATE OF THE VARIANCE OF E(I,J) 00726400
C 00726500
V(I,J)=(1.-COLUM(I)/TOTAL)*(1.-ROW(J)/TOTAL) 00726600
IF(V(I,J).EQ.0.0) GO TO 50 00726700
C 00726800
C COMPUTE AN ADJUSTED RESIDUAL D(I,J) 00726900
C 00727000
DD(I,J)=EE(I,J)/SQRT(V(I,J)) 00727100
GO TO 20 00727200
50 DD(I,J)=0. 00727300
C 00727400
20 CONTINUE 00727500
10 CONTINUE 00727600
C 00727700
XM=99999999. 00727800
XY=99999999. 00727900
IF(II.EQ.2.AND.JJ.EQ.2) CALL CHIMC (CROSS,IA,TOTAL,XY,XM,ICOUNT) 00728000
C 00728100
C COEFFICIENT OF CONTINGENCY BY PEARSON 00728200
C 00728300
PP=SQRT((XX/TOTAL)/(1.+XX/TOTAL)) 00728400
BB=XX/TOTAL 00728500
C 00728600
C KENDALL AND STUART'S COEFFICIENT 00728700
C 00728800
IF( (R-1.).EQ. 0. .OR. (C-1.).EQ. 0. ) GO TO 30 00728900
TT=(XX/TOTAL)/SQRT((R-1)*(C-1)) 00729000

```

C		00729100
C	CRAMER'S COEFFICIENT	00729200
C		00729300
	C1=AMIN1((R-1.),(C-1.))	00729400
	CC=(XX/TOTAL)/C1	00729500
	CC=SQRT(CC)	00729600
	GO TO 35	00729700
30	CONTINUE	00729800
	TT=99999999.	00729900
	CC=99999999.	00730000
35	CONTINUE	00730100
C		00730200
	IF(M1.NE.2) RETURN	00730300
C		00730400
	I1=IK(1)	00730500
	I2=IK(2)	00730600
	I3=JK(1)	00730700
	I4=JK(2)	00730800
	FMT(7)=NUM(I2+1)	00730900
	FMT(19)=NUM(I2+1)	00731000
	FMT1(7)=NUM(I2+1)	00731100
	FMT1(19)=NUM(I2+1)	00731200
	IF(IK(1).EQ.BLANK) GO TO 15	00731300
	FMT(6)=NUM(I1+1)	00731400
	FMT(18)=NUM(I1+1)	00731500
	FMT1(6)=NUM(I1+1)	00731600
	FMT1(18)=NUM(I1+1)	00731700
15	CONTINUE	00731800
	FMT(11)=NUM(I4+1)	00731900
	FMT(23)=NUM(I4+1)	00732000
	FMT1(11)=NUM(I4+1)	00732100
	FMT1(23)=NUM(I4+1)	00732200
	IF(JK(1).EQ.BLANK) GO TO 16	00732300
	FMT(10)=NUM(I3+1)	00732400
	FMT(22)=NUM(I3+1)	00732500
	FMT1(10)=NUM(I3+1)	00732600
	FMT1(22)=NUM(I3+1)	00732700
16	CONTINUE	00732800
	WRITE(IOUT,6000) XX	00732900
6000	FORMAT(10X,'PEARSON',2H'S,' CHI-SQUARE'/13X,'X2 =' ,F12.6)	00733000
	IF(I1.EQ.2.AND.JJ.EQ.2) WRITE(IOUT,6900) XY, XM	00733100
6900	FORMAT(/10X,'CHI SQUARE WITH YATES',1H', ' CONTINUITY CORRECTION',	00733200
*	/13X,'X2 =' ,G15.6//10X,'MCNEMER',2H'S,' CHI SQUARE',	00733300
*	/13X,'X2 =' ,G15.6 /1X)	00733400
	WRITE(IOUT,FMT1)((DD(I,J),J=1,JJ),I=1,II),((V(I,J),J=1,JJ),I=1,II)	00733500
	WRITE(IOUT,FMT)((EE(I,J),J=1,JJ),I=1,II),((E(I,J),J=1,JJ),I=1,II)	00733600
	WRITE(IOUT,6100) PP,BB,TT,CC	00733700
6100	FORMAT(10X,'COEFFICIENT OF CONTINGENCY (CONSISTENCY)'/	00733800
*	13X,'P =' ,F12.6/13X,'F =' ,F12.6//	00733900
*	10X,'KENDALL AND STUART',2H'S/13X,'T =' ,F12.6//	00734000
*	10X,'CRAMER',2H'S/13X,'C =' ,F12.6 ///)	00734100
C		00734200
	RETURN	00734300
	END	00734400

	SUBROUTINE	CORRLB (CROSS,COLUM,ROW,II,JJ,X,Y,TOTAL,IA,M1,R)	00734500
C				00734600
C	CALCULATION OF CORRELATION COEFFICIENT			00734700
C	COMMON	/ OUTIN / INPT, IOUT		00734800
				00734900
C	DIMENSION	CROSS(IA,IA),COLUM(IA),ROW(IA),X(IA),Y(IA)		00735000
				00735100
C				00735200
	SYY2=0			00735300
	SXX2=0			00735400
	SYY1=0			00735500
	SXX1=0			00735600
	SYX1=0			00735700
C				00735800
	DO 10 I=1,II			00735900
	DO 20 J=1,JJ			00736000
	SYX1=SYX1+CROSS(I,J)*X(J)*Y(I)			00736100
	IF(I.GE.2) GO TO 20			00736200
	SXX1=SXX1+ROW(J)*X(J)**2			00736300
	SXX2=SXX2+ROW(J)*X(J)			00736400
	20 CONTINUE			00736500
C				00736600
	SYY1=SYY1+COLUM(I)*Y(I)**2			00736700
	SYY2=SYY2+COLUM(I)*Y(I)			00736800
	10 CONTINUE			00736900
C				00737000
	SYX=SYX1-SXX2*SYY2/TOTAL			00737100
	SXX=SXX1-SXX2**2/TOTAL			00737200
	SYY=SYY1-SYY2**2/TOTAL			00737300
	IF(SXX*SYY .LT.1E-10) GO TO 30			00737400
	R=SYX/SQRT(SXX*SYY)			00737500
	GO TO 35			00737600
	30 R=9999999.			00737700
	35 CONTINUE			00737800
	BYX=SYX/SXX			00737900
	VBY=SYY/(SXX*TOTAL)			00738000
C				00738100
	IF(M1.NE.2) RETURN			00738200
C				00738300
	WRITE(IOUT,6000) SYX,SXX,SYY,R,BYX,VBY			00738400
	6000 FORMAT(10X,'S(Y,X) =',G12.6/10X,'S(X,X) =',G12.6/			00738500
	# 10X,'S(Y,Y) =',G12.6/10X,'R =',F12.6/			00738600
	# 10X,'B(Y,X) =',F12.6/10X,'VB(Y) =',F12.6)			00738700
C				00738800
	RETURN			00738900
	END			00739000

	SUBROUTINE	CATGS1 (CROSS,TOTAL,II,JJ,IA,M1,TB,TC,R,DXY,DYX)	00739100
C	COMMON	/ OUTIN / INPT,IOUT	00739200
	DIMENSION	CROSS(IA,IA)	00739300
C			00739400
	P=0		00739500
	PP=0		00739600
	Q=0		00739700
	QQ=0		00739800
	XX=0		00739900
	YY=0		00740000
	X=0		00740100
	Y=0		00740200
C			00740300
	DO 10 I=1,II		00740400
	DO 20 J=1,JJ		00740500
	IF (I.EQ.II) GO TO 35		00740600
C			00740700
	K=I+1		00740800
	DO 30 J1=K,II		00740900
	XX=XX+CROSS(I1,J)		00741000
30	CONTINUE		00741100
	X=X+XX*CROSS(I,J)		00741200
	XX=0		00741300
35	IF (J.EQ.JJ) GO TO 20		00741400
	K1=J+1		00741500
	DO 40 J1=K1,JJ		00741600
	YY=YY+CROSS(I,J1)		00741700
40	CONTINUE		00741800
	Y=Y+YY*CROSS(I,J)		00741900
	YY=0		00742000
C			00742100
	20 CONTINUE		00742200
10	CONTINUE		00742300
	K3=II-1		00742400
	K4=JJ-1		00742500
	DO 50 I=1,K3		00742600
	DO 60 J=1,K4		00742700
C			00742800
	L1=I+1		00742900
	L2=J+1		00743000
	L3=(JJ+1)-J		00743100
	DO 70 I2=L1,II		00743200
	DO 80 J2=L2,JJ		00743300
	P=P+CROSS(I2,J2)		00743400
80	CONTINUE		00743500
	L4=L3-1		00743600
	DO 90 J3=1,L4		00743700
	Q=Q+CROSS(I2,J3)		00743800
90	CONTINUE		00743900
70	CONTINUE		00744000
C			00744100
	PP=PP+CROSS(I,J)*P		00744200
	QQ=QQ+CROSS(I,L3)*Q		00744300
	P=0		00744400
	Q=0		00744500
C			00744600
	60 CONTINUE		00744700
50	CONTINUE		00744800
C			00744900
	.S=PP-QQ		00745000
C			00745100
C	KENDALL'S TAU=B		00745200
C			00745300
	TB=S/SQRT((PP+QQ+X)*(PP+QQ+Y))		00745400
C			00745500
	K=MIN0(II,JJ)		00745600
			00745700

C		00745800
C	KENDALL'S TAU-C	00745900
C		00746000
	IF((K-1) .EQ. 0) GO TO 45	00746100
	TC=(2*FLOAT(K)*S)/(TOTAL*TOTAL*FLOAT(K-1))	00746200
	GO TO 55	00746300
45	CONTINUE	00746400
	TC=9999999.	00746500
55	CONTINUE	00746600
		00746700
C		00746800
C	GOODMAN AND KRUSKAL'S GAMMA	00746900
C		00747000
	R=S/(PP+QQ)	00747100
C		00747200
C	SOMERS' D(X,Y) AND D(Y,X)	00747300
C		00747400
	DXY=S/(PP+QQ+Y)	00747500
C		00747600
	DYX=S/(PP+QQ+X)	00747700
C		00747800
	IF(M1.NE.2) RETURN	00747900
C		00748000
	WRITE(IOUT,6100)	00748100
6100	FORMAT(1H ,///)	00748200
	WRITE(IOUT,6000) TB,TC,R,DXY,DYX	00748300
6000	FORMAT(10X,'KENDALL',2H'S,' TAU (T)'/15X,	00748400
	'TAU-B =' ,F12.6,/15X,'TAU-C =' ,F12.6,///10X,	00748500
	'GOODMAN AND KRUSKAL',2H'S,' GAMMA' ,/15X,	00748600
	'GAMMA =' ,F12.6,///10X,	00748700
	'SOMERS',1H' ,/15X,	00748800
	'DYX =' ,F12.6,/15X,'DXY =' ,F12.6)	00748900
C		00749000
	RETURN	00749100
	END	

	SUBROUTINE	CATGS2 (CROSS,COLUM,ROW,I,JJ,TOTAL,IA,M1,XA,XB,X)00749200
C			00749300
C	CALCULATE GOODMAN AND KRUSKAL'S LAMBDA		00749400
C			00749500
	COMMON	/ OUTIN / INPT,IOUT	00749600
	DIMENSION	CROSS(IA,IA),COLUM(IA),ROW(IA)	00749700
C			00749800
	RMAX=0		00749900
	CMAX=0		00750000
C			00750100
	C=COLUM(1)		00750200
	DO 10 I=1,II		00750300
	A=CROSS(I,1)		00750400
	DO 20 J=2,JJ		00750500
C			00750600
	IF(A.GE.CROSS(I,J)) GO TO 20		00750700
	A=CROSS(I,J)		00750800
20	CONTINUE		00750900
	RMAX=RMAX+A		00751000
	IF(C.GE.COLUM(I)) GO TO 10		00751100
	C=COLUM(I)		00751200
10	CONTINUE		00751300
C			00751400
	D=ROW(1)		00751500
	DO 30 J=1,JJ		00751600
	B=CROSS(1,J)		00751700
	DO 40 I=2,II		00751800
C			00751900
	IF(B.GE.CROSS(I,J)) GO TO 40		00752000
	B=CROSS(I,J)		00752100
40	CONTINUE		00752200
C			00752300
	CMAX=CMAX+B		00752400
	IF(D.GE.ROW(J)) GO TO 30		00752500
	D=ROW(J)		00752600
30	CONTINUE		00752700
C			00752800
	XB=(RMAX-D)/(TOTAL-D)		00752900
	XA=(CMAX-C)/(TOTAL-C)		00753000
	X=(RMAX+CMAX-C-D)/(2*TOTAL-C-D)		00753100
C			00753200
	IF(M1.NE.2) RETURN		00753300
C			00753400
	WRITE(IOUT,6100)		00753500
6100	FORMAT(///)		00753600
	WRITE(IOUT,6000) XA,XB,X		00753700
6000	FORMAT(10X,'GOODMAN AND KRUSKAL',2H'S, /		00753800
	* 15X,'LAMBDA-A =',F12.6 /		00753900
	* 15X,'LAMBDA-B =',F12.6 /		00754000
	* 15X,'LAMBDA =',F12.6)		00754100
C			00754200
	RETURN		00754300
	END		00754400

	SUBROUTINE	CPRINT (X2,D,F,T,C,R,X,TC,RC,GT,DX,Y,XA,XB,XN,	00754500
	*	YATE,MCNE,N,COFF,MQ,Q,MM,MX,MB,MA,MQ1,MC,	00754600
	*	NUM,KN,M1,IPUNCH,ICOUNT,KA1)	00754700
			00754800
C	THIS IS AN OUTPUT-ROUTINE FOR ASSOCIATION MEASURES.		00754900
C			00755000
	COMMON	/ OUTIN / INPT,IDOUT	00755100
	REAL	MCNE	00755200
	INTEGER	Q(MB),FMT1(13),FMT2(13),FMT3(9),FMT4(9),FMT5(36)	00755300
	DIMENSION	COFF(KA1),X2(N,N),D(N,N),F(N,N),T(N,N),C(N,N),R(N,N),	00755400
	*	TB(N,N),TC(N,N),RG(N,N),DTX(N,N),XA(N,N),XB(N,N),	00755500
	*	XN(N,N),MQ(MX),MM(MA),NUM(KN),DXY(N,N),YATE(N,N),	00755600
	*	MCNE(N,N)	00755700
	DATA	FMT1/'(1X,' ','A1,' ','X,' ','(G10,'.4,1',	00755800
	*	'X),2','X','4H'* ',' ','A1)'/ ,	00755900
	*	FMT2/'(1X,' ','A1,' ','X,' ','(F10,'.6,1',	00756000
	*	'X),2','X','4H'* ',' ','A1)'/	00756100
	DATA	FMT4/'(13,'1X,8','A1,1','X,6F','10.6','/(13',	00756200
	*	'X,6F','10.6',' ')/ ,	00756300
	*	FMT3/'(13,'1X,8','A1,1','X,6G','10.4','/(13',	00756400
	*	'X,6G','10.6',' ')/ ,	00756500
	DATA	FMT5/'(1X,' ','A1,' ','X,' ','G10,'.10',1X,'	00756600
	*	'G10,'.10',1X,'G10,'.10',1X,'G10,'.10',	00756700
	*	'1X,'G10,'.10',1X,'G10,'.10',1X,'G10,'	00756800
	*	'10',1X,'G10,'.10',1X,'G10,'.10',1X,'	00756900
	*	'G10,'.10',1X,' ')/	00757000
			00757100
	DO 10 I=1,MC		00757200
	II=COFF(I)		00757300
			00757400
C	IF(I.NE.1) WRITE(IOUT,600)		00757500
			00757600
C	1 GO TO (11,22,33,44,55,66,77,88,99,111,222),II		00757700
	GO TO 10		00757800
			00757900
	11 CONTINUE		00758000
	WRITE(IOUT,601)		00758100
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT1,X2,N,Q,MB,MM,MA,M1,		00758200
	*	13,IPUNCH,FMT3)	00758300
	GO TO 10		00758400
			00758500
C	22 CONTINUE		00758600
	WRITE(IOUT,615)		00758700
	IF(ICOUNT.EQ.0) GO TO 9		00758800
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT5,YATE,N,Q,MB,MM,MA,M1,		00758900
	*	36,IPUNCH,FMT3)	00759000
	GO TO 10		00759100
			00759200
C	33 CONTINUE		00759300
	WRITE(IOUT,616)		00759400
	IF(ICOUNT.EQ.0) GO TO 9		00759500
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT5,MCNE,N,Q,MB,MM,MA,M1,		00759600
	*	36,IPUNCH,FMT3)	00759700
	GO TO 10		00759800
			00759900
C	44 CONTINUE		00760000
	WRITE(IOUT,602)		00760100
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,D,N,Q,MB,MM,MA,M1,		00760200
	*	13,IPUNCH,FMT4)	00760300
	GO TO 10		00760400
			00760500
C	55 CONTINUE		00760600
	WRITE(IOUT,603)		00760700
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,F,N,Q,MB,MM,MA,M1,		00760800
	*	13,IPUNCH,FMT4)	00760900
	GO TO 10		00761000

C	66 CONTINUE	00761100
	WRITE(IOUT,604)	00761200
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,T,N,Q,MB,MM,MA,M1,	00761300
	* 13,IPUNCH,FMT4)	00761400
	WRITE(IOUT,605)	00761500
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,C,N,Q,MB,MM,MA,M1,	00761600
	* 13,IPUNCH,FMT4)	00761700
	GO TO 10	00761800
C	77 CONTINUE	00761900
	WRITE(IOUT,606)	00762000
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,R,N,Q,MB,MM,MA,M1,	00762100
	* 13,IPUNCH,FMT4)	00762200
	GO TO 10	00762300
C	88 CONTINUE	00762400
	WRITE(IOUT,607)	00762500
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,TB,N,Q,MB,MM,MA,M1,	00762600
	* 13,IPUNCH,FMT4)	00762700
	WRITE(IOUT,608)	00762800
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,TC,N,Q,MB,MM,MA,M1,	00762900
	* 13,IPUNCH,FMT4)	00763000
	GO TO 10	00763100
C	99 CONTINUE	00763200
	WRITE(IOUT,609)	00763300
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,RG,N,Q,MB,MM,MA,M1,	00763400
	* 13,IPUNCH,FMT4)	00763500
	GO TO 10	00763600
C	111 CONTINUE	00763700
	WRITE(IOUT,610)	00763800
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,DTX,N,Q,MB,MM,MA,M1,	00763900
	* 13,IPUNCH,FMT4)	00764000
	WRITE(IOUT,6000)	00764100
	WRITE(IOUT,614)	00764200
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,DX,N,Q,MB,MM,MA,M1,	00764300
	* 13,IPUNCH,FMT4)	00764400
	WRITE(IOUT,6000)	00764500
	GO TO 10	00764600
C	222 CONTINUE	00764700
	WRITE(IOUT,611)	00764800
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,XA,N,Q,MB,MM,MA,M1,	00764900
	* 13,IPUNCH,FMT4)	00765000
	WRITE(IOUT,612)	00765100
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,XB,N,Q,MB,MM,MA,M1,	00765200
	* 13,IPUNCH,FMT4)	00765300
	WRITE(IOUT,613)	00765400
	CALL CROPRI (MQ,MX,MQ1,NUM,KN,FMT2,XN,N,Q,MB,MM,MA,M1,	00765500
	* 13,IPUNCH,FMT3)	00765600
	GO TO 10	00765700
C	9 WRITE(IOUT,617)	00765800
C	10 CONTINUE	00765900
	RETURN	00766000
		00766100
		00766200
		00766300
		00766400
		00766500
		00766600
		00766700
		00766800

```

C
600 FORMAT(1H1) 00766900
601 FORMAT(1H ,///1H ,9('.'),'<< PEARSON',2H'S,' CHI-SQUARE ( X2 ) ',00767000
* ' >>',64('.'),'( LIST 1 )',10('.')) ///) 00767100
602 FORMAT(1H ,///1H ,9('.'),'<< COEFFICIENT OF CONTINGENCY ( P ) >',00767200
* '>',59('.'),'( LIST 1 )',10('.')) ///) 00767300
603 FORMAT(1H ,///1H ,9('.'),'<< COEFFICIENT OF CONTINGENCY ( PHI ',00767400
* ' SQUARE ) >>',50('.'),'( LIST 1 )',10('.')) ///) 00767500
604 FORMAT(1H ,///1H ,9('.'),'<< KENDALL AND STUART',2H'S,' ( T ) >>',00767600
* '66('.'),'( LIST 1 )',10('.')) ///) 00767700
605 FORMAT(1H1,///1H ,9('.'),'<< CRAMER',2H'S,' ( PHI ) >>',00767800
* '76('.'),'( LIST 1 )',10('.')) ///) 00767900
606 FORMAT(1H ,///1H ,9('.'),'<< CORRELATION COEFFICIENT >>',00768000
* '69('.'),'( LIST 1 )',10('.')) ///) 00768100
607 FORMAT(1H ,///1H ,9('.'),'<< KENDALL',2H'S,' TAU ( TAU-B ) >>',00768200
* '69('.'),'( LIST 1 )',10('.')) ///) 00768300
608 FORMAT(1H1,///1H ,9('.'),'<< KENDALL',2H'S,' TAU ( TAU-C ) >>',00768400
* '69('.'),'( LIST 1 )',10('.')) ///) 00768500
609 FORMAT(1H ,///1H ,9('.'),'<< GOODMAN AND KRUSKAL',2H'S,' GAMMA',00768600
* ' >>',64('.'),'( LIST 1 )',10('.')) ///) 00768700
610 FORMAT(1H ,///1H ,9('.'),'<< SOMERS',1H', ' COEFFICIENT D(Y,X) ',00768800
* ' >>',65('.'),'( LIST 1 )',10('.')) ///) 00768900
611 FORMAT(1H ,///1H ,9('.'),'<< GOODMAN AND KRUSKAL',2H'S,' LAMBDA-',00769000
* 'A >>',62('.'),'( LIST 1 )',10('.')) ///) 00769100
612 FORMAT(1H1,///1H ,9('.'),'<< GOODMAN AND KRUSKAL',2H'S,' LAMBDA-',00769200
* 'B >>',62('.'),'( LIST 1 )',10('.')) ///) 00769300
613 FORMAT(1H1,///1H ,9('.'),'<< GOODMAN AND KRUSKAL',2H'S,' LAMBDA ',00769400
* ' >>',64('.'),'( LIST 1 )',10('.')) ///) 00769500
614 FORMAT(1H1,///1H ,9('.'),'<< SOMERS',1H', ' COEFFICIENT D(X,Y) ',00769600
* ' >>',65('.'),'( LIST 1 )',10('.')) ///) 00769700
615 FORMAT(1H ,///1H ,9('.'),'<< CHI SQUARE WITH YATES',1H',00769800
* ' CONTINUITY CORRECTION >>',48('.'),'( LIST 1 )',10('.')) ///) 00769900
616 FORMAT(1H ,///1H ,9('.'),'<< MCNEMER',2H'S,' CHI SQUARE >>',00770000
* '71('.'),'( LIST 1 )',10('.')) ///) 00770100
617 FORMAT(///5X,'--- ALL TABLES ARE NOT (2*2) DIMENSION. ---') 00770200
6000 FORMAT(///10X,'--- EACH COLUMN IN THE ABOVE TABLE IS USED',00770300
* ' AS DEPENDENT VARIABLES.',/15X,'THAT IS SAID ',00770400
* ', EACH COLUMN IS HEAD OF TABLES. ---' ) 00770500
END 00770600
00770700

```

C	SUBROUTINE	CROPRI (MQ,MX,MQ1,NUM,KN,FMT,X2,N,Q,MB,MM,MA,M1,	00770800
	*	MZ,IPUNCH,FMT3)	00770900
C			00771000
C	THIS SUBROUTINE	IMPLEMENTS THE PRINT OF ASSOCIATION MEASURES.	00771100
C			00771200
	COMMON	/ OUTIN / INPT,IOUT	00771300
	INTEGER	FMT(MZ),FMT1(5),FMT2(8),Q(MB),QQ(50,8),QQQ(10,8),	00771400
	*	FMT3(9),BLANK	00771500
	DIMENSION	MQ(MX),MM(MA),X2(N,N),NUM(KN),LL1(50),LL2(50)	00771600
C			00771700
	DATA	FMT1/'(13X',4H,'*',4H',',',',A1)'/,	00771800
	*	BLANK/' ',K/ 4H,'*',/	00771900
	DATA	FMT2/'(1X',',',',4H('(',',',8A1',4H',',',',8A1',	00772000
	*	4H,')',',')'/	00772100
	ERR0=9999999.		00772200
	KK=1		00772300
	J1=1		00772400
C			00772500
	DO 90 II=1,50		00772600
	DO 90 JJ=1,8		00772700
	QQ(II,JJ)=BLANK		00772800
	90 CONTINUE		00772900
C			00773000
C			00773100
C	10 CONTINUE		00773200
C			00773300
	J2=0		00773400
	L1=1		00773500
	KI=MQ(J1)		00773600
	CALL PITEM (MM,MA,M1,KI,L1,L2,KP)		00773700
	M9=L2-L1+1		00773800
	FMT1(4)=NUM(M9+1)		00773900
C			00774000
	WRITE(IOUT,FMT1) (Q(L),L=L1,L2)		00774100
C			00774200
	LL=L1		00774300
	DO 15 I=1,M9		00774400
	QQ(J1,I)=Q(LL)		00774500
	15 LL=LL+1		00774600
C			00774700
	J=MQ1-1		00774800
	DO 20 I=J1,J		00774900
C			00775000
	FMT(2)=BLANK		00775100
	FMT(4)=BLANK		00775200
	IF(MZ.EQ.36) GO TO 14		00775300
	FMT(6)=BLANK		00775400
	FMT(12)=BLANK		00775500
	14 CONTINUE		00775600
C			00775700
	J2=J2+1		00775800
	L1=1		00775900
	KJ=MQ(I+1)		00776000
	CALL PITEM (MM,MA,M1,KJ,L1,L2,KP)		00776100
	M8=L2-L1+1		00776200
	FMT(2)=NUM(M8+1)		00776300
	IF(MZ.EQ.13) FMT(12)=NUM(M8+1)		00776400
	M7=10-M8		00776500
	FMT(4)=NUM(M7+1)		00776600
C			00776700
	I1=I+1		00776800
	LL=L1		00776900
	DO 16 NN=1,M8		00777000
	QQ(I1,NN)=Q(LL)		00777100
	16 LL=LL+1		00777200
C			00777300
	N1=J2		00777400
	IF(N1.GT. 9) N1=9		00777500
	IF(MZ.EQ.13) FMT(6)=NUM(N1+1)		00777600
C			00777700
	JJ=I		00777800
	IF(JJ.GT.9*KK) JJ=9*KK		00777900
C			00778000
	IF(MZ.EQ.13) GO TO 17		00778100
	CALL PRIXX (X2,FMT,Q,MB,I,J,JJ,J1,L1,L2,M8,NUM,KN,N,KZ,KK,I1)	00778200
	GO TO 20		00778300
			00778400

C		00778500
17	LI=0	00778600
	DO 50 LO=J1,JJ	00778700
	IF(X2(I1,LO).NE.ERRD) GO TO 50	00778800
	LI=LI+1	00778900
	LL1(LI)=I1	00779000
	LL2(LI)=LO	00779100
50	CONTINUE	00779200
C		00779300
	IF(I.EQ.J) GO TO 30	00779400
	IF(JJ.EQ.9*KK) GO TO 30	00779500
	WRITE(IOUT,FMT) (Q(L),L=L1,L2),(X2(I1,M),M=J1,JJ),(Q(L),L=L1,L2)	00779600
	GO TO 20	00779700
C		00779800
30	CONTINUE	00779900
	FMT(11)=BLANK	00800000
	WRITE(IOUT,FMT) (Q(L),L=L1,L2),(X2(I1,M),M=J1,JJ)	00800100
	FMT(11)=K	00800200
20	CONTINUE	00800300
	IF((MQ1-1).LE.9*KK) GO TO 40	00800400
C		00800500
	KK=KK+1	00800600
	J1=J1+9	00800700
C		00800800
	WRITE(IOUT,6000) KK	00800900
6000	FORMAT(///1H ,9(' - '), '(CONTINUED) ',86(' - '), '(LIST ',12,' '),	00801000
*	10(' - ') ///)	00801100
C		00801200
	GO TO 10	00801300
C		00801400
40	WRITE(IOUT,6100)	00801500
6100	FORMAT(///1H ,128(' . '))	00801600
C		00801700
C	PUNCH-OPTION FOR ASSOCIATION MEASURES	00801800
C		00801900
	IF(IPUNCH.NE.1) GO TO 70	00802000
	IF(MZ.EQ.13) GO TO 51	00802100
	CALL CARDXX (X2, N, MQ1, QG)	00802200
	GO TO 70	00802300
C		00802400
51	CONTINUE	00802500
	DO 60 I=1,MQ1	00802600
	WRITE(7,FMT3) I,(QG(I,L),L=1,8),(X2(I,NM),NM=1,MQ1)	00802700
60	CONTINUE	00802800
70	IF(L1.EQ.0) RETURN	00802900
C		00803000
	WRITE(IOUT,6400)	00803100
6400	FORMAT(///1H , '--- IT IS IMPOSSIBLE TO COMPUTE AN ASSOCIATION',	00803200
*	'MEASURE FOR THE FOLLOWING PAIRS. ---',/)	00803300
C		00803400
	KJ=1	00803500
	NI=1	00803600
	NJ=LI	00803700
C		00803800
25	CONTINUE	00803900
	IF(NJ.GT.5*KJ) NJ=5*KJ	00804000
	IJK=NJ-NI+1	00804100
	FMT2(2)=NUM(IJK+1)	00804200
C		00804300
	N4=0	00804400
	DO 35 N3=NI,NJ	00804500
	N4=N4+1	00804600
	N5=N4+1	00804700
	M1=LL1(N3)	00804800
	M2=LL2(N3)	00804900
C		00805000
	DO 36 MMM=1,8	00805100
	QQQ(N4,MMM)=QQ(M1,MMM)	00805200
	QQQ(N5,MMM)=QQ(M2,MMM)	00805300
36	CONTINUE	00805400
35	CONTINUE	00805500
C		00805600
	WRITE(IOUT,FMT2)((QQQ(I,J),J=1,8),I=1,N5)	00805700
C		00805800
	IF(NJ.GE.LI) RETURN	00805900
	NI=NJ+1	00806000
	NJ=LI	00806100
	KJ=KJ+1	00806200
	GO TO 25	00806300
C		00806400
C		00806500
	END	00806600

	SUBROUTINE	CARDXX (X2, N, MQ1, QQ	00786700
C			00786800
C	PRERARE THE CARD	OUTPUT OPTION FOR THE MATRIX OF ASSOCIATION MEASURES	00786900
			00787000
	INTEGER	FMS1(17),FMS2(15),TABLE(6),TABLE1(6),TABLE2(2)	00787100
	DIMENSION	X2(N,N),QQ(50,8)	00787200
	DATA	TABLER/6,8,10,12,14,16/,	00787300
	*	TABLER/4,6,8,10,12,14/,TABLE2/'4','10','/',	00787400
	*	FMS1/'(I3','1X,8','A1,1','X','G10.','10','G10.',	00787500
	*	'10','G10.','10','G10.','10','G10.','10',	00787600
	*	'G10.','10','')/',	00787700
	*	FMS2/'(I3X','','G10.','10','G10.','10','G10.',	00787800
	*	'10','G10.','10','G10.','10','G10.','10',	00787900
	*	'') /	00788100
C			00788200
	ERR0=9999999.		00788300
C			00788400
	DO 10 I=1,MQ1		00788500
	L1=1		00788600
C	5 CONTINUE		00788700
	L2=MQ1		00788800
	IF(L2.GT.6) L2=L1+5		00788900
C			00789000
	J5=0		00789100
	DO 18 II=L1,L2		00789200
	J5=J5+1		00789300
	IF(X2(I,II).EQ.ERRO) GO TO 18		00789400
C			00789500
	IF(L1.NE.1) GO TO 19		00789600
	M5=TABLE(J5)		00789700
	FMS1(M5)=TABLE2(1)		00789800
	GO TO 18		00789900
C			00790000
	19 M5=TABLE1(J5)		00790100
	FMS2(M5)=TABLE2(1)		00790200
C	18 CONTINUE		00790300
			00790400
	IF(L1.NE.1) GO TO 15		00790500
C			00790600
	WRITE(7,FMS1) I,(QQ(I,L),L=1,8),(X2(I,NM),NM=L1,L2)		00790700
C			00790800
	DO 21 J=1,6		00790900
	M5=TABLE(J)		00791000
	21 FMS1(M5)=TABLE2(2)		00791100
	GO TO 20		00791200
C			00791300
	15 WRITE(7,FMS2) (X2(I,NM),NM=L1,L2)		00791400
C			00791500
	DO 22 J=1,6		00791600
	M5=TABLE1(J)		00791700
	22 FMS2(M5)=TABLE2(2)		00791800
C			00791900
	20 IF(L2.GE.MQ1) GO TO 10		00792000
	L1=L2+1		00792100
	GO TO 5		00792200
C			00792300
	10 CONTINUE		00792400
C			00792500
	RETURN		00792600
	END		00792700

	SUBROUTINE CHIMC (CROSS, IA, TOTAL, XW, XM, ICOUNT)	00792800
C		00792900
C	CALCULATE THE MEASURES FOR (2 X 2) TABLES.	00793000
C	(YATES' AND MCNEMER'S COEFFICIENTS)	00793100
C		00793200
	COMMON / OUTIN / INPT, IOUT	00793300
	REAL*8 A,B,C,D,XX1,XX2,AIJ	00793400
	DIMENSION CROSS(IA,IA)	00793500
C		00793600
	ICOUNT=ICOUNT+1	00793700
C		00793800
	AIJ=TOTAL	00793900
	A=CROSS(1,1)	00794000
	B=CROSS(1,2)	00794100
	C=CROSS(2,1)	00794200
	D=CROSS(2,2)	00794300
	XM=0	00794400
	XW=0	00794500
C		00794600
C	CHI SQUARE WITH YATES' CONTINUITY CORRECTION	00794700
C		00794800
	XX1=AIJ*(DABS(A*D-B*C)-0.5D00*AIJ)**2	00794900
	XX2=(A+B)*(C+D)*(A+C)*(B+D)	00795000
	XW=XX1/XX2	00795100
C		00795200
C	MCNEMER'S CHI SQUARE	00795300
C		00795400
	XM=(DABS(A-D)-1.0D00)**2/(A+D)	00795500
C		00795600
	RETURN	00795700
	END	00795800

```

SUBROUTINE PRIXX (X2,FMT,Q,MB,I,J,JJ,J1,L1,L2,M8,NUM,KN,N,KZ, 00795900
* KK,I1 ) 00796000
C PRINT YATES' COEFFICIENT AND MCNEMER'S COEFFICIENTS. 00796100
C 00796200
C 00796300
COMMON / OUTIN / INPT,IOUT 00796400
INTEGER FMT(KZ),Q(MB),FMT4(8),BLANK,TABLE(10),TABLE1(2) 00796500
DIMENSION NUM(KN),X2(N,N) 00796600
DATA FMT4/'1H',' ',' ',' ','4HX','*','4H ',' ',' ','A1')/, 00796700
* TABLE/7,10,13,16,19,22,25,28,31,34/, 00796800
* TABLE1/'4','10',BLANK/' '/ 00796900
C 00797000
ERRO=9999999. 00797100
J5=0 00797200
DO 18 I1=J1,JJ 00797300
J5=J5+1 00797400
IF(X2(I1,I1).EQ.ERRO) GO TO 18 00797500
M5=TABLE(J5) 00797600
FMT(M5)=TABLE1(1) 00797700
18 CONTINUE 00797800
C 00797900
WRITE(IOUT,FMT) (Q(L),L=L1,L2),(X2(I1,M),M=J1,JJ) 00798000
C 00798100
IF(I.EQ.J) GO TO 19 00798200
IF(JJ.EQ.9*KK) GO TO 19 00798300
FMT4(7)=NUM(M8+1) 00798400
KL=(JJ-J1+2)*11 00798500
CALL FORM(NUM,KN,FMT4,KL,3,4,8) 00798600
C 00798700
WRITE(IOUT,FMT4) (Q(L),L=L1,L2) 00798800
C 00798900
FMT4(3)=BLANK 00799000
FMT4(4)=BLANK 00799100
19 CONTINUE 00799200
C 00799300
DO 21 IJ1=1,10 00799400
M5=TABLE(IJ1) 00799500
21 FMT(M5)=TABLE1(2) 00799600
C 00799700
RETURN 00799800
END 00799900

SUBROUTINE RANGE (XX,Q,MM,MQ,NO,NOMIT,ICOL,MS,NS,A,B,FT,MA,MB,MX,NN00800000
* ,KN,N,M1,J1,BLANK,INDEV,OUTDEV,WORK,IWORK,LIMIT1,00800100
* LIMIT2 ) 00800200
C 00800300
COMMON NUMDT(20) 00800400
COMMON / OUTIN / INPT, IOUT 00800500
INTEGER Q(MB),FT(KN),BLANK,OUTDEV 00800600
DIMENSION XX(MA),MM(MA),MQ(MX),NO(MX),NOMIT(NN),ICOL(NN),MS(MX), 00800700
* NS(NN),A(MX),B(MX),WORK(LIMIT1),IWORK(LIMIT2) 00800800
C 00800900
I11=1 00801000
I12=I11+MX*NN 00801100
I13=I12+MX*NN 00801200
I14=I13+MX*NN-1 00801300
C 00801400
NN1=1 00801500
NN2=NN1+MX*NN 00801600
NN3=NN2+MX*NN 00801700
NN4=NN3+MX*NN-1 00801800
C 00801900
IF(I14.GT.LIMIT1.OR.NN4.GT.LIMIT2) GO TO 10 00802000
C 00802100
CALL SBRANG(XX,Q,MM,MQ,NO,NOMIT,ICOL,MS,NS,IWORK(NN1),IWORK(NN2), 00802200
* IWORK(NN3),A,B,WORK(I11),WORK(I12),WORK(I13), 00802300
* FT,MA,MB,MX,NN,KN,N,M1,J1,BLANK,INDEV,OUTDEV ) 00802400
C 00802500
RETURN 00802600
C 00802700
10 CONTINUE 00802800
WRITE(IOUT,600) 00802900
600 FORMAT(1H1,///1X,'+-----+ '/1X,'I RANGE CHECK 1', 00803000
* /1X,'+-----+') 00803100
WRITE(IOUT,610) I14,NN4 00803200
610 FORMAT(///1H,'USED WORKING AREA WORK(1) ~ WORK(' ,I6,' ) .' / 00803300
* 1H,'USED WORKING AREA IWORK(1) ~ IWORK(' ,I6,' ) .' ) 00803400
RETURN 00803500
END 00803600

```

C	SUBROUTINE CATGYA (XX,INDEV,MA,X,Y,MQ,MX,M1,N,MQ1,IA,IERO)	00803700
C	CALCULATE THE CATEGORIES AND THEIR NUMBERS.	00803800
C	COMMON / OUTIN / INPT,IOUT	00803900
C	DIMENSION XX(MA),MQ(MX),X(N,IA),Y(N)	00804000
C	DO 4 I=1,MQ1	00804100
C	4 Y(I)=1	00804200
C	DETECT THE CATEGORIES AND COUNT NUMBERS OF EACH CATEGORY	00804300
C	REWIND INDEV	00804400
C	READ(INDEV,END=99,ERR=999) KK,(XX(J),J=1,M1)	00804500
C	DO 5 I=1,MQ1	00804600
C	I2=MQ(I)	00804700
C	X(I,1)=XX(I2)	00804800
C	5 CONTINUE	00804900
C	10 CONTINUE	00805000
C	READ(INDEV,END=99,ERR=999) KK,(XX(J),J=1,M1)	00805100
C	DO 15 K=1,MQ1	00805200
C	I2=MQ(K)	00805300
C	K3=Y(K)	00805400
C	DO 20 K1=1,K3	00805500
C	IF(X(K,K1).EQ.XX(I2)) GO TO 25	00805600
C	20 CONTINUE	00805700
C	Y(K)=Y(K)+1	00805800
C	X(K,K3+1)=XX(I2)	00805900
C	25 CONTINUE	00806000
C	15 CONTINUE	00806100
C	GO TO 10	00806200
C	99 CONTINUE	00806300
C	DO 36 I=1,MQ1	00806400
C	L=Y(I)-1	00806500
C	LM=Y(I)	00806600
C	DO 40 L1=1,L	00806700
C	LL=L1+1	00806800
C	DO 45 L2=LL,LM	00806900
C	IF(X(I,L1).LE.X(I,L2)) GO TO 45	00807000
C	A=X(I,L1)	00807100
C	X(I,L1)=X(I,L2)	00807200
C	X(I,L2)=A	00807300
C	45 CONTINUE	00807400
C	40 CONTINUE	00807500
C	36 CONTINUE	00807600
C	RETURN	00807700
C	999 WRITE(IOUT,900)	00807800
C	IERO=1	00807900
C	RETURN	00808000
C	900 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00808100
C	* 'SPECIFIED BY COMMAND ---')	00808200
C	END	00808300
		00808400
		00808500
		00808600
		00808700
		00808800
		00808900
		00809000
		00809100
		00809200
		00809300
		00809400
		00809500

```

      SUBROUTINE CROSSA(XX,MA,MQ,MQ1,MQH,MQ2,MM,G,MX,MB,WORK,IWORK,
      *                LIMIT1,LIMIT2,INDEV,M1,ITWO,KMZ,MM1,MM2,NN1,
      *                NUM,KN )
C
C   THIS SUBROUTINE GENERATES THE CROSS-CLASSIFIED TABLES AND CARRIES
C   OUT THE PERCENT-TEST.
C
      COMMON          NUMDT(20)
      COMMON          /OUTIN/INPT,IOUT
      INTEGER          Q(MB)
      DIMENSION        XX(MA),MQ(MX),WORK(LIMIT1),IWORK(LIMIT2),MM(MA) ,
      *                NUM(KN),MQH(MX)
C
      WRITE(IOUT,6000)
6000  FORMAT(1H1,///1H ,'+-----+',/1X,
      *      'I CROSS CLASSIFIED TABLES I',/1X,
      *      'I AND I',/1X,
      *      'I PERCENTILE TEST I',/1X,
      *      '+-----+')
C
      KNEW=NUMDT(INDEV)
      IF(KNEW.EQ.0) GO TO 1000
C
      I1=200
      I11=180
      I12=180
      I2=I11*I12
      I3=50
      I4=400
      I15=20
      I5=I15*I3
      I6=15
      I7=I6*I6*I6
C
      N1=1
      N4=N1+I1
      N5=N4+I2
      N6=N5+I3
      N7=N6+I3
      N8=N7+I7
      N9=N8+I15
      N10=N9+I15
      N11=N10+I15-1
C
      MN1=1
      M2=MN1+I4
      M3=M2+I4
      M4=M3+I3
      M5=M4+I3
      M6=M5+I5
      M7=M6+I1
      M8=M7+I15
      M9=M8+I15
      M10=M9+I15
      M11=M10+I15
      M12=M11+I4
      M13=M12+I4-1
C
      IF(LIMIT1.LT.M13. OR .LIMIT2.LT.N11) GO TO 99
      CALL MSORT(XX,MA,MQ,MQ1,MQH,MQ2,MM,G,MX,MB,INDEV,ITWO,M1,
      *          WORK(MN1),WORK(M2),WORK(M3),WORK(M4),WORK(M5),WORK(M6),
      *          WORK(M7),WORK(M8),WORK(M9),WORK(M10),WORK(M11),
      *          WORK(M12),IWORK(N1),IWORK(N4),IWORK(N5),
      *          IWORK(N6),IWORK(N7),IWORK(N8),IWORK(N9),IWORK(N10),
      *          I1,I11,I12,I3,I15,I6,KMZ,MM1,MM2,NN1,NUM,KN )
C
      RETURN
C
99  WRITE(IOUT,6100) M7,N7
6100  FORMAT(///10X,'USED WORKING AREA WORK(1) ~ WORK('I6,')',
      *      /10X,'USED WORKING AREA IWORK(1) ~ IWORK('I6,')' )
      RETURN
C
1000  WRITE(IOUT,6200) INDEV
6200  FORMAT(///1H ,'--- FILE NUMBER IS',I3,'. THIS FILE DOES NOT ',
      *      ' INCLUDE DATA ---' )
      RETURN
      END

```

00809600
 00809700
 00809800
 00809900
 00810000
 00810100
 00810200
 00810300
 00810400
 00810500
 00810600
 00810700
 00810800
 00810900
 00811000
 00811100
 00811200
 00811300
 00811400
 00811500
 00811600
 00811700
 00811800
 00811900
 00812000
 00812100
 00812200
 00812300
 00812400
 00812500
 00812600
 00812700
 00812800
 00812900
 00813000
 00813100
 00813200
 00813300
 00813400
 00813500
 00813600
 00813700
 00813800
 00813900
 00814000
 00814100
 00814200
 00814300
 00814400
 00814500
 00814600
 00814700
 00814800
 00814900
 00815000
 00815100
 00815200
 00815300
 00815400
 00815500
 00815600
 00815700
 00815800
 00815900
 00816000
 00816100
 00816200
 00816300
 00816400
 00816500
 00816600
 00816700
 00816800
 00816900
 00817000
 00817100
 00817200
 00817300

```

SUBROUTINE MSORT(XX,MA,MQ,MQ1,MQH,MQ2,MM,Q,MX,MB,INDEV,ITWO,M1, 00817400
* HYODAI,FHYODA,PT,TPAR,P,IData,IFSC1,IFSC2,YY1, 00817500
* YY2,XX1,XX2,ITEM,IA,TOTALR,TOTALC,IFC,FACE, 00817600
* COUNT1,COUNT2,I1,I11,I12,I3,I15,I6,KMZW,MM1, 00817700
* MM2,NN1,NUM,KN ) 00817800
C 00817900
COMMON /OUTIN/INPT,IOUT 00818000
C 00818100
REAL*8 HYODAI(I1),FHYODA(I1) 00818200
INTEGER Q(MB),TOTALR(I3),TOTALC(I3), 00818300
* COUNT1(I15),COUNT2(I15),FACE(I15) 00818400
REAL IDATA(I1),IFSC1(I15),IFSC2(I15) 00818500
DIMENSION XX(MA),MQ(MX),MM(MA),NUM(KN),ITEM(I1),IA(I11,I12), 00818600
* PT(I3),TPAR(I3),P(I3,I15),XX1(I15,I15),XX2(I15,I15), 00818700
* YY1(I15),YY2(I15),IFC(I6,I6,I6),MQH(MX) 00818800
C 00818900
GO TO (1,2),ITWO 00819000
RETURN 00819100
C 00819200
C GENERATE ONEWAY CROSS-CLASSIFIED TABLES 00819300
C 00819400
1 CONTINUE 00819500
WRITE(IOUT,6300) 00819600
6300 FORMAT(/1H0,5X,' TWO-WAY TABLES ',/6X,16(' ') // ) 00819700
C 00819800
CALL ONEWAY(XX,MA,MQ,MQ1,MQH,MQ2,MM,Q,MX,MB,INDEV,ITWO,M1, 00819900
* HYODAI,FHYODA,PT,TPAR,P,IData,ITEM,IA,TOTALR,TOTALC, 00820000
* I1,I11,I12,I3,I15,I6,KMZW,MM1,MM2,NN1,COUNT1,COUNT2, 00820100
* FACE,IFSC1,IFSC2,IFC,XX1,XX2,YY1,YY2 ) 00820200
C 00820300
RETURN 00820400
C 00820500
C GENERATE TWOWAY CROSS-CLASSIFIED TABLES 00820600
C 00820700
2 CONTINUE 00820800
WRITE(IOUT,6400) 00820900
6400 FORMAT(/1H0,5X,' MULTI-WAY TABLES ',/6X,18(' ') // ) 00821000
C 00821100
CALL TWOWAY(XX,MA,MQ,MQ1,MQH,MQ2,MM,Q,MX,MB,INDEV,ITWO,M1, 00821200
* HYODAI,FHYODA,PT,TPAR,P,IData,ITEM,IA,TOTALR,TOTALC, 00821300
* I1,I11,I12,I3,I15,I6,KMZW,MM1,MM2,NN1,COUNT1,COUNT2, 00821400
* FACE,IFSC1,IFSC2,IFC,XX1,XX2,YY1,YY2,NUM,KN ) 00821500
C 00821600
RETURN 00821700
END 00821800

```

C	SUBROUTINE ONEWAY(XX,MA,MQ,MQ1,MQH,MQ2,MM,Q,MX,MB,INDEV,ITWO,M1,	00821900
*	HYODAI,FHYODA,PT,TPAR,P,IDATA,ITEM,IA,TOTALR,	00822000
*	TOTALC,I1,I11,I12,I3,I15,I6,KMZ,MM1,MM2,NN1,	00822100
*	COUNT1,COUNT2,FACE,IFSC1,IFSC2,IFC,XX1,XX2,YY1,	00822200
*	YY2)	00822300
		00822400
C	GENERATE THE ONEWAY-CROSS TABLES	00822500
C		00822600
C	COMMON / OUTIN /INPT,IOUT	00822700
		00822800
C	REAL*8 HYODAI(I1),FHYODA(I1),I11	00822900
REAL	IDATA(I1),IFSC1(I15),IFSC2(I15)	00823000
INTEGER	Q(MB),TOTALR(I3),TOTALC(I3),	00823100
*	COUNT1(I15),COUNT2(I15),FACE(I15)	00823200
DIMENSION	XX(MA),MQ(MX),MM(MA),	00823300
*	ITEM(I1),IA(I11,I12),PT(I3),TPAR(I3),P(I3,I15),	00823400
*	XX1(I15,I15),XX2(I15,I15),YY1(I15),YY2(I15),	00823500
*	MQH(MX),IFC(I6,I6,I6)	00823600
DATA	I11/' '	00823700
		00823800
C	DO 3 JJJ=1,I1	00823900
	HYODAI(JJJ)=I11	00824000
	FHYODA(JJJ)=I11	00824100
3	CONTINUE	00824200
		00824300
C	KL=MQ2	00824400
	DO 10 I=1,MQ1	00824500
	DO 14 J=1,MQ2	00824600
	IF(MQ(I).EQ.MQH(J)) GO TO 10	00824700
14	CONTINUE	00824800
	KL=KL+1	00824900
	MQH(KL)=MQ(I)	00825000
	GO TO 10	00825100
		00825200
C	10 CONTINUE	00825300
	N=KL	00825400
	LL=MQ1	00825500
		00825600
C	DO 16 II=1,MQ1	00825700
	DO 17 I=1,KL	00825800
	IF(MQ(II).NE.MQH(I)) GO TO 17	00825900
	FACE(II)=1	00826000
17	CONTINUE	00826100
16	CONTINUE	00826200
		00826300
C	DO 50 IJ=1,KL	00826400
	L1=1	00826500
	K1=MQH(IJ)	00826600
	CALL PITEM(MM,MA,M1,K1,L1,L2,KP)	00826700
	K2=0	00826800
	DO 60 L=L1,L2	00826900
	K2=K2+1	00827000
	CALL MOVEC(HYODAI(IJ),K2,Q(L),1,1)	00827100
60	CONTINUE	00827200
50	CONTINUE	00827300
		00827400
C	DO 70 IJ=1,MQ1	00827500
	L1=1	00827600
	K1=MQ(IJ)	00827700
	CALL PITEM(MM,MA,M1,K1,L1,L2,KS)	00827800
	K2=0	00827900
	DO 80 L=L1,L2	00828000
	K2=K2+1	00828100
	CALL MOVEC(FHYODA(IJ),K2,Q(L),1,1)	00828200
80	CONTINUE	00828300
70	CONTINUE	00828400
		00828500
C	CALL CATGYA(XX,INDEV,MA,XX1,YY1,MQH,10,M1,20,KL,20,IERO)	00828600
	CALL CATGYA(XX,INDEV,MA,XX2,YY2,MQ,MX,M1,20,MQ1,20,IERO)	00828700
	IF(IERO.EQ.1) GO TO 9999	00828800
		00828900
C	CALL CROSSB(XX,MA,INDEV,M1,ITWO,TOTALR,TOTALC,MQ,MQH,MX,PT,	00829000
*	TPAR,P,IDATA,HYODAI,FHYODA,ITEM,IA,I1,I11,I12,I3,	00829100
*	I15,N,LL,FACE,IDD1,IDD2,ICR,XX1,XX2,YY1,YY2,IFC,I6,	00829200
*	IFSC1,IFSC2,KMZ,MM1,MM2,NN1,COUNT1,COUNT2,MQ2)	00829300
		00829400
C	RETURN	00829500
		00829600
C	9999 WRITE(IOUT,6000)	00829700
	6000 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00829800
*	'SPECIFIED BY COMMAND ---')	00829900
		00830000
	RETURN	00830100
	END	00830200

C		00830300
	SUBROUTINE TWOWAY(XX,MA,MQ,MQ1,MQH,MQ2,MM,Q,MX,MB,INDEV,ITWO,M1,	00830400
*	HYODAI,FHYODA,PT,TPAR,P,IData,ITEM,IA,TOTALR,	00830500
*	TOTALC,I1,I11,I12,I3,I15,I6,KMZ,MM1,MM2,NN1,	00830600
*	COUNT1,COUNT2,FACE,IFSC1,IFSC2,IFC,XX1,XX2,YY1,	00830700
*	YY2,NUM,KN)	00830800
C		00830900
C	GENERATE THE TWO-WAY CROSS TABLES WITH BREAK-DOWN.	00831000
C		00831100
	COMMON / OUTIN /INPT,IOUT	00831200
C		00831300
	REAL*8 HYODAI(I1),FHYODA(I1),III	00831400
	REAL IDATA(I1),IFSC1(I15),IFSC2(I15)	00831500
	INTEGER Q(MB),TOTALR(I3),TOTALC(I3),	00831600
*	FACE(I15),COUNT1(I15),COUNT2(I15)	00831700
	DIMENSION XX(MA),MQ(MX),MM(MA),NUM(KN),ITEM(I1),IA(I11,I12),	00831800
*	PT(I3),TPAR(I3),P(I3,I15),XX1(I15,I15),XX2(I15,I15),	00831900
*	YY1(I15),YY2(I15),MQH(MX),IFC(I6,I6,I6),JPS(5),	00832000
*	MQI(10),MQJ(10)	00832100
	DATA III/' ' / ,JPS/0,0,0,0,0/	00832200
C		00832300
	KKK=MQ2/2	00832400
C		00832500
	DO 15 I=1,KKK	00832600
C		00832700
	IS1=2*I-1	00832800
	JS1=2*I	00832900
	MQI(1)=MQH(JS1)	00833000
C		00833100
	DO 111 IJ=1,MQ1	00833200
	IJ1=2*IJ-1	00833300
	IJ2=2*IJ	00833400
	MQJ(IJ1)=MQH(IS1)	00833500
	MQJ(IJ2)=MQ(IJ)	00833600
111	CONTINUE	00833700
C		00833800
	DO 112 IJK=1,MQ1	00833900
	IS=2*IJK-1	00834000
	JS=2*IJK	00834100
	MMQ1=1	00834200
C		00834300
	DO 18 J=1,50	00834400
	HYODAI(J)=III	00834500
	FHYODA(J)=III	00834600
18	CONTINUE	00834700
C		00834800
C	STORE THE FACE-SHEET BLOCKS	00834900
C		00835000
	KL=MMQ1	00835100
	DO 71 N1=IS,JS	00835200
	DO 72 N2=1,MMQ1	00835300
	IF(MQJ(N1).EQ.MQI(N2)) GO TO 73	00835400
72	CONTINUE	00835500
	KL=KL+1	00835600
	MQI(KL)=MQJ(N1)	00835700
	N2=KL	00835800
73	N3=N2	00835900
	IF(N1.EQ.IS) GO TO 74	00836000
	IDD2=N3	00836100
	GO TO 71	00836200
74	IDD1=N3	00836300
71	CONTINUE	00836400
C		00836500
	CALL CATGYA(XX,INDEV,MA,XX1,YY1,MQI,10,M1,I15,KL,I15,IERO)	00836600
	IF(IERO.EQ.1) GO TO 9999	00836700
	ICR=YY1(IDD2)	00836800
	NF2=YY1(IDD2)	00836900

	NF1=YY1(1001)	00837000
C		00837100
	N=KL	00837200
	LL=NF2+2	00837300
	DO 35 J=1,LL	00837400
	KJ=KL+J	00837500
	FACE(J)=KJ	00837600
35	CONTINUE	00837700
	DO 1 J=1,NF1	00837800
1	IFSC1(J)=J	00837900
	DO 2 J=1,NF2	00838000
2	IFSC2(J)=J	00838100
	DO 3 K5=1,LL	00838200
	K2=1001	00838300
	IF(K5.EQ.LL) K2=1002	00838400
	YY2(K5)=YY1(K2)	00838500
	K1=YY2(K5)	00838600
	DO 4 I1=1,K1	00838700
4	XX2(K5,I1)=XX1(K2,I1)	00838800
3	CONTINUE	00838900
C		00839000
C	CONSTRUCT THE DUMMY VARIABLES	00839100
C		00839200
	DO 55 I1=1,ICR	00839300
	DO 55 IJ=1,NF2	00839400
	DO 55 IK=1,NF1	00839500
	IF(I1.EQ.IJ) GO TO 56	00839600
	IFC(I1,IJ,IK)=NF1+1	00839700
	GO TO 55	00839800
56	IFC(I1,IJ,IK)=IK	00839900
55	CONTINUE	00840000
C		00840100
C	STORE THE LIST OF ITEM-NAME	00840200
C		00840300
	DO 65 IJ=1,KL	00840400
	L1=1	00840500
	K1=MQI(IJ)	00840600
	CALL PITEM(MM,MA,M1,K1,L1,L2,KZ)	00840700
	K2=0	00840800
	DO 75 L=L1,L2	00840900
	K2=K2+1	00841000
	CALL MOVEC(HYODAI(IJ),K2,Q(L),1,1)	00841100
75	CONTINUE	00841200
65	CONTINUE	00841300
C		00841400
	JJ=NF2+2	00841500
	J3=0	00841600
	DO 86 JK=1,JJ	00841700
	L1=1	00841800
	IF(JK.EQ.1.OR.JK.EQ.JJ) GO TO 88	00841900
	JJ1=0	00842000
	J3=J3+1	00842100
	J4=XX2(LL,J3)	00842200
92	JJ1=JJ1+1	00842300
	J5=J4/10	00842400
	IF(J5.EQ.0) GO TO 87	00842500
	JPS(JJ1)=J4-J5*10	00842600
	J4=J5	00842700
	GO TO 92	00842800
87	CONTINUE	00842900
	JPS(JJ1)=J4	00843000
	DO 91 I1=1,JJ1	00843100
	JJ2=JPS(I1)	00843200
	JJ3=5-I1	00843300
	CALL MOVEC(FHYODA(JK),JJ3,NUM(JJ2+1),1,1)	00843400
91	CONTINUE	00843500
	GO TO 86	00843600

C		00843700
	88 K1=MQJ(IS)	00843800
	IF(JK.EQ.JJ) K1=MQJ(JS)	00843900
	CALL PITEM(MM,MA,M1,K1,L1,L2,KA)	00844000
	K2=0	00844100
	DO 89 L=L1,L2	00844200
	K2=K2+1	00844300
	89 CALL MOVEC(FHYODA(JK),K2,Q(L),1,1)	00844400
	86 CONTINUE	00844500
C		00844600
	N=KL	00844700
	LL=NF2+2	00844800
C		00844900
	IF(I.NE.1.OR.IJK.NE.1) WRITE(IOUT,6200)	00845000
	IPRINT=KMZW	00845100
C		00845200
	CALL CROSSB(XX,MA,INDEV,M1,ITWO,TOTALR,TOTALC,MQI,MQJ,10,PT,	00845300
	* TPAR,P,IData,HYODA1,FHYODA,ITEM,IA,I1,I11,I12,I3,	00845400
	* I15,N,LL,FACE,IDD1,IDD2,ICR,XX1,XX2,YY1,YY2,IFC,I6,	00845500
	* IFSC1,IFSC2,IPRINT,MM1,MM2,NN1,COUNT1,COUNT2,1)	00845600
C		00845700
	112 CONTINUE	00845800
	15 CONTINUE	00845900
C		00846000
	6200 FORMAT(1H1)	00846100
	RETURN	00846200
C		00846300
	9999 WRITE(IOUT,6000)	00846400
	6000 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA ',	00846500
	* 'SPECIFIED BY COMMAND ---')	00846600
	RETURN	00846700
	END	00846800

```

      SUBROUTINE CROSSB(XX,MA,INDEV,M1,ITWO,TOTALR,TOTALC,MQ,MQH,MX,PT, 00846900
      *      TPAR,P,IDATA,HYODAI,FHYODA,ITEM,IA,M9,MM9,MM8, 00847000
      *      M7,MM6,N,L,FACE,IDD1,IDD2,ICR,XX1,XX2,YY1,YY2, 00847100
      *      IFC,M6,IFSC1,IFSC2,KMZw,MM1,MM2,NN1,COUNT1, 00847200
      *      COUNT2,MQ1 ) 00847300
C      00847400
C      THIS SUBROUTINE IS ORIGINALLY WRITTEN BY H.MORO1. 00847500
C      THIS PROGRAM IMPLEMENTS THE CROSS-CLASSIFIED TABLES AND CARRIES OUT 00847600
C      THE PERCENTILE-TEST FOR THEM. 00847700
C      00847800
C      COMMON /OUTIN/INPT,IOUT 00847900
C      00848000
C      INTEGER*2 LITER(13),CHECK(50),IDP(3),PY(50,20) 00848100
C      INTEGER TOTALR(M7),TOTALC(M7),COUNT1(MM6),COUNT2(MM6), 00848200
      *      FACE(MM6) 00848300
      *      HYODAI(M9),FHYODA(M9) 00848400
      *      REAL*8 IDATA(M9),IFSC1(MM6),IFSC2(MM6) 00848500
      *      REAL XX(MA),MQ(MX),MQH(MX),ITEM(M9),IA(MM9,MM8), 00848600
      *      DIMENSION P(M7,MM6),PT(M7),TPAR(M7),XX1(MM6,MM6),XX2(MM6,MM6), 00848700
      *      YY1(MM6),YY2(MM6),IFC(M6,M6,M6), 00848800
      *      IDL(200),AAA(50),FM1(8),FM2(11),FM3(10),FM4(12), 00848900
      *      FM5(7),FM6(11) 00849000
C      00849100
      *      DATA IDP/' ','+', '-' / 00849200
      *      DATA LITER/'1','2','3','4','5','6','7','8','9','10','11','12','13' / 00849300
      *      DATA FM1/'(1H+',',',63X',',',12F',',',6.0',',',6X',',',4H',',',TO,4HTAL',',')' / 00849400
      *      DATA FM2/' ',11',',',21',',',31',',',41',',',51',',',61',',',71',',',81', 00849500
      *      ',91',',',101',',',111' / 00849600
      *      DATA FM3/'(1H+',',',56X',',',F6',',',0,2X',',',',F6.1',',',11(' ',A1,F', 00849700
      *      ',5.1)',',',15)' / 00849800
      *      DATA FM4/' ',1F',',',2F',',',3F',',',4F',',',5F',',',6F',',',7F',',',8F', 00849900
      *      ',9F',',',10F',',',11F',',',12F' / 00850000
      *      DATA FM5/'(1H+',',',59X',',',4H',',',TO,4HTAL',',',12F',',',6.1',',',15)' / 00850100
      *      DATA FM6/' ',1(' ',',',2(' ',',',3(' ',',',4(' ',',',5(' ',',',6(' ',',',7(' ',',',8(' 00850200
      *      ',9(' ',',10(' ',',11(' / 00850300
C      00850400
      *      CALL ERRSET(207,256,-1,1,1,207) 00850500
      *      CALL ERRSET(209,256,-1,1,1,209) 00850600
      *      CALL ERRSET(251,256,-1,1,1,251) 00850700
C      00850800
      *      III=1 00850900
      *      JJJ=0 00851000
      *      NL=N+L 00851100
      *      NN2=1 00851200
      *      IF(ITWO.EQ.2) NN2=0 00851300
      *      NSUBT=0 00851400
      *      IF(NN2.GT.0)NL=N 00851500
C      00851600
C      00851700
      *      DO 15 I=1,NL 00851800
      *      IF(I.GT.N) GO TO 14 00851900
      *      ITEM(I)=YY1(I) 00852000
      *      GO TO 15 00852100
14  *      J=1-N 00852200
      *      ITEM(I)=YY2(J) 00852300
15  *      CONTINUE 00852400
      *      DO 10 I=1,MM9 00852500
      *      DO 10 J=1,MM8 00852600
10  *      IA(I,J)=0 00852700
      *      NSAMPL=0 00852800
C      00852900
      *      REWIND INDEV 00853000
      *      DO 425 K=1,NSAMP 00853100
C      00853200
C      00853300
      *      READ-IN BY LETTER OR INTEGER(IN CASE OF INTEGER, GO TO 1003) 00853400
C      00853500
      *      READ(INDEV,END=99,ERR=999) KK,(XX(J),J=1,M1) 00853600
      *      IF(ITWO.EQ.2) GO TO 97 00853700
      *      DO 98 I=1,N 00853800
      *      K1=MQH(I) 00853900
      *      J=YY1(I) 00854000
      *      DO 13 I1=1,J 00854100
      *      IF(XX(K1).EQ.XX1(I,I1)) GO TO 12 00854200
13  *      CONTINUE 00854300
12  *      IDATA(I)=I1 00854400
98  *      CONTINUE 00854500
      *      GO TO 95

```

C		00854600
	97 DO 96 I=1,N	00854700
	K1=MQ(I)	00854800
	J=YY1(I)	00854900
	DO 1 I1=1,J	00855000
	IF(XX(K1).EQ.XX1(I,I1)) GO TO 2	00855100
	1 CONTINUE	00855200
	2 IDATA(I)=I1	00855300
	96 CONTINUE	00855400
	95 CONTINUE	00855500
C		00855600
	IF(NN1.GT.0)GO TO 1003	00855700
C		00855800
	DO 94 I=1,N	00855900
	94 IDL(I)=IDATA(I)	00856000
	DO 4001 J=1,N	00856100
	DO 4002 JJ=1,13	00856200
	IF(IDL(J).NE.LITER(JJ))GO TO 4002	00856300
	IDATA(J)=JJ	00856400
	GO TO 4001	00856500
	4002 CONTINUE	00856600
	4001 CONTINUE	00856700
	GO TO 1006	00856800
	1003 CONTINUE	00856900
C		00857000
	WHEN CALCULATE ONLY SIMPLE CROSS, GO TO 1004	00857100
C		00857200
	1006 IF(NN2.GT.0)GO TO 1004	00857300
	ID1=IDATA(IDD1)	00857400
	ID2=IDATA(IDD2)	00857500
	ID1=IFSC1(ID1)	00857600
	ID2=IFSC2(ID2)	00857700
	IDATA(N+1)=ID1	00857800
	DO 497 I=1,ICR	00857900
	NI=N+I+1	00858000
	AAA(I)=ID2	00858100
	497 IDATA(NI)=IFC(I,ID2,ID1)	00858200
	NI=N+ICR+2	00858300
	IDATA(NI)=ID2	00858400
	IDATA(IDD1)=ID1	00858500
	IDATA(IDD2)=ID2	00858600
	1004 CONTINUE	00858700
	DO 11 I=1,N	00858800
	ZZ=FLOAT(ITEM(I))	00858900
	11 IF(IDATA(I).LE.0.0.OR.IDATA(I).GT.ZZ) IDATA(I)=ITEM(I)	00859000
	ISS=1	00859100
	NSAMPL=NSAMPL+ISS	00859200
	I1=0	00859300
C		00859400
	SORT AND STORE IN ARRAY IA FOR EACH FACE SHEET	00859500
C		00859600
	DO 25 I2=1,L	00859700
	I3=FACE(I2)	00859800
	I=IDATA(I3)+FLOAT(I1)	00859900
	JJ=0	00860000
	IF(NN2.GT.0) GO TO 81	00860100
	IF(I2.EQ.1.OR.I2.EQ.L) GO TO 81	00860200
	IF(I2-1.NE. AAA(I2-1)) GO TO 25	00860300
	81 CONTINUE	00860400
	DO 20 J2=1,N	00860500
	J=IDATA(J2)+FLOAT(JJ)	00860600
	IA(I,J)=IA(I,J)+ISS	00860700
	JJ=JJ+ITEM(J2)	00860800
	20 CONTINUE	00860900
	25 I1=I1+ITEM(I3)	00861000
	425 CONTINUE	00861100
	99 CONTINUE	00861200

C	NSAMP=NSAMPL	00861300
	S=100./NSAMP	00861400
318	KMZW=KMZW+1	00861500
	J1=1	00861600
	J2=ITEM(1)	00861700
	NQUES=0	00861800
C		00861900
C	PRINT OUT CROSS TABULATIONS FOR EACH ITEM	00862000
C		00862100
	DO 70 K=1,N	00862200
	IF(NN2.EQ.0.AND.K.GE.2) GO TO 998	00862300
	IF(K.GT.MQ1) GO TO 998	00862400
	IR=ITEM(K)	00862500
26	IF(IR.GT.9.OR.KMZW.EQ.2) GO TO 27	00862600
	NQUES=NQUES+1	00862700
	IF(NQUES.LE.1)GO TO 597	00862800
	WRITE(6,596)	00862900
596	FORMAT(1H1,///)	00863000
	I1=I1+1	00863100
	WRITE(IOUT,6100) I1	00863200
6100	FORMAT(110X,'--- PAGE 1-',I3,' ---' /)	00863300
1005	NQUES=1	00863400
	GO TO 599	00863500
597	WRITE(6,598)	00863600
598	FORMAT(///)	00863700
599	WRITE(6,1001) HYODAI(K),HYODAI(K)	00863800
1001	FORMAT(' (',AB,')',40X,'(',AB,')')	00863900
	WRITE(IOUT,1000) (XX1(K,I10),I10=1,IR)	00864000
1000	FORMAT(10X,10F5.0)	00864100
	FM1(3)=FM4(IR)	00864200
	WRITE(IOUT,FM1) (XX1(K,I10),I10=1,IR)	00864300
27	I1=1	00864400
	I3=FACE(1)	00864500
	I2=ITEM(I3)	00864600
	DO 28 IK1=1,50	00864700
	DO 28 IK2=1,20	00864800
28	PY(IK1,IK2)=IDP(1)	00864900
C		00865000
C	PRINT OUT CROSS TABLES WITH PRCENTAGES FOR EACH FACE SHEET	00865100
C		00865200
	DO 60 IK=1,L	00865300
	IF(NN2.EQ.1.AND.K.EQ.I3) GO TO 61	00865400
	JJ=0	00865500
	DO 65 J=J1,J2	00865600
	JJ=JJ+1	00865700
65	TOTALR(JJ)=0	00865800
	I2P=I2-NSUBT	00865900
	ISUB=0	00866000
	I1=0	00866100
C		00866200
C	CALCULATE ROW TOTAL IN 'TOTALC' AND COLUMN TOTAL IN 'TOTALR'	00866300
C		00866400
	DO 50 I=I1,I2P	00866500
	IC=0	00866600
	JJ=0	00866700
	DO 40 J=J1,J2	00866800
	JJ=JJ+1	00866900
	IC=IC+IA(I,J)	00867000
40	TOTALR(JJ)=TOTALR(JJ)+IA(I,J)	00867100
	I1=I1+1	00867200
	ISUB=ISUB+IC	00867300
50	TOTALC(I1)=IC	00867400
	I1=0	00867500
		00867600

C		00867700
C	CALCULATE PERCENTAGES	00867800
C		00867900
	DO 55 I=I1,I2P	00868000
	TP=0	00868100
	II=II+1	00868200
	IF(TOTALC(II).NE.0) TP=100./TOTALC(II)	00868300
	TC=0.	00868400
	JJ=0	00868500
	DO 56 J=J1,J2	00868600
	JJ=JJ+1	00868700
	P(II,JJ)=FLOAT(IA(I,J))#TP	00868800
56	TC=TC+P(II,JJ)	00868900
55	TPAR(II)=TC	00869000
	IF(ISUB.NE.0)S=100./ISUB	00869100
	PTT=0	00869200
	JJ=0	00869300
	DO 57 J=J1,J2	00869400
	JJ=JJ+1	00869500
	PT(JJ)=TOTALR(JJ)#S	00869600
57	PTT=PTT+PT(JJ)	00869700
C		00869800
C	MARGINAL PERCENT-TEST OR NOT	00869900
C		00870000
	IF(MM1.EQ.0)GO TO 51	00870100
	CALL SIGINF(II,JJ,P,PT,PY,IDP,TOTALC,M7,MM6)	00870200
51	CONTINUE	00870300
	K1=0	00870400
	GO TO (59,80,59),KMZW	00870500
59	IF(IR.GT.9.AND.KMZW.EQ.3) GO TO 80	00870600
	IF(IR.GT.9.AND.KMZW.EQ.1) GO TO 60	00870700
	WRITE(6,1100) FHYODA(IK), FHYODA(IK)	00870800
1100	FORMAT(1H0,'('A8,')',40X,'('A8,')')	00870900
	DO 58 I=I1,I2P	00871000
	K1=K1+1	00871100
	WRITE(IOUT,1200) XX2(IK,K1),(IA(I,J),J=J1,J2),TOTALC(K1)	00871200
	FM3(7)=FM6(IR)	00871300
58	WRITE(IOUT,FM3) XX2(IK,K1),(P(K1,J),PY(K1,J),J=1,IR),	00871400
*	TPAR(K1),TOTALC(K1)	00871500
	WRITE(6,1202) (TOTALR(J),J=1,IR),ISUB	00871600
	FM5(5)=FM4(IR+1)	00871700
	WRITE(6,FM5)(PT(J),J=1,IR),PTT,ISUB	00871800
1200	FORMAT(1H ,F6.0,3X,10I5)	00871900
1202	FORMAT(1H ,3X,'TOTAL ',10I5)	00872000
	IFORM=1	00872100
	GO TO 62	00872200
80	JJJ=JJJ+1	00872300
	WRITE(IOUT,6300) JJJ	00872400
6300	FORMAT(1H1//110X,'--- PAGE 2-'I3,' ---' /)	00872500
	WRITE(IOUT,2000) FHYODA(K)	00872600
2000	FORMAT(/1H0,'('A8,')')	00872700
	WRITE(IOUT,2101) (XX1(K,I10),I10=1,IR)	00872800
2101	FORMAT(10X,20F6.0)	00872900
	WRITE(6,2100)FHYODA(IK)	00873000
2100	FORMAT(1H ,'('A8,')')	00873100
	KK1=K1	00873200
	DO 85 I=I1,I2P	00873300
	K1=K1+1	00873400
	WRITE(IOUT,2001) XX2(IK,K1),(IA(I,J),J=J1,J2),TOTALC(K1)	00873500
85	WRITE(6,2002)(P(K1,J),PY(K1,J),J=1,IR),TPAR(K1)	00873600
	KK2=K1	00873700
	WRITE(6,2003)(TOTALR(J),J=1,IR),ISUB	00873800
	WRITE(6,2006)(PT(J),J=1,IR),PTT	00873900
2001	FORMAT(1H0,F6.0,3X,20I6)	00874000
2002	FORMAT(1H ,11X,F6.1,19(A1,F5.1))	00874100
2003	FORMAT(1H0, 9X,20I6)	00874200
2006	FORMAT(1H ,11X,20F6.1)	00874300
	IF(KMZW.NE.2) GO TO 63	00874400
	IF(IR.EQ.1) GO TO 63	00874500
	ISEQ=10*(I2P-I1+1)#3	00874600
	LC=60	00874700
	IIK=IK	00874800

IF(NN2.GT.0) GO TO 64	00874900
IF(IK.EQ.1.OR.IK.EQ.L) GO TO 64	00875000
I1K=1	00875100
64 CONTINUE	00875200
CALL PICT(P,M7,MM6,IR,KK1,KK2,ISEQ,LC,HYODA1,FHYODA,M9,	00875300
* K,I1K,JJJ)	00875400
63 CONTINUE	00875500
C IFORM=2	00875600
C PERCENT TEST FOR PAIRED-ENTRIES OR NOT	00875700
C	00875800
62 IF(MM2.EQ.0)GO TO 61	00875900
M1=COUNT1(IK)	00876000
M2=COUNT2(IK)	00876100
K1=M1+I1-1	00876200
K2=M2+I1-1	00876300
I=0	00876400
DO 90 J=J1,J2	00876500
I=I+1	00876600
A=IA(K1,J)+IA(K2,J)	00876700
PSTAR=A/(TOTALC(M1)+TOTALC(M2))	00876800
VARI= PSTAR*(1.-PSTAR)	00876900
AA=1./TOTALC(M1)+1./TOTALC(M2)	00877000
AA=SQRT(10.)*SQRT(VARI*AA)	00877100
DOFP=ABS(P(M1,I)-P(M2,I))/100.	00877200
CHECK(I)=IDP(1)	00877300
IF(DOFP.LE.AA)GO TO 90	00877400
CHECK(I)=IDP(2)	00877500
IF(P(M1,I).GT.P(M2,I))GO TO 90	00877600
CHECK(I)=IDP(3)	00877700
90 CONTINUE	00877800
IF(IFORM.EQ.2)GO TO 91	00877900
WRITE(6,2004)(CHECK(I),I=1,IR)	00880000
GO TO 61	0088100
91 WRITE(6,2005)(CHECK(I),I=1,IR)	0088200
2004 FORMAT(1H ,61X,10(3X,A1,2X))	0088300
2005 FORMAT(1H ,11X,20(3X,A1,2X))	0088400
61 I1=I2+1	0088500
IF(IK.EQ.L) GO TO 666	0088600
I3=FACE(IK+1)	0088700
60 I2=I2+ITEM(I3)	0088800
666 CONTINUE	0088900
71 J1=J2+1	0089000
70 J2=J2+ITEM(K+1)	0089100
998 CONTINUE	0089200
GO TO (318,319,319),KMZW	0089300
319 CONTINUE	0089400
REWIND 10	0089500
C	0089600
9999 CONTINUE	0089700
RETURN	0089800
C	0089900
999 WRITE(IOUT,6000)	0080000
6000 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA',	0080100
* ' SPECIFIED BY COMMAND ---')	0080200
RETURN	0080300
C	0080400
END	0080500
	0080600
	0080700
SUBROUTINE SIGINF(I1,J1,P,PT,PY,IDP,TC,M7,MM6)	00880800
INTEGER*2 PY,IDP	00880900
INTEGER TC	00881000
DIMENSION P(M7,MM6),PT(M7),TC(M7),PY(50,20),IDP(3)	00881100
C	00881200
C 2-SIGMA CHECK	00881300
C	00881400
DO 10 J=1,J1	00881500
VARI=PT(J)*(100.-PT(J))	00881600
DO 10 I=1,I1	00881700
AA=2.0*SQRT(VARI/TC(I))	00881800
DOFP=ABS(P(I,J)-PT(J))	00881900
PY(I,J)=IDP(1)	00882000
IF(DOFP.LE.AA)GO TO 10	00882100
PY(I,J)=IDP(2)	00882200
IF(P(I,J).GT.PT(J))GO TO 10	00882300
PY(I,J)=IDP(3)	00882400
10 CONTINUE	00882500
RETURN	00882600
END	00882700

```

C
C THIS SUBROUTINE ILLUSTRATES THE PROFILE PATTERNS FOR THE GENERATED
C CROSS-CLASSIFIED TABLES.
C WRITTEN BY K.NAKAMURA.
C
SUBROUTINE PICT(FCLT,M7,MM6,IR,KK1,KK2,ISEQ,LC,
* HYODAI,FHYODA,M9,IJK,KJI,JJJ )
DIMENSION FCLT(M7,MM6)
INTEGER*2 IPLT(50,20),IGPC(51),IGPH(23,105),IGPM(51,50),L(60)
REAL*8 GAZE(7),HYODAI(M9),FHYODA(M9)
COMMON / OUTIN / INUNIT,IPRINT
DATA L/ 1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,
* 1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,
* 1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,
* 1HU,1HV,1HW,1HX,1HY,1HZ,1H ,1H=,1H+,1H-,
* 1H*,1H/,1H(,1H),1H.,1H',1H#,1H$,1H@,
* 1H%,1H~,1H~,1H~,1H~,1H~,1H~,1H~/
DATA ISH,ISW/20,100/
DATA GAZE/'100 +',' 75 +',' 50 +',' 25 +',' 0 +',' I',' ' /
IF(IR.GT.20) RETURN
IGZ=20
KK3=KK2-KK1
IF(IGZ*KK3.GT.70) IGZ=70/KK3
N=0
ITV=ISH+3
FS=100/ISH
KKK=0
DO 10 I=1,KK3
DO 10 J=1,IR
F=FCLT(I,J)/FS
IF(F-AINT(F).GE.0.5) F=F+1.0
10 IPLT(I,J)=ISH-INT(F)+1
DO 20 I=1,ITV
DO 20 J=1,ISW
20 IGPH(I,J)=L(37)
DO 40 I=1,50
DO 30 J=1,50
30 IGPM(I,J)=L(37)
40 IGPC(I)=0
DO 50 I=1,KK3
DO 50 J=1,IR
K=IPLT(I,J)
KK=I*IGZ
IF(IGPH(K,KK).NE.L(37)) GO TO 1
IGPH(K,KK)=L(J+10)
GO TO 50
1 IF(IGPH(K,KK).EQ.L(41)) GO TO 3
IF(N.LE.50) N=N+1
IGPC(N)=2
IGPM(N,7)=IGPH(K,KK)
IGPM(N,8)=L(45)
IGPH(N,9)=L(J+10)
IGPH(K,KK)=L(41)
IGPM(N,1)=L(41)
L1=MOD(N,10)
IGPH(K,KK+1)=L(L1+1)
IGPM(N,2)=L(L1+1)
L2=N/10
IF(L2.LE.0) GO TO 2
IGPH(K,KK+1)=L(L2+1)
IGPH(K,KK+2)=L(L1+1)
IGPM(N,2)=L(L2+1)
IGPM(N,3)=L(L1+1)
2 IGPM(N,5)=L(38)
GO TO 50
3 IF(IGPH(K,KK+1).LT.L(1).OR.IGPH(K,KK+1).GT.L(10)) RETURN
LL=0
4 LL=LL+1
IF(IGPH(K,KK+1).GT.L(LL)) GO TO 4
NL=LL-1
IF(IGPH(K,KK+2).LT.L(1).OR.IGPH(K,KK+2).GT.L(10)) GO TO 6
LL=0
5 LL=LL+1
IF(IGPH(K,KK+2).GT.L(LL)) GO TO 5
NL=NL*10+LL-1
6 LL=IGPC(NL)*2+6
IGPM(NL,LL)=L(45)
LL=LL+1
IGPM(NL,LL)=L(J+10)
IGPC(NL)=IGPC(NL)+1
50 CONTINUE
IF(N.GT.50) KKK=1
IF(N.GT.50) N=50

```


C		00891200
C	SET UP SORT-ROUTINE AND ADJUST LINE	00891300
C		00891400
	DO 60 I=1,N	00891500
	M=IGPC(I)*2+6	00891600
	DO 60 J=7,M,2	00891700
	DO 60 K=7,M,2	00891800
	IF(IGPM(I,J).GE.IGPM(I,K)) GO TO 60	00891900
	IWORK=IGPM(I,J)	00892000
	IGPM(I,J)=IGPM(I,K)	00892100
	IGPM(I,K)=IWORK	00892200
60	CONTINUE	00892300
C		00892400
	JA=KK3*IGZ+3	00892500
	DO 70 J=1,JA	00892600
	IF(IGPH(ITV-2,J).NE.L(37)) GO TO 70	00892700
	IGPH(ITV-2,J)=L(40)	00892800
	IF(MOD(J,IGZ).EQ.0) IGPH(ITV-2,J)=L(39)	00892900
70	CONTINUE	00893000
	I=0	00893100
7	I=I+1	00893200
	J=IGZ*I	00893300
	K=MOD(I,10)	00893400
	IGPH(ITV,J)=L(K+1)	00893500
	K=I/10	00893600
	IF(K.EQ.0) GO TO 8	00893700
	IGPH(ITV,J-1)=L(K+1)	00893800
8	IF(I.LT.KK3) GO TO 7	00893900
C		00894000
C	PRINT THE PROFILE PATTERN GRAPH	00894100
C		00894200
	ISEQ=ISEQ+MAX(N,23)+7	00894300
	IF(ISEQ.LE.LC) GO TO 9	00894400
	WRITE(IPRINT,6000)	00894500
	WRITE(IPRINT,6300) JJJ	00894600
6300	FORMAT(/110X,'--- PAGE 2-',I3,'-1 ---' /)	00894700
9	WRITE(IPRINT,6100) HYODAI(IJK)	00894800
	ISI=ISH+1	00894900
	JA=KK3*IGZ+7	00895000
	I=0	00895100
	II=N+1	00895200
	INIT=1	00895300
	K=(ISH-N)/2	00895400
11	IF(I.GE.K) GO TO 12	00895500
	I=I+1	00895600
	GO TO 80	00895700
12	INIT=2	00895800
	II=0	00895900
13	I=I+1	00896000
	IF(I.GT.ITV) GO TO 14	00896100
	II=II+1	00896200
	GO TO 80	00896300
14	IF(I.GE.N) GO TO 99	00896400
	INIT=3	00896500
	I=ITV+1	00896600
15	IF(II.GE.N) GO TO 99	00896700
	II=II+1	00896800
80	IG=6	00896900
	IF(MOD(ISI-I,5).EQ.0) IG=5-(ISI-I)/5	00897000
	IF(I.GT.ISI) IG=7	00897100
	JB=IGPC(II)*2+6	00897200
	WRITE(IPRINT,6200) GAZE(IG),(IGPH(I,J),J=1,JA),	00897300
	* (IGPM(II,JJ),JJ=1,JB)	00897400
	GO TO (11,13,15),INIT	00897500
99	CONTINUE	00897600
	WRITE(IPRINT,6400) FHYODA(KJI),HYODAI(IJK)	00897700
	IF(KKK.EQ.1) WRITE(IPRINT,6500)	00897800
6400	FORMAT(////40X,'<< 'A8,' VS. 'A8,' >>')	00897900
6500	FORMAT(/1H,19X,'--- OVERLAPPED POINTS ARE MORE THAN 50 . ---')	00898000
	RETURN	00898100
6000	FORMAT(1H1)	00898200
6100	FORMAT(/1H0,4X,'--- PROFILE PATTERN FOR EACH CATEGORY OF',	00898300
	* ' ITEM 'A8,' ---'///)	00898400
6200	FORMAT(1H,10X,A5,120A1)	00898500
	END	00898600

	SUBROUTINE ACHECK(AA, III)	00898700
C		00898800
C	THIS SUBROUTINE DETECTS THE USELESS PARENTHESES AND DELETES THEM.	00898900
C		00899000
	INTEGER AA(100,80),BLANK,SLASH,OPC,CLC,COMMA	00899100
	DIMENSION ICON(5),KCON(6)	00899200
	DATA ICON/'O','R','A','N','D',/	00899300
	* KCON/'I','F','T','H','E','N',/	00899400
	* BLANK/' ', SLASH/'/', OPC/'(', CLC/')',	00899500
	* COMMA/',',/	00899600
C		00899700
	K=0	00899800
	KK=0	00899900
	IOPC=0	00900000
	ICLC=0	00900100
	IIF=0	00900200
	IRET=0	00900300
	II=III	00900400
C		00900500
	5 CONTINUE	00900600
	DO 10 I=16,80	00900700
	IF(AA(II,I).EQ.OPC) GO TO 11	00900800
	IF(AA(II,I).EQ.CLC) GO TO 12	00900900
	IF(AA(II,I).EQ.BLANK.OR.AA(II,I).EQ.SLASH.OR.AA(II,I).EQ.COMMA)	00901000
	* GO TO 13	00901100
	K=K+1	00901200
	GO TO 10	00901300
	11 IOPC=1	00901400
	GO TO 13	00901500
	12 ICLC=1	00901600
	13 IF(K.EQ.0) GO TO 17	00901700
	I1=-K	00901800
	L=I1	00901900
	DO 14 J=1,2	00902000
	IF(KCON(J).NE.AA(II,L)) GO TO 15	00902100
	14 L=L+1	00902200
	IIF=1	00902300
	GO TO 17	00902400
	15 L=I1	00902500
	DO 16 J=3,6	00902600
	IF(KCON(J).NE.AA(II,L)) GO TO 18	00902700
	16 L=L+1	00902800
	IIF=1	00902900
	GO TO 17	00903000
	18 L=I1	00903100
	DO 19 J=1,2	00903200
	IF(ICON(J).NE.AA(II,L)) GO TO 20	00903300
	19 L=L+1	00903400
	IF(IRET.EQ.1) RETURN	00903500
	KK=0	00903600
	GO TO 17	00903700
	20 L=I1	00903800
	DO 21 J=3,5	00903900
	IF(ICON(J).NE.AA(II,L)) GO TO 17	00904000
	21 L=L+1	00904100
	IF(IRET.EQ.1) RETURN	00904200
	KK=0	00904300
	GO TO 17	00904400
	17 K=0	00904500
	IF(IOPC.EQ.0) GO TO 23	00904600
	KK=1	00904700
	IOPC=0	00904800
	23 IF(IIF.EQ.0) GO TO 22	00904900
	IIF=0	00905000
	KK=0	00905100
	22 IF(ICLC.EQ.0) GO TO 10	00905200
	ICLC=0	00905300
	IF(KK.EQ.0) GO TO 10	00905400
	IF(IRET.EQ.0) GO TO 9	00905500
	AA(II,KK)=BLANK	00905600
	AA(II,I)=BLANK	00905700
	RETURN	00905800
	9 AA(II,KK)=BLANK	00905900
	AA(II,I)=BLANK	00906000
	KK=0	00906100
	10 CONTINUE	00906200
	IF(KK.EQ.0) RETURN	00906300
	II=II+1	00906400
	IRET=1	00906500
	GO TO 5	00906600
	END	00906700

	SUBROUTINE	INVPRI(NUM,KN,LF,INA,K,L)	00906800
C			00906900
C	THIS SUBROUTINE	CONTROLS THE OUTPUT OF INVALID DATA IN CONSISTENCY	00907000
C	CHECKING.		00907100
C			00907200
	COMMON	/ OUTIN / INPT,IOUT	00907300
	DIMENSION	NUM(KN),IPRINT(16),IFMT(8)	00907400
	DATA	IFMT/'(10X',' ',' ' '(2X','I4,1','X,I5',	00907500
	*	4H,'/',' ') ' ' /	00907600
			00907700
C	IF(L.EQ.1)	GO TO 10	00907800
	IF(INA.EQ.201)	WRITE(IOUT,600)	00907900
	LF=LF+1		00908000
	LF1=LF#2-1		00908100
	LF2=LF#2		00908200
	IPRINT(LF1)=INA		00908300
	IPRINT(LF2)=K		00908400
	IF(LF.LT.8)	RETURN	00908500
10	CONTINUE		00908600
	IFMT(3)=NUM(LF+1)		00908700
	WRITE(IOUT,IFMT)	(IPRINT(1),I=1,LF2)	00908800
	IJ=INA-200		00908900
	IF(IJ/40#40 .EQ. IJ)	WRITE(IOUT,610)	00909000
	LF=0		00909100
	RETURN		00909200
600	FORMAT(//// 10X, '< NUMBER OF INVALID DATA >' ///		00909300
*	18X,8('DATA',9X) / 13X,8('NO.',3X,'NO.',4X) //)		00909400
610	FORMAT(1H)		00909500
	END		00909600

```

      SUBROUTINE RECOD(AA,II,BLANK,COMMA,SLASH,NUM,Q,MQ1,MQ,MM,M1,KN,00909700
      *          KA1,KA2,MB,MX,MA,ABC,KA,AM1,AM2,MM3,MM4,NUMB) 00909800
C
C THIS IS A DECODER OF RECODE PROCEDURE. 00909900
C
      COMMON / OUTIN / INPT,IOUT 00910000
      INTEGER AA(100,80),Q(MB),COMMA,BLANK,SLASH,ABC(KA), 00910100
      * CLOSK,OPENK,EQUAL,CORON,SCORON 00910200
      DIMENSION MM(MA),MQ(MX),NUM(KN),NUMB(KN), 00910300
      * AM1(KA1,KA2),AM2(KA1,KA2),MM3(KA1,KA2),MM4(KA1) 00910400
      DATA NAMI/'~'/'/CORON/'/'/SCORON/'/'/ 00910500
      * CLOSK/'/'/OPENK/'/'/EQUAL/'/'/MINUS/'-'/' 00910600
C 00910700
C 00910800
C 00910900
C 00911000
C 00911100
      INITIALIZATION 00911200
C
      N1=1 00911300
      K2=0 00911400
      FFA=0 00911500
      INAM=0 00911600
      FF=0 00911700
      INAM1=0 00911800
      FG=0 00911900
      L1=1 00912000
      K=0 00912100
      FFG=0 00912200
      IERO=0 00912300
C 00912400
C 00912500
      DECODE THE ITEM-NAME 00912600
C
      DO 33 I=16,80 00912700
      IF(AA(II,I).EQ.SLASH) GO TO 34 00912800
      IF(FG.EQ.1.0) GO TO 18 00912900
      IF(AA(II,I).EQ.OPENK) GO TO 12 00913000
      I2=1 00913100
      IF(AA(II,I).EQ.BLANK.OR.AA(II,I).EQ.COMMA) GO TO 13 00913200
      K=K+1 00913300
      GO TO 33 00913400
12 FG=1 00913500
13 IF(K.EQ.0) GO TO 33 00913600
      KK1=I-K 00913700
      KK2=I-1 00913800
      DO 15 L=1,M1 00913900
      M9=MM(L) 00914000
      L2=L1+M9-1 00914100
      IF(K.EQ.M9) GO TO 16 00914200
14 L1=L2+1 00914300
15 CONTINUE 00914400
      K=0 00914500
      IF(IERO.EQ.0) WRITE(IOUT,6100) 00914600
      WRITE(IOUT,6200) (AA(II,IM),IM=KK1,KK2) 00914700
      IERO=1 00914800
6100 FORMAT(1H1,///1X,'--- THIS ITEM IS UNDEFINED ---') 00914900
6200 FORMAT(1H,'ERROR ITEM = ',8A1) 00915000
      L1=1 00915100
      GO TO 33 00915200
16 J=KK1 00915300
      DO 17 LL=L1,L2 00915400
      IF(Q(LL).NE.AA(II,J)) GO TO 14 00915500
17 J=J+1 00915600
      MQ1=MQ1+1 00915700
      MQ(MQ1)=L 00915800
      K=0 00915900
      I1=MQ1 00916000
      L1=1 00916100
      GO TO 33

```

C		00916200
C	DECODE AND STORE THE NUMERIC VALUES	00916300
C		00916400
	18 IF(AA(I1,I).EQ.CLOSK) GO TO 19	00916500
	IF(AA(I1,I).EQ.COMMA.OR.AA(I1,I).EQ.BLANK) GO TO 22	00916600
	IF(AA(I1,I).EQ.CORON.OR.AA(I1,I).EQ.SCURON) GO TO 22	00916700
	IF(AA(I1,I).EQ.EQUAL) GO TO 20	00916800
	IF(AA(I1,I).EQ.NAMI) GO TO 21	00916900
	K2=K2+1	00917000
	GO TO 33	00917100
	19 FG=0	00917200
	FFG=1	00917300
	GO TO 22	00917400
	20 FF=1	00917500
	GO TO 22	00917600
	21 INAMI=1	00917700
C		00917800
	22 IF(K2.EQ. 0) GO TO 32	00917900
	KJ1=1-K2	00918000
	KJ2=1-1	00918100
	DO 23 JJ=1,KA	00918200
	IF(AA(I1,KJ1).EQ.ABC(JJ)) GO TO 31	00918300
	23 CONTINUE	00918400
	K2=0	00918500
	IA=0	00918600
	IFZ=1	00918700
	DO 26 J=KJ1,KJ2	00918800
	IF(AA(I1,J).NE.MINUS) GO TO 80	00918900
	IFZ=-1	00919000
	GO TO 26	00919100
	80 IA=IA+1	00919200
	DO 24 JN=1,KN	00919300
	IF(AA(I1,J).EQ.NUM(JN)) GO TO 25	00919400
	24 CONTINUE	00919500
	25 JN=JN-1	00919600
	NUMB(IA)=JN	00919700
	26 CONTINUE	00919800
C		00919900
	JA=0	00920000
	DO 27 JJ=1,IA	00920100
	J1=IA-JJ	00920200
	27 JA=JA+NUMB(JJ)*10**J1	00920300
	JA=IFZ*JA	00920400
C		00920500
	IF(FFA.EQ.1.0) GO TO 29	00920600
	IF(INAM.EQ. 1) GO TO 28	00920700
	AM1(I1,I2)=JA	00920800
	AM2(I1,I2)=JA	00920900
	MM4(I1)=I2	00921000
	I2=I2+1	00921100
	GO TO 32	00921200
	28 I12=I2-1	00921300
	AM2(I1,I12)=JA	00921400
	INAM=0	00921500
	INAMI=0	00921600
	GO TO 32	00921700
	29 N2=I2-1	00921800
	DO 30 JB=N1,N2	00921900
	MM3(I1,JB)=JA	00922000
	30 CONTINUE	00922100
	N1=I2	00922200
	FF=0	00922300
	FFA=0	00922400
	GO TO 32	00922500
	31 K2=0	00922600
	N1=KA2	00922700
	I2=KA2+1	00922800
	AM1(I1,KA2)=99.	00922900
	AM2(I1,KA2)=99.	00923000
	32 IF(FF.EQ.1.0) FFA=1	00923100
	IF(INAMI.EQ. 1) INAM=1	00923200
	IF(FFG.NE.1.0) GO TO 33	00923300
	FFG=0	00923400
	N1=1	00923500
	33 CONTINUE	00923600
	34 IF(IERO.EQ. 1) MQ1=0	00923700
	RETURN	00923800
	END	00923900

	SUBROUTINE	RECODE(XX,MQ,INDEV,OUTDEV,AM1,AM2,MM3,MM4,	00924000
*		MQ1,N,MA,M1,MX,KA1,KA2,MM,Q,MB,NUM,KN,MMM,REC)	00924100
C	COMMON	/ OUTIN / INPT,IOUT	00924200
	COMMON	NUMDT(20)	00924300
	INTEGER	OUTDEV,FMT(4),BLANK,COMMA,AM5(30),ELOS(5)	00924400
	DIMENSION	N1(20),N2(20),NUM(KN),QQ(20,8),IJK(6)	00924500
	DIMENSION	AM1(KA1,KA2),AM2(KA1,KA2),MM3(KA1,KA2),MM4(KA1),	00924600
		XX(MA),MQ(MX),Q(MB),MM(MA),MMM(KA1,KA2)	00924700
*	DATA	BLANK/' ',COMMA/' ',NAMI/'~'/',	00924800
*		ELOS/'0','T','H','E','R'/',	00924900
*		IJK/'(','X','I')',' ',' ',' ',' '	00925000
	DATA	FMT/'(1H0','5X',' ','A1)'/	00925100
C			00925200
	IF(OUTDEV.NE. 0)	GO TO 99	00925300
	OUTDEV=10		00925400
	REC=1		00925500
99	CONTINUE		00925600
C			00925700
	WRITE(IOUT,6000)		00925800
6000	FORMAT(1H1,///1H '+'-----+',/1X,'I RECODE 'I',/1X,		00925900
*	'+'-----+',)		00926000
C			00926100
	KNUM=NUMDT(INDEV)		00926200
	IF(KNUM.EQ. 0)	GO TO 1000	00926300
C			00926400
C	SPECIFY THE INITIAL PAPAMETERS		00926500
C			00926600
	DO 11 I1=1,KA1		00926700
	DO 12 I2=1,KA2		00926800
	MMM(I1,I2)=0		00926900
12	CONTINUE		00927000
11	CONTINUE		00927100
	ID=0		00927200
	NOS=0		00927300
	DO 42 IK=1,20		00927400
	DO 41 IJ=1,8		00927500
	QQ(IK,IJ)=BLANK		00927600
41	CONTINUE		00927700
42	CONTINUE		00927800
	DO 24 JJ=1,MQ1		00927900
	N1(JJ)=0		00928000
	N2(JJ)=0		00928100
24	CONTINUE		00928200
C			00928300
C	EXECUTE THE RECODE OPTION		00928400
C			00928500
	REWIND INDEV		00928600
	DO 40 I=1,N		00928700
	READ(INDEV,END=50,ERR=70) KK,(XX(J),J=1,M1)		00928800
	NOS=NOS+1		00928900
	DO 30 J=1,MQ1		00929000
	K=MQ(J)		00929100
	NN=MM4(J)		00929200
C			00929300
	DO 10 JJ=1,NN		00929400
	IF(XX(K).GE.AM1(J,JJ).AND.XX(K).LE.AM2(J,JJ))	GO TO 20	00929500
10	CONTINUE		00929600
	JJ=KA2		00929700
	IF(AM1(J,KA2).EQ.99.)	GO TO 20	00929800
	N2(J)=N2(J)+1		00929900
	GO TO 30		00930000
C			00930100
	20 CONTINUE		00930200
	XX(K)=MM3(J,JJ)		00930300
	N1(J)=N1(J)+1		00930400
	MMM(J,JJ)=1		00930500
30	CONTINUE		00930600
	IF(OUTDEV.EQ. 0)	GO TO 40	00930700
	WRITE(OUTDEV) KK,(XX(J),J=1,M1)		00930800
	ID=ID+1		00930900
40	CONTINUE		00931000
50	CONTINUE		00931100
C			00931200
	IF(OUTDEV.EQ. 0)	GO TO 60	00931300
C			00931400
	NUMDT(OUTDEV)=ID		00931500
			00931600

C		00931700
C	PRINT THE RESULT OF RECODE	00931800
C		00931900
	60 WRITE(IOUT,6100) NOS	00932000
6100	FORMAT(1H0,6X,'SAMPLES',5X,'= ',15)	00932100
	DO 25 J1=1,MQ1	00932200
	L1=1	00932300
	K=MQ(J1)	00932400
	CALL PITEM(MM,MA,M1,K,L1,L2,KP)	00932500
	M9=L2-L1+1	00932600
	FMT(3)=NUM(M9+1)	00932700
	WRITE(IOUT,FMT) (Q(L),L=L1,L2)	00932800
	KJ=0	00932900
	WRITE(IOUT,6300)	00933000
	DO 37 IU=L1,L2	00933100
	KJ=KJ+1	00933200
	QQ(J1,KJ)=Q(IU)	00933300
37	CONTINUE	00933400
	WRITE(IOUT,6250) N1(J1),N2(J1)	00933500
6250	FORMAT(1H ,6X,'RECODE',6X,'= ',15/7X,'NON-RECODE',2X,'= ',15)	00933600
	WRITE(IOUT,6300)	00933700
6300	FORMAT(1H)	00933800
25	CONTINUE	00933900
	WRITE(IOUT,6300)	00934000
	WRITE(IOUT,6400)	00934100
6400	FORMAT(1H0,9X,76('.'/1H /15X,'ITEM',19X,'PRE-CODE',25X,	00934200
	* 'AFTER-CODE'/1H /10X,76('.'/))	00934300
C		00934400
	DO 28 L0=1,MQ1	00934500
	DO 35 I1=1,30	00934600
	AM5(I1)=BLANK	00934700
35	CONTINUE	00934800
	WRITE(IOUT,6500) (QQ(L0,K),K=1,8)	00934900
6500	FORMAT(1H /17X,8A1)	00935000
	NN=MM4(L0)	00935100
	I1=0	00935200
	DO 29 LP=1,NN	00935300
	A=AM1(L0,LP)	00935400
	CALL RECI1K(A,NUM,KN,I1,AM5)	00935500
	IF(AM1(L0,LP).EQ.AM2(L0,LP)) GO TO 31	00935600
	I1=I1+1	00935700
	AM5(I1)=NANI	00935800
	A1=AM2(L0,LP)	00935900
	CALL RECI1K(A1,NUM,KN,I1,AM5)	00936000
31	MM6=MM3(L0,LP)	00936100
	IF(MM3(L0,LP).EQ.MM3(L0,LP+1)) GO TO 26	00936200
	LP2=LP	00936300
	IF(MMM(L0,LP).EQ. 1) GO TO 15	00936400
13	LP1=LP2	00936500
	IF(MM3(L0,LP2).NE.MM3(L0,LP1-1)) GO TO 14	00936600
	IF(MMM(L0,LP1-1).EQ. 1) GO TO 15	00936700
	LP2=LP1-1	00936800
	GO TO 13	00936900
14	IKK=1	00937000
	GO TO 16	00937100
15	IKK=4	00937200
16	IK1=IKK+2	00937300
	WRITE(IOUT,6800) AM5, MM6, (1JK(JK),JK=IKK,IK1)	00937400
	DO 34 I1=1,30	00937500
	AM5(I1)=BLANK	00937600
34	CONTINUE	00937700
	I1=0	00937800
	GO TO 29	00937900
26	I1=I1+1	00938000
	AM5(I1)=COMMA	00938100
29	CONTINUE	00938200

11=11+1	00938300
IF(AM1(L0,KA2),NE.99.) GO TO 36	00938400
DO 38 K=1,5	00938500
AM5(K)=ELOS(K)	00938600
38 CONTINUE	00938700
MM6=MM3(L0,KA2)	00938800
IF(MMM(L0,KA2).EQ. 1) GO TO 17	00938900
IKK=1	00939000
GO TO 18	00939100
17 IKK=4	00939200
18 IK1=IKK+2	00939300
WRITE(IOUT,6800) AM5, MM6, (IJK(JK),JK=IKK,IK1)	00939400
6800 FORMAT(40X,30A1,2X,I5,3X,3A1)	00939500
36 WRITE(IOUT,6600)	00939600
6600 FORMAT(1H /10X,76('.'))	00939700
28 CONTINUE	00939800
WRITE(IOUT,6700)	00939900
6700 FORMAT(//10X,'--- MESSAGE ---'//12X,'(1) SYMBOL (*)'5X,	00940000
* 'MEANS NO APPEARED CODE.')	00940100
C	00940200
C VERIFY THE DUPLICATED CODES	00940300
C	00940400
CALL RECERR(AM1,AM2,MM4,KA1,KA2,NUM,KN,QQ,MQ1)	00940500
C	00940600
RETURN	00940700
C	00940800
1000 WRITE(IOUT,6001) INDEV	00940900
6001 FORMAT(///1H , '--- FILE NUMBER IS'13,'. THIS FILE DOES NOT',	00941000
* ' INCLUDE DATA ---')	00941100
RETURN	00941200
C	00941300
70 WRITE(IOUT,6200)	00941400
6200 FORMAT(1H1,///1X,'--- IT IS IMPOSSIBLE TO FIND DATA SPECIFIED ',	00941500
* 'BY COMMAND ---')	00941600
RETURN	00941700
END	00941800

	SUBROUTINE RECERR(AM1,AM2,MM4,KA1,KA2,NUM,KN,QQ,MQ1)	00941900
C		00942000
C	THIS SUBROUTINE VERIFIES AND DETECTS THE DUPLICATED CODES.	00942100
C		00942200
	COMMON / DUTIN / INPT,IOUT	00942300
	INTEGER FMS(8)	00942400
	DIMENSION MMQ(10),NUM(KN),QQ(20,8),QW(20,8),	00942500
	* AM1(KA1,KA2),AM2(KA1,KA2),MM4(KA1)	00942600
	* DATA FMS/'(1H+', '43X', '4H'(' ', ' ', '8A1', '4H', ' ',	00942700
	* 4H), ' ', '4H') /	00942800
	I=0	00942900
C		00943000
	DO 20 LP=1,MQ1	00943100
	NN=MM4(LP)	00943200
	DO 10 LO=1,NN	00943300
	DO 30 LQ=1,NN	00943400
	IF(LQ.EQ.LQ) GO TO 30	00943500
	IF(AM1(LP,LO).LE.AM1(LP,LQ).AND.AM2(LP,LO).GE.AM1(LP,LQ)) GO TO 40	00943600
	IF(AM1(LP,LQ).EQ.AM2(LP,LQ)) GO TO 30	00943700
	IF(AM1(LP,LO).LE.AM2(LP,LQ).AND.AM2(LP,LO).GE.AM2(LP,LQ)) GO TO 40	00943800
	30 CONTINUE	00943900
	10 CONTINUE	00944000
	GO TO 20	00944100
	40 I=I+1	00944200
	MMQ(I)=LP	00944300
	20 CONTINUE	00944400
C		00944500
C	DETECTION AND PRINT THE CHECKING RESULTS.	00944600
C		00944700
	IF(I.EQ.0) RETURN	00944800
C		00944900
	DO 50 J=1,I	00945000
	L=MMQ(J)	00945100
	DO 60 JJ=1,8	00945200
	QW(J,JJ)=QQ(L,JJ)	00945300
	60 CONTINUE	00945400
	50 CONTINUE	00945500
C		00945600
	WRITE(IOUT,6100)	00945700
	6100 FORMAT(1H)	00945800
	WRITE(IOUT,6000)	00945900
	6000 FORMAT(12X,'(2) OVERLAPPING-CODE APPEAR IN ')	00946000
	KK=NUM(I+1)	00946100
	FMS(4)=KK	00946200
	WRITE(IOUT,FMS) ((QW(J,JJ),JJ=1,8),J=1,I)	00946300
C		00946400
	RETURN	00946500
	END	00946600

	SUBROUTINE	RECIIK(A,NUM,KN,II,AM5)	00946700
C			00946800
C	TRANSFORM THE CHARACTER MODE INTO THE NUMERIC MODE.		00946900
C			00947000
	COMMON /OUTIN /	INPT, IOUT	00947100
	INTEGER	AM5(30)	00947200
	DIMENSION	IIK(5),NUM(KN)	00947300
	DATA	IIK/5#0/,MINUS/'-'/	00947400
C			00947500
	IFZ=0		00947600
	I=0		00947700
	IK=A		00947800
	IF(IK.GE.0) GO TO 15		00947900
	IK=IK*(-1)		00948000
	IFZ=1		00948100
15	CONTINUE		00948200
10	CONTINUE		00948300
	IK1=IK/10		00948400
	IF(IK1.EQ. 0) GO TO 20		00948500
	I=I+1		00948600
	IIK(I)=IK-IK1*10		00948700
	IK=IK1		00948800
	GO TO 10		00948900
20	I=I+1		00949000
	IIK(I)=IK		00949100
C			00949200
	IF(IFZ.EQ.0) GO TO 25		00949300
	II=II+1		00949400
	AM5(II)=MINUS		00949500
25	CONTINUE		00949600
	DO 30 K=1,I		00949700
	II=II+1		00949800
	KK=I+1-K		00949900
	JJ=IIK(KK)		00950000
	AM5(II)=NUM(JJ+1)		00950100
30	CONTINUE		00950200
	RETURN		00950300
	END		00950400

〔参考資料〕

A. 主な参考文献

- 〔1〕 Akaike, H, Kitagawa, G. and others. (1979) : TIMSAC-78, Computer Science Monographs No.11, The Inst. Stat. Math.
- 〔2〕 Brown, M. B. (1976) : Screening Effects in Multidimensional Contingency Tables, *Appl. Statist.* Vol. 25, No. 1, 37-46.
- 〔3〕 Everitt, B. S. (1977) : *The Analysis of Contingency Tables*, Chapman and Hall.
- 〔4〕 Fienberg, S. E. (1977) : *The Analysis of Cross-Classified Categorical Data*, MIT Press.
- 〔5〕 Goodman, L. A. (1971) : The Analysis of Multidimensional Contingency Tables, *Technometrics*, Vol. 13, No. 1, 33-61.
- 〔6〕 Haberman, S. J. (1972) : Log-linear fit for Contingency Tables, *Appl. Statist.* Vol. 21, 218-225.
- 〔7〕 Haberman, S. J. (1978, 1979) : *Analysis of Qualitative Data*, Vol I, II, Academic Press.
- 〔8〕 Naus, J. I. (1975) : *Data Quality Control and Editing*, Marcel Dekker.
- 〔9〕 Payne, C. (1977) : *The Preparation and Processing of Survey Data, The Analysis of Survey Data*, Vol. 1 : *Exploring Data Structures*, (eds.) O' Muirheartaigh, C. A., Payne, C., John Wiley.
- 〔10〕 Sakamoto, Y., Akaike, H. (1978) : Analysis of Classified Data by AIC, *Ann. Inst. Statist. Math.*, Vol. 30, No. 1, part B.
- 〔11〕 Sonquist, J. A., Dunkelberg, W. C. (1977) : *Survey and Opinion Research: Procedures for Processing and Analysis*, Prentice-Hall.
- 〔12〕 大隅 昇 (1978) : 多次元クロス表と対数線形モデル, 「統計情報の地方における多目的利用に関する調査研究報告書」, 全国統計協会連合会。
- 〔13〕 大隅 昇 (1979) : 多重クロス表による社会調査データのモデル解析, 「多次元統計解析の数理解析」, 京大数理解析研講究録 345。
- 〔14〕 大隅 昇 (1979) : データ解析と管理技法, 現代人の統計 6, 朝倉書店。
- 〔15〕 奥野忠一他 (1971) : 多変量解析法, 日科技連。
- 〔16〕 丹後俊郎, 刈谷丈治 (1979) : 統計パッケージ SPMS と BMDP, SPSS, SAS との比較 (第1報), 日本統計学会第47回大会報告集。
- 〔17〕 水野欽司, 大隅 昇, 桂 康一 (1978, 1979) : 統計パッケージ①～⑦, ビット, Vol. 10, No. 8, 9, 11, 12, 15; Vol. 11, No. 1, 2, .

- [18] 矢島敬二, 大隅 昇 (1977. 7): 統計, 「アプリケーション・プログラム」, ビット, 臨時増刊, (小野勝章 他編)。
- [19] 行動計量学のための統計解析用プログラム・パッケージの開発 (昭和50年度文部省科研費試験研究: 駒澤 勉ほか)
- [20] 統計的データ解析と統計プログラム・パッケージ (1979. 3), 文部省科研費特定研究A-4 班研究報告書。

B. 参考にした主な統計パッケージ・マニュアルのリスト

- [1] Armor, D. J., Couch, A. S. (1972): An Introduction to Computerized Social Data Analysis — DATA-TEXT PRIMER, The Free Press.
- [2] Barr, A. J. and others (1976): A User's Guide to SAS 76, SAS Institute, 1976.
- [3] Buhler, S, Buhler, R. (1976): P-STAT Introductory Manual, Princeton Univ. Computer Center, Nov. 29, 1976.
- [4] Computing Services Office (1976): SOUPAC Program Description, August 1, 1976, Univ. of Illinois.
- [5] Dixon, W. J. (editor) (1975, 1977): BMDP; Biomedical Computer Programs, Univ. of California Press, 1975, 1977.
- [6] Fox, D. J., Guire, K. E. (1976): Documentation for MIDAS, Statistical Research Lab. (SRL), The Univ. of Michigan: 3rd. ed. Sep., 1976.
- [7] Hilsenrath, J., Ziegler, G. G. and others. (1966): OMNITAB-A Computer Program for Statistical and Numerical Analysis, National Bureau of Standards Handbook 101.
- [8] Nie, N. and others (1975): SPSS: Statistical Package The Social Sciences (2nd ed.), McGraw-Hill Book Com.
- [9] OSIRIS III (1973): Univ. of Michigan, ISR, 1973, Vol. 1~ Vol. 5.
- [10] Rohlf, F. J., Kishpaugh, J. Kirk, D; NT-SYS (1974): Numerical Taxonomy System of Multivariate Statistical Programs.
- [11] Ryan, T. A., Joiner, B. L. and Ryan, B. F. (1974): MINITAB Student Handbook, Duxbury Press.
- [12] Ryan, T. A. and others (1978): MINITAB II Reference Manual Preliminary Edition, Prof. T. Ryan, Jr., Statistical Department 215 Pond Laboratory. The Pennsylvania State Univ. Park, Pa, 16802.
- [13] Statistical Department, Rothamsted Experimental Station (1973, 1974, 1975): GENSTAT Users Guide No. 1 — No. 9.

- [14] Statistical Department, Rothamsted Experimental Station (1976) : The Rothamsted General Survey Program (RGSP), Part I (1975), Part II (1973), Introductory Guide to RGSP (1976)。
- [15] Statistical Department, Rothamsted Experimental Station (1975) : GENKEY (Manual by Payne, R. W.)。
- [16] Wishart, D (1975, 1978) : CLUSTAN 1C User Manual, Computer Center, Univ. of Colledge London.
- [17] 中川 徹, 小柳義夫, 戸川隼人 (1979) : 最小二乗法標準プログラム SALS (第2版) 利用の手引き (第1, 2部)
- [18] 三宅一郎, 水野欽司他 (1977) : SPSS 統計パッケージ II 解析編, 東洋経済新報社。
- [19] 三宅一郎, 山本嘉一郎 (1976) : SPSS 統計パッケージ I 基礎編。
- [20] 情報処理振興事業協会 (1978) : UMS 利用者マニュアル (Universal Mathematical Software System)

はしがき

MINERVAの利用法

基本的な使い方と一般規則

MINERVA命令文

READ, FORMAT, ITEM〔分析データの入力指示〕

INPUT DEVICE, OUTPUT DEVICE〔一時作業ファイルの利用〕

END

RANGE〔範囲の検証, データの選出〕

CONSISTENCY〔項目間の論理性の検証〕

FIND〔サンプルの参照〕

PATTERN〔回答パターンとその度数表〕

RECODE〔数値変換, 新コードの付与〕

ASSOCIATION〔項目間の関係係数の算出〕

BREAK DOWN〔データの多段層別化〕

QCHART〔多変量管理図〕

MULTIWAY TABLES〔多重クロス分析, 対数線形モデル解析〕

CROSS TABULATION〔クロス表作成と項目間比率の検定〕

BMDP〔BMDP解析課題の連結処理〕

MINERVAの構成

MINERVAプログラムの構成図

MINERVAジョブの処理概略

出力メッセージ一覧表

MINERVA命令文一覧表

例題集

MINERVAプログラム・リスト

参考資料

Research Report

General Series No. 48

USERS GUIDE TO MINERVA

MINI-PACKAGE FOR EVALUATING

and

RATING THE VALIDITY OF SURVEY DATA

(VERSION - V01)

FEBRUARY 1980

by

THE SIXTH DIVISION

—THE INSTITUTE OF STATISTICAL MATHEMATICS—

TÔKEI-SÛRI KENKYÛZYO

Institute of Statistical Mathematics

4-6-7 Minami-Azabu, Minato-ku,

Tôkyô 106, Japan