

JAERI-M

7228

OLYMPUSシステムの導入と
プリ・プロセッサの開発

1977年8月

岡本 正雄・竹田 辰興・田中 正俊

浅井 清・中野 鴻^{*}・川上 一郎^{**}

日本原子力研究所
Japan Atomic Energy Research Institute

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

OLYMPUS システムの導入とプリ・プロセッサの開発

日本原子力研究所東海研究所核融合研究部

岡本正雄・竹田辰興・田中正俊

浅井 清⁺・中野 鴻^{*}・川上一郎^{**}

(1 9 7 7 年 7 月 2 7 日受理)

K.V.Roberts 等により開発された OLYMPUS システムを変換して原研計算センタに導入し、同システムを動かせるためのプリ・プロセッサを開発した。この OLYMPUS システムの概要を解説し、原研 OLYMPUS システム (J A E R I OLYMPUS SYSTEM) の説明と使用法を述べた。OLYMPUS システムは核融合研究やプラズマ物理におけるプログラム作成の標準化や、プログラム交換などに役立つものである。開発されたプリ・プロセッサは、原研 OLYMPUS システムにとって不可欠のものであるが、それ自体、ファイル操作、ファイル作成、ファイル修正などに役立つものである。

+ 日本原子力研究所東海研究所計算センタ

* センチュリ・リサーチ・センタ

** 客員研究員・日本大学理工学部物理学教室

OLYMPUS System and Development of its Pre-Processor

Masao OKAMOTO, Tatsuoki TAKEDA, Masatoshi TANAKA,
Kiyoshi ASA⁺I, Koh NAKANO* and Ichiro KAWAKAMI**

Division of Thermonuclear Fusion Research,
Tokai Research Establishment, JAERI

(Received July 27, 1977)

The OLYMPUS SYSTEM developed by K. V. Roverts et al. was converted and introduced in computer system FACOM 230/75 of the JAERI Computing Center. A pre-processor was also developed for the OLYMPUS SYSTEM. The OLYMPUS SYSTEM is very useful for development, standardization and exchange of programs in thermonuclear fusion research and plasma physics. The pre-processor developed by the present authors is not only essential for the JAERI OLYMPUS SYSTEM, but also useful in manipulation, creation and correction of program files.

Keywords: Computer Program, OLYMPUS System, Pre-processor, Thermonuclear Fusion Research, Program Manipulation

+ Computing Center, Tokai, JAERI

* Century Research Center Corporation, Nihon-bashi,
Chuo-ku, Tokyo

** Guest Staff of JAERI: Department of Physics,
Nihon University, Surugadai, Kanda, Tokyo

目 次

1. はじめに	1
2. OLYMPUS システムの概要	3
3. JAERI OLYMPUS システム	14
4. プリ・プロセッサー「UPDATE」	23
5. おわりに	35
謝　　辞	36
参考文献	36
付録 A	37
付録 B	65

1. はじめに

最近における、計算機コードの作成の為に払われる研究者の努力の急激な増加には目覚しいものがある。これは、計算機のための技術・科学の発達と、計算機を用いて行う計算の需要が歩調をそろえて伸びて来たためと言えよう。実際、計算速度、記憶容量の飛躍的増加やソフトウェア技術の発達のおかげで、プラズマ・核融合研究の分野のみに限っても、かつては夢のようだった多次元でかつ現実的幾何形状のプラズマ・シミュレーションが少くとも原理的には可能になって来た。この結果、計算コードの大型化、精密化をもたらし計算コードが広い範囲の専門化した研究者達によって作成される傾向を生み出して来た。このような傾向は、必然的に2つの問題を提起する。即ち、まず第一は、大規模で複雑な計算コードを作成する際に要求される生産性の向上に対する要求である。大規模計算コードを数人以上のチームで共同作成する場合はもちろんのこと、そうでなくとも、対象とする問題にふさわしい、ソフトウェア体系の中で計算コードを作成するならば、コード開発の能率上昇が期待できるであろう。第二の点は、コード開発努力の無意味な重複を避ける事と研究者間の情報流通の円滑化についての要求である。元来、FORTRAN, ALGOL, PL/1 のような言語は、計算機相互間の言語障壁を取り除きプログラムの交換を容易にすることがその一つの大きな使命であるが、実際には個々の計算コードをその作者以外の人が利用するのはかなり困難であるというのが実情である。これは、これら言語が汎用言語であることからして、やむを得ないことであるが、今日、計算コードの開発の全世界的協力の必要性が指摘されていることを考えれば、何らかの対策を講ずる必要がある。〔文献1, 2〕。

このような問題に応える方策には、唯單に、ソフトウェア開発技術の改善といった技術的問題をとりあげるだけでなく、研究者間の協力体制等制度的な問題を考慮することも重要であることは言うまでもない。しかし、純粹に技術的問題に関する改善によって多くの点が解決されるであろう。英國の Culham 研究所の Roberts らによって開発された OLYMPUS^{注1)} システムは、まさにこの観点に立って作られたシステムである〔3〕。このシステムは、もともとは次のような一般形であらわせる初期値問題を解く為に作られたものである〔3, 4〕。

$$\frac{\partial f}{\partial t} + G(f) = 0 \quad . \quad (1)$$

ここで、 G は線形、非線形演算子で f は解ベクトルである。言うまでもなく、この形の方程式は、流体力学、プラズマ物理学、天体物理学等多くの物理学の諸分野でしばしば現われてくる。後の章で詳述されるように、OLYMPUS システムは、計算コードの構造等について明確な規則を持っており、また、コード開発に際して有効に利用できる数多くの機能を備えている。プログラム構造を規定したり、これらの機能を働かせる上で重要な役割を果しているのが、標準的

注1) ギリシャ Thessaly の山、ギリシャの12神が住んでいたと伝えられる。

に用意されたファイルの集合である。これらのファイルは、主プログラム、サブプログラム、COMMON ブロック等を含んでいる。プログラム交換という点から重要な点は、個々の計算機システムに依存する諸量に関しては、プログラム作成者ではなく、OLYMPUS システム自体が設定するようになっている点である。従って、ひとたび OLYMPUS システムが、ある特定の計算機システムに組込まれたならば、チャネル番号、語長、文字コード等のように各計算機システムに固有の数値について意識せずに、コード作成や交換が可能になる。このようなわけで、OLYMPUS システムのもとで働くように作られた計算コード（“olympian”）は、計算機システムの種類を問わず、OLYMPUS システムの用意されているところならばどこに於ても、変更なしに使用できる。

OLYMPUS システムは、最初、Culham 研究所の計算機 ICL 4/70 用に作られたが、その後、IBM 360/370 シリーズ用、および CDC 6000/7000 シリーズ用に変換され、特にヨーロッパの各研究機関の間で計算コード開発と交換の為に実用に供されている。日本においては、著者の一人（川上）により、日本大学理工学部計算センタ、および東京大学大型計算機センタにすでに OLYMPUS システムが導入されている。OLYMPUS システムの新しい版や新たに開発された “olympian” は、Computer Physics Communication Library (CPC Library) を通じて交換されるのが普通である。すでに作成された代表的な “olympian”^{注1)} に、レーザ・コード “MEDUSA”^{注1)} [5] や、トカマク一次元輸送コード “ICARUS”^{注2)} [6]、ピンチプラズマの一次元コード “IRIS”^{注3)} [7] 等がある。

我々はプログラムの標準化の試みとして、上記 OLYMPUS システムを若干変更して、原研計算センタに導入した。OLYMPUS システムに必要なプリ・プロセッサは独自で開発した。本報告書は、OLYMPUS システムの概要を解説し、原研計算センタの OLYMPUS システム (JAERI OLYMPUS SYSTEM) の説明およびその使用法を述べる。JAERI OLYMPUS SYSTEMのために開発したプリ・プロセッサは、“olympian” をコンパイル可能なプログラムに変換する機能を持つのみでなく、OLYMPUS システムを原研計算センタ FACOM 230/75 で効率的に動かせるために、ファイル編集・結合を行なうユーティリティとして活用でき、さらに、OLYMPUS システムとは無関係に、ファイルの編集・結合・修正等を一般的に行うことのできるものである。

§ 2 ではオリンパス・システムの概要を解説し、§ 3 では JAERI OLYMPUS SYSTEM の説明を行いその使用法も述べ、「CRONUS」、「CYCLOPS」、「MINOS」のテスト・ランの結果を掲げる。§ 4 では、プリ・プロセッサ「UPDATE」のパラメータ、コマンド、使用法を説明し、さらにそのファイル形式についても簡単に述べる。§ 5 で全体のまとめと注意を与える。付録 A に、「CRONUS」や「CYCLOPS」などの OLYMPUS システムのプログラムリストを掲げる。このリストを読むだけでも、OLYMPUS システムは理解できる。プリ・プロセッサ「UPDATE」のソース・プログラム・リストを付録 B に掲げる。

注1) ギリシャ神 頭髪がへびでできていて見る者を石に化したといふ三人姉妹の女怪の一人。Perseus に殺された。

注2) ギリシャ神 ろうで翼をつけ空中高く昇ったが太陽に近づきすぎてろうがとけ海についらしくして死んだ。

注3) ギリシャ神 夜の神

2. OLYMPUS システムの概要

2.1 OLYMPUS システムの概要

OLYMPUS システムの概要を記すにあたって、まず、このシステムが作られるに至った背景をまとめてみて、このシステムによって問題点がどれ程解決し得るかについて考えてみよう。前章にも記したように、OLYMPUS システム開発の動機は、次の 2 点に要約できる。即ち、

- (1) 各国、各研究機関（各研究室においてさえ）の間で計算コード開発努力の重複がある。
研究的要素の強いコード開発においては、ある程度の重複は避けられないが、現在重複を避けられないものとしている理由はむしろ各研究者が独自のプログラム構成規則のもとでプログラムを構成しているので情報伝達の非能率性から研究者間の協力がしにくくなっている事にある。
- (2) 通常、一人一人の研究者が扱っている計算コードは、かなり狭い範囲の問題に限られているが、研究の対象をかなり広い範囲にひろげても、計算コードの形式には似たような問題が沢山ある。これらは、FORTRAN のような汎用計算機言語が対象とし得る問題の中でごく特殊な一部を為している。従って、扱い得る問題の種類を狭く限定し、汎用言語を限定された問題向き言語に改良する（例えば FORTRAN 言語の制限をきびしくする）ことによってコード作成の能率向上が期待できる。

OLYMPUS システムではこれらの点に対し、次のように応えている。

(a) 説明文の完備

olympian の作成にあたって守るべき文書化法を詳しく規定している。たとえば、コメント・カードを多数挿入する事によって、コード自身一つの科学上の論文のように理解できるものとするなどである。さらに、各ステートメントは、その機能によって書き出しのコラムの位置に制限を加えプログラム・リストを視覚的に理解しやすいものとする。

(b) 異種計算機間のコード交換の容易さ

数値計算の為の汎用言語である FORTRAN においては、各計算機毎に便利な“方言”が作られたが、OLYMPUS システムでは、利用目的に応じた“方言”を作る一方、計算機毎の特殊性については、各プログラム作成者は意識しないでもすむようにしてある。

(c) 記号、構造の標準化およびユーティリティの整備

プログラム構造の標準化が、OLYMPUS システムの中心を為す部分である。標準化によって、プログラム作成時の自由度が減るが、きまりきった手続を簡略化できるのでプログラム作成の能率化がはかれる。これらは、具体的には、次の 4 点に要約できる。

- 1) 計算コードのプログラム構造を明確に規定してある。

- II) サブプログラム間のデータの受け渡しは、原則として COMMON 変数によるものとし、COMMON が利用しやすいような配慮がほどこされている。
- III) サブプログラム "EXPERT" および "REPORT" の導入。
- IV) ユーティリティ・プログラムの整備。

これら I) ~ IV) については、2.2 ~ 2.5において、詳しく説明する。なお、更に詳しい説明については文献 [4] を参照されたい。

2.2 OLYMPUS システムの構造

OLYMPUS システムの最も大きな特長は、プログラムが、あらかじめ用意された明確な構造を持つように構成される事である。

基本的には、個々のプログラムは、"DATA" と、それらに作用する "INSTRUCTIONS" の集合体として理解する事が可能である。即ち、

```
<PROGRAM> ::= <INSTRUCTIONS><DATA>
<INSTRUCTIONS> ::= subprograms
<DATA>      ::= COMMON blocks
```

ここで、各 subprogram は、その機能によって、0, 1, 2, 3, 4, 5, U の class に分類される（第1表）。それぞれの class に含まれる個々の subprogram は、各 class 每に subprogram 番号が与えられる。この結果、全ての subprograms は 2 つの数字 < I, J > によって識別される。例えば、

```
<0, 0>    class 0, subprogram 0 ..... MAIN,
<0, 3>    class 0, subprogram 3 ..... COTROL.
```

個々の副プログラムのうちで、全体の制御、入力、初期設定等一般的な操作に関するものは、その副プログラム相互間の関係が固定されている（第1図）。これらの副プログラムのうちには、相互関係のみならず、その内容も予め与えられているものもある。このような OLYMPUS システムの構造のみを与えて実際の計算を行わないプログラムが、null program "CRONUS"^{注1)} である（付録 A にそのリストを掲げる）。このプログラムは、第2図に示すような構造を持っており、OLYMPUS システムの構造を与えるものである。このプログラムを実行させると第2図に示した流れに従って制御が移行し、それに従って制御の移ったサブルーチンの名称のみが、出力される（文献 [4] の p 249 または第5図）。利用者の作成したプログラムは、"CRONUS" に含まれるダミー・サブプログラムを実際に必要なプログラムに置き換える事によって実行される。

注1) ギリシャ神 Titan (巨人)の一人、Zeus の父、Zeus に退けられるまでは宇宙の支配者であった。
(ローマ神話では Saturn)

2.3 サブプログラム間のデータ受け渡し

OLYMPUS システムでは、サブプログラムと同様、データに関しても分類整理が行われ、形式的な構造が与えられている。この為、全てのデータは可能な限り、COMMON変数としてサブプログラム間で受け渡される。サブルーチンの引数としてデータの授受を行うとこのようないくつかの分類が著しく困難となり、その結果、データ構造の明瞭度を損うからである。COMMON ブロックは、第2表に示すような8つのグループに分類され、グループに対応する番号 r とブロックに対応する番号 s によってサブプログラムと同様に、[Cr.s] と表示される。例えば、

[C1.1] COMBAS 基本的システム・パラメータ
 [C1.9] COMDDP 開発診断用パラメータ

COMMON ブロックのラベルは、全て、COM で始まる名称を有する。グループ 1 の COMMON ブロックは OLYMPUS システムを構成する為のシステム用データの為に使われる。現在、COMBAS と COMDDP が用意されているが、これらの内容は第3表及び第4表に示されている。

OLYMPUS システムにおける COMMON の取扱いに関して最も重要な事は、各サブプログラムにおいて必要な COMMON ブロックを完全なフォートラン文の形式で記述する必要がない事である。プログラム全体の前に、COMMON ブロックを完全な形で登録しておけば、各サブプログラムの中では、*CALL で、COMMON のラベルを呼ぶ事によって COMMON 文が生成される。これは、COMMON 変数を変更したい場合にきわめて有効な方法である。なお、COMMON 文の生成に関する機能だけ持ったプリ・プロセッサは既に FACOM 230/75 用に用意されている〔8〕。

COMMON 変数に関する指摘しなければならない事は、各変数についての命名法の規則である。OLYMPUS システムにおいては、各変数名の頭文字によって変数の機能を分類する事を推奨している(第5表)。この命名法をとる事によって、ある変数が COMMON 変数か、局部変数かの区別が直ちにつき、またその変数の型もすぐわかり、プログラムの拡張などの際によくおかず誤りがさけられるようになる。

2.4 "EXPERT" および "REPORT"

標準化を進める事によってプログラムを他人にもわかりやすいものにする事が、OLYMPUS システムの一つの目標ではあるが、標準化推進を余り徹底しすぎるとプログラム開発過程における柔軟性がなくなる危険性がある。また、出来上ったプログラムを利用する際に付け加えたごくわずかの変更に対して、その変更の数だけの新しい版のプログラムができる事になる。このような事を避ける為、プログラム作成時に重要と思われる点に、外部ループを挿入する為の手法がサブプログラム "EXPERT" によって実現される。EXPERT は、呼ばれた点を指定する情報、即ち、KCLASS(サブルーチンの class 番号)、KSUB(サブルーチン番号)、KPOINT(呼ばれた点を指定する番号) を引数とするサブルーチンであって、標準的には何もせずに呼ばれた点へもどるダミー・サブルーチンである。上に記した 3 つの引数によって、

EXPERTが呼ばれた点が一意的に定まるので、これらの引数の値を使って判断し必要な処理を行って戻るならば、基本的なプログラムの構造には手を触れずにプログラムの修正や変更が可能になる。具体的には、第3図に示したようにEXPERTを構成すればよい。

EXPERTは、class 0に属するサブプログラムであるが、診断用サブプログラムの属するclass5にも、同様の機能を持つサブプログラムREPORTが備えられている。EXPERT, REPORTおよびCYCLOPSユーティリティ(次節参照)等による標準的な診断用プログラム構成を第4図に示す。ここで、CLIST, ARRAYSはclass5に属する診断用サブルーチンで、それぞれ指定されたCOMMONブロックのCOMMON変数とCOMMON array変数の内容を出力するものである。

2.5 ユーティリティ・プログラムの整備

コーディングの際にしばしば生ずる問題は、出力フォーマットを定めたり、行列の初期設定を行ったり、単純ではあるが煩雑で、また、相互によく似ている操作を何度も行わなければならぬ事である。このような事をなるべく避ける事によってプログラム作成の効率化を企りました、標準化によってプログラムを見やすくするため、OLYMPUSシステムではclass U(CYCLOPS)^{注1)}を設けている。現在の所、class Uに含まれているサブプログラムは27個であって、これらは、出力用、配列操作用、計時用、診断用サブルーチンから構成されている(第6表)。出力用サブルーチンを例にとれば、これらは、「改頁する」、「線を引く」等のように簡単なものであるが、これらのユーティリティ・プログラムを活用する事によって、コーディングに際しての煩雑な単純作業が大巾に節減できるであろう。

注1) ギリシャ神 一つ目の巨人

第1表 OLYMPUS システムの副プログラムの分類

Classification of OLYMPUS subprograms.

-
- 0. Control
 - 1. Prologue
 - 2. Calculation
 - 3. Output
 - 4. Epilogue
 - 5. Diagnostics
 - U. CYCLOPS utilities
-

To allow for future expansion we reserve further classes:

- | | |
|----------------|--------------------------------|
| 6, 7, N-T, V-Z | for general OLYMPUS use |
| 8, A-F | for individual programs |
| 9, G-M | for use by local installations |

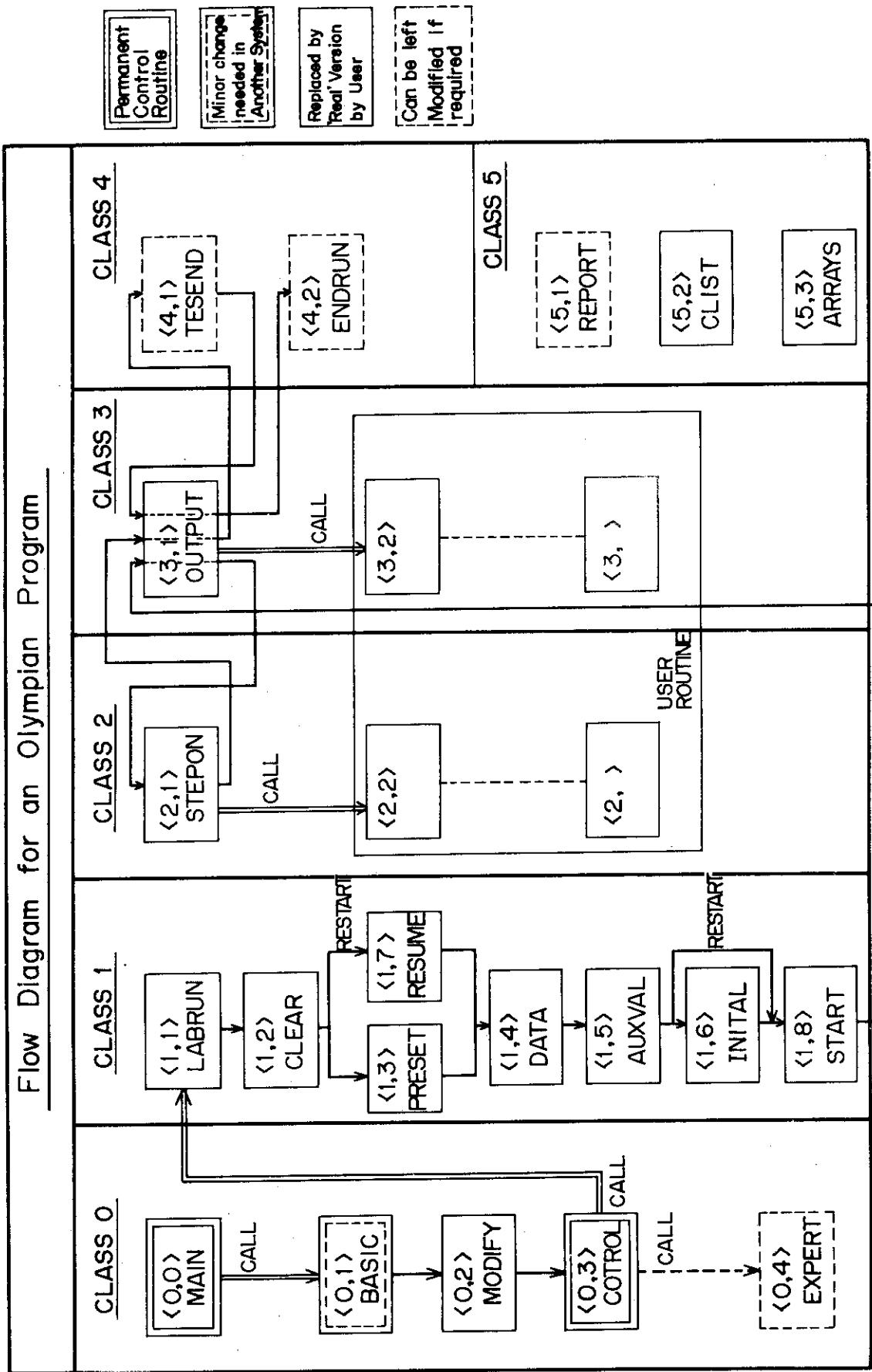
(Reprint from J.P. Christiansen and K.V. Roberts: Computer Phys. Commun. 7(1974)245.)

第2表 名前つき COMMON群

Labelled common groups.

-
- 1. General OLYMPUS data
 - 2. Physical problem
 - 3. Numerical scheme
 - 4. Housekeeping
 - 5. Input-output and diagnostics
 - 6. Text manipulation
 - 7. Arbitrary purposes by an individual programmer
 - 8. General purpose blocks introduced by a local installation
-

(Reprint from J.P. Christiansen and K.V. Roberts: Computer Phys. Commun. 7(1974)245.)



第1 図 OLYMPUS システムのもとでの FORTRAN プログラムの構成

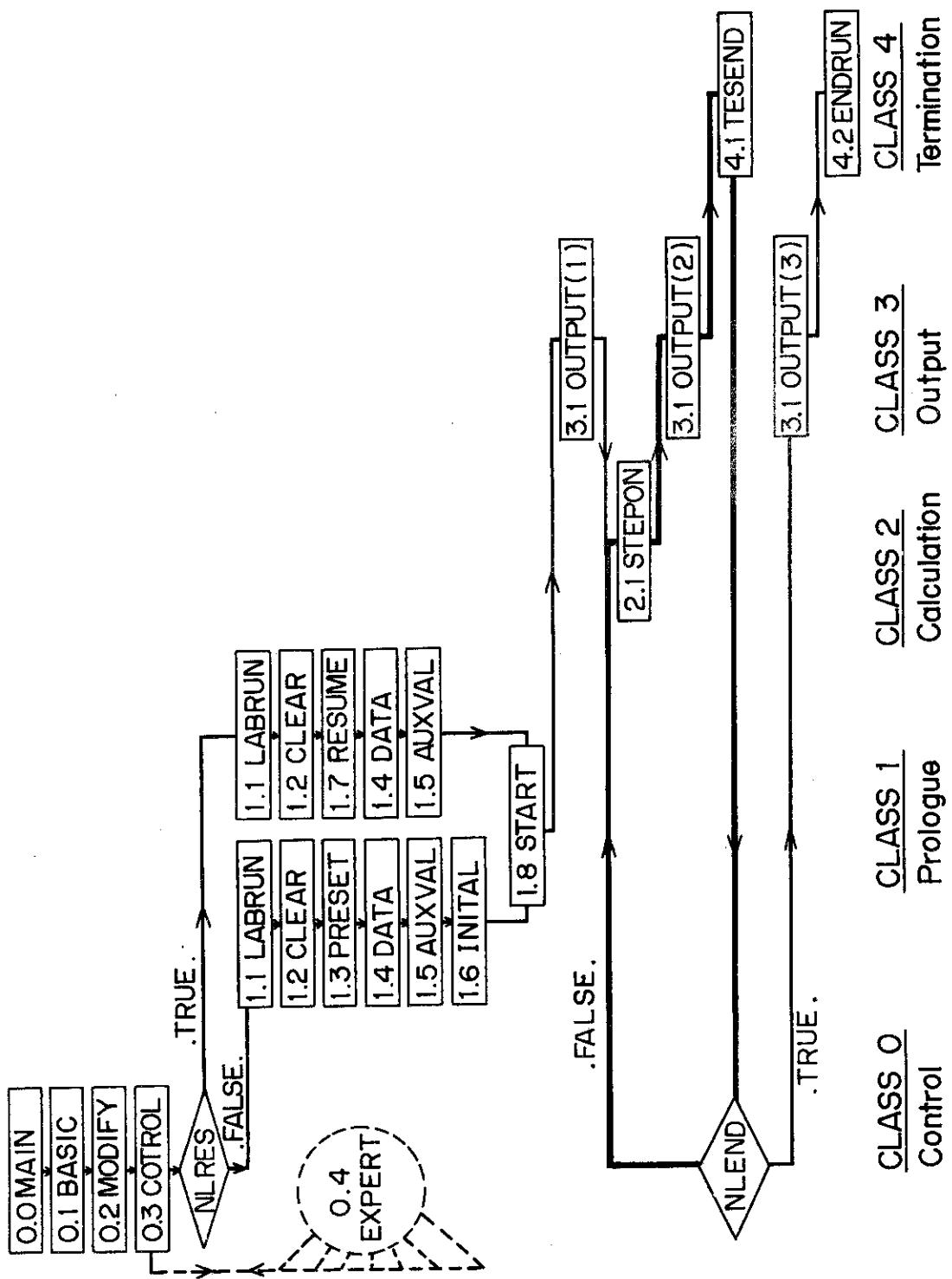


Fig. 1. Flow diagram for CRONUS. The thick line indicates the main loop of the step-by-step calculations.
(Reprint from J.P. Christiansen and K.V. Roberts : Computer Phys. Commun. 7 (1974) 245.)

第3表 COMMON文「COMBAS」の内容。〔C1.1〕に属し、基本的システム・パラメータを与える。

[C1.1] COMBAS basic system parameters.

Name	Type	Dimension	Purpose	Preset value
Variable names in version 1A				
ALTIME	R		CPU time allocated for job,(secs)	Supervisor Call
CPTIME	R		CPU time used so far,(secs)	0.0
NLEDGE	I		Channel for restart records	30
NLEND	L		.TRUE. if run to be terminated	.FALSE.
NLRES	L		.TRUE. if run being resumed	.FALSE.
NONLIN	I		Channel for on-line input/output	1
NOUT	I		Current output channel	NPRINT
NPRINT	I		Channel for printed output	99
NREAD	I		Channel for card input,(or equivalent)	97
NREC	I		Current record number	1
NRESUM	I		Resume from record on this channel	NLEDGE
NSTEP	I		Current step number	0
STIME	R		Start time,(secs)	0.0
Variable and array names added in version 2A				
LABEL 1	IA	12	Labels used to describe the run set by program in LABRUN	BLANK
LABEL 2	IA	12		
LABEL 3	IA	12		
LABEL 4	IA	12		
LABEL 5	IA	12	Labels available to programmer	
LABEL 6	IA	12		
LABEL 7	IA	12	Labels reserved for system use	
LABEL 8	IA	12		
NDIARY	I		Channel for diary	NPUNCH
NIN	I		Current input channel	NREAD
NPUNCH	I		Channel for punched card output (or equivalent)	98
NRUN	I		Maximum number of steps	1

(Reprint from J.P. Christiansen and K.V. Roberts : Computer Phys. Commun. 7(1974)245.)

第4表 COMMON文「COMDDP」の内容。〔C1.9〕に属し、開発診断用のパラメータを与える。

[C1.9] COMDDP development and diagnostic parameters.

Name	Type	Dimension	Purpose	Preset value
MAXDUM	I		Maximum dimension of dump arrays	20
MXDUMP	I		Actual dimension of dump arrays	10
NADUMP	IA	20	Codes for array dumps	0
NCLASS	I		Most recent class reported	0
NPDUMP	IA	20	Codes for dumping points	0
NPOINT	I		Most recent point reported	0
NSUB	I		Most recent subprogram reported	1
NVDUMP	IA	20	Codes for dumping variables	0
NLCHED	L		.TRUE. if report heads needed for control class 0	.FALSE.
NLHEAD	LA	9	.TRUE. if report heads needed for class 1-9	.FALSE.
NLOMT1	LA	50	.TRUE. if subprogram in class 1 to be omitted	.FALSE.
NLOMT2	LA	50	" " " " " " " " " "	.FALSE.
NLOMT3	LA	50	" " " " " " " " " "	.FALSE.
NLREPT	L		.TRUE. if any reports required	.FALSE.

(Reprint from J.P. Christiansen and K.V. Roberts : Computer Phys. Commun. 7(1974)245.)

第5表 OLYMPUS システムにおける変数命名法

Initial letters and array names

	Real, complex	Integer (and Hollerith)	Logical
Subprogram dummy arguments	P	K	KL
Common variable and array names	A-H,O,Q-Y	L,M,N	LL,ML,NL
Local variable and array names	Z	I	IL
Loop indexes		J	

(Reprint from K.V. Roberts : Computer Phys. Commun. 7(1974) 237.)

```

SUBROUTINE EXPERT(KCLASS,KSUB,KPOINT)
.....Title, common block.....
NCLASS = KCLASS
NSUB = KSUB
NPOINT = KPOINT
IF (NLREPT) CALL REPORT(KCLASS,KSUB,KPOINT)
RETURN
END
.....
```

```

      ICODE=10000 * KCLASS+100*KSUB+KPOINT
      IF(ICODE.EQ.22102) GO TO 22102
      IF(ICODE.EQ.30311) GO TO 30311
      RETURN
22102 CONTINUE
      (Statements for point 2.21.2)
      RETURN
30311 CONTINUE
      (Statements for point 3.3.11)
      RETURN
      END

```

(a)

(b)

(Reprint from J.P. Christiansen and K.V. Roberts : Computer Phys. Commun. 7(1974) 245.)

第3図 EXPERT のコーディング

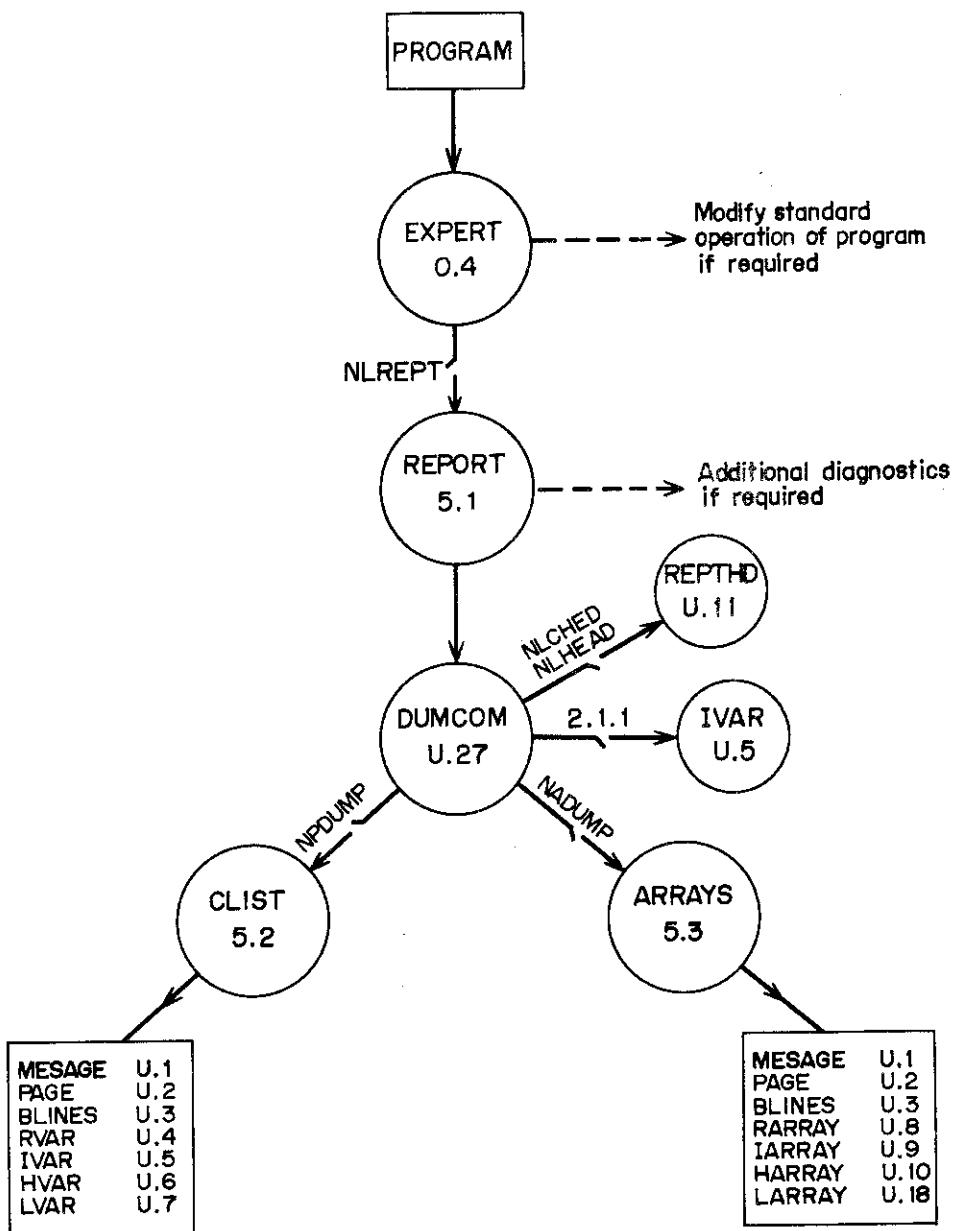


Fig.4. Standard diagnostic and control facilities.

(Reprint from J.P.Christiansen and K.V.Roberts : Computer Phys. Commun. 7(1974)245.)

第4図 標準的な診断用プログラム構成

第6表 ユーティリティ CYCLOPS JIC用意されているプログラム群

CYCLOPS CLASS U: subprogram names and identification numbers.

Name	No.	Dummy arguments	Title
MESSAGE	U. 1	KMESS	Print 48-character message on output channel
PAGE	U. 2	K	Fetch new page on output channel
BLINES	U. 3	KNAME, PVALUE	Insert blank lines on output channel!
RVAR	U. 4	KNAME, KVALUE	Print name and value of real variable
IVAR	U. 5	KNAME, KVALUE	Print name and value of integer variable
HVAR	U. 6	KNAME, KIVAL	Print name and value of Hollerith variable
LVAR	U. 7	KNAME, PA, KDIM	Print name and value of logical variable
RARRAY	U. 8	KNAME, KA, KDIM	Print name and values of real array
IARRAY	U. 9	KNAME, KA, KDIM	Print name and values of integer array
HARRAY	U. 10	KNAME, KA, KDIM	Print name and values of Hollerith array
REPTHD	U. 11	KCLASS, KSUB, KPOINT	Print heading for diagnostic report
RUNTIM	U. 12		Update CPU time (secs) and print it
DAYTIM	U. 13		Print date and time
RESETR	U. 14	PA, KDIM, PVALUE	Reset real array to specified value
RESETI	U. 15	KA, KDIM, KVALUE	Reset integer array to specified value
RESETH	U. 16	KA, KDIM, KVALUE	Reset Hollerith array to specified value
JOBTIM	U. 17	PTIME	Fetch allocated jobtime, (secs)
RESETL	U. 18	KLA, KDIM, KLVAL	Reset logical array to specified value
LARRAY	U. 19	KNAME, KLA, KDIM	Print name and values of logical array
RARRAY2	U. 20	KNAME, PA, KDIMX, KX, KY	Print doubly-subscripted real array
SCALER	U. 21	PA, KDIM, PC	Scale a real array by a real value
SCALEI	U. 22	KA, KDIM, KC	Scale an integer array by an integer value
COPYR	U. 23	PA1, K1, PA2, K2, KDIM	Copy one real array into another
COPYI	U. 24	KA1, K1, KA2, K2, KDIM	Copy one integer array into another
SIGNR	U. 25	PA, KDIM	Change the sign of a real array
SIGNI	U. 26	KA, KDIM	Change the sign of an integer array
DUMCOM	U. 27	KCLASS, KSUB, KPOINT	Dump selected common blocks

* Additions for Version 2.

(Reprint from J.P. Christiansen and K.V.Roberts : Computer Phys. Commun. 7(1974)245.)

3. JAERI OLYMPUS システム

本報告書で紹介する OLYMPUS システムは、CPC Library を通じて入手した CDC 版の OLYMPUS システムを、原研計算センタの FACOM 230/75 計算機システムにおいて利用可能な形に変換したものである。OLYMPUS システムを動かせるのに必要なプリ・ブロセッサ（プログラム「UPDATE」と命名）は新たに開発したが、OLYMPUS システムの核を成す、"CRONUS" と "CYCLOPS" に関しては、プログラムを変更した個所は、極くわずかである。（日付、時刻、cpu 時間を測るルーチンの変更、計算機による語長の違いの修正等のみ。）従って、ここで "JAERI" OLYMPUS と呼ぶのは便宜上のことにしかすぎない。

はじめに、"CRONUS" のテスト結果を第5図に掲げる。"CRONUS" は第2図に示されるような構造とプログラム単位から成っており、ダミのルーチンを多く含むがこれ自体1つのcompleteプログラムを形成しており、"CRONUS" をテストすることにより、先ず OLYMPUS システムが試される。第5図の結果は、文献〔4〕のFig. 2 と比較されたい。

以下、"olympian" の組み方および実行のさせ方について最も基本的な事柄のみに限って説明する。

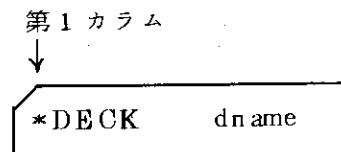
3.1 "olympian" の書き方

① プログラムの流れ・構造

§2で述べたように、これは、第2図に従わねばならない。第2図のプログラム単位（主および副プログラム）は、それぞれ役割が決っており、"CRONUS" ではその多くがダミであるが、各副プログラムに内容を与える。副プログラム名は第2図のものと同じでなければならない。これらの副プログラムの中で更に新しい副プログラムをCALLしてもよいことは言うまでもない。"CYCLOPS" のユーティリティプログラムは、いつどこで呼んでもよい。その他細かい規則は、§2で述べた通りである。

② デック

作成されたプログラムをいくつかに分割してそれぞれにデック名をつける。（デックとは、FACOM 230/75 でいうエレメントと考えてよい。）普通各副プログラムを1つのデックとして扱うと便利であるが、いくつかの副プログラムを1つのデックとしてもよい。デック名はデックの先頭に次のような形で書く。



ここで dname は8文字以内の英数字である。1つのデックの最後は END 文で終らせないと

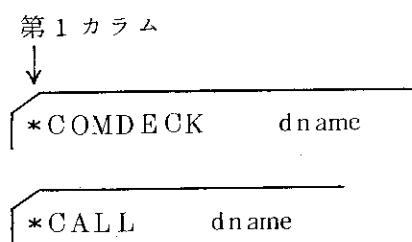
(END 文と*DECKの間に文があってはいけない。)コンパイル時にエラが生ずることがある。第6図を参照のこと。

(3) コモンデック

COMMON文をコモンデックに登録すること。第6図を例にとって説明すると、COMABCという名前のコモンデック(COMDECK)の内容は、

COMMON//COMABC//A, B(10), C(5, 100)

である。OUTPUTというデックの中のサブルーチン、OUTPUT(K)の中で*CALL COMABCによりCOMABCという名のCOMDECKの内容すべてが呼ばれる。この処理をするのがブリ・プロセッサであり、ブリ・プロセス後、*CALL COMABCのところは、COMMON//COMABC//A, B(10), C(5, 100)に置きかえられる。*COMDECKおよび*CALLは次のように書く。



ここでdnameはコモンデック名で8文字以内の英数字から成る。

OLYMPUSシステムではCOMMON文はすべて、COMDECKに登録して、*CALLで呼び出すのが通例である。このようにすると、同一名のCOMMON文の中の変数名は、これが呼ばれるルーチンですべて一致することになる。しかしこれがどうしても守りたくない時、例えば

```

SUBROUTINE STEPON
.....
COMMON//COMABC//A, B(10), C(5, 100)
.....
END
SUBROUTINE DATA
.....
COMMON//COMABC//A, X(10), Y(5, 100)
.....
END
    
```

としたい時(FORTRANではこれが許される)、*CALLを使わずに、直接COMMON文を書いてもよい。また、COMDECKに登録されるCOMMON文は1つでなくとも2つ以上でもよいし、またCOMMON文に限らずどんな文を登録してもよい。第6図の例では、SUBR-

OUTINE OUTPUT の中の *CALL MESA はプリ・プロセス後
 CALL MESAGE(48H JAERI OLYMPUS SYSTEM)
 におきかえられる。

尙、ユーザーは、[C1, 1]のCOMBASSと[C1, 9]のCOMDDPの内容は書いて登録する必要がない。この2つのCOMDECKはランの時自動的に呼ばれる。

以上の他、細かい規則は§2または付録Bを参照すること。しかし細かい規則は守らなくても“olympian”は実行できる。また③をやめればプリ・プロセッサなしで実行できる。

3.2 “olympian” の実行

3.1に従って作成された“olympian”を原研計算センタFACOM230/75のシステムで実行する方法を述べる。“CRONUS”と“CYCLOPS”はすでにプリ・プロセッサで前処理をした後コンパイルされ、RBファイルに保存されている。ユーザ作成の“olympian”は、第7図に従って処理される。はじめに“olympian”とCOMMON LIBRARY(COMBASS, COMDDP)を合せて1つのLIBE 8ビットモードファイルを作成し、これを9ビットモードに変換する。プリ・プロセッサを通ったあとコンパイルされRBファイルを作成する。“CRONUS”と“CYCLOPS”はすでに別のRBファイルに保存されているので、これら2つのRBファイルを結合し編集する。この時、“CRONUS”の中のダミのルーチンは“olympian”的のルーチンに置きかえられる。こうしてできたEB形式のプログラムは、データを与えると実行される。第8図に“olympian”的の1つである“MINOS”を例に取つて原研計算センタFACOM230/75システムでのジョブ制御文の並びを書いた。“MINOS”はK.V.Robertsによって書かれたOLYMPUSシステムのテストプログラムでその詳細は、文献[4]に書かれている。我々の“MINOS”的のテストの結果は、第9図に示す。文献[4]の“MINOS”的のテストの結果と比較されたい。

OLYMPUSシステムに必要なプログラムは、現在次のファイルに登録されている。

◦ J3033.OLYMPUS ソース形式

◦ ELM=COMMENT(コメント)

◦ ELM=COMMON(COMBASS, COMDDP)

◦ ELM=CRONUS

◦ ELM=CYCLOPS

◦ J3033.CRONUSR RB形式 CRONUS, CYCLOPS

◦ J3033.UPDATE EB形式 プリ・プロセッサ

J3033.OLYMPUSのリストを取りたい時は次のようにするとよい。

¥NO

¥GJOB

¥LIBE

PRINT DDOLD0

FIN

¥DISKTO DDOLD0, J3033.OLYMPUS

OLYMPUS SYSTEM
 JAERI VERSION 1 APRIL 15, 1977
 0,1 BASIC CONTROL DATA SET
 0,2 STANDARD DATA ACCEPTED
 DATE 77-07-14 TIME 14:30:14
 0,3 ENTER RUN CONTROL
 1,1 LABEL THE RUN
 1,2 CLEAR VARIABLES AND ARRAYS
 1,3 SET DEFAULT VALUES
 1,4 DEFINE DATA SPECIFIC TO RUN
 1,5 SET AUXILIARY VALUES
 1,6 DEFINE PHYSICAL INITIAL CONDITIONS
 1,8 START OR RESTART THE RUN
 3,1 CONTROL THE OUTPUT
 CALL = 1
 INITIAL OUTPUT (1)
 2,1 STEP ON THE CALCULATION
 3,1 CONTROL THE OUTPUT
 CALL = 2
 PERIODIC OUTPUT (2)
 4,1 TEST FOR COMPLETION OF RUN
 3,1 CONTROL THE OUTPUT
 CALL = 3
 FINAL OUTPUT (3)

4,2 TERMINATE THE RUN

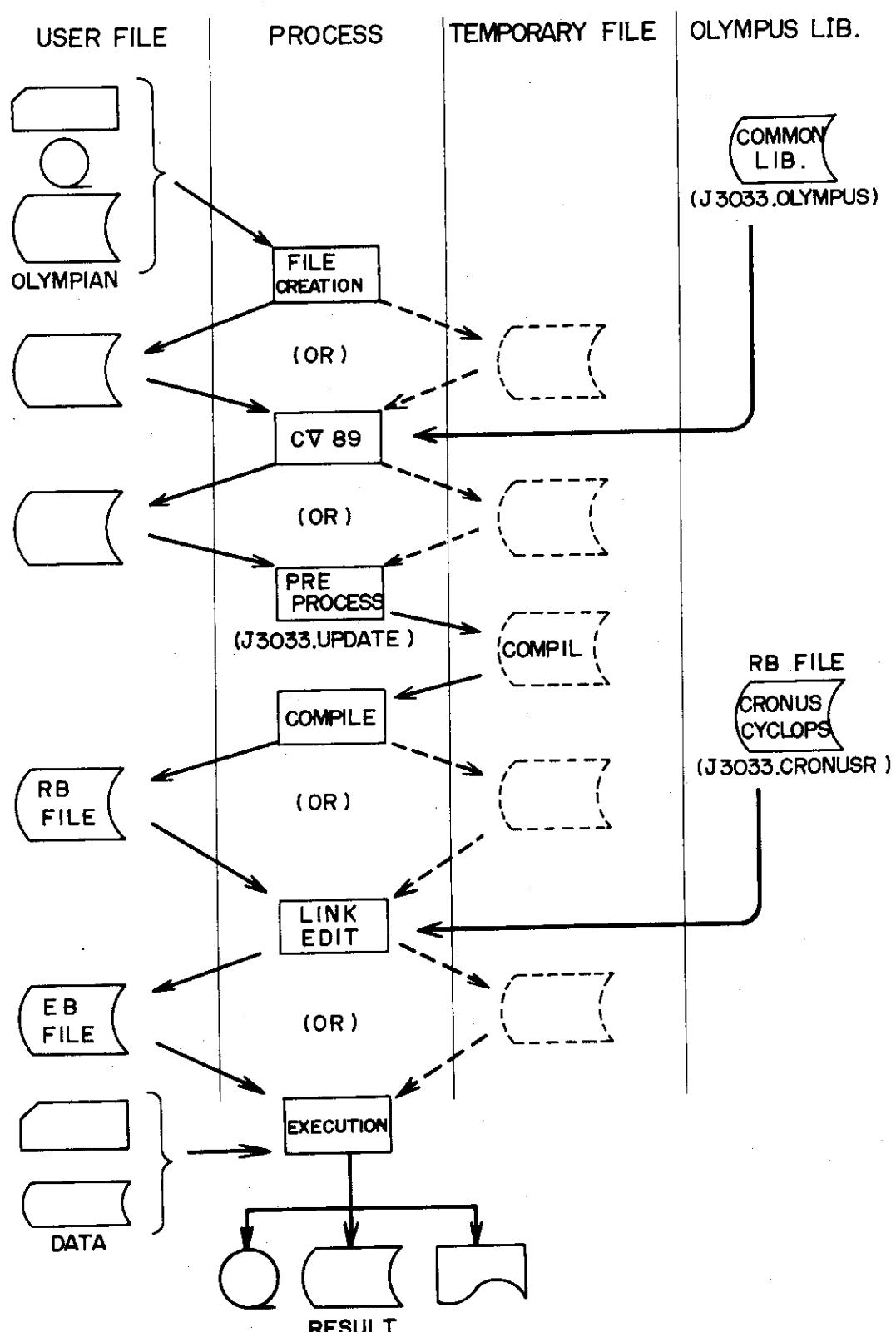
DATE 77-07-14 TIME 14:30:14
 CPU TIME USED SO FAR = 1,4000E-02 SECS

第5図 「CRONUS」のテスト結果

```
*COMDECK COMABC
  COMMON/COMABC/ A, B(10), C(5,100)
*COMDECK COMXYZ
  COMMON/COMX/ X1,X2
  COMMON/COMY/ Y1
  COMMON/COMZ/ Z1,Z2(100)
  DOUBLE PRECISION Y1,Z1
*COMDECK MESA
  CALLMESSAGE(48H JAERI OLYMPUS SYSTEM
  .....
  .....
*DECK STEPON
  SUBROUTINE STEPON
  .....
  .....
*CALL COMBAS
*CALL COMXYZ
  .....
  .....
  END
*DECK OUTPUT
  SUBROUTINE OUTPUT(K)
  .....
  .....
*CALL COMBAS
*CALL COMDDP
*CALL COMABC
  .....
  .....
*CALL MESA
  END
  .....
  .....
  .....
```

第6図 olympian の書き方

JAERI OLYMPUS SYSTEM



第7図 JAERI OLYMPUS SYSTEMのシステム構成

*NO N515, /
 P,0/PCH 0
 C,1/CORE 64
 T,2/TIME 1M

*GJOB 2113033,M.OKAMOTO,411.12,VLASOVMD /00331
 *LIBE
 EDIT DDNEW0,DDOLD1(COMMON),DDOLD2
 CV89 DDNEW0,DDOLD0
 FIN
 *LIBEDISK DDNEW0,WORK9
 *LIBEDISK DDOLD0,WORK8
 *DISKTO DDOLD1,J3033,OLYMPUS
 *DISKTO DDOLD2,J3033,MINOS
 *HRUN EFNAME=J3033,UPDATE
 *TPDISK F09,COMPIL,TMOD=9,RSIZE=80,BSIZE=800
 *LIBEDISK F11,WORK9
 *DISK F08
 *DISK F98
 *DISK F99
 *DATA
 *UPDATE COMPIL=9 *END
 *HFORT TFNAME=COMPIL,RFNAME=J3033,MINOSR
 *LIBE NAME=2
 *DISKTN DDNEW0,J3033,OLYMPIAN
 *DISKTO DDOLD0,J3033,CRONUSR
 *DISKTO DDOLD1,J3033,MINOSR
 *HLIEDRUN RFNAME=J3033,OLYMPIAN,ENTRY=MINOS
 *DATA
 *JEND

第8図 JAERI OLYMPUS SYSTEM 実行のためのジョブ制御文の並び。この例では「MINOS」が olympian である。

C1.1. BLOCK COMBAS

```

ALTIME = 0.0
CPTIME = 0.0
STIME = 1.0000E+00
NLEDGE = 30
NONLIN = 1
NOUT = 6
NPRINT = 6
NREAD = 5
NREC = 1
NRESUM = 30
NSTEP = 0
NDIARY = 6
NIN = 5
NPUNCH = 7
NRUN = 2
NLEND = .FALSE.
NLRES = .FALSE.

```

C1.9. BLOCK COMDOP

```

MAXDUM = 20
MXDUMP = 10
NCLASS = 0
NPOINT = 13
NSUB = 3
NLCHED = .FALSE.
NLREPT = .TRUE.

```

C4.1. BLOCK COMTES

```

XSCALE = 5.5000E-01
MAXTES = 12
MAXTX = 5
MAXTY = 5
N1 = 7
N2 = 4
N3 = 3
N4 = 4
N5 = 6
NDIM = 8
NDX = 2
NSCALE = 2
NX1 = 7
NX2 = 5
NX3 = 4
NX4 = 4
NX5 = 2

```

C4.1. BLOCK COMTES

XTEST1

1.234E+00							
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

XTEST2

5.678E+00							
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

XTES
TD

9.012E+00	9.012E+00

NTEST1

1	12345	12345	12345	12345	12345	12345	12345
---	-------	-------	-------	-------	-------	-------	-------

NTEST2

67890	67890	67890	67890	67890	67890	67890	67890
-------	-------	-------	-------	-------	-------	-------	-------

NHTEST

ABCD	ABCD	ABCD	ABCD
------	------	------	------

NLTE
ST

T	T	T	T	F	F	F	F
---	---	---	---	---	---	---	---

STEP = 1

NHTEST 1 =ABCD
NHTEST 1 =EFGH

JAERI-M 7228

C4.1. BLOCK COMTES

XTEST1

6.787E-01 6.787E-01 6.787E-01 6.787E-01 1.234E+00 1.234E+00 1.234E+00 1.234E+00

XTEST2

5.678E+00 5.678E+00 5.678E+00 5.678E+00 1.234E+00 1.234E+00 0.0 0.0

XTES
TD9.012E+00 9.012E+00
9.012E+00 9.012E+00
9.012E+00 9.012E+00
9.012E+00 9.012E+00
9.012E+00 9.012E+00

NTEST1

2 24690 24690 24690 12345 12345 12345 12345

NTEST2

67890 67890 67890 12345 12345 0 67890 67890

NHTEST

EFGH ABCD ABCD ABCD

NLTE
STT T T F F F F
CPU TIME USED SO FAR = 8.7000E-02 SECS
DATE 77-07-21 TIME 16:22:54
4.1 TEST FOR COMPLETION OF RUNSTEP = 2
*****NHTEST 1 =EFGH
NHTEST 1 =EFGH
CPU TIME USED SO FAR = 9.1000E-02 SECS
DATE 77-07-21 TIME 16:22:54
4.1 TEST FOR COMPLETION OF RUN

C4.1. BLOCK COMTES

XTEST1

-6.787E-01 -6.787E-01 6.787E-01 6.787E-01 1.234E+00 1.234E+00 1.234E+00 1.234E+00

XTEST2

5.678E+00 5.678E+00 5.678E+00 5.678E+00 1.234E+00 1.234E+00 0.0 0.0

XTES
TD9.012E+00 9.012E+00
9.012E+00 9.012E+00
9.012E+00 9.012E+00
9.012E+00 9.012E+00
9.012E+00 9.012E+00

NTEST1

-2 -24690 -24690 -24690 -12345 -12345 12345 12345

NTEST2

67890 67890 67890 12345 12345 0 67890 67890

NHTEST

EFGH ABCD ABCD ABCD

NLTE
ST

T T T T F F F F

4.2 TERMINATE THE RUN

DATE 77-07-21 TIME 16:22:54
CPU TIME USED SO FAR = 1.0900E-01 SECS

4. プリ・プロセッサー「UPDATE」

前節で述べたように、オリンパス・システムでは、プリ・プロセッサーに通すことにより、オリジナル・プログラムが、コンパイル可能なプログラムに変換される。プリ・プロセッサーは*CALLで呼ばれたCOMDECKの内容を、呼ばれた所に挿入する。我々の開発したプリ・プロセッサー「UPDATE」は、この機能を有するのみではなく、オリンパス・システムを原研計算センターFACOM 230/7-5で有効に動かせるためにファイル編集、結合を行うユーティリティとして活用でき、さらに、オリンパス・システムとは無関係に、ファイルの編集・結合・修正等を一般的に行うことのできるものである。この節では、プリ・プロセッサー「UPDATE」の機能の説明を行い、また使用法について簡単に説明する。最後にプログラム「UPDATE」の構造について述べる。また「UPDATE」のプログラム・リストは付録Bに掲げる。

プリ・プロセッサー「UPDATE」の機能を大別すると、(i)ファイル操作、(ii)ファイル作成、(iii)ファイル修正、の3つに分けられる。プリ・プロセッサーのシステムの概略は第10図のとおりである。

4.1 プリ・プロセッサー「UPDATE」のパラメータ

NAMELIST入力によりプリ・プロセッサーで用いるファイルの割付けと処理を指示する。NAMELIST名はUPDATEで以下のパラメータがあり、プリ・プロセッサーの実行に際して最初に与える。

(1) OLDPL=1fn

プリ・プロセッサーにより作成された旧プログラム・ライブラリーで、修正の場合は必ず指定する。1fnはファイル番号でF01～F04のいずれかを指定する。

(2) NEWPL=1fn

プリ・プロセッサーにより作成される新プログラム・ライブラリーで、1fnはファイル番号でF01～F04で、指定のない時は作らない。

(3) INPUT=1fn

プリ・プロセッサーのテキストを入力するファイルで、標準ファイル番号はF05=READである。

(4) OUTPUT=1fn

プリ・プロセッサーの処理情報を出力するファイルで、標準ファイル番号は、F06=PRINTである。

(5) COMPILE=1fn

プリ・プロセッサーにより作成されカード・イメージで書かれたコンパイル用のファイル。1fnはファイル番号で、F07=PUNCH, F09～F97のいずれ

かを指定する。指定のない時は作らない。

(6) SOURCE=1fn

プリ・プロセッサーにより作成され、カード・イメージで書かれたNEWPL 再創造のためのファイルで、1fnはファイル番号でF07=PUNCH, F09~F97のいずれかを指定する。指定のない場合は作らない。

(7) F=.TRUE.

コンパイル用のファイル(COMPIL ファイル)にすべての登録デックを書き出す。標準値は.FALSE. が与えられている。

F=.FALSE. の時は、修正を受けたデックのみCOMPIL ファイルに書き出される。ただし、新プログラム・ライブラリー NEWPL を初めて作成する時(創造ラン、例えば第12図)、このパラメータの指定に関係なく、全登録デックがCOMPIL に書き出される。

(8) D=.FALSE.

コンパイル用のファイルに書き出す登録デックに識別符と行番号を付ける。D=.TRUE. ではこれらを付けない。標準値は、D=.FALSE. である。

(9) MASTER= *

プリ・プロセッサーの各コマンドの最初の文字(マスター・コントロール)で、標準は * である。

上記パラメータで指定したファイル以外に補助ファイルとして、F08, F98, F99を必ず指定しなければならない。

4.2 コマンドとテキスト

第1カラムにマスター・コントロール(標準は *)のついたものをコマンドと呼び、その他をテキストという。それらの組み合せをセットと呼ぶ。各コマンドは一般的に次の形式であり、カンマ、ポイント、ブランクによって区切られる。

* ident · seq

* コマンド

ident 英数字からなる認識名または識別符

seq テキストの行番号など

* INSERT, *DELETE, *RESTORE では、ident を省略することも許され、seq のみを書いた場合には、これ以前に使われた ident が使われる。以下に「UPDATE」が有するコマンドを説明する。第2行目に書いたコマンドは、第1行目のコマンドの省略形である。

(1) *DECK dname

*DK

dname は新たに登録するテキスト群に対する認識名で以前に作られたものと同じではいけない。*DECKの終りは他の *DECK, *COMDECK, または * である。ただし修正ラン(例えば第13図)で *INSERT, *DELETE に続く場合はテキストの扱いとなる。これによりデックの分割(LIBE で言うエレメント・セパレーション)が可能になる。

(2) *COMDECK dname

*CD

*DECKと同じ扱いであるが、*CALLコマンドによりテキスト群の内容を呼び出すことができる。オリンパス・システムでは、COMMON文はCOMDECKのテキスト(群)として扱い、必要な所で、必要なCOMMON文を*CALLで呼び出す。しかし、COMDECKの中のテキストはCOMMON文でなくとも、何を書いててもよい。

(3) *IDENT ident

*ID

修正ランで修正セットに対する認識名で*ADDFILE 以外の修正は、*IDENT から始まり以前に作られたものと同じではいけない。*IDENTの終りは、他の *IDENT, *ADDFILE, または * である。

(4) *INSERT ident 1 · seq 1

*I

このコマンドに続くテキスト群を ident 1 · seq 1 で示される識別符と行番号を持つテキストの後に挿入する。*INSERT の終りは他の修正コマンドまたは * である。

(5) *DELETE ident 1 · seq 1 [, ident 2 · seq 2]

*D

このコマンドに続くテキスト群を ident 1 · seq 1 (から ident 2 · seq 2) で示される識別符と行番号を持つテキストを消去した後に挿入する。テキストの無い時は、消去のみ行う。*DELETE の終りは他の修正コマンドまたは * である。

(6) *RESTORE ident 1 · seq 1 [, ident 2 · seq 2]

*R

このコマンドに続くテキスト群を ident 1 · seq 1 (から ident 2 · seq 2) で示される識別符と行番号を持つテキストを再生した後に挿入する。テキストの無い場合は再生のみ行う。*RESTORE の終りは他の修正コマンド、または * である。

(7) *ADDFILE lfn, dname

*AF

lfnで示されるファイルより読み込んだセットを dname という認識名を持つテキスト群の後に登録する。*ADDFILE に続く最初のコマンドは *DECK

または *COMDECK であり、 *REWIND, *SKIP, *READ, *ADDFILE を含んではならない。登録すみデックすべての先頭に登録したい時は dname を YANK¥¥¥とするとい。 *ADDFILE の終りはジョブ制御文または *である。(この修正は、他のすべての修正より後で行わねばならない。)

(8) *SEQUENCE dname 1, dname 2, ..., dname n

*S

*SEQUENCE dname a . dname b

*S

dname という認識名で登録されているテキスト群の行番号を付け直す。カンマで区切られている場合にはその各々について、ポイントで区切られている時は dname a から dname b までについて処理する。

(9) *READ lfn

*RD

lfn で示されるファイルよりセットを読み込む。 *REWIND, *READ, *SKIP, *ADDFILE を含んではいけない。 *READ の終りは * である。

(10) *SKIP lfn, n

*SK

lfn で示されるファイルの現在の位置から n で示されるレコード数だけスキップする。

(11) *REWIND lfn

*RW

lfn で示されるファイルを巻きもどす。

(12) *COMPILE dname 1, dname 2, ..., dname n

*C

*COMPILE dname a . dname b

*C

dname という識別符で登録されているテキスト群をコンパイル用ファイル (COMPILE) に書き出す。カンマで区切られた場合には各々について、またポイントで区切られたときは、 dname a から dname b までについて処理する。

(13) *WEOR

*W

コンパイル用ファイル (COMPILE) を作成中このコマンドが現われた時点で COMPILE に EOF を書き込む。

(14) *CALL dname

*CA

dname という認識名で登録されているコモン・デック (COMDECK) の内容をコンパイル用ファイルに書き出す。

(15) *

コマンド及びテキストからなるセットの終了を意味し、新しいNAME LIST ¥UPDATE から次のプリ・プロセッサの処理が始まる。新しい¥UPDATE がなければこのコマンドでプリ・プロセッサの処理はすべて終了する。

(16) /* comments

プリ・プロセッサのアウトプットファイル(OUTPUT)にコメントを出す。
プリ・プロセッサの処理はない。

4.3 プリ・プロセッサー「UPDATE」の実行

原研計算センタのFACOM 230/75でプリ・プロセッサ「UPDATE」を実行する方法を簡単に説明する。^{注1)} プログラム「UPDATE」でのファイルおよびセットの取り扱いは全て9ビットモードであるため第11図に示すようなモード変換が必要である。第11図では、(1)にLIBE 8ビットモードのファイルに作成されたソース・プログラムJ****.PROGを9ビットモードに変換して、J****.INPUTという名前のファイルに格納した例を示している。ここで作られたDDNEW0(J****.INPUT)を「UPDATE」の読み込みファイルとすることができます。(2)では、「UPDATE」により作成されたCOMPILE またはSOURCEファイルをLIBE 8ビットモードのファイルに登録する方法を示した。(1), (2)では、J****.INPUTやCOMPILE, SOURCEファイルは一時ファイルとしたが永久ファイルでもよい。

プリ・プロセッサ「UPDATE」の実行には、創造ラン(creation run)と修正ラン(correction run)の2種類ある。創造ランでは、修正・更新したいプログラムをプリ・プロセッサを通して、プリ・プロセッサの修正ランの対象となるファイルを作成する(NEWPLファイルを作成する)と同時に全デックをコンパイル用ファイル(COMPILE)に書き出す。修正ランでは、創造ラン、または前回の修正ランで作成されたNEWPLファイルをOLDPLファイルとして読み込み、更に、修正のコマンドのセットを読み込むことにより、プログラムの修正を行う。

創造ランの例を第12図に示す。(1)では修正・更新したいプログラムがカードで組まれている例である。図の中で(1)はプリ・プロセッサ「UPDATE」の実行を示す。(2)は創造ランの後作成される新プログラム・ライブラリNEWPLファイルを定め、(3)はコンパイル用ファイル、(4)は作業のための補助ファイルをそれぞれ割り付けている。今後の修正ランのためNEWPLファイルは永久ファイルとし他は一時ファイルにした。(5)でプリ・プロセッサのパラメータを指定した。(6)は修正・更新したいプログラムのカード群で、それぞれのコモン・デックおよびデックには、先頭に*COMDECK, *DECKをつけねばならない。コモン・デックおよびデックの順序は「UPDATE」では任意でよい。(7)はプリ・プロセッサにより作成されたコンパイル用ファイルCOMPILEをHFORTでコンパイルすることを示している。第12図の(2)では修正・更新したいプログラムがソース・ファイルに入っている場合を示す。修正・更新したいプログラム(例えばJ****.PROGとする。)がLIBE8ビットモードのファイルにある時は第11図の(1)に従って9ビットモード(J****.INPUT)に変換せねばならない。

注1) 原研計算センタのジョブ制御文については文献[9]を参照

次に修正ランの例を第13図に示す。図の中で(1)は、旧プログラム・ライブラリ(OLDPL)にファイル番号F01を割り付けた。修正ランの時は必ずOLDPLファイルを指定しなければならない。(OLDPLファイルは、プリ・プロセッサを通して作成されない。)(2)はプリ・プロセス後に作成される新プログラム・ライブラリで今後の修正ランのために永久ファイルに保存しておく。(3)はプリ・プロセッサのパラメータを与える。Fに対する指定がないので標準値F=.FALSE.がセットされ、修正を受けたデックのみコンパイル用ファイル(COMPIL)に書き出される。最初に述べたように、このようにして作成されたCOMPILファイルやSOURCEファイルをLIBE 8ビット・モードファイルに保存したい時は、第11図の(2)のようにするとよい。

4.4 プリ・プロセッサ「UPDATE」の構造

プリ・プロセッサ「UPDATE」の構造について、特にそのファイル形式を簡単に述べる。

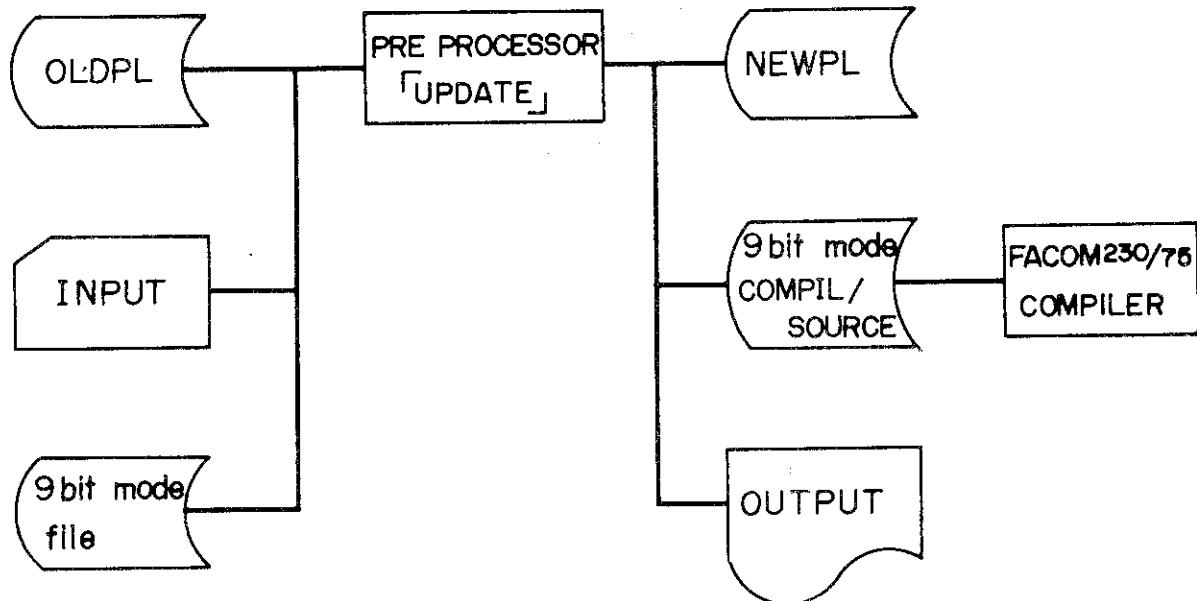
プリ・プロセッサ「UPDATE」で扱うファイルには、2種類の形式があり、ひとつは、**DEFINEFILE**文によって宣言されるランダム・ファイルでプリ・プロセッサによって作成されるプログラム・ライブラリ(OLDPL, NEWPL)，および補助ファイルF08である。もうひとつは、通常のシーケンシャル・ファイルで9ビット・モードのカード・イメージで書かれるファイルである。ランダム・ファイルは、ディレクトリ部分とデータ部分に分けられ、1レコードは32語(128バイト)で10レコードを1ブロックとしてファイルへの入出力を行う。最初の100レコードがディレクトリ部分で101レコード以後がデータ部分として創造および修正に用いられたコマンドやテキストの登録に使われる。ディレクトリ部分もデータ部分もレコード形式は同じであるが、パラメータの扱いが若干異なっている。ディレクトリ部分は第7表のような形式で*DECK, *COMDECK, *IDENT等のコマンドにより作成および参照が行なわれる。またデータ部分のレコードは第8表のようになる。

これらのことについては、詳しくは文献[10]を参照されたい。また、プログラム「UPDATE」のフロー・チャートを第14図に掲げる。

4.5 プリ・プロセッサ「UPDATE」の制限事項

プリ・プロセッサ「UPDATE」の制限事項を列挙する。

- 1) *DECK, *COMDECK, *IDENTによる登録デックの総数は100個までである。
- 2) *DECK, *COMDECK, *IDENTによるデック名は8文字以内の任意の英数字からなる。(数字で始まるデック名も許される。)
- 3) *INSERT, *DELETE 等による修正の重複度は、10回までである。
- 4) *ADDFILEの中に修正ランのコマンドが入ってはいけない。
- 5) 同じ名前の*DECK, *COMDECK, *IDENTがあってはならない。
- 6) *ADDFILEで指示されたファイルの最初のコマンドは*DECKまたは*COMDECKとする。
- 7) プログラム・ライブラリはF01からF04のいずれかとすること。



第10図 プリ・プロセッサ「UPDATE」のシステム概略

(1) CONVERSION OF 8-BIT MODE TO 9-BIT MODE

```

*****
*LIBE
CV89      DDNEWO,DDOLDO
FIN
*DISKTO   DDOLDO,J****,PROG
*LIBEDISK DDNEWO,J****,INPUT
*****

```

(2) CONVERSION OF 9-BIT MODE TO 8-BIT MODE

```

*****
*LIBE
CV98      DDNEWO(ELM...),DDOLDO
FIN
*TPDISK   DDOLDO,COMPIL
*DISKTN   DDNEWO,J****,LIBE8
*****

```

第11図 モード変換 (1)は8ビットから9ビットモードへの変換、(2)はその逆である。

```

*HRUN      EFNAME=J3033.UPDATE          (1)
*DISKTN   F02,J3033.PROG1            (2)
*SORTFDD  F09,,COMPILE,WKUNIT,WORK,PASS,9,,80,800 (3)
*DISK     F08
*DISK     F98
*DISK     F99
*DATA
  *UPDATE NEWPL=2, COMPILE=9 *END          (5)
*COMDECK COMABC
  COMMON/COMABC/ A1 B(10), C(5,100)
*COMDECK COMXYZ
  COMMON/COMX/   X1,X2
  COMMON/COMY/   Y1
  COMMON/COMZ/   Z1,Z2(100)
  DOUBLE PRECISION Y1,Z1
*COMDECK MESA
  CALLMESSAGE(48H JAERI OLYMPUS SYSTEM
  .....
  .....
*DECK STEPON
  SUBROUTINE STEPON
  .....
  .....
*CALL COMBAS
*CALL COMXYZ
  .....
  .....
  END
*DECK OUTPUT
  SUBROUTINE OUTPUT(K)          (6)
  .....
  .....
*CALL COMBAS
*CALL COMDDP
*CALL COMABC
  .....
  .....
*CALL MESA
  END
  .....
  .....
  .....
*
*HFORT      TFNAME=COMPILE          (7)
*JEND

```

第12図 (1) プリ・プロセッサ「UPDATE」の実行(創造ラン)。カード入力の時、

```

      *****
      *****
*HRUN      EFNAME=J3033,UPDATE
*DISKTN   F02,J3033,PRELIB1
*SORTFDD  F09,,COMPILE,WKUNIT,WORK,PASS,9,,80,800
*LIBEDISK F11,J3033,INPUT
*DISK     F08
*DISK     F98
*DISK     F99
*DATA
  *UPDATE  NEWPL=2,  COMPILE=9  *END
*REWIND   11
*READ    11
*
*HFORT    TFNAME=COMPILE
      *****
      *****

```

第12図 (2) プリ・プロセッサ「UPDATE」の実行(創造ラン)。

ディスク・ファイルからの入力の場合を示す。

```

      *****
      *****
*HRUN      EFNAME=J3033,UPDATE
*DISKTO   F01,J3033,LHNEWPL2          (1)
*DISKTN   F02,J3033,LHNEWPL3          (2)
*SORTFDD  F09,,COMPILE,WKUNIT,WORK,PASS,9,,80,800
*DISK     F08
*DISK     F98
*DISK     F99
*DATA
  *UPDATE  OLDPPL=1,  NEWPL=2,  COMPILE=9  *END          (3)
*IDENT    CRRCT1
*DELETE   BLOCKD,6,19
*CALL     CMMN
*D       ICOND,5,45
  A=B+C
  GO TO 654
  X=Y+Z
      *****
      *****
*IDENT    CRRCT2
*INSERT   STEPON,56
  IF(FIRST) GO TO 123          (4)
      *****
      *****
*INSERT   OUTPUT,89
*DECK    OUT
*ADDFILE  5,MAIN
*DECK    DUMMY
  SUBROUTINE DUMMY
      *****
      *****
      *****
  END          (5)
*
*HFORT    TFNAME=COMPILE
      *****
      *****

```

第13図 プリ・プロセッサ「UPDATE」の実行(修正ラン)

第7表 ランダム・ファイル。ディレクトリ部分の構成とパラメータ

	20	22	24	25	26	27	28	29	30	31	32	(語)
M TAB			M NAME	M DENT	M SEQ	M REC	M IND	M NUM	M TYP	M YAK	M COM	MDUM

M TAB ; ダミー。

M NAME; DECK名, COMDECK名, IDENT名。

M DENT; M NAMEと同じ。

M SEQ ; ダミー。

M REC ; M NAMEで登録されたデックの開始番地(レコード番号)。

M IND ; M NAMEで登録されたデックの終了番地(レコード番号)。

M NUM ; M NAMEで登録されたデックの順番。

M TYP ; M NAMEで登録された形式。DECKは1, COMDECKは2, IDENTは3。

M YAK ; 登録内容が活性か不活性かの指示。

M COM ; COMPILEファイルの出力指示。

MDUM ; ダミー。

第8表 ランダム・ファイル。データ部分の構成とパラメータ

	20	22	24	25	26	27	28	29	30	31	32 (語)
K TAB		K NAME	K DENT	K SEQ	K REC	K IND	K NUM	K TYP	K YAK	K COM	K DUM

K TAB ; 登録されたカード・イメージ。

K NAME ; 修正の属する DECK名, 又は COMDECK名。

K DENT ; 修正 IDENT名(創造ランの場合は K NAMEと同じ)。

K SEQ ; K DENTという名前を持つカードの順番。

K REC ; 次に処理するカード・イメージの入っている基準番地(レコード番号)。

K IND ; 基準番地(K REC)に対する相対番地のインデックス。実際に処理されるカード・イメージの入っている番地は K REC + K INDになる。

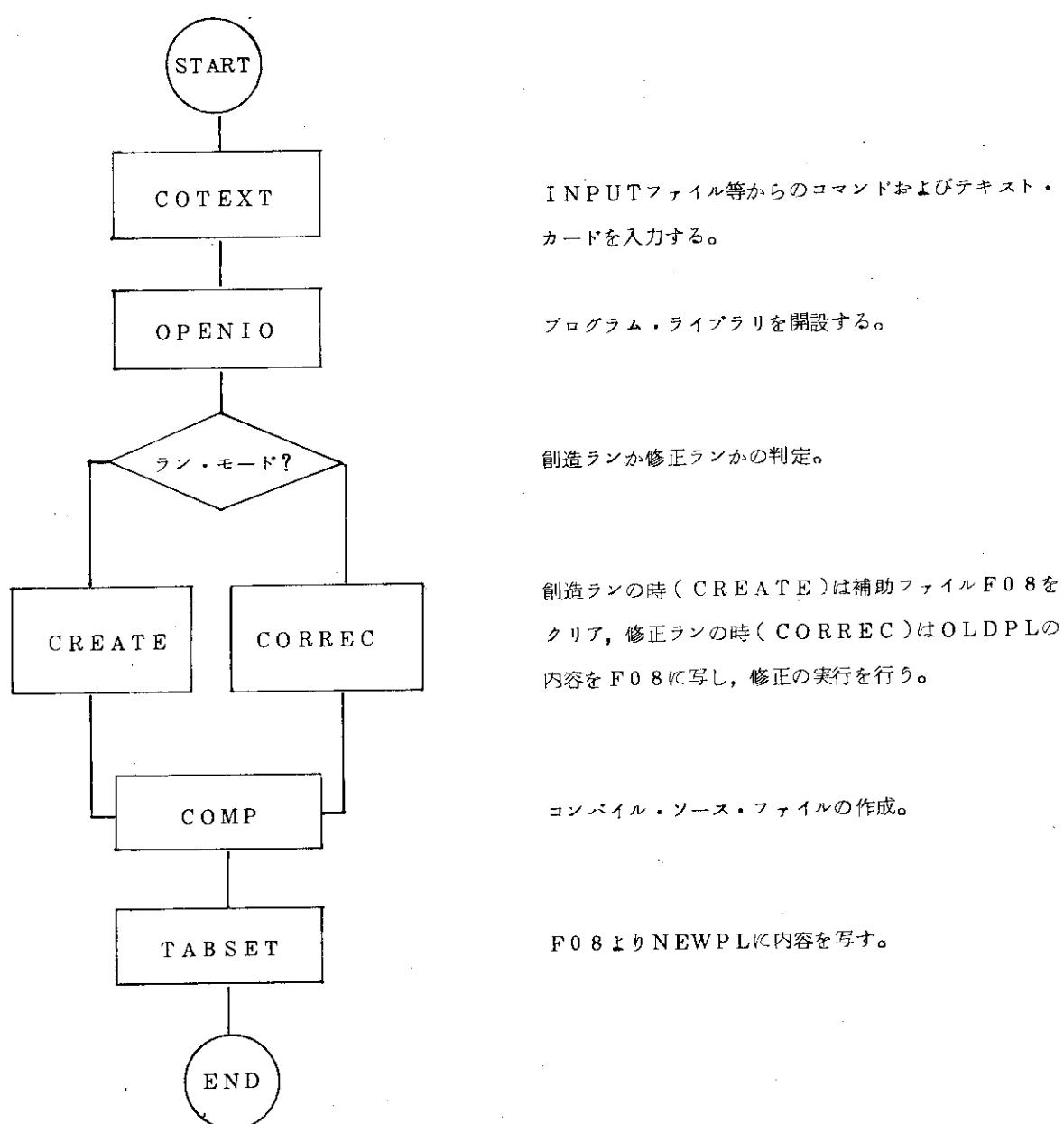
K NUM ; ダミー。

K TYP ; K TABの内容がコントロール・カードの時その型の番号が入る。

K YAK ; カード・イメージが活性か不活性かの指示。

K COM ; ダミー。

K DUM ; カード・イメージの登録形式。INSERT, DELETE, RESTOREの区别。



第14図 プリ・プロセッサ「UPDATE」のプログラム・フロー

5. おわりに

最後に、OLYMPUS システムの問題点と評価、今後の発展について考える。先ず、核融合関係の計算コード開発の現状と比較して、OLYMPUS システムの位置付けをしてみよう。これまでの議論から理解できるように、もし国際的にも OLYMPUS システムが相当普及したと仮定するならば、計算コードの作成を OLYMPUS システムの規則に忠実に従って行う事によって、研究機関の間でのコード交換は円滑に行われるようになり、コード作成の能率も大巾に改善されるであろう。問題は、OLYMPUS システムによる計算コード作成が今後普及するか否かという点にある。この点について検討してみる。

- (a) OLYMPUS システムを導入する事はそれ程むづかしいことではない。
- (b) OLYMPUS システムの規則の学習は容易である。
- (c) 中規模、小規模のプログラム開発に際し、OLYMPUS システムの規則に忠実に従うのは、制限が多すぎてわずらわしいと感じるかも知れない。
- (d) 大規模プログラムを数人のチームで開発する場合には、OLYMPUS システムを利用する事は有効であろう。
- (e) Subroutine の呼出しが多くなるので計算時間はやや増加する可能性がある。

OLYMPUS システムは、現在の所、完成したシステムというよりは発展過程にあるコード開発システムの一つの試みとみなされる。とは言え、このシステムは、これ自身で閉じたシステムとして最初に設定された開発の動機に関してはかなりの水準までその目的を達成しているものと考えられる。今後は、システム内に用意されている多数のスペアのクラスについて具体的な意味付けを行いライブラリを整備してシステムの機能を充実させていくことが重要であろう。

JAERI OLYMPUS SYSTEM にとっては、ジョブ制御文のマクロ化が肝要であろう。現在もっとも効果的なジョブ制御文の並びを検討中で、早々に、バッチ処理用のマクロ制御文および cps マクロを開発する予定である。

謝 辞

英國 Culham 研究所の K.V.Roberts 氏には、1974年来日された際、OLYMPUS システムの導入に対して格別の援助を頂いた。同氏の御厚意に深く感謝いたします。また核融合研究部の常松俊秀氏、滝塚知典氏、および計算センタの富山峰秀氏の御親切で有意義な討論、御教示に対し感謝いたします。

参 考 文 献

- 1) K.V.Roberts: Computer Phys.Commun. 1 (1969) 1.
- 2) 田中正俊・川上一郎:「計算機プログラム交換に関する国際会議参加報告」
(unpublished).
- 3) K.V.Roberts: Computer Phys.Commun. 7 (1974) 237.
- 4) J.P.Christiansen and K.V.Roberts: Computer Phys.Commun. 7 (1974)
245.
- 5) J.P.Christiansen, D.E.T.F.Ashby and K.V.Roberts: Computer Phys.Commun.
7 (1974) 271.
- 6) M.L.Watkins, M.H.Hughes, K.V.Roberts, P.M.Keeping and J.Killeen:
ICARUS-A One - Dimensional Plasma Diffusion Code, Methods in
Computational Physics, Vol. 16, p 166-209 (edited B. Alder et
al., Academic Press, 1976).
- 7) 川上一郎・田中幸夫・村田孝治: IRIS, 核融合・プラズマプログラムパッケージ
(FPPP), 1976. (昭和51年度 文部省 特定研究(1))
- 8) 滝塚知典・岡本正雄:「COMMON 文を活用するためのプリ・プロセッサの作成およびそ
の使用法」(unpublished).
- 9) 山崎和彦:「ジョブ制御文の手引」(unpublished).
- 10) CYBER 70 Series Computer Systems, "Update Reference Manual",
(Control Data Corporation, Minnesota, U.S.A., 1971).

謝 辞

英國 Culham 研究所の K.V.Roberts 氏には、1974年来日された際、OLYMPUS システムの導入に対して格別の援助を頂いた。同氏の御厚意に深く感謝いたします。また核融合研究部の常松俊秀氏、滝塚知典氏、および計算センタの富山峰秀氏の御親切で有意義な討論、御教示に対し感謝いたします。

参 考 文 献

- 1) K.V.Roberts: Computer Phys.Commun. 1 (1969) 1.
- 2) 田中正俊・川上一郎:「計算機プログラム交換に関する国際会議参加報告」
(unpublished).
- 3) K.V.Roberts: Computer Phys.Commun. 7 (1974) 237.
- 4) J.P.Christiansen and K.V.Roberts: Computer Phys.Commun. 7 (1974)
245.
- 5) J.P.Christiansen, D.E.T.F.Ashby and K.V.Roberts: Computer Phys.Commun.
7 (1974) 271.
- 6) M.L.Watkins, M.H.Hughes, K.V.Roberts, P.M.Keeping and J.Killeen:
ICARUS-A One-Dimensional Plasma Diffusion Code, Methods in
Computational Physics, Vol. 16, p 166-209 (edited B. Alder et
al., Academic Press, 1976).
- 7) 川上一郎・田中幸夫・村田孝治: IRIS, 核融合・プラズマプログラムパッケージ
(FPPP), 1976. (昭和51年度 文部省 特定研究(1))
- 8) 滝塚知典・岡本正雄:「COMMON 文を活用するためのプリ・プロセッサの作成およびそ
の使用法」(unpublished).
- 9) 山崎和彦:「ジョブ制御文の手引」(unpublished).
- 10) CYBER 70 Series Computer Systems, "Update Reference Manual",
(Control Data Corporation, Minnesota, U.S.A., 1971).

付録A OLYMPUS システムの「COMBAS, COMDDP」および「CRONUS」,
 「CYCLOPS」のプログラム・リスト

```

C C C C C
C ***** * ***** CMT00010
C * * * * * CMT00020
C * * * * * CMT00030
C * * * * * CMT00040
C * * * * * CMT00050
C * * * * * CMT00060
C * * * * * CMT00070
C * * * * * CMT00080
C * * * * * CMT00090
C * * * * * CMT00100
C * * * * * CMT00110
C
C A STANDARD CONTROL AND UTILITY PACKAGE CMT00120
C ----- CMT00130
C FOR INITIAL-VALUE FORTRAN PROGRAMS CMT00140
C ----- CMT00150
C (CDC VERSION 1) CMT00160
C
C M.H.HUGHES, G.G.LISTER AND K.V.ROBERTS CMT00170
C
C COMPUTATIONAL PHYSICS GROUP FOM INSTITUUT VOOR PLASMAFYSICA CMT00180
C CULHAM LABORATORY RIJNHUIZEN CMT00190
C ABINGDON JUTPHAAST CMT00200
C BERKSHIRE THE NETHERLANDS CMT00210
C ENGLAND CMT00220
C
C FURTHER COPIES OF THE PUBLISHED VERSION OF THIS PROGRAM, CMT00230
C INCLUDING ANY SUBSEQUENT CORRECTIONS AND AMENDMENTS NOTIFIED CMT00240
C BY THE AUTHORS, SHOULD BE OBTAINED ON MAGNETIC TAPE DIRECTLY CMT00250
C FROM CMT00260
C CPC PROGRAM LIBRARY, QUEENS UNIVERSITY, BELFAST, N.IRELAND. CMT00270
C CMT00280
C AN APPLICATION FORM APPEARS IN EACH ISSUE OF CMT00290
C CMT00300
C COMPUTER PHYSICS COMMUNICATIONS. CMT00310
C CMT00320
C CATALOGUE NUMBER CMT00330
C WRITEUP REFERENCE COMPUTER PHYS. COMMUN. CMT00340
C
C ----- CMT00350
C CMT00360
C CMT00370
C CMT00380
C CMT00390
C CMT00400
C CMT00410
C CMT00420
C CMT00430
C CMT00440
C CMT00450
C CMT00460
C CMT00470
C CMT00480
C CMT00490
C CMT00500
C CMT00510
C CMT00520

```

C C----- CMT00530
 C----- CMT00540
 C----- CMT00550
 C----- CMT00560
 C----- CMT00570
 C----- CMT00580
 C----- CMT00590
 C----- CMT00600
 C----- CMT00610
 C----- CMT00620
 C----- CMT00630
 C----- CMT00640
 C----- CMT00650
 C----- CMT00660
 C----- CMT00670
 C----- CMT00680
 C----- CMT00690
 C----- CMT00700
 C----- CMT00710
 C----- CMT00720
 C----- CMT00730
 C----- CMT00740
 C----- CMT00750
 C----- CMT00760
 C----- CMT00770
 C----- CMT00780
 C----- CMT00790
 C----- CMT00800
 C----- CMT00810
 C----- CMT00820
 C----- CMT00830
 C----- CMT00840
 C----- CMT00850
 C----- CMT00860
 C----- CMT00870
 C----- CMT00880
 C----- CMT00890
 C----- CMT00900
 C----- CMT00910
 C----- CMT00920
 C----- CMT00930
 C----- CMT00940
 C----- CMT00950
 C----- CMT00960
 C----- CMT00970
 C----- CMT00980
 C----- CMT00990
 C----- CMT01000
 C----- CMT01010
 C----- CMT01020
 C----- CMT01030
 C----- CMT01040
 C----- CMT01050
 C----- CMT01060

C----- JAERI VERSION 1 -----
 M.OKAMOTO, T.TAKEDA, M.TANAKA
 THEORETICAL AND COMPUTATIONAL GROUP
 DIVISION OF THERMONUCLEAR FUSION RESARCH
 JAPAN ATOMIC ENERGY RESEARCH INSTITUTE (JAERI)
 TOKAI ESTABLISHMENT
 TOKAI, IBARAKI, JAPAN
 K.ASAI
 DATA PROCESSING CENTER
 JAERI,
 TOKAI ESTABLISHMENT
 TOKAI, IBARAKI, JAPAN
 K.NAKANO
 CENTURY RESEARCH CENTER CORPORATION
 NIHON-BASHI, CHUO-KU
 TOKYO, JAPAN
 I.KAMAKAMI
 PHYSICS DEPARTMENT AND DATA PROCESSING CENTER
 NIHON UNIVERSITY
 KANDA-SURUGADAI, CHIYODA-KU
 TOKYO, JAPAN
 JAERI VERSION 1
 COMPLETED ON APRIL 15, 1977.
 COPIED FROM CPC PROGRAM LIBRARY AND
 CONVERTED TO THE COMPUTER SYSTEM OF JAERI.
 REQUIRES THE PRE-PROCESSOR *UPDATE*.
 JAERI DATA PROCESSING CENTER
 A-SYSTEM FACOM-230/75 * 2 (2-CPU, 1MWORDS) FOR TSS
 CLOSED BATCH CMT00980
 PLOTTER CMT00990
 GRAPHICS CMT01000
 B-SYSTEM FACOM-230/75 * 2 (2-CPU, 1MWORDS) FOR OPEN BATCH CMT01010
 CMT01020
 FACOM-230/75 IS A WORD MACHINE. 1-WORD = 4-CHARACTERS
 1-CH, = 9-BITS CMT01030
 CMT01040
 CMT01050
 CMT01060

C-----			CMT01070
C			CMT01080
C			CMT01090
C	INDEX OF SUBPROGRAMS		CMT01100
C			CMT01110
C			CMT01120
C	MAIN CONTROL.	CLASS 0	CMT01130
C			CMT01140
C CRONUS	FORTRAN MAIN PROGRAM		0.0 CMT01150
C BASIC	INITIALIZE BASIC CONTROL DATA		0.1 CMT01160
C MODIFY	MODIFY BASIC DATA IF REQUIRED		0.2 CMT01170
C COTROL	CONTROL THE RUN		0.3 CMT01180
C EXPERT(3)	MODIFY STANDARD OPERATION OF PROGRAM		0.4 CMT01190
C	PROLOGUE.	CLASS 1	CMT01200
C			CMT01210
C LABRUN	LABEL THE RUN		1.1 CMT01230
C CLEAR	CLEAR VARIABLES AND ARRAYS		1.2 CMT01240
C PRESET	SET DEFAULT VALUES		1.3 CMT01250
C DATA	DEFINE DATA SPECIFIC TO RUN		1.4 CMT01260
C AUXVAL	SET AUXILIARY VALUES		1.5 CMT01270
C INITIAL	DEFINE PHYSICAL INITIAL CONDITIONS		1.6 CMT01280
C RESUME	RESUME FROM PREVIOUS RECORD		1.7 CMT01290
C START	START OR RESTART THE RUN		1.8 CMT01300
C	CALCULATION.	CLASS 2	CMT01310
C			CMT01320
C STEPON	STEP ON THE CALCULATION		2.1 CMT01340
C	OUTPUT.	CLASS 3	CMT01350
C			CMT01360
C OUTPUT(1)	CONTROL THE OUTPUT		3.1 CMT01380
C	EPILOGUE.	CLASS 4	CMT01390
C			CMT01400
C TESEND	TEST FOR COMPLETION OF RUN		4.1 CMT01420
C ENDRUN	TERMINATE THE RUN		4.2 CMT01430
C	DIAGNOSTICS.	CLASS 5	CMT01440
C			CMT01450
C REPORT(3)	CONTROL THE DIAGNOSTICS		5.1 CMT01470
C CLIST(2)	PRINT COMMON VARIABLES		5.2 CMT01480
C ARRAYS(2)	PRINT COMMON ARRAYS		5.3 CMT01490
C	UTILITIES.	CLASS U	CMT01500
C			CMT01510
C MESSAGE(1)	PRINT 48-CHARACTER MESSAGE ON OUTPUT CHANNEL		U.1 CMT01530
C PAGE	FETCH NEW PAGE ON OUTPUT CHANNEL		U.2 CMT01540
C BLINES(1)	INSERT BLANK LINES ON OUTPUT CHANNEL		U.3 CMT01550
C RVAR(2)	PRINT NAME AND VALUE OF REAL VARIABLE		U.4 CMT01560
C IVAR(2)	PRINT NAME AND VALUE OF INTEGER VARIABLE		U.5 CMT01570
C HVAR(2)	PRINT NAME AND VALUE OF HOLLERITH VARIABLE		U.6 CMT01580
C LVAR(2)	PRINT NAME AND VALUE OF LOGICAL VARIABLE		U.7 CMT01590
C RARRAY(3)	PRINT NAME AND VALUES OF REAL ARRAY		U.8 CMT01600

C IARRAY(3)	PRINT NAME AND VALUES OF INTEGER ARRAY	U.9 CMT01610
C HARRAY(3)	PRINT NAME AND VALUES OF HOLLERITH ARRAY	U.10CMT01620
C REPTHD(3)	PRINT HEADING FOR DIAGNOSTIC REPORT.	U.11CMT01630
C RUNTIM	UPDATE CPU TIME AND PRINT IT	U.12CMT01640
C DAYTIM	PRINT DATE AND TIME	U.13CMT01650
C RESETR(3)	RESET REAL ARRAY TO SPECIFIED VALUE	U.14CMT01660
C RESETI(3)	RESET INTEGER ARRAY TO SPECIFIED VALUE	U.15CMT01670
C RESETH(3)	RESET HOLLERITH ARRAY TO SPECIFIED VALUE	U.16CMT01680
C JOBTIM	FETCH ALLOCATED JOBTIME	U.17CMT01690
C LARRAY(3)	PRINT NAME AND VALUES OF LOGICAL ARRAY	U.19CMT01700
C RESETL(3)	RESET LOGICAL ARRAY TO SPECIFIED VALUE	U.19CMT01710
C RRAY2	PRINT DOUBLY-SUBSCRIPTED ARRAY	U.20CMT01720
C SCALER(3)	SCALE A REAL ARRAY BY A REAL VALUE	U.21CMT01730
C SCALEI(3)	SCALE AN INTEGER ARRAY BY AN INTEGER VALUE	U.22CMT01740
C COPYR(5)	COPY ONE REAL MATRIX INTO ANOTHER	U.23CMT01750
C COPYI(5)	COPY ONE INTEGER MATRIX INTO ANOTHER	U.24CMT01760
C -SIGNR(2)	CHANGE THE SIGN OF A REAL MATRIX	U.25CMT01770
C SIGNI(2)	CHANGE THE SIGN OF AN INTEGER MATRIX	U.26CMT01780
C DUMCOM(3)	DUMP SELECTED COMMON BLOCKS	U.27CMT01790
C		CMT01800
C		CMT01810
C		CMT01820
CL	C1.1. BASIC SYSTEM PARAMETERS	CMT01830
C	COMMON/COMBAS/	CMT01840
C		CMT01850
C		CMT01860
C		CMT01870
C ALTIME	TIME ALLOCATED TO JOB, (SECS)	R 1.1 CMT01880
C CPTIME	CPU TIME USED SO FAR ON THIS TASK, (SECS)	R 1.1 CMT01890
C LABEL1(12)	LABEL DESCRIBING THE RUN	HA 1.1 CMT01900
C LABEL2(12)	LABEL DESCRIBING THE RUN	HA 1.1 CMT01910
C LABEL3(12)	LABEL DESCRIBING THE RUN	HA 1.1 CMT01920
C LABEL4(12)	LABEL DESCRIBING THE RUN	HA 1.1 CMT01930
C LABEL5(12)	LABEL AVAILABLE TO PROGRAMMER	HA 1.1 CMT01940
C LABEL6(12)	LABEL AVAILABLE TO PROGRAMMER	HA 1.1 CMT01950
C LABEL7(12)	LABEL RESERVED FOR SYSTEM USE	HA 1.1 CMT01960
C LABEL8(12)	LABEL RESERVED FOR SYSTEM USE	HA 1.1 CMT01970
C NDIARY	CHANNEL FOR DIARY	I 1.1 CMT01980
C NIN	CURRENT INPUT CHANNEL	I 1.1 CMT01990
C NLEDGE	CHANNEL FOR RESTART RECORDS	I 1.1 CMT02000
C NLEND	.TRUE., IF RUN TO BE TERMINATED	L 1.1 CMT02010
C NLRES	,TRUE., IF RUN TO BE RESTARTED	L 1.1 CMT02020
C NONLIN	CHANNEL FOR ONLINE INPUT-OUTPUT	I 1.1 CMT02030
C NOUT	CURRENT OUTPUT CHANNEL	I 1.1 CMT02040
C NPRINT	CHANNEL FOR PRINTED OUTPUT	I 1.1 CMT02050
C NPUNCH	CHANNEL FOR CARD OUTPUT (OR EQUIVALENT)	I 1.1 CMT02060
C NREAD	CHANNEL FOR CARD INPUT	I 1.1 CMT02070
C NREC	CURRENT RECORD NUMBER	I 1.1 CMT02080
C NRESUM	RESUME FROM RECORD ON THIS CHANNEL	I 1.1 CMT02090
C NRUN	MAXIMUM NUMBER OF STEPS	I 1.1 CMT02100
C NSTEP	CURRENT STEP NUMBER	I 1.1 CMT02110
C STIME	START TIME, (SECS)	R 1.1 CMT02120
C		CMT02130
C		CMT02140

C-----				CMT02150
C				CMT02160
C				CMT02170
CL	C1.9. DEVELOPMENT AND DIAGNOSTIC PARAMETERS			CMT02180
C	COMMON/COMDDP/			CMT02190
C				CMT02200
C				CMT02210
C				CMT02220
C				CMT02230
C MAXDUM	MAXIMUM DIMENSION OF DUMP ARRAYS	I 1.9		CMT02240
C MXDUMP	ACTUAL DIMENSION OF DUMP ARRAYS	I 1.9		CMT02250
C NADUMP(MAXDUM)	CODES FOR ARRAY DUMPS	IA 1.9		CMT02260
C NCLASS	MOST RECENT CLASS REPORTED	I 1.9		CMT02270
C NLCHED	.TRUE. IF CLASS 0 REPORT HEADS REQUIRED	L 1.9		CMT02280
C NLHEAD(9)	.TRUE. IF CLASSES 1-9 REPORT HEADS REQUIRED	LA 1.9		CMT02290
C NLOMT1(50)	CLASS 1 SUBPROGRAM SELECTOR	LA 1.9		CMT02300
C NLOMT2(50)	CLASS 2 SUBPROGRAM SELECTOR	LA 1.9		CMT02310
C NLOMT3(50)	CLASS 3 SUBPROGRAM SELECTOR	LA 1.9		CMT02320
C NRREPT	.TRUE. IF ANY REPORT REQUIRED	L 1.9		CMT02330
C NPDUMP(MAXDUM)	CODES FOR DUMPING POINTS	IA 1.9		CMT02340
C NPOINT	MOST RECENT POINT REPORTED	I 1.9		CMT02350
C NSUB	MOST RECENT SUBPROGRAM REPORTED	I 1.9		CMT02360
C NVDUMP(MAXDUM)	CODES FOR DUMPING VARIABLES	IA 1.9		CMT02370
C				CMT02380
C				CMT02390
C				CMT02400
C	ALPHABETIC INDEX OF COMMON VARIABLES			CMT02410
C				CMT02420
C				CMT02430
C ALTIME	TIME ALLOCATED TO JOB. (SECS)	R 1.1		CMT02440
C CPTIME	CPU TIME USED SO FAR ON THIS TASK. (SECS)	R 1.1		CMT02450
C LABEL1(12)	LABEL DESCRIBING THE RUN	HA 1.1		CMT02460
C LABEL2(12)	LABEL DESCRIBING THE RUN	HA 1.1		CMT02470
C LABEL3(12)	LABEL DESCRIBING THE RUN	HA 1.1		CMT02480
C LABEL4(12)	LABEL DESCRIBING THE RUN	HA 1.1		CMT02490
C LABEL5(12)	LABEL AVAILABLE TO PROGRAMMER	HA 1.1		CMT02500
C LABEL6(12)	LABEL AVAILABLE TO PROGRAMMER	HA 1.1		CMT02510
C LABEL7(12)	LABEL RESERVED FOR SYSTEM USE.	HA 1.1		CMT02520
C LABEL8(12)	LABEL RESERVED FOR SYSTEM USE	HA 1.1		CMT02530
C MAXDUM	MAXIMUM DIMENSION OF DUMP ARRAYS	I 1.9		CMT02540
C MXDUMP	ACTUAL DIMENSION OF DUMP ARRAYS	I 1.9		CMT02550
C NADUMP	CODES FOR ARRAY DUMPS	IA 1.9		CMT02560
C NCLASS	MOST RECENT CLASS REPORTED	I 1.9		CMT02570
C NDIARY	CHANNEL FOR DIARY	I 1.1		CMT02580
C NIN	CURRENT INPUT CHANNEL	I 1.1		CMT02590
C NLCHED	.TRUE. IF CLASS 0 REPORT HEADS REQUIRED	L 1.9		CMT02600
C NLEDGE	CHANNEL FOR RESTART RECORDS	I 1.1		CMT02610
C NLEND	.TRUE. IF RUN TO BE TERMINATED	L 1.1		CMT02620
C NLHEAD	.TRUE. IF CLASSES 1-9 REPORT HEADS REQUIRED	LA 1.9		CMT02630
C NLOMT1	CLASS 1 SUBPROGRAM SELECTOR	LA 1.9		CMT02640
C NLOMT2	CLASS 2 SUBPROGRAM SELECTOR	LA 1.9		CMT02650
C NLOMT3	CLASS 3 SUBPROGRAM SELECTOR	LA 1.9		CMT02660
C NRREPT	.TRUE. IF ANY REPORT REQUIRED	L 1.9		CMT02670
C NRRES	.TRUE. IF RUN TO BE RESTARTED	L 1.1		CMT02680

C	NONLIN	CHANNEL FOR ONLINE INPUT-OUTPUT	I	1.1	CMT02690
C	NOUT	CURRENT OUTPUT CHANNEL	I	1.1	CMT02700
C	NPDUMP	CODES FOR DUMPING POINTS	IA	1.9	CMT02710
C	NPOINT	MOST RECENT POINT REPORTED	I	1.9	CMT02720
C	NPRINT	CHANNEL FOR PRINTED OUTPUT	I	1.1	CMT02730
C	NPUNCH	CHANNEL FOR CARD OUTPUT (OR EQUIVALENT)	I	1.1	CMT02740
C	NREAD	CHANNEL FOR CARD INPUT	I	1.1	CMT02750
C	NREC	CURRENT RECORD NUMBER	I	1.1	CMT02760
C	NRESUM	RESUME FROM RECORD ON THIS CHANNEL	I	1.1	CMT02770
C	NRUN	MAXIMUM NUMBER OF STEPS	I	1.1	CMT02780
C	NSTEP	CURRENT STEP NUMBER	I	1.1	CMT02790
C	NSUB	MOST RECENT SUBPROGRAM REPORTED	I	1.9	CMT02800
C	NVDUMP	CODES FOR DUMPING VARIABLES	IA	1.9	CMT02810
C	STIME	START TIME, (SECS)	R	1.1	CMT02820
C					CMT02830
C					CMT02840
C-----					CMT02850
C					CMT02860
C					CMT02870

*COMDECK COMBAS

CL-----	C1.1. BASIC SYSTEM PARAMETERS					
C	COMMON/COMBAS/					
+ ALTIME, CPTIME,	NLEDGE,	NLEND,	NLRES,	NONLIN,		COM00010
+ NOUT, NPRINT,	NREAD,	NREC,	NRESUM,	NSTEP,		COM00020
+ STIME, LABEL1,	LABEL2,	LABEL3,	LABEL4,	LABEL5,		COM00030
+ LABEL6, LABEL7,	LABEL8,	NDIARY,	NIN,	NPUNCH,		COM00040
+ NRUN	LOGICAL	NLEND,	NLRES			COM00050
	DIMENSION					COM00060
H LABEL1(12),		LABEL2(12),		LABEL3(12),		COM00070
H LABEL4(12),		LABEL5(12),		LABEL6(12),		COM00080
H LABEL7(12),		LABEL8(12)				COM00090

*COMDECK COMDDP

CL-----	C1.9. DEVELOPMENT AND DIAGNOSTIC PARAMETERS					
C	COMMON/COMDDP/					
I MAXDUM, MXDUMP,	NADUMP,	NCLASS,	NPDUMP,	NPOINT,		COM00110
I NSUB, NVDUMP,						COM00120
L NLCHED, NLHEAD,	NLOMT1,	NLOMT2,	NLOMT3,	NLREPT		COM00130
LOGICAL						COM00140
L NLCHED, NLHEAD,	NLOMT1,	NLOMT2,	NLOMT3,	NLREPT		COM00150
DIMENSION						COM00160
I NADUMP(20),	NPDUMP(20),		NVDUMP(20),			COM00170
L NLHEAD(9),	NLOMT1(50),		NLOMT2(50),			COM00180
L NLOMT3(50)						COM00190

```

*DECK CRONUS
ELEMENT CRONUS
C
C
C      VERSION 1C          29/4/75      KVR/MHH/GGL      JUTPHAAS
C
C      CALL MASTER
STOP.
END
*DECK COS0
C
SUBROUTINE MASTER
C
C 0.0  MASTER SUBPROGRAM
C
C      VERSION 2C          20/5/75      KVR/MHH/GGL      JUTPHAAS
C      JAERI VERSION 1        APRIL 15.1977
C
*CALL COMBAS
C-----.
C
C      TIME ALLOCATED TO JOB
C      CALL JOBTIM(ALTIME)
C      ( FACOM 230/75 HAS NO ROUTINE GIVING THE ALLOCATED CPU TIME )CRN00250
C
C      ALTIME=0.0
C
C      START TIME FOR THIS TASK
C      *CLOKM* IS A FACOM ROUTINE GIVING THE CPU-TIME (MSEC)
C
C      CALL CLOCKM(KCPU)
STIME=FLOAT(KCPU)
C      ZERO CPTIME FOR THIS TASK
CPTIME=0.0
C
C      SET UP THE BASIC CONTROL DATA
CALL BASIC
C
C      PRINT DATE AND TIME
CALL DAYTIM
C
C      CONTROL THE RUN
CALL COTROL
C
RETURN
END
*DECK COS1
C
SUBROUTINE BASIC
C
C 0.1  INITIALIZE BASIC DATA
CRN00010
CRN00020
CRN00030
CRN00040
CRN00050
CRN00060
CRN00070
CRN00080
CRN00090
CRN00100
CRN00110
CRN00120
CRN00130
CRN00140
CRN00150
CRN00160
CRN00170
CRN00180
CRN00190
CRN00200
CRN00210
CRN00220
CRN00230
CRN00240
CRN00250
CRN00260
CRN00270
CRN00280
CRN00290
CRN00300
CRN00310
CRN00320
CRN00330
CRN00340
CRN00350
CRN00360
CRN00370
CRN00380
CRN00390
CRN00400
CRN00410
CRN00420
CRN00430
CRN00440
CRN00450
CRN00460
CRN00470
CRN00480
CRN00490
CRN00500
CRN00510
CRN00520

```

C			CRN00530		
C	VERSION 4C	20/5/75	KVR/MHH/GGL	JUTPHAS	CRN00540
C					CRN00550
*CALL COMBAS					CRN00560
*CALL COMDDP					CRN00570
C-----					CRN00580
	DATA	IBLANK/4H /			CRN00590
C					CRN00600
C-----					CRN00610
CL	1.	GENERAL OLYMPUS DATA			CRN00620
C					CRN00630
CL	1.1	BASIC SYSTEM PARAMETERS			CRN00640
C					CRN00650
C	CLEAR ALL 8 LABEL ARRAYS				CRN00660
	IL = 8*12				CRN00670
	CALL RESETH(LABEL1,IL,IBLANK)				CRN00680
C					CRN00690
C	INPUT-OUTPUT CHANNELS				CRN00700
	NLEDGE = 30				CRN00710
	NONLIN = 1				CRN00720
	NPUNCH=7				CRN00730
	NPRINT=6				CRN00740
	NREAD=5				CRN00750
	NIN=NREAD				CRN00760
	NOUT=NPRINT				CRN00770
	NDIARY=NOUT				CRN00780
C					CRN00790
C	TIMESTEP CONTROL.				CRN00800
	NRUN = 1				CRN00810
	NSTEP=0				CRN00820
C					CRN00830
C	RESTART CONTROL				CRN00840
	NREC = 1				CRN00850
	NRESUM = NLEDGE				CRN00860
C					CRN00870
C	LOGICAL SWITCHES				CRN00880
	NLEND=.FALSE.				CRN00890
	NLRES=.FALSE.				CRN00900
C					CRN00910
CL	1.9	DIAGNOSTIC AND DEVELOPMENT PARAMETERS			CRN00920
C	MAXIMUM DIMENSIONS OF DUMP ARRAYS				CRN00930
	MAXDUM = 20				CRN00940
	MXDUMP = 10				CRN00950
C	RESET DUMP ARRAYS				CRN00960
	CALL RESETI(NADUMP,MAXDUM,0)				CRN00970
	CALL RESETI(NPDUMP,MAXDUM,0)				CRN00980
	CALL RESETI(NVDUMP,MAXDUM,0)				CRN00990
C	TRACER VARIABLES				CRN01000
	NCLASS = 0				CRN01010
	NSUB = 1				CRN01020
	NPOINT = 1				CRN01030
C	LOGICAL SWITCHES				CRN01040
	NLCHED = .FALSE.				CRN01050
	NLREPT = .FALSE.,				CRN01060

```

C      REPORT HEADS FOR CLASSES 1-9          CRN01070
C      CALL RESETL(NLHEAD,9,,FALSE,)          CRN01080
C      RESET CLASS 1,2,3 SUBPROGRAM SELECTOR ARRAY CRN01090
C      CALL RESETL(NLOMT1,50,,FALSE,)          CRN01100
C      CALL RESETL(NLOMT2,50,,FALSE,)          CRN01110
C      CALL RESETL(NLOMT3,50,,FALSE,)          CRN01120
C
C      CALL PAGE                            CRN01130
C      CALLMESSAGE(48H    OLYMPUS SYSTEM        )CRN01140
C      CALLMESSAGE(48H    JAERI VERSION 1 APRIL 15, 1977 )CRN01150
C      CALLMESSAGE(48H    0,1 BASIC CONTROL DATA SET )CRN01160
C
C      USER INTERFACE                      CRN01170
C      CALL MODIFY                         CRN01180
C
C      RETURN                             CRN01190
C      END                                CRN01200
*CDECK COS2                           CRN01210
C
C      SUBROUTINE MODIFY                   CRN01220
C
C 0.2  MODIFY BASIC DATA IF REQUIRED   CRN01230
C
C      VERSION 2C           17/10/73       KVR/MHH      CULHAM CRN01240
C
C-----CRN01250
C
C      CALLMESSAGE(48H    0.2 STANDARD DATA ACCEPTED )CRN01260
C      RETURN                           CRN01270
C      END                               CRN01280
*CDECK COS3                           CRN01290
C
C      SUBROUTINE COTROL                  CRN01300
C
C 0.3  CONTROL THE RUN                 CRN01310
C
C      VERSION 2C           17/10/73       KVR/MHH      CULHAM CRN01320
C
C-----CRN01330
C
C      CALL COMBAS                         CRN01340
C      CALL COMDDP                         CRN01350
C
C-----CRN01360
C
C      DATA ICLASS,ISUB/0.3/             CRN01370
C      CALLMESSAGE(48H    0.3 ENTER RUN CONTROL )CRN01380
C
C-----CRN01390
C
CL      1.          PROLOGUE            CRN01400
C
C-----CRN01410
C
C      IF(NLRES) GO TO 170                CRN01420
C
C-----CRN01430
C
C      A.          NEW RUN              CRN01440
C
C-----CRN01450
C
CL      1.1         LABEL THE RUN      CRN01460
C
C-----CRN01470
C
110     CALL LABRUN                     CRN01480
C
C-----CRN01490
C
C-----CRN01500
C
C-----CRN01510
C
C-----CRN01520
C
C-----CRN01530
C
C-----CRN01540
C
C-----CRN01550
C
C-----CRN01560
C
C-----CRN01570
C
C-----CRN01580
C
C-----CRN01590
C
C-----CRN01600

```

C			CALL EXPERT(ICLASS,ISUB,1)	CRN01610
CL		1.2	CLEAR VARIABLES AND ARRAYS	CRN01620
	120	CALL CLEAR	CALL EXPERT(ICLASS,ISUB,2)	CRN01630
C		1.3	SET DEFAULT VALUES	CRN01640
CL		130	CALL PRESET	CRN01650
C		1.4	DEFINE DATA SPECIFIC TO RUN	CRN01660
CL		140	CALL DATA	CRN01670
C		1.5	SET AUXILIARY VALUES	CRN01680
CL		150	CALL AUXVAL	CRN01690
C		1.6	DEFINE PHYSICAL INITIAL CONDITIONS	CRN01700
CL		160	CALL INITAL	CRN01710
C			CALL EXPERT(ICLASS,ISUB,4)	CRN01720
C			SET AUXILIARY VALUES	CRN01730
CL			CALL EXPERT(ICLASS,ISUB,5)	CRN01740
C			DEFINE PHYSICAL INITIAL CONDITIONS	CRN01750
CL			CALL EXPERT(ICLASS,ISUB,6)	CRN01760
C			GO TO 180	CRN01770
C		B.	RESUME A PREVIOUS RUN	CRN01780
C			PICK UP RECORD, MODIFY REQUIRED PARAMETERS	CRN01790
CL		1.7	CONTINUE	CRN01800
	170		CALL EXPERT(ICLASS,ISUB,7)	CRN01810
C			LABEL THE CONTINUATION RUN	CRN01820
			CALL LABRUN	CRN01830
C			CLEAR VARIABLES AND ARRAYS	CRN01840
			CALL CLEAR	CRN01850
C			PICK UP RECORD AND PRINT DETAILS	CRN01860
			CALL RESUME	CRN01870
C			PICK UP RECORD AND PRINT DETAILS	CRN01880
			CALL EXPERT(ICLASS,ISUB,8)	CRN01890
C			READ ANY NEW DATA NEEDED	CRN01900
			CALL DATA	CRN01910
C			CALL EXPERT(ICLASS,ISUB,9)	CRN01920
C			MODIFY AUXILIARY VARIABLES AS REQUIRED	CRN01930
			CALL AUXVAL	CRN01940
C			CALL EXPERT(ICLASS,ISUB,10)	CRN01950
C			CALL EXPERT(ICLASS,ISUB,11)	CRN01960
C			CALL EXPERT(ICLASS,ISUB,12)	CRN01970
C			PRELIMINARY OPERATIONS	CRN01980
C			START OR RESTART THE RUN	CRN01990
CL		1.8	CALL START	CRN02000
	180		CALL EXPERT(ICLASS,ISUB,13)	CRN02010
C			INITIAL OUTPUT	CRN02020
			CALL OUTPUT(1)	CRN02030
C			CALL EXPERT(ICLASS,ISUB,14)	CRN02040
C				CRN02050
C				CRN02060
				CRN02070
				CRN02080
C				CRN02090
C				CRN02100
C				CRN02110
C				CRN02120
C				CRN02130
C				CRN02140

```

C----- CRN02150
CL      2.      CALCULATION CRN02160
C----- CRN02170
CL      2.1     STEP ON THE CALCULATION CRN02180
C----- CRN02190
210    CALL STEPON      CALL EXPERT(ICLASS,ISUB,15) CRN02200
C----- CRN02210
CL      3.      OUTPUT CRN02220
C----- CRN02230
CL      3.1     PERIODIC PRODUCTION OF OUTPUT CRN02240
310    CALL OUTPUT(2)      CALL EXPERT(ICLASS,ISUB,16) CRN02250
C----- CRN02260
C----- CRN02270
CL      4.      EPILOGUE CRN02280
C----- CRN02290
CL      4.1     TEST FOR COMPLETION OF RUN CRN02300
C----- CRN02310
CL      410    CALL TSEND      CALL EXPERT(ICLASS,ISUB,17) CRN02320
C----- CRN02330
      IF(,NOT,NLEND) GO TO 210 CRN02340
C----- CRN02350
C----- CRN02360
C----- CRN02370
C----- CRN02380
C----- CRN02390
C----- CRN02400
CL      4.2     TERMINATE THE RUN CRN02410
C----- CRN02420
420    CALL ENDRUN      CRN02430
C----- CRN02440
      RETURN      CRN02450
      END      CRN02460
*DECK C054      CRN02470
C----- CRN02480
C----- CRN02490
SUBROUTINE EXPERT(KCLASS,KSUB,KPOINT)      CRN02500
C----- CRN02510
C 0.4 MODIFY STANDARD OPERATION OF PROGRAM      CRN02520
C----- CRN02530
C----- CRN02540
VERSION 2C          17/10/73      KVR/MHH      CULHAM      CRN02550
C----- CRN02560
*CALL COMDDP      CRN02570
C----- CRN02580
C----- CRN02590
C----- CRN02600
SET TRACER VARIABLES      CRN02610
      NCLASS = KCLASS      CRN02620
      NSUB = KSUB      CRN02630
      NPOINT = KPOINT      CRN02640
C----- CRN02650
C----- CRN02660
ARE DIAGNOSTICS REQUIRED ?      CRN02670
      IF(NLREPT) CALL REPORT(KCLASS,KSUB,KPOINT)      CRN02680
C----- CRN02690
C----- CRN02700
      RETURN      CRN02710
      END      CRN02720
*DECK C1S1
C----- CRN02730

```

```

SUBROUTINE LABRUN CRN02690
C CRN02700
C 1.1 LABEL THE RUN CRN02710
C CRN02720
C VERSION 2C 17/10/73 KVR/MHH CULHAM CRN02730
C CRN02740
C----- CRN02750
C CRN02760
C CALLMESSAGE(48H 1.1 LABEL THE RUN )CRN02770
C CRN02780
C RETURN CRN02790
C END CRN02800
*DECK C1S2 CRN02810
C CRN02820
C SUBROUTINE CLEAR CRN02830
C CRN02840
C 1.2 CLEAR VARIABLES AND ARRAYS CRN02850
C CRN02860
C VERSION 2C 17/10/73 KVR/MHH CULHAM CRN02870
C CRN02880
C----- CRN02890
C CRN02900
C CALLMESSAGE(48H 1.2 CLEAR VARIABLES AND ARRAYS )CRN02910
C CRN02920
C RETURN CRN02930
C END CRN02940
*DECK C1S3 CRN02950
C CRN02960
C SUBROUTINE PRESET CRN02970
C CRN02980
C 1.3 SET DEFAULT VALUES CRN02990
C CRN03000
C VERSION 2C 17/10/73 KVR/MHH CULHAM CRN03010
C CRN03020
C----- CRN03030
C CRN03040
C CALLMESSAGE(48H 1.3 SET DEFAULT VALUES )CRN03050
C CRN03060
C RETURN CRN03070
C END CRN03080
*DECK C1S4 CRN03090
C CRN03100
C SUBROUTINE DATA CRN03110
C CRN03120
C 1.4 DEFINE DATA SPECIFIC TO RUN CRN03130
C CRN03140
C VERSION 2C 17/10/73 KVR/MHH CULHAM CRN03150
C CRN03160
C----- CRN03170
C CRN03180
C CALLMESSAGE(48H 1.4 DEFINE DATA SPECIFIC TO RUN )CRN03190
C CRN03200
C RETURN CRN03210
C END CRN03220

```

```

*DECK C155
C
    SUBROUTINE AUXVAL
C 1.5 SET AUXILIARY VALUES
C
C     VERSION 2C      17/10/73      KVR/MHH      CULHAM
C----->
C     CALLMESSAGE(48H  1.5 SET AUXILIARY VALUES
C
C         RETURN
C         END
*DECK C156
C
    SUBROUTINE INITIAL
C 1.6 DEFINE PHYSICAL INITIAL CONDITIONS
C
C     VERSION 2C      17/10/73      KVR/MHH      CULHAM
C----->
C     CALLMESSAGE(48H  1.6 DEFINE PHYSICAL INITIAL CONDITIONS
C
C         RETURN
C         END
*DECK C157
C
    SUBROUTINE RESUME
C 1.7 RESUME FROM PREVIOUS RECORD
C
C     VERSION 2C      17/10/73      KVR/MHH      CULHAM
C----->
C     CALLMESSAGE(48H  1.7 RESUME FROM PREVIOUS RECORD
C     CALL IVAR(8HCHANNEL +NRESUM)
C     CALL IVAR(8HRECORD  +NREC)
C
C         RETURN
C         END
*DECK C158
C
    SUBROUTINE START
C 1.8 START OR RESTART THE RUN
C
C     VERSION 2C      17/10/73      KVR/MHH      CULHAM
C----->
C
CRN03230
CRN03240
CRN03250
CRN03260
CRN03270
CRN03280
CRN03290
CRN03300
CRN03310
CRN03320
)CRN03330
CRN03340
CRN03350
CRN03360
CRN03370
CRN03380
CRN03390
CRN03400
CRN03410
CRN03420
CRN03430
CRN03440
CRN03450
CRN03460
)CRN03470
CRN03480
CRN03490
CRN03500
CRN03510
CRN03520
CRN03530
CRN03540
CRN03550
CRN03560
CRN03570
CRN03580
CRN03590
CRN03600
CRN03610
)CRN03620
CRN03630
CRN03640
CRN03650
CRN03660
CRN03670
CRN03680
CRN03690
CRN03700
CRN03710
CRN03720
CRN03730
CRN03740
CRN03750
CRN03760

```

```

C           CALLMESSAGE(48H  1,8 START OR RESTART THE RUN          CRN03770
C           RETURN                                              )CRN03780
C           END                                                 CRN03790
*DECK C2S1
C           SUBROUTINE STEPON                                     CRN03800
C 2.1  STEP ON THE CALCULATION                                CRN03810
C           VERSION 2C      17/10/73      KVR/MHH      CULHAM    CRN03820
C           NSTEP=NSTEP+1                                         CRN03830
*CALL COMBAS
C-----                                          CRN03840
C           CALLMESSAGE(48H  2.1 STEP ON THE CALCULATION          CRN03850
C           NSTEP=NSTEP+1                                         CRN03860
C           RETURN                                              CRN03870
C           END                                                 CRN03880
*DECK C3S1
C           SUBROUTINE OUTPUT(K)                                 CRN03890
C 3.1  CONTROL THE OUTPUT                                    CRN03900
C           VERSION 2C      17/10/73      KVR/MHH      CULHAM    CRN03910
C-----                                          CRN03920
C           CALLMESSAGE(48H  3.1 CONTROL THE OUTPUT          CRN03930
C           CALL IVAR(8HCALL   ,K)                           CRN03940
C           IF(K.EQ.1)                                         CRN03950
C           +  CALLMESSAGE(48H  INITIAL OUTPUT (1)           CRN03960
C           IF(K.EQ.2)                                         CRN03970
C           +  CALLMESSAGE(48H  PERIODIC OUTPUT (2)          CRN03980
C           IF(K.EQ.3)                                         CRN03990
C           +  CALLMESSAGE(48H  FINAL OUTPUT (3)            CRN04000
C           RETURN                                              CRN04010
C           END                                                 CRN04020
*DECK C4S1
C           SUBROUTINE TESEND                                     CRN04030
C 4.1  TEST FOR COMPLETION OF RUN                            CRN04040
C           VERSION 2C      17/10/73      KVR/MHH      CULHAM    CRN04050
C-----                                          CRN04060
*CALL COMBAS
C-----                                          CRN04070
C           CALLMESSAGE(48H  3.1 CONTROL THE OUTPUT          CRN04080
C           CALL IVAR(8HCALL   ,K)                           CRN04090
C           IF(K.EQ.1)                                         CRN04100
C           +  CALLMESSAGE(48H  INITIAL OUTPUT (1)           CRN04110
C           IF(K.EQ.2)                                         CRN04120
C           +  CALLMESSAGE(48H  PERIODIC OUTPUT (2)          CRN04130
C           IF(K.EQ.3)                                         CRN04140
C           +  CALLMESSAGE(48H  FINAL OUTPUT (3)            CRN04150
C           RETURN                                              CRN04160
C           END                                                 CRN04170
*DECK C5S1
C           SUBROUTINE TESEND                                     CRN04180
C 5.1  TEST FOR COMPLETION OF RUN                            CRN04190
C           VERSION 2C      17/10/73      KVR/MHH      CULHAM    CRN04200
C-----                                          CRN04210
*CALL COMBAS
C-----                                          CRN04220
C           CALLMESSAGE(48H  5.1 TEST FOR COMPLETION OF RUN    CRN04230
C           RETURN                                              CRN04240
*DECK C6S1
C           SUBROUTINE TESEND                                     CRN04250
C 6.1  TEST FOR COMPLETION OF RUN                            CRN04260
C           VERSION 2C      17/10/73      KVR/MHH      CULHAM    CRN04270
C-----                                          CRN04280
*CALL COMBAS
C-----                                          CRN04290
C-----                                          CRN04300

```

```

C      CALLMESSAGE(48H  4.1 TEST FOR COMPLETION OF RUN          CRN04310
C      IF(NSTEP.GE.NRUN) NLEND = .TRUE.                      )CRN04320
C
C      RETURN                                                 CRN04330
C      END                                                   CRN04340
*DECK C4S2                                              CRN04350
C
C      SUBROUTINE ENDRUN                                     CRN04360
C
C 4.2 TERMINATE THE RUN                                 CRN04370
C
C      VERSION 3C           14/5/75       KVR/MHH/GGL   CULHAM CRN04380
C
*CALL COMBAS                                         CRN04390
C-----                                             CRN04400
C      DATA ICLASS,ISUB/4,2/                               CRN04410
C-----                                             CRN04420
C
C      IF(.NOT.NLEND) GO TO 100                         CALL EXPERT(ICLASS,ISUB,1) CRN04430
C
C      CALL BLINES(2)                                     CRN04440
C      CALLMESSAGE(48H  4.2 TERMINATE THE RUN          )CRN04450
C      GO TO 101                                         CRN04460
C
C 100     CONTINUE                                         CRN04470
C      CALL OUTPUT(3)                                     CRN04480
C
C      CALL BLINES(2)                                     CRN04490
C      CALLMESSAGE(48H  4.2 ABNORMAL EXIT          )CRN04500
C
C      PRINT THE FINAL TIMES                            CRN04510
C
C 101     CALL BLINES(2)                                     CRN04520
C      CALL DAYTIM                                       CRN04530
C      CALL RUNTIM                                       CRN04540
C
C      STOP                                              CRN04550
C      END                                               CRN04560
*DECK C5S1                                              CRN04570
C
C      SUBROUTINE REPORT(KCLASS,KSUB,KPOINT)            CRN04580
C
C 5.1 DIAGNOSTIC REPORTING SUBROUTINE                 CRN04590
C
C      VERSION 2C           17/10/73       KVR/MHH       CULHAM CRN04600
C
C-----                                             CRN04610
C
C      CALL DUMCOM(KCLASS,KSUB,KPOINT)                  CRN04620
C
C      RETURN                                           CRN04630
C      END                                              CRN04640
*DECK C5S2                                              CRN04650
C

```

SUBROUTINE CLIST(KGROUP,KBLOCK)	CRN04850
C	CRN04860
C 5.2 PRINT COMMON VARIABLES	CRN04870
C	CRN04880
C VERSION 2C 17/10/73 KVR/MHH CULHAM	CRN04890
C-----	CRN04900
C	CRN04910
C-----	CRN04920
C CALLMESSAGE(48H 5.2 PRINT COMMON VARIABLES)CRN04930
IF(KGROUP.NE.0) CALL IVAR(8HGROUP 'KGROUP)	CRN04940
IF(KBLOCK.NE.0) CALL IVAR(8HBLOCK 'KBLOCK)	CRN04950
C	CRN04960
RETURN	CRN04970
END	CRN04980
*DECK C5S3	CRN04990
C	CRN05000
SUBROUTINE ARRAYS(KGROUP,KBLOCK)	CRN05010
C	CRN05020
C 5.3 PRINT COMMON ARRAYS	CRN05030
C	CRN05040
C VERSION 2C 17/10/73 KVR/MHH CULHAM	CRN05050
C-----	CRN05060
C	CRN05070
C-----	CRN05080
C	CRN05090
C CALLMESSAGE(48H 5.3 PRINT COMMON ARRAYS)CRN05100
IF(KGROUP.NE.0) CALL IVAR(8HGROUP 'KGROUP)	CRN05110
IF(KBLOCK.NE.0) CALL IVAR(8HBLOCK 'KBLOCK)	CRN05120
C	CRN05130
RETURN	CRN05140
END	CRN05150

```

*DECK U1
C
C      SUBROUTINE MESSAGE(KMESS)
C
C U.1 PRINT 48-CHARACTER MESSAGE ON OUTPUT CHANNEL
C
C      VERSION 1C          17/10/73      KVR/MHH      CULHAM
C
*CALL COMBAS
C-----DIMENSION KMESS(12)
C-----WRITE(NOUT,9900) (KMESS(J),J=1,12)
C
C      RETURN
9900   FORMAT(4X,12A4)
END

*DECK U2
C
C      SUBROUTINE PAGE
C
C U.2 FETCH NEW PAGE ON OUTPUT CHANNEL
C
C      VERSION 1C          17/10/73      KVR/MHH      CULHAM
C
*CALL COMBAS
C-----WRITE(NOUT,9900)
C
C      RETURN
9900   FORMAT(1H1)
END

*DECK U3
C
C      SUBROUTINE BLINES(K)
C
C U.3 INSERT BLANK LINES ON OUTPUT CHANNEL
C
C      VERSION 1C          17/10/73      KVR/MHH      CULHAM
C
*CALL COMBAS
C-----DO 100 J=1,K
100    WRITE(NOUT,9900)
C
C      RETURN
9900   FORMAT(1H )
END

*DECK U4

```

CYC00010
CYC00020
CYC00030
CYC00040
CYC00050
CYC00060
CYC00070
CYC00080
CYC00090
CYC00100
CYC00110
CYC00120
CYC00130
CYC00140
CYC00150
CYC00160
CYC00170
CYC00180
CYC00190
CYC00200
CYC00210
CYC00220
CYC00230
CYC00240
CYC00250
CYC00260
CYC00270
CYC00280
CYC00290
CYC00300
CYC00310
CYC00320
CYC00330
CYC00340
CYC00350
CYC00360
CYC00370
CYC00380
CYC00390
CYC00400
CYC00410
CYC00420
CYC00430
CYC00440
CYC00450
CYC00460
CYC00470
CYC00480
CYC00490
CYC00500
CYC00510
CYC00520

```

C          SUBROUTINE RVAR(KNAME,PVALUE) CYC00530
C          U.4  PRINT NAME AND VALUE OF REAL VARIABLE CYC00540
C          VERSION 1C      17/10/73      KVR/MHH      CULHAM CYC00550
C          *CALL COMBAS CYC00560
C----- CYC00570
C          DIMENSION KNAME(2) CYC00580
C          WRITE(NOUT,9900) KNAME, PVALUE CYC00590
C          RETURN CYC00600
C          9900 FORMAT(4X,2A4,2H =,1PE12.4) CYC00610
C          END CYC00620
*DECK U5 CYC00630
C          SUBROUTINE IVAR(KNAME,KVALUE) CYC00640
C          U.5  PRINT NAME AND VALUE OF INTEGER VARIABLE CYC00650
C          VERSION 1C      17/10/73      KVR/MHH      CULHAM CYC00660
C          *CALL COMBAS CYC00670
C----- CYC00680
C          DIMENSION KNAME(2) CYC00690
C----- CYC00700
C          WRITE(NOUT,9900) KNAME, KVALUE CYC00710
C          RETURN CYC00720
C          9900 FORMAT(4X,2A4,2H =,I12) CYC00730
C          END CYC00740
*DECK U6 CYC00750
C          SUBROUTINE HVAR(KNAME,KVALUE) CYC00760
C          U.6  PRINT NAME AND VALUE OF HOLLERITH VARIABLE CYC00770
C          VERSION 1C      17/10/73      KVR/MHH      CULHAM CYC00780
C          *CALL COMBAS CYC00790
C----- CYC00800
C          DIMENSION KNAME(2) CYC00810
C----- CYC00820
C          WRITE(NOUT,9900) KNAME, KVALUE CYC00830
C          RETURN CYC00840
C          9900 FORMAT(4X,2A4,2H =,2A4) CYC00850
C          END CYC00860
*DECK U7 CYC00870
C          SUBROUTINE LVAR(KNAME,KLVAL) CYC00880
C----- CYC00890
C          RETURN CYC00900
C          9900 FORMAT(4X,2A4,2H =,2A4) CYC00910
C----- CYC00920
C          END CYC00930
C----- CYC00940
C----- CYC00950
C----- CYC00960
C----- CYC00970
C----- CYC00980
C----- CYC00990
C----- CYC01000
C----- CYC01010
C----- CYC01020
C----- CYC01030
C----- CYC01040
C----- CYC01050
C----- CYC01060

```

```

C
C U.7 PRINT NAME AND VALUE OF LOGICAL VARIABLE          CYC01070
C
C      VERSION 1C           17/10/73       KVR/MHH        CULHAM      CYC01080
C
C *CALL COMBAS
C----- CYC01090
C----- CYC01100
C----- CYC01110
C----- CYC01120
C----- CYC01130
C----- CYC01140
C----- CYC01150
C----- CYC01160
C----- CYC01170
C----- CYC01180
C----- CYC01190
C----- CYC01200
C----- CYC01210
C----- CYC01220
C----- CYC01230
C----- CYC01240
C----- CYC01250
C----- CYC01260
C----- CYC01270
C----- CYC01280
C----- CYC01290
C----- CYC01300
C----- CYC01310
C----- CYC01320
C----- CYC01330
C----- CYC01340
C----- CYC01350
C----- CYC01360
C----- CYC01370
C----- CYC01380
C----- CYC01390
C----- CYC01400
C----- CYC01410
C----- CYC01420
C----- CYC01430
C----- CYC01440
C----- CYC01450
C----- CYC01460
C----- CYC01470
C----- CYC01480
C----- CYC01490
C----- CYC01500
C----- CYC01510
C----- CYC01520
C----- CYC01530
C----- CYC01540
C----- CYC01550
C----- CYC01560
C----- CYC01570
C----- CYC01580
C----- CYC01590
C----- CYC01600

*DECK U8
C
C      SUBROUTINE RARRAY(KNAME,PA,KDIM)
C
C U.8 PRINT NAME AND VALUES OF REAL ARRAY
C
C      VERSION 2C           17/10/73       KVR/MHH        CULHAM      CYC01280
C
C *CALL COMBAS
C----- CYC01290
C----- CYC01300
C----- CYC01310
C----- CYC01320
C----- CYC01330
C----- CYC01340
C----- CYC01350
C----- CYC01360
C----- CYC01370
C----- CYC01380
C----- CYC01390
C----- CYC01400
C----- CYC01410
C----- CYC01420
C----- CYC01430
C----- CYC01440
C----- CYC01450
C----- CYC01460
C----- CYC01470
C----- CYC01480
C----- CYC01490
C----- CYC01500
C----- CYC01510
C----- CYC01520
C----- CYC01530
C----- CYC01540
C----- CYC01550
C----- CYC01560
C----- CYC01570
C----- CYC01580
C----- CYC01590
C----- CYC01600

*DECK U9
C
C      SUBROUTINE IARRAY(KNAME,KA,KDIM)
C
C U.9 PRINT NAME AND VALUES OF INTEGER ARRAY
C
C      VERSION 2C           17/10/73       KVR/MHH        CULHAM      CYC01510
C
C *CALL COMBAS
C----- CYC01520
C----- CYC01530
C----- CYC01540
C----- CYC01550
C----- CYC01560
C----- CYC01570
C----- CYC01580
C----- CYC01590
C----- CYC01600

```

```

CALL BLINES(1) CYC01610
WRITE(NOUT,9900) KNAME CYC01620
CALL BLINES(1) CYC01630
WRITE(NOUT,9901) (KA(J),J=1,KDIM) CYC01640
CYC01650
CYC01660
CYC01670
CYC01680
CYC01690
CYC01700
CYC01710
CYC01720
CYC01730
CYC01740
CYC01750
CYC01760
CYC01770
CYC01780
CYC01790
CYC01800
CYC01810
CYC01820
CYC01830
CYC01840
CYC01850
CYC01860
CYC01870
CYC01880
CYC01890
CYC01900
CYC01910
CYC01920
CYC01930
CYC01940
CYC01950
CYC01960
CYC01970
CYC01980
CYC01990
CYC02000
CYC02010
CYC02020
CYC02030
CYC02040
CYC02050
CYC02060
CYC02070
CYC02080
CYC02090
CYC02100
CYC02110
CYC02120
CYC02130
CYC02140
C      RETURN
9900  FORMAT(4X,2A4)
9901  FORMAT((6X,10(I12)))
END
*DECK U10
C      SUBROUTINE HARRAY(KNAME,KA,KDIM)
C U.10 PRINT NAME AND VALUES OF HOLLERITH ARRAY
C      VERSION 2C      17/10/73      KVR/MHH      CULHAM
C
*CALL COMBAS
C----- DIMENSION KNAME(2)
DIMENSION KA(KDIM)
C----- CALL BLINES(1)
WRITE(NOUT,9900) KNAME
CALL BLINES(1)
WRITE(NOUT,9901) (KA(J),J=1,KDIM)
C      RETURN
9900  FORMAT(4X,2A4)
9901  FORMAT((6X,10(2X,A10)))
END
*DECK U11
C      SUBROUTINE REPTHD(KCLASS,KSUB,KPOINT)
C U.11 PRINT HEADING FOR DIAGNOSTICS REPORT
C      VERSION 1C      17/10/73      KVR/MHH      CULHAM
C
*CALL COMBAS
C----- WRITE(NOUT,9900) KCLASS,KSUB,KPOINT
C      RETURN
9900  FORMAT(4X,7HCLASS =,I6,2H, ,12HSUBPROGRAM =,I6,9H, POINT =,I6)
END
*DECK U12
C      SUBROUTINE RUNTIM
C U.12 UPDATE CPU TIME (SECS) AND PRINT IT
C

```

```

C      VERSION 2C          4/11/74      KVR/MHH/GGL      JUTPHAAS      CYC02150
C                                         CYC02160
C                                         CYC02170
C                                         CYC02180
C                                         CYC02190
C                                         CYC02200
C                                         CYC02210
C                                         CYC02210
C                                         CYC02220
C                                         CYC02230
C                                         CYC02240
C                                         CYC02250
C                                         CYC02260
C                                         CYC02270
C                                         CYC02280
C                                         CYC02290
C                                         CYC02300
C                                         CYC02310
C                                         CYC02320
C                                         CYC02330
C                                         CYC02340
C                                         CYC02350
C                                         CYC02360
C                                         CYC02370
C                                         CYC02380
C                                         CYC02390
C                                         CYC02400
C                                         CYC02410
C                                         CYC02420
C                                         CYC02430
C                                         CYC02440
C                                         CYC02450
C                                         CYC02460
C                                         CYC02470
C                                         CYC02480
C                                         CYC02490
C                                         CYC02500
C                                         CYC02510
C                                         CYC02520
C                                         CYC02530
C                                         CYC02540
C                                         CYC02550
C                                         CYC02560
C                                         CYC02570
C                                         CYC02580
C                                         CYC02590
C                                         CYC02600
C                                         CYC02610
C                                         CYC02620
C                                         CYC02630
C                                         CYC02640
C                                         CYC02650
C                                         CYC02660
C                                         CYC02670
C                                         CYC02680
C-----*CALL COMBAS
C-----*CLOCKM* IS A FACOM ROUTINE GIVING THE CPU-TIME (MSEC)
C-----CALL CLOCKM(KCPU)
C-----CPTIME=FLOAT(KCPU)
C-----CPTIME=CPTIME-STIME
C-----CPTIME=CPTIME/1000.0
C-----WRITE(NOUT,9900) CPTIME
C-----RETURN
C-----9900 FORMAT(5X,22HCPU TIME USED SO FAR =,1PE12.4,5H SECS)
C-----END
*DECK U13
C-----SUBROUTINE DAYTIM
C-----U.13 PRINT DATE AND TIME
C-----VERSION 2C          17/10/73      KVR/MHH      CULHAM      CYC02370
C-----*CALL COMBAS
C-----DIMENSION KDATE(2)
C-----*DATE* AND *TIME* ARE FACOM ROUTINES GIVING THE DATE AND TIME
C-----CALL DATE(KDATE)
C-----CALL TIME(KTIME)
C-----ZHOUR=FLOAT(KTIME)/3600000.0
C-----IHOUR=ZHOUR
C-----ZMINU=(ZHOUR-FLOAT(IHOUR))*60.0
C-----IMINU=ZMINU
C-----ISEC =(ZMINU-FLOAT(IMINU))*60.0
C-----WRITE(NOUT,9900) KDATE,IHOUR,IMINU,ISEC
C-----RETURN
C-----9900 FORMAT(5X,4HDATE,4X,2A4,8X,4HTIME,4X,I2,1H:,I2,1H:,I2)
C-----END
*DECK U14
C-----SUBROUTINE RESETR(PA,KDIM,PVALUE)
C-----U.14 RESET REAL ARRAY TO SPECIFIED VALUE
C-----VERSION 1C          17/10/73      KVR/MHH      CULHAM      CYC02640
C-----DIMENSION PA(KDIM)
C-----
```

```

C          DO 100 J=1,KDIM      CYC02690
C          PA(J) = PVALUE      CYC02700
100      CONTINUE            CYC02710
C          RETURN             CYC02720
C          END                CYC02730
C          RETURN             CYC02740
C          END                CYC02750
*DECK U15           RETURN             CYC02760
C          SUBROUTINE RESETICKA,KDIM,KVALUE) CYC02770
C          U,15 RESET INTEGER ARRAY TO SPECIFIED VALUE CYC02780
C          VERSION 1C          17/10/73      KVR/MHH      CULHAM CYC02790
C----- DIMENSION KA(KDIM) CYC02800
C----- CYC02810
C          DO 100 J=1,KDIM      CYC02820
C          KA(J) = KVALUE      CYC02830
100      CONTINUE            CYC02840
C          RETURN             CYC02850
C          END                CYC02860
*DECK U16           RETURN             CYC02870
C          SUBROUTINE RESETHKA,KDIM,KVALUE) CYC02880
C          U,16 RESET HOLLERITH ARRAY TO SPECIFIED VALUE CYC02890
C          VERSION 1C          17/10/73      KVR/MHH      CULHAM CYC02900
C----- DIMENSION KA(KDIM), KVALUE(1) CYC02910
C----- CYC02920
C          DO 100 J=1,KDIM      CYC02930
C          KA(J) = KVALUE(1)    CYC02940
100      CONTINUE            CYC02950
C          RETURN             CYC02960
C          END                CYC02970
*DECK U17           RETURN             CYC02980
C          SUBROUTINE JOBTIM(PTIME) CYC02990
C          U,17 FETCH ALLOCATED JOBTIME (SECS) CYC03000
C          VERSION 2C          4/11/74      KVR/MHH/GGL      JUTPHAA CYC03010
C          JAERI MACHINE FACOM-230/75 HAS NO ROUTINE GIVING CYC03020
C          THE ALLOCATED CPU TIME. SO THAT THIS ROUTINE IS DUMMY. CYC03030
C----- CYC03040
C----- CYC03050
C----- CYC03060
C----- CYC03070
C----- CYC03080
C----- CYC03090
C----- CYC03100
C----- CYC03110
C----- CYC03120
C----- CYC03130
C----- CYC03140
C----- CYC03150
C----- CYC03160
C----- CYC03170
C----- CYC03180
C----- CYC03190
C----- CYC03200
C----- CYC03210
C----- CYC03220
C-----
```

```

INTEGER TIMCPU CYC03230
C CYC03240
C TIMCPU IS A COMPASS SUBROUTINE WHICH USES SYSTEM MACRO CYC03250
C *TIME* TO RETURN ALLOCATED JOBTIME CYC03260
C IT = TIMCPU(0) CYC03270
C PTIME = FLOAT(IT) CYC03280
C CYC03290
C RETURN CYC03300
C END CYC03310
*DECK U18 CYC03320
C SUBROUTINE LARRAY(KNAME,KLA,KDIM) CYC03330
C U,18 PRINT NAME AND VALUES OF LOGICAL ARRAY CYC03340
C VERSION 1C 17/10/73 KVR/MHH CULHAM CYC03350
C CYC03360
C *CALL COMBAS CYC03370
C -----
C DIMENSION KNAME(2) CYC03380
C LOGICAL KLA CYC03390
C DIMENSION KLA(KDIM) CYC03400
C -----
C CALL BLINES(1) CYC03410
C WRITE(NOUT,9900) KNAME CYC03420
C CALL BLINES(1) CYC03430
C WRITE(NOUT,9901) (KLA(J),J=1,KDIM) CYC03440
C -----
C RETURN CYC03450
9900 FORMAT(4X,A8) CYC03460
9901 FORMAT((6X,10(L12))) CYC03470
C -----
*DECK U19 CYC03480
C SUBROUTINE RESETL(KLA,KDIM,KLVAL) CYC03490
C U,19 RESET LOGICAL ARRAY TO SPECIFIED VALUE CYC03500
C VERSION 1C 17/10/73 KVR/MHH CULHAM CYC03510
C CYC03520
C -----
C LOGICAL KLA, KLVAL CYC03530
C DIMENSION KLA(KDIM) CYC03540
C -----
C DO 100 J=1,KDIM CYC03550
C KLA(J) = KLVAL CYC03560
C CONTINUE CYC03570
100
C RETURN CYC03580
C END CYC03590
*DECK U20 CYC03600
C CYC03610
C -----

```

```

SUBROUTINE RARAY2(KNAME,PA,KDIMX,KX,KY) CYC03770
C CYC03780
C U,20 PRINT DOUBLY-SUBSCRIPTED ARRAY CYC03790
C CYC03800
C VERSION 2C 17/10/73 KVR/MHH CULHAM CYC03810
C CYC03820
C *CALL COMBAS CYC03830
C----- CYC03840
C DIMENSION KNAME(2) CYC03850
C DIMENSION PA(1) CYC03860
C----- CYC03870
C----- CYC03880
C----- CYC03890
C----- CYC03900
C----- CYC03910
C----- CYC03920
C----- CYC03930
C----- CYC03940
C----- CYC03950
C----- CYC03960
C----- CYC03970
C----- CYC03980
C----- CYC03990
C----- CYC04000
C----- CYC04010
C----- CYC04020
C----- CYC04030
C----- CYC04040
C----- CYC04050
C----- CYC04060
C----- CYC04070
C----- CYC04080
C----- CYC04090
C----- CYC04100
C----- CYC04110
C----- CYC04120
C----- CYC04130
C----- CYC04140
C----- CYC04150
C----- CYC04160
C----- CYC04170
C----- CYC04180
C----- CYC04190
C----- CYC04200
C----- CYC04210
C----- CYC04220
C----- CYC04230
C----- CYC04240
C----- CYC04250
C----- CYC04260
C----- CYC04270
C----- CYC04280
C----- CYC04290
C----- CYC04300
*DECK U21
C----- SUBROUTINE SCALER(PA,KDIM,PC)
C----- CYC04100
C----- U,21 SCALE A REAL ARRAY BY A REAL VALUE CYC04110
C----- CYC04120
C----- VERSION 1C 17/10/73 KVR/MHH CULHAM CYC04130
C----- CYC04140
C----- CYC04150
C----- CYC04160
C----- CYC04170
C----- CYC04180
C----- CYC04190
C----- CYC04200
C----- CYC04210
C----- CYC04220
C----- CYC04230
C----- CYC04240
C----- CYC04250
C----- CYC04260
C----- CYC04270
C----- CYC04280
C----- CYC04290
C----- CYC04300
*DECK U22
C----- SUBROUTINE SCALEI(KA,KDIM,KC)
C----- CYC04100
C----- U,22 SCALE AN INTEGER ARRAY BY AN INTEGER VALUE CYC04110
C----- CYC04120
C----- CYC04130
C----- CYC04140
C----- CYC04150
C----- CYC04160
C----- CYC04170
C----- CYC04180
C----- CYC04190
C----- CYC04200
C----- CYC04210
C----- CYC04220
C----- CYC04230
C----- CYC04240
C----- CYC04250
C----- CYC04260
C----- CYC04270
C----- CYC04280
C----- CYC04290
C----- CYC04300

```

```

C      VERSION 1C          17/10/73      KVR/MHH      CULHAM      CYC04310
C
C-----DIMENSION KA(KDIM)-----CYC04320
C-----CYC04330
C-----CYC04340
C-----CYC04350
C-----CYC04360
C-----CYC04370
C-----DO 100 J=1,KDIM      CYC04380
C-----KA(J) = KA(J) * KC
C-----CONTINUE
C-----100
C-----RETURN
C-----END
*DECK U23      SUBROUTINE COPYR(PA1,K1,PA2,K2,KDIM)
C
C-----U,23 COPY ONE REAL ARRAY INTO ANOTHER-----CYC04420
C-----CYC04430
C-----CYC04440
C-----CYC04450
C-----CYC04460
C-----CYC04470
C-----CYC04480
C-----VERSION 1C          17/10/73      KVR/MHH      CULHAM      CYC04490
C
C-----DIMENSION PA1(KDIM),PA2(KDIM)-----CYC04500
C-----CYC04510
C-----CYC04520
C-----CYC04530
C-----CYC04540
C-----CYC04550
C-----DO 100 J=1,KDIM      CYC04560
C-----I1 = K1 + J - 1
C-----I2 = K2 + J - 1
C-----PA2(I2) = PA1(I1)
C-----100
C-----CONTINUE
C-----RETURN
C-----END
*DECK U24      SUBROUTINE COPYI(KA1,K1,KA2,K2,KDIM)
C
C-----U,24 COPY ONE INTEGER ARRAY INTO ANOTHER-----CYC04620
C-----CYC04630
C-----CYC04640
C-----CYC04650
C-----CYC04660
C-----CYC04670
C-----CYC04680
C-----VERSION 1C          17/10/73      KVR/MHH      CULHAM      CYC04690
C
C-----DIMENSION KA1(KDIM),KA2(KDIM)-----CYC04700
C-----CYC04710
C-----CYC04720
C-----CYC04730
C-----CYC04740
C-----CYC04750
C-----DO 100 J=1,KDIM      CYC04760
C-----I1 = K1 + J - 1
C-----I2 = K2 + J - 1
C-----KA2(I2) = KA1(I1)
C-----100
C-----CONTINUE
C-----RETURN
C-----END
*DECK U25
C

```

```

SUBROUTINE SIGNR(PA,KDIM) CYC04850
C CYC04860
C U.25 CHANGE THE SIGN OF A REAL ARRAY CYC04870
C CYC04880
C VERSION 1C 17/10/73 KVR/MHH CULHAM CYC04890
C CYC04900
C----- CYC04910
C----- CYC04920
C----- CYC04930
C----- CYC04940
C----- CYC04950
C----- CYC04960
C----- CYC04970
100 CONTINUE CYC04980
C CYC04990
C----- CYC05000
C----- CYC05010
*DECK U26 CYC05020
C CYC05030
C----- CYC05040
C----- CYC05050
C----- CYC05060
C----- CYC05070
C----- CYC05080
C----- CYC05090
C----- CYC05100
C----- CYC05110
C----- CYC05120
C----- CYC05130
C----- CYC05140
C----- CYC05150
100 CONTINUE CYC05160
C CYC05170
C CYC05180
C CYC05190
*DECK U27 CYC05200
C CYC05210
C----- CYC05220
C----- CYC05230
C----- CYC05240
C----- CYC05250
C----- CYC05260
C----- CYC05270
*CALL COMBAS CYC05280
*CALL COMDDP CYC05290
C----- CYC05300
LOGICAL ILREPT CYC05310
DIMENSION IDIGIT(8) CYC05320
DATA IDMAX/8/, ILREPT/, TRUE,/ CYC05330
C----- CYC05340
C----- CYC05350
CL 1. INITIALIZE AND PRINT STEP NUMBER CYC05360
C----- CYC05370
ICODE = 10000 * KCLASS + 100 * KSUB + KPOINT CYC05380
IF(ICODE.NE.20101) GO TO 100

```

```

CALL BLINES(1) CYC05390
CALL IVAR(8HSTEP ,NSTEP) CYC05400
CALLMESSAGE(48H******) CYC05410
100 CONTINUE CYC05420
IF(KCLASS,NE,0) GO TO 101 CYC05430
IF (NLCHED) GO TO 102 CYC05440
GO TO 200 CYC05450
C CYC05460
101 IF(.NOT.NLHEAD(KCLASS))GO TO 200 CYC05470
102 CALL REPTHD(KCLASS,KSUB,KPOINT) CYC05480
    ILREPT=.FALSE. CYC05490
C----- CYC05500
CL 2. SCAN OVER LIST CYC05510
C----- CYC05520
200 CONTINUE CYC05530
DO 800 J=1,MXDUMP CYC05540
IF(NPDUMP(J),NE,1CODE) GO TO 800 CYC05550
C----- CYC05560
CL 3. DUMPING POINT FOUND CYC05570
C----- CYC05580
C PRINT HEADING ONLY ONCE CYC05590
    IF(ILREPT)CALL REPTHD(KCLASS,KSUB,KPOINT) CYC05600
    ILREPT=.FALSE. CYC05610
C----- CYC05620
CL 4. ARE VARIABLES TO BE DUMPED CYC05630
C----- CYC05640
CL I=1 CYC05650
ID=NVDUMP(J) CYC05660
IF(ID,EQ,0)GO TO 500 CYC05670
IF(ID,NE,100)GO TO 600 CYC05680
CALL CLIST(0,0) CYC05690
C----- CYC05700
CL 5. ARE ARRAYS TO BE DUMPED CYC05710
C----- CYC05720
500 CONTINUE CYC05730
I=2 CYC05740
ID=NADUMP(J) CYC05750
IF(ID,EQ,0)GO TO 800 CYC05760
IF(ID,NE,100)GO TO 600 CYC05770
CALL ARRAYS(0,0) CYC05780
GO TO 800 CYC05790
C----- CYC05800
CL 6. DISENTANGLE CODE CYC05810
C----- CYC05820
600 CONTINUE CYC05830
CALL RESET1(IDIGIT,IMAX,0) CYC05840
DO 601 JD=1,IMAX CYC05850
IDIV=ID/10 CYC05860
IDIGIT(JD)=ID-IDIV*10 CYC05870

```

SOURCE ELEMENT LIST (8BIT MODE)

```

IN=JD CYC05930
IF(IDIV,EQ,0)GO TO 700 CYC05940
ID=IDIV CYC05950
CONTINUE CYC05960
CYC05970
CYC05980
CL----- CYC05990
CL 7. ISSUE CALLS CYC06000
C----- CYC06010
700 CONTINUE CYC06020
C----- CYC06030
MAKE *IN* EVEN CYC06040
IN=2*(IN/2) CYC06050
IF(IN,EQ,0)GO TO 800 CYC06060
DO 701 JD=1,IN*2 CYC06070
IJ=IN-JD+1 CYC06080
IG=1DIGIT(IJ) CYC06090
IB=1DIGIT(IJ-1) CYC06100
IF(IG,EQ,0)GO TO 701 CYC06110
IF(I,EQ,1)CALL CLIST(IG,IB) CYC06120
IF(I,EQ,2)CALL ARRAYS(IG,IB) CYC06130
701 CONTINUE CYC06140
IF(I,EQ,1)GO TO 500 CYC06150
CYC06160
CL----- CYC06170
CL 8. NEXT ENTRY IN LIST CYC06180
C----- CYC06190
800 CONTINUE CYC06200
RETURN CYC06210
END CYC06220

```

付録B プリ・プロセッサ「UPDATE」のプログラム・リスト

```

*DECK CHECKS
1   SUBROUTINE CHECKS  C1TAB  :INAME  :IDENT  :ISEBU  :ITYP    UPD00010
2   COMMON   INPUT   :OUTPUT :OLDPL  :NEWPL  :COMPL  :SOURCE  UPD00020
3     1      :IO     :ICOMP :ITEAT :IXOLD  :IXNEW  :IX     UPD00030
4     2      :MASTER :ID    :IFILE :NUM    :IUPDAT UPD00040
5   INTEGER  INPUT   :OUTPUT :OLDPL  :NEWPL  :COMPL  :SOURCE  UPD00050
6   LOGICAL  F       :D     :       :       :       :       UPD00060
7   DIMENSION ITAB   ( 90) :INAME  ( 9, 10) :IDENT  ( 9, 10) :UPD00080
8     1      ISEBU  ( 9, 10) :NEWPL  :       :       UPD00090
9   DIMENSION LTAB   ( 90) :LNAME  ( 90) :LDENT   ( 9) :UPD00100
10    1      LSEBU ( 9) :       :       UPD00110
11   DIMENSION NTAB   ( 9, 30) :       :       UPD00120
12   COMMENT UPDATE CONTROL CHARACTORS SET
13   DATA    ((NTAB(1,J),J=1,8),J=1,10) :       :       UPD00130
14   DATA    ((NTAB(1,J),J=1,8),J=1,10)
15     1      '/ DEC', 'K', '' :       :       UPD00150
16     2      ' COM', 'DECK', '' :       :       UPD00160
17     3      ' IDE', 'INT', '' :       :       UPD00170
18     4      ' INS', 'ERT', '' :       :       UPD00180
19     5      ' BEF', 'ORE', '' :       :       UPD00190
20     6      ' DEL', 'ETE', '' :       :       UPD00200
21     7      ' RES', 'TORY', '' :       :       UPD00210
22     8      ' COP', 'Y', '' :       :       UPD00220
23     9      ' CHA', 'NGE', '' :       :       UPD00230
24     0      ' YAN', 'K', '' :       :       UPD00240
25   DATA    ((NTAB(1,J),J=1,8),J=1,20) :       :       UPD00250
26     1      '/ YAN', 'KDEC', 'K :       :       UPD00260
27     2      ' SEL', 'YANK', '' :       :       UPD00270
28     3      ' PUR', 'GE :       :       UPD00280
29     4      ' PUR', 'DECK', '' :       :       UPD00290
30     5      ' SEL', 'PURG', 'E :       :       UPD00300
31     6      ' ADD', 'FILE', '' :       :       UPD00310
32     7      ' SEQ', 'UENCY', 'E :       :       UPD00320
33     8      ' MOV', 'IE', '' :       :       UPD00330
34     9      ' REA', 'ID', '' :       :       UPD00340
35     0      ' SK', 'IP', '' :       :       UPD00350
36   DATA    ((NTAB(1,J),J=1,8),J=21,27) :       :       UPD00360
37     1      '/ REV', 'IND', '' :       :       UPD00370
38     2      ' COM', 'PILE', '' :       :       UPD00380
39     3      ' WED', 'IR', '' :       :       UPD00390
40     4      ' CAL', 'IL', '' :       :       UPD00400
41     5      ' /', '' :       :       UPD00410
42     6      ' ABB', 'REV', '' :       :       UPD00420
43     7      ' NDA', 'BBRE', 'Y :       :       UPD00430
44   COMMENT
45     TYP = 0 :       :       UPD00450
46     INA = 0 :       :       UPD00460
47     IDE = 0 :       :       UPD00470
48     ISE = 0 :       :       UPD00480
49     IKE = 0 :       :       UPD00490
50     IRE = 0 :       :       UPD00500
51   COMMENT CHECK FIRST COLUMN OF ITAB ARRAY IS OR NOT
52   IF((ITAB(1).NE.MASTER)) GO TO7000 UPD00510
53   CALL   RESETH ((INAME ,90 ,6H ) :       :       UPD00520
54   CALL   RESETH ((IDENT ,90 ,6H ) :       :       UPD00530
55   CALL   RESETH ((ISEQ ,90 ,6H ) :       :       UPD00540
56   DO 4000 N=1,10 :       :       UPD00550
57   INAME(9,N)=0 :       :       UPD00560
58   IDENT(9,N)=0 :       :       UPD00570
59   ISEQ(9,N)=0 :       :       UPD00580
60   4000 CONTINUE :       :       UPD00590
61   DO 3000 N=1,27 :       :       UPD00600
62   LREC = NTAB(7,N) :       :       UPD00610
63   LTYP = NTAB(8,N) :       :       UPD00620
64   CALL   RESETH ((LTAB ,9 ,6H ) :       :       UPD00630
65   CALL   GETBCD (NTAB ( 4, N),LREC ,LTAB ) :       :       UPD00640
66   LREC1 = LREC+1 :       :       UPD00650
67   DO 3010 L=2,LREC1 :       :       UPD00660
68   L = L :       :       UPD00670
69   IF((ITAB(1).NE.LTAB(L))) GOTO 3012 :       :       UPD00680
70   3010 CONTINUE :       :       UPD00690
71   GOTO 3020 :       :       UPD00700
72   3012 CONTINUE :       :       UPD00710
73   LREC = NTAB(6,N) :       :       UPD00720
74   CALL   RESETH ((LTAB ,9 ,6H ) :       :       UPD00730
75   CALL   GETBCD (NTAB (1,(N)+LREC ,LTAB ) :       :       UPD00740
76   DO 3014 L=2,LREC :       :       UPD00750
77   IF((ITAB(1).NE.LTAB(L))) GOTO 3000 :       :       UPD00760
78   3014 CONTINUE :       :       UPD00770
79   3020 CONTINUE :       :       UPD00780
80   TYP = N :       :       UPD00790
81   I = 1+1 :       :       UPD00800
82   IF(I,GT,80) GO TO TD3030 UPD00810
83   IF((ITAB(1).EQ,1H)) GO TO3030 UPD00820
84   IF((ITAB(1).EQ,1H)) GO TO3020 UPD00830
85   I = I-1 :       :       UPD00840
86   3030 CONTINUE :       :       UPD00850
87   INA = 0 :       :       UPD00860
88   IDE = 0 :       :       UPD00870
89   ISE = 0 :       :       UPD00880
90   IKE = 0 :       :       UPD00890
91   IRE = 0 :       :       UPD00900
92   GO TO ( 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000,
93   1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000,
94   2100, 2200, 2300, 2400, 2500, 2600, 2700),ITYP UPD00910
95   COMMENT DNAM1,(DNAM2,...,DNAMEN)
96   COMMENT DNAM1,NAME2,...,:
97   100 CONTINUE :       :       UPD00920
98   200 CONTINUE :       :       UPD00930
99   1100 CONTINUE :       :       UPD00940
100  1400 CONTINUE :       :       UPD00950
101  1700 CONTINUE :       :       UPD00960
102  1800 CONTINUE :       :       UPD00970
103  2200 CONTINUE :       :       UPD00980
104  2400 CONTINUE :       :       UPD00990
105  INA = 0 :       :       UPD01000
106  IDE = 0 :       :       UPD01010
107  ISE = 0 :       :       UPD01020
108  IKE = 0 :       :       UPD01030
109  IRE = 0 :       :       UPD01040
110  GO TO3110 :       :       UPD01050
111  I = 1+1 :       :       UPD01060
112  IF(I,GT,80) GO TO TD3170 UPD01070
113  IF((ITAB(1).EQ,1H)) GO TO3140 UPD01080
114

```

```

76      IF(ITAB(I),EQ,1H) GO TO3170
77      IF(ITAB(I),EQ,1H) GO TO3170
78      IF(IKE,LT,0) GO TO3160
79      IF(IKE,EQ,0) GO TO3125
80      IF(IKE,GT,0) GO TO3130
81      3120 CONTINUE
82      I = I+1
83      3125 CONTINUE
84      INA = INA+1
85      IKE = 1
86      IRE = 0
87      3130 CONTINUE
88      IRE = IRE+1
89      IF(IRE,GT,8) STOP 100
90      3135 CONTINUE
91      INAME(IRE,INAJ=ITAB(I)
92      GO TO3110
93      3140 CONTINUE
94      INAME(9,INA)=IRE
95      I = I+1
96      3150 CONTINUE
97      INA = INA+1
98      IKE = -1
99      IRE = 0
100     3160 CONTINUE
101     IRE = IRE+1
102     IF(IKE,GT,8) STOP 100
103     IDENT(IRE,INA)=ITAB(I)
104     GO TO3135
105     3170 CONTINUE
106     IF(INA,EW,0) STOP 100
107     INAME(9,INA)=IRE
108     IF(IKE,LT,0)
109     IDENT(9,INA)=IRE
110     IF(I,GT,80) GO TO3195
111     IF(ITAB(I),EQ,1H) GO TO3195
112     GO TO3120
113     3195 CONTINUE
114     GO TO7000
115     COMMENT IDENT1(),IDENT2,...,IDENTN)
116     COMMENT IDENT1,IDENT2...
117     300 CONTINUE
118     900 CONTINUE
119     1000 CONTINUE
120     1300 CONTINUE
121     INA = 0
122     IDE = 0
123     3210 CONTINUE
124     I = I+1
125     IF(I,GT,80) GO TO3270
126     IF(ITAB(I),EQ,1H,) GO TO3240
127     IF(ITAB(I),EW,1H+) GO TO3270
128     IF(ITAB(I),EW,1H-) GO TO3270
129     IF(IKE,LT,0) GO TO3260
130     IF(IKE,EW,0) GO TO3225
131     IF(IKE,GT,0) GO TO3230
132     3220 CONTINUE
133     I = I+1
134     3225 CONTINUE
135     IDE = IDE+1
136     IKE = 1
137     IRE = 0
138     3230 CONTINUE
139     IRE = IRE+1
140     IF(IRE,GT,8) STOP 200
141     3235 CONTINUE
142     IDENT(IRE,IDE)=ITAB(I)
143     GO TO3210
144     3240 CONTINUE
145     IDENT(9,IDE)=IRE
146     I = I+1
147     3250 CONTINUE
148     IDE = IDE+1
149     IKE = -1
150     IRE = 0
151     3260 CONTINUE
152     IRE = IRE+1
153     IF(IRE,GT,8) STOP 200
154     INAME(IRE,IDE)=ITAB(I)
155     GO TO3235
156     3270 CONTINUE
157     IF(IDE,EW,0) STOP 200
158     IDENT(9,IDE)=IRE
159     IF(IKE,LT,0)
160     IDENT(9,IDE)=IRE
161     IF(I,GT,80) GO TO3295
162     IF(ITAB(I),EQ,1H) GO TO3295
163     GO TO3220
164     3295 CONTINUE
165     GO TO7000
166     COMMENT DNAMES,IDENT1(),DNAMES2,IDENT2,...,DNAMEN,IDENTN)
167     1200 CONTINUE
168     1300 CONTINUE
169     INA = 0
170     IDE = 0
171     ISE = 0
172     3310 CONTINUE
173     I = I+1
174     IF(I,GT,80) GO TO3370
175     IF(ITAB(I),EQ,1H,) GO TO3340
176     IF(ITAB(I),EW,1H+) GO TO3370
177     IF(ITAB(I),EW,1H-) GO TO3370
178     IF(IKE,LT,0) GO TO3360
179     IF(IKE,EW,0) GO TO3325
180     IF(IKE,GT,0) GO TO3330
181     3320 CONTINUE
182     I = I+1
183     3325 CONTINUE
184     INA = INA+1

```

```

185      IKE = 1
186      IRE = 0
187      3330 CONTINUE
188      IRE = IRE+1
189      IF(IRE,GT,8) STOP 300
190      INAME(IRE,INA)=ITAB(1)
191      GO TO3310
192      3340 CONTINUE
193      INAME(9,INA)=IRE
194      3345 CONTINUE
195      I = I+1
196      3350 CONTINUE
197      IDE = IDE+1
198      IKE = -1
199      IRE = 0
200      3360 CONTINUE
201      IRE = IRE+1
202      IF(IRE,GT,8) STOP 300
203      IDENT(IRE,IDE)=ITAB(1)
204      GO TO3310
205      3370 CONTINUE
206      IF(INE,NE,IDE) STOP 300
207      IDENT(9,IDE)=IRE
208      IF(I,GT,80) GO TO3395
209      IF(ITAB(1),EQ,1H) GO TO3395
210      IF(IKE,LT,0) GO TO3320
211      IF(IKE,GT,0) GO TO3345
212      3395 CONTINUE
213      GO TO7000
COMMENT IDENT1,SE01
COMMENT SE01
COMMENT IDENT1,SE01+IDENT2+SE02
COMMENT IDENT1,SE01,SE02
COMMENT SE01,IDENT2,SE02
COMMENT SE01,SE02
214      400 CONTINUE
215      500 CONTINUE
216      600 CONTINUE
217      700 CONTINUE
218      INA = 0
219      IDE = 0
220      ISE = 0
221      IRE = 0
222      IKE = 0
223      3410 CONTINUE
224      I = I+1
225      IF(I,GT,80) GO TO3470
226      IF(ITAB(1),EQ,1H) GO TO3440
227      IF(ITAB(1),EQ,1H+) GO TO3470
228      IF(ITAB(1),EQ,1H-) GO TO3470
229      IF(IKE,LT,0) GO TO3460
230      IF(IKE,ES,0) GO TO3423
231      IF(IKE,GT,0) GO TO3430
232      3420 CONTINUE
233      I = I+1
234      3425 CONTINUE
235      IDE = IDE+1
236      IKE = 1
237      IRE = 0
238      3430 CONTINUE
239      IRE = IRE+1
240      IF(IRE,GT,8) STOP 400
241      IDENT(IRE,IDE)=ITAB(1)
242      GO TO3410
243      3440 CONTINUE
244      IDENT(9,IDE)=IRE
245      I = I+1
246      3450 CONTINUE
247      ISE = ISE+1
248      IF(IDE,LT,ISE) STOP 400
249      IKE = -1
250      IRE = 0
251      3460 CONTINUE
252      IRE = IRE+1
253      IF(IRE,GT,8) STOP 400
254      ISE01(IRE,ISE)=ITAB(1)
255      GO TO3410
256      3470 CONTINUE
257      IF(IKE,GT,0) IDENT(9,IDE)=IRE
258      IF(IKE,LT,0) ISE01(9,ISE)=IRE
259      IF(IDE,EQ,ISE) GO TO3490
260      ISE = ISE+1
261      CALL COPYH (IDENT ( 1+IDE)+1 ,ISE01 ( 1+ISE),UPD03110
262      1   1   1   1   ) UPD03120
263      IF(IDE,EW,1) GO TO3480 UPD03130
264      CALL COPYH (IDENT ( 1,IDE-1)+1 ,IDENT ( 1,IDE),UPD03140
265      1   1   1   1   ) UPD03150
266      GO TO3490 UPD03160
267      3480 CONTINUE
268      CALL COPYH (LDENT ,1 ,IDENT ( 1, 1),UPD03180
269      1   1   1   1   ) UPD03190
270      IF(I,GT,80) GO TO3495 UPD03200
271      IF(ITAB(1),EQ,1H) GO TO3495 UPD03210
272      IF(IDE,EQ,2) GO TO3495 UPD03220
273      3495 CONTINUE
274      CALL COPYH (IDENT ( 1,IDE)+1 ,LDENT ,UPD03240
275      1   1   1   1   ) UPD03250
276      IKE = 0 UPD03260
277      IF(LTYP,EW,5) GO TO3610 UPD03270
278      GO TO7000 UPD03280
COMMENT DNAME1,IDENT1,SE01,IDENT2,SE02,(FILE1)
COMMENT DNAME1,IDENT1,SE01,SE02,(FILE1)
COMMENT DNAME1,SE01,IDENT2,SE02,(FILE1)
COMMENT DNAME1,SE01,SE02,(FILE1)
279      800 CONTINUE
280      INA = 0 UPD03360
281      IDE = 0 UPD03370
282      ISE = 0 UPD03380
283      IRE = 0 UPD03390
284      IKE = 0 UPD03400
285      3510 CONTINUE
286      I = I+1 UPD03410

```

```

285      IF((),GT,00)      GO  T03570      UPD03430
286      IF((),TAB(1),EQ,1H,)  GO  T03540      UPD03440
287      IF((),TAB(1),EQ,1H,)  GO  T03570      UPD03450
288      IF((),TAB(1),EQ,1H,)  GO  T03570      UPD03460
289      IF((),KE,LT,0)      GO  T03560      UPD03470
290      IF((),KE,EQ,0)      GO  T03525      UPD03480
291      IF((),KE,GT,0)      GO  T03530      UPD03490
292      3520 CONTINUE      UPD03500
293      3525 CONTINUE      UPD03510
294          INA = INA+1      UPD03520
295          IKE = 1           UPD03530
296          IRE = 0           UPD03540
297      3530 CONTINUE      UPD03550
298          IRE = IRE+1      UPD03560
299          IF((IRE,GT,8))    STOP  500      UPD03570
300          INAME(IRE,INA)=ITAB(1)      UPD03580
301          GO  T03510      UPD03590
302      3540 CONTINUE      UPD03600
303          STOP  500      UPD03610
304      3550 CONTINUE      UPD03620
305      3560 CONTINUE      UPD03630
306      3570 CONTINUE      UPD03640
307          IF((INA,EQ,0))    STOP  500      UPD03650
308          INAME(9,INA)=IRE      UPD03660
309          IF((),GT,80)      GO  T03595      UPD03670
310          IF((),TAB(1),EQ,1H,)  GO  T03595      UPD03680
311          IKE = 0           UPD03690
312          GO  T03410      UPD03700
313      3595 CONTINUE      UPD03710
314          STOP  500      UPD03720
COMMENT FILE1
315      1900 CONTINUE      UPD03730
316      2100 CONTINUE      UPD03740
317      3600 CONTINUE      UPD03750
318          INA = 0           UPD03760
319          IDE = 0           UPD03770
320          ISE = 0           UPD03780
321          IRE = 0           UPD03790
322          IKE = 0           UPD03800
323      3610 CONTINUE      UPD03810
324          I = I+1           UPD03820
325          IF((),GT,80)      GO  T03670      UPD03830
326          IF((),TAB(1),EQ,1H,)  GO  T03640      UPD03840
327          IF((),TAB(1),EQ,1H,)  GO  T03670      UPD03850
328          IF((),TAB(1),EQ,1H,)  GO  T03670      UPD03860
329          IF((),KE,LT,0)      GO  T03650      UPD03870
330          IF((),KE,EQ,0)      GO  T03625      UPD03880
331          IF((),KE,GT,0)      GO  T03630      UPD03890
332      3620 CONTINUE      UPD03900
333      3625 CONTINUE      UPD03910
334          INA = INA+1      UPD03920
335          IKE = 1           UPD03930
336          IRE = 0           UPD03940
337      3630 CONTINUE      UPD03950
338          IRE = IRE+1      UPD03960
339          IF((IRE,GT,8))    STOP  600      UPD03970
340          INAME(IRE,INA)=ITAB(1)      UPD03980
341          GO  T03610      UPD03990
342      3640 CONTINUE      UPD04000
343          STOP  600      UPD04010
344      3650 CONTINUE      UPD04020
345      3660 CONTINUE      UPD04030
346      3670 CONTINUE      UPD04040
347          IF((),GT,80)      GO  T03695      UPD04050
348          IF((),TAB(1),EQ,1H,)  GO  T03695      UPD04060
349          IF(LTYP,EQ,8)      GO  T03695      UPD04070
350          STOP  600      UPD04080
351      3695 CONTINUE      UPD04090
352          IF(LTYP,NE,5)      GO  T03696      UPD04100
353          IF((DE,EQ,0))    STOP  600      UPD04110
354      3696 CONTINUE      UPD04120
355          CALL  SETNUM,  (INAME( 1,INA),IRE,  ,FILE  )      UPD04130
356          IF(LTYP,EQ,6)      GO  T03697      UPD04140
357          IF(LTYP,EQ,7)      GO  T03700      UPD04150
358          IF(LTYP,EQ,8)      GO  T0 100      UPD04160
359      3697 CONTINUE      UPD04170
360          GO  T07000      UPD04180
COMMENT FILE1+NUM
361      2000 CONTINUE      UPD04190
362          GO  T03600      UPD04200
363      3700 CONTINUE      UPD04210
364          INA = 0           UPD04220
365          IDE = 0           UPD04230
366          ISE = 0           UPD04240
367          IRE = 0           UPD04250
368          IKE = 0           UPD04260
369      3710 CONTINUE      UPD04270
370          I = I+1           UPD04280
371          IF((),GT,80)      GO  T03770      UPD04290
372          IF((),TAB(1),EQ,1H,)  GO  T03740      UPD04300
373          IF((),TAB(1),EQ,1H,)  GO  T03770      UPD04310
374          IF((),TAB(1),EQ,1H,)  GO  T03770      UPD04320
375          IF((),KE,LT,0)      GO  T03750      UPD04330
376          IF((),KE,EQ,0)      GO  T03725      UPD04340
377          IF((),KE,GT,0)      GO  T03730      UPD04350
378      3720 CONTINUE      UPD04360
379      3725 CONTINUE      UPD04370
380          INA = INA+1      UPD04380
381          IKE = 1           UPD04390
382          IRE = 0           UPD04400
383      3730 CONTINUE      UPD04410
384          IRE = IRE+1      UPD04420
385          IF((IRE,GT,8))    STOP  700      UPD04430
386          INAME(IRE,INA)=ITAB(1)      UPD04440
387          GO  T03710      UPD04450
388      3740 CONTINUE      UPD04460
389          STOP  700      UPD04470
390      3750 CONTINUE      UPD04480
391      3760 CONTINUE      UPD04490
392      3770 CONTINUE      UPD04500
393          IF((),GT,80)      GO  T03795      UPD04510
394          IF((),TAB(1),EQ,1H,)  GO  T03795      UPD04520
395          STOP  700      UPD04530
396      3795 CONTINUE      UPD04540
397          STOP  700      UPD04550
398          STOP  700      UPD04560

```

```

397      CALL      SETNUM   CINAME  ( 1,(NA)+IRE    ,INUM    ) UPD04570
398      GO TO700
399      COMMENT IFILE1,DNAME1 UPD04580
400      1600 CONTINUE UPD04590
400      GO TO3600 UPD04600
400      COMMENT UPD04610
401      2300 CONTINUE UPD04620
402      2500 CONTINUE UPD04630
403      2600 CONTINUE UPD04640
404      2700 CONTINUE UPD04650
405      GO TO7000 UPD04660
406      3000 CONTINUE UPD04670
407      STOP 1000 UPD04680
408      7000 CONTINUE UPD04690
409      RETURN UPD04700
410      END UPD04710
410      UPD04720

*DECK CGMP
1      SUBROUTINE COMP UPD04730
2      COMMON INPUT  +OUTPUT +OLDPL +NEWPL +COMPIL +SOURCE UPD04740
3      1          IO    +ICOMP +ITEXT +IXOLD +IXNEW +IX UPD04750
4      2          F     +D     +FILE  +NUM   +IUPDAT UPD04760
5      INTEGER INPUT  +OUTPUT +OLDPL +NEWPL +COMPIL +SOURCE UPD04770
6      LOGICAL F     ,D     ,KREC  +KIND  +KNUM  +KYAK UPD04780
7      COMMON/TABK/KTAB ( 90),KNAME ( 9),KDENT ( 9),UPD04790
8      1          KSEQU ( 9) UPD04800
9      COMMON/TABT/KSEGU ( 9) UPD04810
10     1          KREC  +KIND  +KNUM  +KYAK UPD04820
11     2          KCOM  +KDUM  ,MNAME ( 9,100),MDENT ( 9,100),UPD04830
12     1          MSEQ ( 100),MREC ( 100),MIND ( 100),UPD04840
13     1          MNUM ( 100),MTYP ( 100),MYAK ( 100),UPD04850
14     1          MCOM ( 100),MDUM ( 100) UPD04860
15     8          DIMENSION ITAB ( 90),INAME ( 9, 10),IDENT ( 9, 10),UPD04870
16     1          ISERU ( 9, 10),LNAME ( 9, 10),LDENT ( 9, 10),UPD04880
17     9          LTAB  ( 90),LNAM ( 9, 10),LPNT ( 9, 10),UPD04890
18     1          LSEQU ( 9, 10) UPD04900
19     10         DIMENSION LREC ( 100),LTND ( 10) UPD04910
20     11         LNUM = 0 UPD04920
21     12         DO 2000 M=2,100 UPD04930
22     13         IF(LNUM.LE.MNUM(M)) LNUM=MNUM(M) UPD04940
23     14         2000 CONTINUE UPD04950
24     15         LNUM = LNUM UPD04960
25     16         ICOMP = 98 UPD04970
26     17         IF(IUPDAT,EQ,1) F = .TRUE. UPD04980
27     18         IF(.NOT.F) GO TO3200 UPD04990
28         COMMENT FULL COMPILE UPD05000
29         DO 3110 M=2,100 UPD05010
30         IF(MTYP(M).NE,1) GO TO3110 UPD05020
31         IF(MYAK(M).NE,0) GO TO3110 UPD05030
32         MCOM(M)= 1 UPD05040
33         3110 CONTINUE UPD05050
34         COMMENT PARTIAL COMPILE UPD05060
35         3200 CONTINUE UPD05070
36         REWIND ICOMP UPD05080
37         3210 CONTINUE UPD05090
38         READ (ICOMP,6000,END=2800) (ITAB(I80),(I80=1,80) UPD05100
39         WRITE (OUTPUT,9010) (ITAB(I80),(I80=1,80) UPD05110
40         COMMENT CHECK COMPILE DECKS UPD05120
41         3220 CONTINUE UPD05130
42         CALL      CHECKS (ITAB ,INAME ,IDENT ,ISERU ,ITYP ) UPD05140
43         IF(ITYP.EQ,1?) GO TO1700 UPD05150
44         IF(F) GO TO3210 UPD05160
45         COMMENT COMPILE UPD05170
46         2200 CONTINUE UPD05180
47         DO 3230 IRE=1,8 UPD05190
48         IF(IDENT(9,2),EQ,0) GO TO3300 UPD05200
49         IF(INAME(IRE,2).NE,IDENT(IRE,2)) GO TO3300 UPD05210
50         3230 CONTINUE UPD05220
51         I     = 1 UPD05230
52         LNUM = 0 UPD05240
53         3241 CONTINUE UPD05250
54         DO 3240 M=2,100 UPD05260
55         IF(I,EQ,3) GO TO3280 UPD05270
56         IF(MNUM(M).NE,LNUM) GO TO3240 UPD05280
57         UPD05290

```

```

44      LNUM = LNUM+1          UPD05300
45      IF(MTYP(M),NE,1)      GO T03240    UPD05310
46      IF(MYAK(M),NE,0)      GO T03240    UPD05320
47      DO 3250 IRE=1,8       UPD05330
48      IF(MNAME(IRE,M),NE,I NAME(IRE,1)) GO T03270    UPD05340
49      3250 CONTINUE         UPD05350
50      I = I+1              UPD05360
51      3260 CONTINUE         UPD05370
52      MCOM(M)= 1           UPD05380
53      GO T03240            UPD05390
54      3270 CONTINUE         UPD05400
55      IF(I,EQ,1)            GO T03240    UPD05410
56      GO T03260            UPD05420
57      3240 CONTINUE         UPD05430
58      IF(LNUM,LE,LLNUM)    GO T03241    UPD05440
59      3280 CONTINUE         UPD05450
60      GO T03210            UPD05460
61      3300 CONTINUE         UPD05470
62      DO 3310 I=1,10        UPD05480
63      DO 3320 IRE=1,8       UPD05490
64      IF(INAME(IRE,I),NE,1H ) GO T03329    UPD05500
65      3320 CONTINUE         UPD05510
66      GO T03210            UPD05520
67      3329 CONTINUE         UPD05530
68      DO 3330 M=2,100       UPD05540
69      IF(MTYP(M),NE,1)      GO T03330    UPD05550
70      IF(MYAK(M),NE,0)      GO T03330    UPD05560
71      DO 3340 IRE=1,8       UPD05570
72      IF(MNAME(IRE,M),NE,I NAME(IRE,1)) GO T03330    UPD05580
73      3340 CONTINUE         UPD05590
74      MCOM(M)= 1           UPD05600
75      GO T03310            UPD05610
76      3330 CONTINUE         UPD05620
77      STOP 3330            UPD05630
78      3310 CONTINUE         UPD05640
79      GO T03210            UPD05650
80      COMMENT SEQUENCE
81      1700 CONTINUE         UPD05660
82      IF(IDENT(9,2),EQ,0)   GO T03500    UPD05670
83      DO 3430 IRE=1,8       UPD05680
84      IF(INAME(IRE,2),NE,IDENT(IRE,2)) GO T03500    UPD05690
85      3430 CONTINUE         UPD05700
86      I = 1                 UPD05710
87      LNUM = 0               UPD05720
88      3441 CONTINUE         UPD05730
89      DO 3440 M=2,100       UPD05740
90      IF(I,EQ,3)            GO T03480    UPD05750
91      IF(MNUM(M),NE,LNUM)  GO T03440    UPD05760
92      LNUM = LNUM+1          UPD05770
93      IF(MTYP(M),NE,1)      GO T03440    UPD05780
94      IF(MYAK(M),NE,0)      GO T03440    UPD05790
95      DO 3450 IRE=1,8       UPD05800
96      IF(MNAME(IRE,M),NE,I NAME(IRE,1)) GO T03470    UPD05810
97      3450 CONTINUE         UPD05820
98      I = I+1              UPD05830
99      3460 CONTINUE         UPD05840
100     MRE = MREC(M)         UPD05850
101     MND = 0               UPD05860
102     LSEQ = 0               UPD05870
103     3451 CONTINUE         UPD05880
104     CALL READT (MRE ,MND ) > UPD05890
105     IF(KDUM,NE,0)          GO T03452    UPD05900
106     LSEQ = LSEQ+1          UPD05910
107     KSEQ = LSEQ            UPD05920
108     CALL COPYH (KNAME ,1 ,KDENT ,1 ,9 ) > UPD05930
109     CALL WRITET (MRE ,MND ) > UPD05940
110     MRE = KREC            UPD05950
111     MND = KIND             UPD05960
112     IF((MRE+MND),EQ,M)   GO T03440    UPD05970
113     GO T03451            UPD05980
114     3470 CONTINUE         UPD05990
115     IF(I,EQ,1)            GO T03440    UPD06000
116     GO T03460            UPD06010
117     3440 CONTINUE         UPD06020
118     IF(LNUM,LE,LLNUM)    GO T03441    UPD06030
119     3480 CONTINUE         UPD06040
120     GO T03210            UPD06050
121     3500 CONTINUE         UPD06060
122     DO 3511 I=1,10        UPD06070
123     DO 3521 IRE=1,8       UPD06080
124     IF(INAME(IRE,I),NE,1H ) GO T03529    UPD06090
125     3521 CONTINUE         UPD06100
126     GO T03210            UPD06110
127     3529 CONTINUE         UPD06120
128     DO 3531 M=1,100       UPD06130
129     IF(MTYP(M),NE,1)      GO T03531    UPD06140
130     IF(MYAK(M),NE,0)      GO T03531    UPD06150
131     DO 3540 IRE=1,8       UPD06160
132     IF(MNAME(IRE,M),NE,I NAME(IRE,1)) GO T03531    UPD06170
133     3540 CONTINUE         UPD06180
134     MRE = MREC(M)         UPD06190
135     MND = 0               UPD06200
136     LSEQ = 0               UPD06210
137     3551 CONTINUE         UPD06220
138     CALL READT (MRE ,MND ) > UPD06230
139     IF(KDUM,NE,0)          GO T03552    UPD06240
140     LSEQ = LSEQ+1          UPD06250
141     KSEQ = LSEQ            UPD06260
142     CALL COPYH (KNAME ,1 ,KDENT ,1 ,9 ) > UPD06270
143     CALL WRITET (MRE ,MND ) > UPD06280
144     3552 CONTINUE         UPD06290
145     MRE = KREC            UPD06300
146     MND = KIND             UPD06310
147     IF((MRE+MND),EQ,M)   GO T03551    UPD06320
148     GO T03551            UPD06330
149     3531 CONTINUE         UPD06340
150     STOP 3530            UPD06350
151     3511 CONTINUE         UPD06360
152     GO T03210            UPD06370
153     2800 CONTINUE         UPD06380
154     3400 CONTINUE         UPD06390
155     IF(UPDAT,EQ,1)        GO T05002    UPD06400
156     WRITE(OUTPUT,9028)

```

```

157      CALL      RESETH    (LDENT   ,90   ,6H   )          UPD06440
158      CALL      RESETI    (LSEQUU  ,90   ,0     )          UPD06450
159      LNUM    = 1                                UPD06460
160 5001 CONTINUE
161      DO 5000 M=2,100
162      IF(LNUM(M),NE,LNUM) GO TO5000
163      LNUM = LNUM+1
164      IF(HTYP(M),NE,1)   GO TO5000
165      IF(MYAK(M),NE,0)   GO TO5000
166      IF(MCOM(M),NE,1)   GO TO5000
167      L     = 1
168      KRE    = MREC(M)
169      KND    = 0
170 5050 CONTINUE
171      CALL      READT    (KRE     ,KND   )          UPD06470
172      LREC(L)= KREC
173      LIND(L)= KIND
174      IF(KDUM,EQ,0)   GO TO4055
175      WRITE(OUTPUT,9029)
176      KDUM   = 0
177      IF(KYAK,NE,0)   GO TO4052
178      DO 5051 I=1,10
179      DO 5052 IRE=1,8
180      IF(LDENT(IRE,1),NE,KDENT(IRE))GO TO5051
181 5052 CONTINUE
182      GO TO5054
183 5051 CONTINUE
184      DO 5053 I=1,10
185      IF(LSEQU(1,I),EQ,0) GO TO5054
186 5053 CONTINUE
187      STOP 5053
188 5054 CONTINUE
189      CALL      COPYH    (KDENT   ,1     ,LDENT(1,1),1   ,8 )  )UPD06760
190      LSEQU(1,1)=LSEQU(1,1)+1
191      KSE0   = LSEQU(1,1)
192      KDUM   = 0
193      IF(KYAK,NE,0)   GO TO4052
194      WRITE(OUTPUT,9030) (KNAME(KB),KB=1,B),
195      1           (KTAB(KB),KB=1,B),
196      2           (KDENT(KB),KB=1,B),KSE0
197      4052 CONTINUE
198      IF(KYAK,NE,1)   GO TO4053
199      KYAK   = 1
200      WRITE(OUTPUT,4031) (KNAME(KB),KB=1,B),
201      1           (KTAB(KB),KB=1,B),
202      2           (KDENT(KB),KB=1,B),KSE0
203      4053 CONTINUE
204      IF(KYAK,NE,2)   GO TO4055
205      KYAK   = 0
206      WRITE(OUTPUT,9032) (KNAME(KB),KB=1,B),
207      1           (KTAB(KB),KB=1,B),
208      2           (KDENT(KB),KB=1,B),KSE0
209      4055 CONTINUE
210      IF(KYAK,GT,0)   GO TO5200
211      IF(KTYP,NE,23)  GO TO5010
212 5010 CONTINUE
213      IF(HTYP,NE,24)  GO TO5100
214      CALL      CHECKS   (KTAB   ,INAME  ,+IDENT  ,LSEQU  ,+ITYP )  )UPD07070
215      DO 5020 MM=2,100
216      IF(MTYP(MM),NE,2) GO TO5020
217      DO 5030 IRE=1,8
218      IF(MNAME(IRE,MM),NE,INAME(IRE,1))  GO TO5020
219 5030 CONTINUE
220      GO TO5040
221 5020 CONTINUE
222      STOP 5020
223 5040 CONTINUE
224      L     = L+1
225      KRE    = MREC(MM)
226      KND    = 0
227      GO TO5050
228 5100 CONTINUE
229 5200 CONTINUE
230      KRE    = LREC(L)
231      KND    = LIND(L)
232      CALL      INDEX    (KRE     ,KND   ,KRE1   ,KND1   )  )UPD07240
233      IF(KRE1,GT,100) GO TO5050
234      L     = L-1
235      IF(L,LE,0)   GO TO5000
236      GO TO5200
237 5000 CONTINUE
238      IF(LNUM,LE,LLNUM) GO TO5001
239 5002 CONTINUE
240      WRITE(OUTPUT,9025)
241      WRITE(OUTPUT,9020)
242      I     = 0
243      CALL      RESETH   (INAME   ,90   ,6H   )          UPD07350
244      DO 3510 M=1,100
245      IF(MTYP(M),EQ,1)   GO TO3520
246      IF(MTYP(M),EQ,2)   GO TO3520
247      IF(MTYP(M),EQ,3)   GO TO3520
248      GO TO3525
249 3520 CONTINUE
250      I     = I+1
251      CALL      COPYH    (CMNAME(1,M),1   ,INAME(1,1),1   ,9 )  )UPD07440
252 3525 CONTINUE
253      IF(M,LE,100)   GO TO3530
254      IF(L,LT,10)   GO TO3510
255 3530 CONTINUE
256      IF(I,LE,0)   GO TO3510
257      WRITE(OUTPUT,9024) ((INAME(1,1),18=1,B),I10=1,I)
258      I     = 0
259      CALL      RESETH   (INAME   ,90   ,6H   )          UPD07520
260 3510 CONTINUE
261      WRITE(OUTPUT,9025)
262      WRITE(OUTPUT,9021)
263      I     = 0
264      CALL      RESETH   (INAME   ,90   ,6H   )          UPD07570

```

```

265      LNUM = 0                                     UPD07580
266  3601 CONTINUE
267      DO 3610 M=1,100                            UPD07590
268      IF(MNUM(M),NE,LNUM) GO TO3625             UPD07600
269      LNUM = LNUM+1                               UPD07610
270      IF(MTYP(M),EQ,-1) GO TO3620              UPD07620
271      IF(MTYP(M),EQ,13) GO TO3620              UPD07630
272      IF(MTYP(M),EQ,23) GO TO3620              UPD07640
273      GO TO3625                                 UPD07650
274  3620 CONTINUE
275      I = I+1
276      CALL COPYH  (MDENT(1,M),1, +INAME(1,I)+1, +9)  UPD07660
277  3625 CONTINUE
278      IF(LNUM,GT,LLNUM) GO TO3630              UPD07700
279      IF(I,LT,10) GO TO3610                   UPD07710
280  3630 CONTINUE
281      IF(I,EQ,0) GO TO3610                   UPD07720
282      WRITE (OUTPUT,9024) ((INAME(1,I),10),18=1,8),10=1,I  UPD07730
283      I = 0
284      CALL RESETH (INAME ,90 ,+6H )
285  3610 CONTINUE
286      IF(LNUM,LE,LLNUM) GO TO3601              UPD07740
287      WRITE (OUTPUT,9025)
288      WRITE (OUTPUT,9022)
289      I = 0
290      CALL RESETH (INAME ,90 ,+6H )
291      DO 3710 M=1,100                            UPD07750
292      IF(MTYP(M),EQ,2) GO TO3720              UPD07760
293      GO TO3725                                 UPD07770
294  3720 CONTINUE
295      I = I+1
296      CALL COPYH  (MNAME(1,M),1, +INAME(1,I)+1, +9)  UPD07780
297  3725 CONTINUE
298      IF(M,EQ,100) GO TO3730                  UPD07790
299      IF(I,LT,10) GO TO3710                  UPD07910
300  3730 CONTINUE
301      IF(I,EQ,0) GO TO3710                  UPD07920
302      WRITE (OUTPUT,9024) ((INAME(1,I),10),18=1,8),10=1,I  UPD07930
303      I = 0
304      CALL RESETH (INAME ,90 ,+6H )
305  3710 CONTINUE
306      WRITE (OUTPUT,9025)
307      WRITE (OUTPUT,9023)
308      I = 0
309      CALL RESETH (INAME ,90 ,+6H )
310  3801 CONTINUE
311      DO 3810 M=1,100                            UPD08000
312      IF(MNUM(M),NE,LNUM) GO TO3825             UPD08010
313      LNUM = LNUM+1                               UPD08020
314      IF(MTYP(M),NE,1) GO TO3825              UPD08030
315      IF(MYAK(M),NE,0) GO TO3825              UPD08040
316      IF(MCOM(M),NE,1) GO TO3825              UPD08050
317      I = I+1
318      CALL COPYH  (MNAME(1,M),1, +INAME(1,I)+1, +9)  UPD08060
319  3825 CONTINUE
320      IF(LNUM,GT,LLNUM) GO TO3830              UPD08070
321      IF(I,LT,10) GO TO3810                  UPD08080
322  3830 CONTINUE
323      IF(I,EQ,0) GO TO3810                  UPD08090
324      WRITE (OUTPUT,9024) ((INAME(1,I),10),18=1,8),10=1,I  UPD08100
325      I = 0
326      CALL RESETH (INAME ,90 ,+6H )
327  3810 CONTINUE
328      IF(LNUM,LE,LLNUM) GO TO3801              UPD08110
329      WRITE (OUTPUT,9025)
330      IF(COMPIL,NE,0) REWIND COMPIL           UPD08120
331      IF(SOURCE,NE,0) REWIND SOURCE          UPD08130
332      LNUM = 1
333  4001 CONTINUE
334      DO 4000 M=2,100                            UPD08140
335      IF(MNUM(M),NE,LNUM) GO TO4000             UPD08150
336      LNUM = LNUM+1                               UPD08160
337      IF(MTYP(M),NE,1) GO TO4300              UPD08170
338      IF(MYAK(M),NE,0) GO TO4000              UPD08180
339      L = 1
340      KRE = MREC(M)
341      KND = 0
342  4050 CONTINUE
343      CALL READT (KRE ,KND )
344      LREC(L)= KREC
345      LIND(L)= KIND
346      IF(KYAK,GT,0) GOTO 4200                UPD08190
347      IF(KTYP,EQ,23) GOTO 4060                UPD08200
348      IF(SOURCE,NE,0)                         UPD08210
349      1 WRITE (SOURCE,8000) ((KTAB(K80),K80=1,80)          UPD08220
350  4060 CONTINUE
351      IF(MCOM(M),NE,1) GO TO4200              UPD08230
352      IF(KTYP,NE,23) GO TO4010              UPD08240
353      WRITE (OUTPUT,9027) (KNAME(K8),K8=1,8),
354      1 (KTAB(K80),K80=1,80),
355      2 (KDENT(K8),K8=1,8),KSE0            UPD08250
356      ENDFILE COMPIL
357      GO TO4200
358  4010 IF(KTYP,NE,24) GO TO4100              UPD08260
359      CALL CHECKS (KTAB ,INAME ,+IDENT ,+ISEQU ,+ITYP)  UPD08270
360      DO 4020 MM=2,100
361      IF(MTYP(MM),NE,23) GO TO4020             UPD08280
362      DO 4030 IRE=1,8
363      IF(MNAME(IRE,MM),NE,INAME(IRE,1)) GO TO4020   UPD08290
364  4030 CONTINUE
365      GO TO4040
366  4020 CONTINUE
367  4040 STOP 4020
368  4040 CONTINUE
369      1 (KTAB(K80),K80=1,80),
370      2 (KDENT(K8),K8=1,8),KSE0            UPD08300
371      L = L+1
372      KRE = MREC(M)
373      KND = 0
374      GO TO4050
375  4100 CONTINUE
376      IF(KTYP,EW,0) GO TO4101              UPD08310

```

```

374      IF(KTYP.EQ.0) GO TO4101          UPD08720
375      IF(KTAB(1).EQ.MASTER) GO TO4200          UPD08730
376      4101 CONTINUE                      UPD08740
377      IF(D) GO TO4110                      UPD08750
378      KLEN = 0                           UPD08760
379      CALL GETNUM  (KSEQ ,KLEN  ,KTAB(73))          UPD08770
380      CALL COPYH   (KDENT ,1    ,KTAB ,73 ,B-KLEN)          UPD08780
381      CALL GETNUM  (KSEQ ,+KLEN  ,KTAB(81-KLEN))          UPD08790
382      4110,CONTINUE                      UPD08800
383      IF(COMPILE.NE.0)                   UPD08810
384      1 WRITE (COMPILE,8000) (KTAB(KBU),KBU=1,80)          UPD08820
385      4200 CONTINUE                      UPD08830
386      KRE = LREC(L)                      UPD08840
387      KND = LIND(L)                      UPD08850
388      CALL INDEX  (KRE ,KND  ,KRE1 ,KND1 )          UPD08860
389      IF(KRE1.GT.100) GO TO4050          UPD08870
390      L = L-1                           UPD08880
391      IF(L,EQ.0) GO TO4000          UPD08890
392      4300 CONTINUE                      UPD08910
393      IF(MTYP(M).NE.2) GOTO 4000          UPD08920
394      IF(NYAK(M).NE.0) GOTO 4000          UPD08930
395      KRE = MREC(M)                      UPD08940
396      KND = 0                           UPD08950
397      4310 CONTINUE                      UPD08960
398      CALL READT  (KRE ,KND )          UPD08970
399      IF(SOURCE.NE.0)                   UPD08980
400      1 WRITE(SOURCE,8000) (KTAB(KBD)+KBD=1,80)          UPD08990
401      KRE = KREC                      UPD09000
402      KND = KND                      UPD09010
403      IF((KRE<KND),EQ,M) GOTO 4000          UPD09020
404      4000 CONTINUE                      UPD09040
405      IF(LLNUM.LE.LLNUM) GO TO4001          UPD09050
406      IF(COMPILE.NE.0) ENDFILE COMPILE          UPD09060
407      IF(COMPILE.NE.0) REWIND COMPILE          UPD09070
408      IF(SOURCE.NE.0) ENDFILE SOURCE          UPD09080
409      IF(SOURCE.NE.0) REWIND SOURCE          UPD09090
410      IUPDAT = 3                         UPD09100
411      RETURN                            UPD09110
412      9010 FORMAT('***** ',80A1)          UPD09120
413      9020 FORMAT('1           CORRECTION IDENTS ARE LISTED IN CHRONOL')          UPD09130
414      1           'LOGICAL ORDER OF INSERTION'          UPD09140
415      9021 FORMAT('1           DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A')          UPD09150
416      1           'NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE'          UPD09160
417      9022 FORMAT('1           COMMON DECKS ENCOUNTERED'          UPD09170
418      9023 FORMAT('1           DECKS WRITTEN TO COMPILE FILE'          UPD09180
419      9024 FORMAT(10X,10(8A1,2X))          UPD09190
420      9025 FORMAT(//)                      UPD09200
421      9026 FORMAT(20X,8A1)                UPD09210
422      9027 FORMAT(1X,8A1,1X,8A1,2X,8A1,2X,18)          UPD09220
423      9028 FORMAT(//,9X,' MODIFICATIONS / CONTROL CARDS ')          UPD09230
424      9029 FORMAT(//)                      UPD09240
425      9030 FORMAT(1X,8A1,1X,8A1,2X,8A1,2X,18,4H I )          UPD09250
426      9031 FORMAT(1X,8A1,1X,8A1,2X,8A1,2X,18,4H D )          UPD09260
427      9032 FORMAT(1X,8A1,1X,8A1,2X,8A1,2X,18,4H R )          UPD09270
428      8000 FORMAT(80A1)                UPD09280
429      9000 FORMAT(2X,8A1,2X,8A1,2X,8A1)          UPD09290
430      END                               UPD09300

*DECK COPYH
1      SUBROUTINE COPYH  (HOL1 ,IH01 ,HOL2 ,IH02 ,IHOL )          UPD09310
2      DIMENSION HOL1 ( 1 ),HOL2 ( 1 ),IHOL ( 1 )          UPD09320
3      INTEGER HOL1 ,HOL2 ,IHOL ,IH01 ,IH02
4      IH01 = IHOL
5      IH02 = IHOL
6      DO 1000 IH01=HOL1,IH02
7      HOL2(12)=HOL1(IH01)
8      IH01 = IH01+1
9      IH02 = IH02+1
10     1000 CONTINUE
11     RETURN
12     END

*DECK COPYI
1      SUBROUTINE COPYI  (INT1 ,IN1 ,INT2 ,IN2 ,INT )          UPD09440
2      DIMENSION INT1 ( 1 ),IN1 ( 1 ),INT2 ( 1 ),IN2 ( 1 ),INT ( 1 )          UPD09450
3      INTEGER INT1 ,IN1 ,INT2 ,IN2 ,INT ,INT1 ,INT2
4      IN1 = INT1
5      IN2 = INT2
6      DO 1000 INT1=INT1,INT2
7      INT2(12)=INT1(INT1)
8      INT1 = INT1+1
9      INT2 = INT2+1
10     1000 CONTINUE
11     RETURN
12     END

```

```

1      *DECK CORREC          UPD09570
2      SUBROUTINE CORREC      UPD09580
3      COMMENT UPDATE CORRECTION RUN      UPD09590
4      COMMON INPUT OUTPUT OLDPL NEWPL COMPIL SOURCE ,UPD09600
5      1      IO  JCOMP JTEXT IXOLD IXNEW IX ,UPD09610
6      2      MASTER F D  FILE NUM IUPDAT ,UPD09620
7      INTEGER INPUT OUTPUT OLDPL NEWPL COMPIL SOURCE UPD09630
8      LOGICAL F D ,UPD09640
9      COMMON/TABK/KTAB ( 90),KNAME ( 9),KDENT ( 9),UPD09650
10     1      KSE9U ( 9) ,UPD09660
11     COMMON/TABT/KSE9 KREC KIND KNUM KTYP KYAK ,UPD09670
12     1      KCOM KDUM ,UPD09680
13     COMMON/TABM/MTAB ,MNAME ( 9,100),MDENT ( 9,100),UPD09690
14     1      MSE9 ( 100),MREC ( 100),MIND ( 100),UPD09700
15     1      MNUM ( 100),MTYP ( 100),MYAK ( 100),UPD09710
16     1      MCOM ( 100),MDUM ( 100),UPD09720
17     DIMENSION ITAB ( 90),INAME ( 9, 10),IDENT ( 9, 10),UPD09730
18     1      ISE9U ( 9, 10) ,UPD09740
19     DIMENSION LTAB ( 90),LNAME ( 9),LDENT ( 9),UPD09750
20     1      LSE9U ( 9) ,UPD09760
21     DIMENSION TODAY(2) ,UPD09770
22     COMMENT SET INITIAL TABLE PARAMETOR      UPD09780
23     IEOF = 0 ,UPD09790
24     LNUM = 0 ,UPD09800
25     LTYP = 0 ,UPD09810
26     MRE = 0 ,UPD09820
27     IITYP = 28 ,UPD09830
28     MMREC = 0 ,UPD09840
29     MMIND = 0 ,UPD09850
30     IADD = -1 ,UPD09860
31     CALL DATE(TODAY) ,UPD09870
32     WRITE(OUTPUT,9000) TODAY ,UPD09880
33     COMMENT CORRECTION TEXT PRINT      UPD09890
34     REWIND ITEXT ,UPD09900
35     2900 CONTINUE      UPD09910
36     READ (ITEXT,8000,END=2801) (ITAB(180),180=1,80) ,UPD09920
37     IF(ITAB(1),NE,MASTER) GO TO2901 ,UPD09930
38     WRITE (OUTPUT,9010) (ITAB(180),180=1,80) ,UPD09940
39     IF(IADD,EQ,0) GO TO2900 ,UPD09950
40     CALL CHECKS (ITAB ,INAME ,IDENT ,ISE9U ,ITYP) ,UPD09960
41     IF(ITYP,EQ,16) IADD=0 ,UPD09970
42     GO TO2900 ,UPD09980
43     2901 CONTINUE      UPD09990
44     IF(IADD,EQ,0) GO TO2900 ,UPD10000
45     WRITE (OUTPUT,9011) (ITAB(180),180=1,80) ,UPD10010
46     GO TO2900 ,UPD10020
47     2801 CONTINUE      UPD10030
48     IADD = 0 ,UPD10040
49     IDENT = 0 ,UPD10050
50     ICOP = 0 ,UPD10060
51     WRITE(OUTPUT,9012) ,UPD10070
52     REWIND ITEXT ,UPD10080
53     DO 4000 M=2,100 ,UPD10090
54     IF(LNUM,LE,MNUM(M)) LNUM=MNUM(M) ,UPD10100
55     4000 CONTINUE      UPD10110
56     COMMENT READ ITAB FROM ITEXT      UPD10120
57     3000 CONTINUE      UPD10130
58     READ (ITEXT,8000,END=2800) (ITAB(180),180=1,80) ,UPD10140
59     COMMENT CHECK UPDATE CONTROL CHARACTER      UPD10150
60     CALL CHECKS (ITAB ,INAME ,IDENT ,ISE9U ,ITYP) ,UPD10160
61     IF(ITYP,EQ,0) GO TO3500 ,UPD10170
62     GO TO ( 100, 200, 300, 400, 500, 600, 700, 800, 900,1000,
63     1           1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,
64     2           2100,2200,2300,2400,2500,2600,2700),ITYP ,UPD10180
65     2           1800,1900,2000,2100,2200,2300,2400,2500,2600,2700,ITYP ,UPD10190
66     2           2100,2200,2300,2400,2500,2600,2700,ITYP ,UPD10200
67     COMMENT DECK      UPD10210
68     100 CONTINUE      UPD10220
69     COMMENT COMDECK      UPD10230
70     200 CONTINUE      UPD10240
71     IF(IDEN,EQ,0,AND,IADD,EQ,0) STOP ,UPD10250
72     IF(IDEN,NE,0,AND,IADD,EQ,0) GO TO2400 ,UPD10260
73     IF(IDEN,EQ,0,AND,IADD,EQ,2) GO TO3630 ,UPD10270
74     IADD = 2 ,UPD10280
75     GO TO309 ,UPD10290
76     COMMENT IDENT      UPD10300
77     300 CONTINUE      UPD10310
78     IDENT = 1 ,UPD10320
79     GO TO2808 ,UPD10330
80     308 IITYP = ITYP ,UPD10340
81     MMREC = 0 ,UPD10350
82     MMIND = 0 ,UPD10360
83     IDENT = 2 ,UPD10370
84     IADD = 0 ,UPD10380
85     309 CONTINUE      UPD10390
86     IF(ITYP,EQ,3)      UPD10400
87     1CALL COPYH (IDENT +1 ,+INAME +1 ,+9 ) ,UPD10410
88     DO 301 N=1,100 ,UPD10420
89     IF(MNUM(N),EQ,0) GO TO301 ,UPD10430
90     DO 302 IRE=1,8 ,UPD10440
91     IF(MDENT(IRE,N),NE,(INAME(IRE,1))) GO TO301 ,UPD10450
92     302 CONTINUE      UPD10460
93     CALL ERRORC(INAME,4) ,UPD10470
94     301 CONTINUE      UPD10480
95     COMMENT KEEP SUB TABLE PARAMETORS      UPD10490
96     3001 CONTINUE      UPD10500
97     CALL COPYH (INAME :1 ,+LNAME :1 ,+9 ) ,UPD10510
98     CALL COPYH (INAME :1 ,+LDENT :1 ,+9 ) ,UPD10520
99     LSE9 = 0 ,UPD10530
100    LREC = 0 ,UPD10540
101    LIND = 0 ,UPD10550
102    LNUM = LNUM+1 ,UPD10560
103    LTYP = ITYP ,UPD10570
104    LYAK = 0 ,UPD10580
105    LCOM = 1 ,UPD10590
106    LDUM = 0 ,UPD10600
107    IF(ITYP,EQ,3) LDUM=1 ,UPD10610
108    COMMENT RESET SUB TABLE PARAMETORS      UPD10620
109    3100 CONTINUE      UPD10630
110    IF(MRE,EQ,0) GO TO3300 ,UPD10640
111    KIND = M-KRE ,UPD10650
112    KND = KND-1 ,UPD10660
113    GO TO3600 ,UPD10670
114    COMMENT SET MAIN TABLE PARAMETORS      UPD10680
115    3200 CONTINUE      UPD10690
116    CALL COPYH (KNAME :1 ,+MNAME(1,M):1 ,+9 ) ,UPD10700

```

```

91      CALL      COPYH     CKNAME    +1      ,MDENT(1,M),1    ,9   ) UPD10710
92      MSE(M)= 0
93      MRECK(M)= KRE
94      MIND(M)= KRE+KND
95      MNUM(M)= LLNUM
96      HTYP(M)= LLTYP
97      HYAK(M)= LLYAK
98      MCOM(M)= LLCOM
99      MDUM(M)= LDUM
100     COMMENT RESET MAIN TABLE PARAMETER
101     DO 3210 M=2,100
102     IF(MIND(M).NE.0) GO TO3210
103     MRECK(M)= KRE+KND+1
104     MRE = 0
105     IF(IEOF,EQ.1) GOTO 7000
106     COMMENT SEARCH MAIN TABLE
107     3300 CONTINUE
108     IF(IDEN.EQ.1) GO TO 308
109     IF(ADD,EQ.1) GO TO5608
110     LLNUM = LNUM
111     LLTYP = LTYP
112     LLYAK = LYAK
113     LLCOM = LCOM
114     LDUM = LDUM
115     KRE = 0
116     KND = 0
117     DO 3310 M=2,100
118     IF(MRECK(M).LE.KRE) GO TO3310
119     KRE = MRECK(M)
120     MRE = M
121     3310 CONTINUE
122     M = MRL
123     KSEQ = LSEQ
124     KREC = KRE
125     KIND = LIND
126     KNUM = LNUM
127     KTYP = LTYP
128     KYAK = LYAK
129     KCOM = LCOM
130     KDUM = LDUM
131     IF(ITYP,EQ.3) GO TO3000
132     3400 CONTINUE
133     COMMENT SET SUB TABLE PARAMETER
134     3500 CONTINUE
135     ITYP = ITYP
136     XTYP = LTYP
137     3510 CONTINUE
138     KSEQ = LSEQ
139     KREC = KRL
140     KIND = LIND
141     KNUM = LNUM
142     KTYP = LYAK
143     KYAK = LYAK
144     KCOM = LCOM
145     KDUM = LDUM
146     CALL      COPYH     CITAB    +1      ,KTAB    +1      ,80    ) UPD11260
147     CALL      COPYH     CLNAME   +1      ,KNAME   +1      ,9     ) UPD11270
148     CALL      COPYH     CKDENT   +1      ,KDENT   +1      ,9     ) UPD11280
149     KSEQ = KSEQ+1
150     LIND = LIND+1
151     KNUM = KNU4
152     KTYP = KTYP
153     KYAK = KYAK
154     KCOM = KCOM
155     KDUM = KDUM
156     COMMENT WRITE SUB TABLE PARAMETER
157     3600 CONTINUE
158     IF(KTYP,EQ.0) GO TO3610
159     IF(KTYP,EQ.23) GO TO3610
160     IF(KTAB(1).NE.MASTER) GO TO3610
161     WRITE(OUTPUT,9010) ((KTAB(K80),K80=1,80),
162     1          (KDENT(K8),K8=1,8),
163     1          KSEQ
164     3610 CONTINUE
165     CALL      WRITET   (KRE    ,KND    )
166     IF(KIND,LT.0) GO TO3200
167     KND = KND+1
168     IF(COP,NE.0) GO TO4842
169     GO TO3000
170     COMMENT INSERT
171     400 CONTINUE
172     COMMENT IDENT,ISEQ
173     IF(ITYP,NE.0) GO TO4401
174     KREC = MMREC
175     KIND = MMIND
176     IF(KREC,EQ.0) GO TO4401
177     KND = KND-1
178     CALL      WRITET   (KRE    ,KND    )
179     KREC = KKREC
180     KIND = KKIND
181     CALL      WRITET   (KRE    ,MND    )
182     MMREC = 0
183     MMIND = 0
184     DO 4410 N=1,100
185     IF(MNUM(N),EQ.0) GO TO4410
186     DO 4411 IRE=1,8
187     IF(MDENT(IRE,N),NE.IDENT(IRE,1)) GO TO4410
188     4411 CONTINUE
189     4412 CONTINUE
190     MRE = MRECN
191     MND = 0
192     CALL      SETNUM   (ISEQ(  1, 12),ISEQ(  9, 12),ISEQ(  1, 12),ISEQ(  9, 12))
193     4420 CONTINUE
194     CALL      READT    (MRE    ,MND    )
195     IF(KSEQ,EQ.ISEQ) GO TO4421
196     MRE = MRE

```

```

197      MND = MND+1          UPD11850
198      IF(MND.LE.MIND(N)) GO TO4420    UPD11860
199      4410 CONTINUE          UPD11870
200      STOP 4410            UPD11880
201      4421 CONTINUE          UPD11890
202      ISEQ = 0              UPD11900
203      4440 CONTINUE          UPD11910
204      DO 4430 N=1,100        UPD11920
205      IF(MNUM(N),EQ,0) GO TO4430    UPD11930
206      DO 4431 IRE=1,8         UPD11940
207      IF(MDENT(IRE,N),NE,KNAME(IRE)) GO TO4430    UPD11950
208      4431 CONTINUE          UPD11960
209      HCOM(N)= 1             UPD11970
210      GO TO4432            UPD11980
211      4430 CONTINUE          UPD11990
212      STOP 4430            UPD12000
213      4432 CONTINUE          UPD12010
214      CALL  COPYH   (KNAME +1     +LNAME +1   ,9  ) >UPD12020
215      MMREC = KREC           UPD12030
216      MMIND = KIND           UPD12040
217      KKREC = KRE            UPD12050
218      KKIND = KND            UPD12060
219      GO TO3000              UPD12070
220      COMMENT BEFORE        UPD12080
221      500 CONTINUE          UPD12090
222      STOP 500              UPD12100
223      COMMENT DELETE        UPD12110
224      600 CONTINUE          UPD12120
225      COMMENT IDENT,ISEQ    UPD12130
226      IF(IITYP,NE,0) GO TO4601    UPD12140
227      KREC = MMREC           UPD12150
228      KIND = MMIND           UPD12160
229      IF(KREC,EQ,0) GO TO4601    UPD12170
230      KND = KND-1            UPD12180
231      CALL  WRITET  (KRE   ,KND  ) >UPD12190
232      KREC = KKREC           UPD12200
233      KIND = KKIND           UPD12210
234      CALL  WRITET  (MRE   ,MND  ) >UPD12220
235      4601 CONTINUE          UPD12230
236      IITYP = ITYP           UPD12240
237      MMREC = 0               UPD12250
238      MMIND = 0               UPD12260
239      DO 4610 N=1,100        UPD12270
240      IF(MNUM(N),EQ,0) GO TO4610    UPD12280
241      DO 4611 IRE=1,8         UPD12290
242      IF(MDENT(IRE,N),NE,IDENT(IRE+1)) GO TO4610    UPD12300
243      4611 CONTINUE          UPD12310
244      4612 CONTINUE          UPD12320
245      MRE = MREC(N)           UPD12330
246      MND = 0               UPD12340
247      4620 CONTINUE          UPD12350
248      CALL  SETNUM  (ISEQU ( 1, 1)+ISEQU ( 9, 1)+ISEQ ) >UPD12360
249      CALL  READT   (MRE   ,MND  ) >UPD12370
250      IF(KSEQ,EQ,ISEQ) GO TO4621    *UPD12380
251      MRE = MRE              UPD12390
252      MND = MND+1            UPD12400
253      4610 CONTINUE          UPD12410
254      STOP 4610              UPD12420
255      4621 CONTINUE          UPD12430
256      CALL  SETNUM  (ISEQU ( 1, 2),ISEQU ( 9, 2)+ISEQ ) >UPD12440
257      4640 CONTINUE          UPD12450
258      DO 4630 N=1,100        UPD12460
259      IF(MNUM(N),EQ,0) GO TO4630    UPD12470
260      DO 4631 IRE=1,8         UPD12480
261      IF(MDENT(IRE,N),NE,KNAME(IRE)) GO TO4630    UPD12490
262      4631 CONTINUE          UPD12500
263      HCOM(N)= 1             UPD12510
264      GO TO4632            UPD12520
265      4630 CONTINUE          UPD12530
266      STOP 4630              UPD12540
267      4632 CONTINUE          UPD12550
268      CALL  COPYH   (KNAME +1     +LNAME +1   ,9  ) >UPD12560
269      MMREC = KREC           UPD12570
270      MMIND = KIND           UPD12580
271      KKREC = KRE            UPD12590
272      KKIND = KND            UPD12600
273      KYAK = 1               UPD12610
274      KDUM = 1               UPD12620
275      CALL  WRITET  (MRE   ,MND  ) >UPD12630
276      IF(ISEQ,LE,0) GO TO3000    UPD12640
277      MRE = MREC              UPD12650
278      MND = MMIND             UPD12660
279      CALL  READT   (MRE   ,MND  ) >UPD12670
280      DO 4641 IRE=1,8         UPD12680
281      IF(KDENT(IRE),NE,IDENT(IRE+2)) GO TO4640    UPD12690
282      4641 CONTINUE          UPD12700
283      IF(KSEQ,NE,ISEQ) GO TO4640    UPD12710
284      ISEQ = 0               UPD12720
285      GO TO4640              UPD12730
286      COMMENT RESTORE        UPD12740
287      700 CONTINUE          UPD12750
288      IF(IITYP,NE,0) GO TO4701    UPD12760
289      KREC = MMREC           UPD12770
290      KIND = MMIND           UPD12780
291      IF(KREC,EQ,0) GO TO4701    UPD12790
292      KND = KND-1            UPD12800
293      CALL  WRITET  (KRE   ,KND  ) >UPD12810
294      CALL  READT   (MRE   ,MND  ) >UPD12820
295      KREC = KKREC           UPD12830
296      KIND = KKIND           UPD12840
297      CALL  WRITET  (MRE   ,MND  ) >UPD12850
298      4701 CONTINUE          UPD12860
299      IITYP = ITYP           UPD12870
300      MMREC = 0               UPD12880
301      MMIND = 0               UPD12890
302      DO 4710 N=1,100        UPD12900
303      IF(MNUM(N),EQ,0) GO TO4710    UPD12910
304      DO 4711 IRE=1,8         UPD12920
305      IF(MDENT(IRE,N),NE,IDENT(IRE,1)) GO TO4710    UPD12930
306      4711 CONTINUE          UPD12940

```

```

307   4712 CONTINUE
308     MRE = MREC(N)
309     MND = 0
310     CALL SETNUM ((ISEQU ( 1, 1)+ISEQU ( 9, 1)+ISEQ
311   4720 CONTINUE
312     CALL READT (MRE +MND )
313     IF(KSEQ,E0,ISEQ) GO TO4721
314     MRE = MRE
315     MND = MND+1
316     IF(MND.LE.MIND(N)) GO TO4720
317   4710 CONTINUE
318     STOP 4710
319   4721 CONTINUE
320     CALL SETNUM ((ISEQU ( 1, 2)+ISEQU ( 9, 2)+ISEQ
321   4740 CONTINUE
322     DO 4730 N=1,100
323     IF(MNUM(N),E0,0) GO TO4730
324     DO 4731 IRE=1,8
325     IF(KDENT(IRE,N).NE.KNAME(IRE))GO TO4730
326   4731 CONTINUE
327     MCOM(N)= 1
328     GO TO4732
329   4730 CONTINUE
330     STOP 4730
331   4732 CONTINUE
332     CALL COPYH (KNAME +1      +LNAME +1      +9
333     MMREC = KREC
334     MMIND = KIND
335     KKREC = KRE
336     KKIND = KND
337     KYAK = 2
338     KDUM = 1
339     CALL WRITET (MRE +MND )
340     IF((ISEQ,LE,0) GO TO3000
341     MRE = MMREC
342     MND = MMIND
343     CALL READT (MRE +MND )
344     DO 4741 IRE=1,8
345     IF(KDENT(IRE).NE.(DENT(IRE,2))GO TO4740
346   4741 CONTINUE
347     IF(KSEQ,NE,ISEQ) GO TO4740
348     ISEQ = 0
349     GO TO4740
COMMENT COPY
350   800 CONTINUE
351   4801 CONTINUE
352     ICOP = 1
353     DO 4810 N=1,100
354     IF(MNUM(N),E0,0) GO TO4810
355     DO 4811 IRE=1,8
356     IF(KDENT(IRE,N).NE.INAME(IRE,1)) GO TD4810
357   4811 CONTINUE
358   4812 CONTINUE
359     LRE = MREC(N)
360     LND = 0
361     CALL SETNUM ((ISEQU ( 1, 1)+ISEQU ( 9, 1)+ISEQ
362   4820 CONTINUE
363     CALL READT (LRE +LND )
364     DO 4813 IRE=1,8
365     IF(KDENT(IRE).NE.(DENT(IRE,1))GO TO4814
366   4813 CONTINUE
367     IF(KSEQ,E0,ISEQ) GO TO4821
368   4814 LRE = KREC
369     LND = KIND
370     IF(LRE+LND).NE.N) GO TO4820
371   4810 CONTINUE
372     STOP 4810
373   4821 CONTINUE
374     CALL SETNUM ((ISEQU ( 1, 2)+ISEQU ( 9, 2)+ISEQ
375   4840 CONTINUE
376     LRE = KREC
377     LND = KIND
378     ITYP = 0
379     CALL COPYH (KTAB +1      +ITAB +1      +80
380     GO TO3500
381   4842 CONTINUE
382     IF((ISEQ,LE,0) GO TO3000
383     CALL READT (LRE +LND )
384     DO 4841 IRE=1,8
385     IF(KDENT(IRE).NE.(DENT(IRE,2))GO TO4840
386   4841 CONTINUE
387     IF(KSEQ,NE,ISEQ) GO TO4840
388     ISEQ = 0
389     ICOP = 0
390     GO TO4840
391   900 CONTINUE
392   1000 CONTINUE
393   1200 CONTINUE
394   1100 CONTINUE
395   1300 CONTINUE
396   1400 CONTINUE
397   1500 CONTINUE
398     STOP 1500
COMMENT ADDFILE
399   1600 CONTINUE
400     IADD = 1
401     GO TO2808
402   5608 ITYP = ITYP
403     MMREC = 0
404     MMIND = 0
405     IDEN = 0
406     IADD = 1
407   5609 CONTINUE
408     DO 5610 N=1,100
409     IF(MNUM(N),E0,0) GO TO5610
410     DO 5611 IRE=1,8
411     IF(KNAME(IRE,N).NE.INAME(IRE,1)) GO TO5610
412   5611 CONTINUE
413     LNUM = MNUM(N)
414     GO TO5612
415   5610 CONTINUE
416     STOP 5610
417   5612 CONTINUE
418     DO 5620 N=1,100

```

```

419      IF(MNUM(N).GT.LNUM) MNUM(N)=MNUM(N)+1          UPD14130
420      5620 CONTINUE                                     UPD14140
421      IF(IFILE,E9,INPUT) GO TO3000                   UPD14150
422      ITEXT = IFILE                                    UPD14160
423      REWIND ITEXT                                    UPD14170
424      GO TO3000                                     UPD14180
425      5630 CONTINUE                                     UPD14190
426      DO 5640 N=1,100                                UPD14200
427      IF(MNUM(N).GT.LNUM) MNUM(N)=MNUM(N)+1          UPD14210
428      5640 CONTINUE                                     UPD14220
429      GO TO 309                                     UPD14230
430      1700 CONTINUE                                     UPD14240
431      1800 CONTINUE                                     UPD14250
432      1900 CONTINUE                                     UPD14260
433      2000 CONTINUE                                     UPD14270
434      2100 CONTINUE                                     UPD14280
435      2200 CONTINUE                                     UPD14290
436      STOP 2200                                     UPD14300
437      COMMENT WEDR                                     UPD14310
438      2300 CONTINUE                                     UPD14320
439      KTyp = ITyp                                    UPD14330
440      ITyp = 0                                       UPD14340
441      GO TO3510                                     UPD14350
442      COMMENT CALL                                     UPD14360
443      2400 CONTINUE                                     UPD14370
444      KTyp = ITyp                                    UPD14380
445      ITyp = 0                                       UPD14390
446      IF(CTYP,E9,24) GO TO3510                   UPD14400
447      COMMENT /                                     UPD14410
448      2500 CONTINUE                                     UPD14420
449      WRITE(OUTPUT,9010) (ITAB(IRO),IRO=1,80)        UPD14430
450      GO TO3000                                     UPD14440
451      2600 CONTINUE                                     UPD14450
452      2700 CONTINUE                                     UPD14460
453      GO TO3000                                     UPD14470
454      2800 CONTINUE                                     UPD14480
455      IEOF = 1                                      UPD14490
456      IF(IDEN,EN,0,AND,IADD,EQ,0) GO TO7000       UPD14500
457      2808 CONTINUE                                     UPD14510
458      IF(CTYP,NE,3) GO TO3100                      UPD14520
459      IF((ITYP,NE,0) GO TO2811                   UPD14530
460      KREC = MMREC                                 UPD14540
461      KIND = MMIND                                UPD14550
462      IF(KREC,EW,0) GO TO2811                   UPD14560
463      KIND = KND-1                                UPD14570
464      CALL WRITET (KRE ,KND )                      UPD14580
465      KIND = KND+1                                UPD14590
466      CALL READT (MRE ,MND )                      UPD14600
467      KREC = KKREC                                UPD14610
468      KIND = KKIND                                UPD14620
469      CALL WRITET (MRE ,MND )                      UPD14630
470      2811 CONTINUE                                     UPD14640
471      IF(MRE,EW,0) GO TO3300                      UPD14650
472      KIND = M-KRE                                UPD14660
473      KIND = KND-1                                UPD14670
474      IF(KND,LT,0) KND=0                           UPD14680
475      CALL COPYH (LDENT ,1 ,XNAME ,1 ,+9)           UPD14690
476      GO TO3200                                     UPD14700
477      7000 CONTINUE                                     UPD14710
478      REWIND ITEXT                                    UPD14720
479      RETURN                                         UPD14730
480      8000 FORMAT(8A1)                                UPD14740
481      9000 FORMAT(1H1,/,1 CORRECTION RUN           UPD14750
482      1 ,20X,2A4/)                                UPD14760
483      9010 FORMAT(' **** ',8A1,2X,8A1,1B)          UPD14770
484      9011 FORMAT(' ',8A1,2X,8A1,1B)                UPD14780
485      9012 FORMAT(//)                                UPD14790
486      END                                           UPD14800
487

```

```

*DECK COTEXT
1   SUBROUTINE COTEXT          UPD14830
2     COMMON   INPUT  >OUTPUT >OLDPL  >NEWPL  >COMPIL  >SOURCE  UPD14840
3       1      IO    >ICOMP  >ITEXT  >IXOLD  >IXNEW  >IX    UPD14850
4       2      MASTER >F     >D     >FILE   >NUM    >UPDAT  UPD14870
5     INTEGER  INPUT  >OUTPUT >OLDPL  >NEWPL  >COMPIL  >SOURCE  UPD14880
6     LOGICAL   F     >D     >FILE   >NUM    >UPDAT  UPD14890
7     DIMENSION ITAB  ( 90) ,NAME ( 9, 10) ,IDENT ( 9, 10) ,ISEQU ( 9, 10) UPD14900
8       1      ISEQU ( 9, 10)
9     IF(INPUT,NE,5)           UPD14910
10    IREWIND  INPUT           UPD14920
11    REWIND   ITEXT           UPD14930
12    REWIND   ICOMP           UPD14940
13    1000 CONTINUE           UPD14950
14    READ (INPUT,8000,END=7000) (ITAB(180),I80=1,80) UPD14960
15    CALL    CHECKS (ITAB,  NAME ,IDENT ,ISEQU ,ITYP) JUPD14980
16    IF(ITYP,EQ,26)          GO T07000 UPD14990
17    IF(ITYP,NE,17)          GO T01010 UPD15000
18    WRITE(ICOMP,8000) (ITAB(180),I80=1,80) UPD15010
19    GO T01000 UPD15020
20    1010 CONTINUE           UPD15030
21    IF(ITYP,NE,19)          GO T01200 UPD15040
22    READ (FILE,8000,END=1000) (ITAB(180),I80=1,80) UPD15050
23    WRITE (ITEXT,8000) (ITAB(180),I80=1,80) JUPD15060
24    CALL    CHECKS (ITAB,  NAME ,IDENT ,ISEQU ,ITYP) JUPD15070
25    IF(UPDAT,NE,0)          GO T01100 UPD15080
26    IF(ITYP,EQ,1)           IUPDAT=1 UPD15090
27    IF(ITYP,EQ,2)           IUPDAT=1 UPD15100
28    IF(ITYP,EQ,3)           IUPDAT=1 UPD15110
29    IF(ITYP,EQ,16)          IUPDAT=2 UPD15120
30    IF(ITYP,EQ,19)          CALL    ERROR1(2) UPD15130
31    IF(ITYP,EQ,20)          CALL    ERROR1(15) UPD15140
32    IF(ITYP,EQ,21)          CALL    ERROR1(15) UPD15150
33    IF(ITYP,NE,20)          GO T01400 UPD15160
34    DO 1300 N=1,NUM          UPD15170
35    READ (FILE,8000) (ITAB(180),I80=1,80) UPD15180
36    1300 CONTINUE           UPD15190
37    GO T01000 UPD15200
38    1400 CONTINUE           UPD15210
39    IF(ITYP,NE,21)          GO T01500 UPD15220
40    REWIND   FILE            UPD15230
41    GO T01000 UPD15240
42    1500 CONTINUE           UPD15250
43    IF(ITYP,NE,22)          GO T01600 UPD15260
44    WRITE (ICOMP,8000) (ITAB(180),I80=1,80) UPD15270
45    GO T01000 UPD15280
46    1600 CONTINUE           UPD15290
47    WRITE (ITEXT,8000) (ITAB(180),I80=1,80) UPD15300
48    IF(UPDAT,NE,0)          GO T01000 UPD15310
49    IF(ITYP,EQ,1)           IUPDAT=1 UPD15320
50    IF(ITYP,EQ,2)           IUPDAT=1 UPD15330
51    IF(ITYP,EQ,3)           IUPDAT=2 UPD15340
52    IF(ITYP,EQ,16)          IUPDAT=2 UPD15350
53    GO T01000 UPD15360
54    7000 CONTINUE           UPD15370
55    IF(UPDAT,EQ,0)          IUPDAT =2 UPD15380
56    ENDFILE  ITEXT           UPD15390
57    REWIND   ITEXT           UPD15400
58    ENDFILE  ICOMP           UPD15410
59    REWIND   ICOMP           UPD15420
60    RETURN             UPD15430
61    8000 FORMAT(80A1)        UPD15440
62    END                   UPD15450
63

```

```

1 *DECK      CREATE          UPD15490
2   SUBROUTINE CREATE          UPD15500
3   COMMENT UPDATE CREATION RUN          UPD15510
4
5   COMMON    INPUT    *OUTPUT  *OLDPL  *NEWPL  *COMPIL  *SOURCE  UPD15520
6   1        ID      *ICOMP  *ITEXT  *IXOLD  *IXNE*  *IX      UPD15530
7   2        MASTER  *F      *D      *IFILE  *NUM    *IUPDAT  UPD15540
8   INTEGER   INPUT    *OUTPUT  *OLDPL  *NEWPL  *COMPIL  *SOURCE  UPD15550
9   LOGICAL   F      *D      *KNAME  *99,KDENT  *99      UPD15560
10  COMMON/TABK/KTAB  ( 90),KNAME  ( 99,KDENT  ( 99,         UPD15570
11  1        KSEQU  ( 99      *KREC  *KIND  *KNUM  *KTYP  *KYAK  UPD15580
12  1        KCOM  *KDUM      *MNAME  ( 9,100),MDENT  ( 9,100)       UPD15590
13  1        MSE@  ( 100),MREC  ( 100),MIND  ( 100)       UPD15600
14  1        MNUM  ( 100),MTYP  ( 100),MYAK  ( 100)       UPD15610
15  1        MCOM  ( 100),MDUM  ( 100)       UPD15620
16  DIMENSION ITAB  ( 90),INAME  ( 9,10),IDENT  ( 9,10)       UPD15630
17  1        ISEQU  ( 9, 10      *LNAME  ( 99,LIDENT  ( 99,         UPD15640
18  1        LTAB  ( 90),LNAME  ( 99,LIDENT  ( 99,         UPD15650
19  1        LSEQU  ( 99      *TODAY(2)          *DATE(TODAY)          UPD15660
20  COMMENT SET INITIAL TABLE PARAMETOR          UPD15670
21  1EOF  = 0          *WRITE(OUTPUT,9000), TODAY          UPD15680
22  LNUM  = 0          UPD15690
23  LTYP  = 0          UPD15700
24  MRE  = 0          UPD15710
25  MMREC  = 0         UPD15720
26  MMIND  = 0         UPD15730
27  CALL    DATE(TODAY)          UPD15740
28  WRITE(OUTPUT,9000), TODAY          UPD15750
29  CALL    DATE(TODAY)          UPD15760
30  COMMENT READ ITAB FROM ITEXT          UPD15770
31  3000 CONTINUE          UPD15780
32  READ  (ITEXT,8000,END=2800)  (ITAB(180),180*1,80)          UPD15790
33  COMMENT CHECK UPDATE CONTROL CHARACTER          UPD15800
34  CALL    CHECKS  (ITAB,INAME  *IDENT  ,ISEQU  *ITYP  )       UPD15810
35  IF(ITYP,EQ,0)  GO TO3500          UPD15820
36  GO TO  ( 100, 200, 300, 400, 500, 600, 700, 800, 900,1000,       UPD15830
37  1        1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,       UPD15840
38  2        2100,2200,2300,2400,2500,2600,2700),ITYP          UPD15850
39  COMMENT DECK          UPD15860
40  100 CONTINUE          UPD15870
41  COMMENT COMDECK          UPD15880
42  200 CONTINUE          UPD15890
43  DO 301 N=1,100          UPD15900
44  IF(EMNUM(N),EQ,0)  GO TO 301          UPD15910
45  DO 302 IRE=1,8          UPD15920
46  IF(MDENTIRE,N),NE,INAME(IRE+1))  GO TO301          UPD15930
47  302 CONTINUE          UPD15940
48  301 CONTINUE          UPD15950
49  CALL    ERROHC(INAME,4)          UPD15960
50  COMMENT KEEP SUB TABLE PARAMETERS          UPD15970
51  CALL    COPYH  (INAME  ,1  ,LNAME  ,1  ,9      )       UPD15980
52  CALL    COPYH  (INAME  ,1  ,LIDENT  ,1  ,9      )       UPD15990
53  LSE@  = 0          UPD16000
54  LREC  = 0          UPD16010
55  LIND  = 0          UPD16020
56  LNUM  = LNUM+1          UPD16030
57  LTYP  = ITYP          UPD16040
58  LYAK  = 0          UPD16050
59  LCOM  = 1          UPD16060
60  LDUM  = 0          UPD16070
61  COMMENT RESET SUB TABLE PARAMETERS          UPD16080
62  3100 CONTINUE          UPD16090
63  IF(MRE,EQ,0)  GO TO3300          UPD16100
64  KIND  = M-KRE          UPD16110
65  KND  = KND-1          UPD16120
66  GO TO3600          UPD16130
67  COMMENT SET MAIN TABLE PARAMETERS          UPD16140
68  3200 CONTINUE          UPD16150
69  CALL    COPYH  (KNAME  ,1  ,MNAME(1+M)+1  ,9      )       UPD16160
70  CALL    COPYH  (KNAME  ,1  ,MDENT(1,M)+1  ,9      )       UPD16170
71  MSE@(M)= 0          UPD16180
72  MREC(M)= KRE          UPD16190
73  MIND(M)= KRE-KND+1          UPD16200
74  3210 CONTINUE          UPD16210
75  MRE  = 0          UPD16220
76  IF(CEOF,EQ,1)  GOTO 7000          UPD16230
77  COMMENT SEARCH MAIN TABLE          UPD16240
78  3300 CONTINUE          UPD16250
79  LLNUM  = LNUM          UPD16260
80  LLTYP  = LTYP          UPD16270
81  LLYAK  = LYAK          UPD16280
82  LLCOM  = LCOM          UPD16290
83  LLDDUM  = LDUM          UPD16300
84  KRE  = 0          UPD16310
85  KND  = 0          UPD16320
86  DO 3310 M=2,100          UPD16330
87  IF(MRECM),LE,KRE, GO TO3310          UPD16340
88  KRE  = MREC(M)          UPD16350
89  MRE  = M
90  3310 CONTINUE          UPD16360
91  M  = MRE          UPD16370
92  KSEQ  = LSE@          UPD16380
93  KREC  = KRE          UPD16390
94  KIND  = LIND          UPD16400
95  KNUM  = LNUM          UPD16410
96  KTYP  = LTYP          UPD16420
97  KYAK  = LYAK          UPD16430
98  KCOM  = LCOM          UPD16440
99  KDUM  = LDUM          UPD16450
100 3400 CONTINUE          UPD16460
101  COMMENT SET SUB TABLE PARAMETOR          UPD16470
102  3500 CONTINUE          UPD16480
103  KTYP  = LTYP          UPD16490
104  3510 CONTINUE          UPD16500

```

```

91      CALL    COPYH    C1TAB   *1     *XTAB    *1     *80    JUPD16630
92      CALL    COPYH    CLNAME   *1     *KNAME    *1     *9     JUPD16640
93      CALL    COPYH    CLDENT   *1     *KDENT    *1     *9     JUPD16650
94      KSEQ = KSEQ+1
95      KREC = KREC
96      KIND = KIND+1
97      KNUM = KNUM
98      KTYP = KTYP
99      KYAK = KYAK
100     KCOM = KCOM
101     KDUM = KDUM
102     COMMENT WRITE SUB TABLE PARAMETR
103     3600 CONTINUE
104     IF(KTYP,EQ,0) GO TO3610
105     IF(KTYP,EQ,23) GO TO3610
106     IF(KTYP,EQ,24) GO TO3610
107     IF(KTAB(1),NE,MASTER) GOTO 3610
108     WRITE(OUTPUT,9010) (KTAB(K8),K8=1,80),
109     (KDENT(K8),K8=1,8),
110     1
111     1
112     KSEQ
113     3610 CONTINUE
114     CALL    WRITET    (KRE ,KND )
115     IF(KIND,LT,0) GO TO3200
116     KND = KND+1
117     GO TO3000
118     300 CONTINUE
119     400 CONTINUE
120     500 CONTINUE
121     600 CONTINUE
122     700 CONTINUE
123     800 CONTINUE
124     900 CONTINUE
125     1000 CONTINUE
126     1200 CONTINUE
127     1400 CONTINUE
128     1600 CONTINUE
129     1800 CONTINUE
130     1900 CONTINUE
131     2000 CONTINUE
132     2100 CONTINUE
133     2200 CONTINUE
134     CALL    ERROR1(32)
135     COMMENT WDR
136     2300 CONTINUE
137     IF(LTYP,EQ,0) STOP 2300
138     LTYP = ITYP
139     GO TO3510
140     COMMENT CALL
141     2400 CONTINUE
142     IF(LTYP,EQ,0) STOP 2400
143     LTYP = ITYP
144     GO TO3510
145     COMMENT /
146     2500 CONTINUE
147     WRITE(OUTPUT,9010) (ITAB(180),I80=1,80)
148     GO TO3000
149     2600 CONTINUE
150     2700 CONTINUE
151     2800 CONTINUE
152     2900 CONTINUE
153     3000 FORMAT(8A1)
154     3100 FORMAT(1H1//,' CREATION RUN           FACOM 230-75   UPDATE 1.0  'UPD17330
155     3200 FORMAT(' ****  ',8A1,2X,8A1,1B)
156     3300 FORMAT(' ****  ',8A1,2X,8A1,1B)
157     END

```

```

*DECK   ERROR
1      SUBROUTINE  ERROR0  (IDENT  ,IOP  )          UPD17370
2      COMMON      INPUT   ,OUTPUT ,OLDPL ,NEWPL ,COMPILE ,SOURCE  UPD17380
3      1          IO     ,ICOMP ,ITEXT ,IXOLD ,IXNEW ,IX           UPD17390
4      2          MASTER ,F     ,D     ,FILE  ,NUM   ,IUPDAT    UPD17400
5      INTEGER     INPUT   ,OUTPUT ,OLDPL ,NEWPL ,COMPILE ,SOURCE  UPD17420
6      LOGICAL     F     ,D     ,FILE  ,NUM   ,IUPDAT    UPD17430
7      DIMENSION   ITAB   ( 90) ,INAME ( 9),IDENT ( 9),ISEQU ( 9) UPD17440
8      DATA NER/ 0 /
9      CALL       SETBCD (IDENT  ,B   ,INAME  )          UPD17450
10     IF(IOP.LE.0,OR,IOP.GT.10) STOP ' ERROR0 '
11     NER =NER+1
12     GOTO ( 10, 11, 12, 13, 14, 15, 16, 17, 18, 19) ,IOP UPD17500
13     10 CONTINUE
14     WRITE(OUTPUT,600) (INAME(I),I=1,2)          UPD17510
15     11 CONTINUE
16     WRITE(OUTPUT,601) (INAME(I),I=1,2)          UPD17520
17     12 CONTINUE
18     WRITE(OUTPUT,602) (INAME(I),I=1,2)          UPD17530
19     13 CONTINUE
20     WRITE(OUTPUT,603) (INAME(I),I=1,2)          UPD17540
21     14 CONTINUE
22     WRITE(OUTPUT,604) (INAME(I),I=1,2)          UPD17550
23     15 CONTINUE
24     WRITE(OUTPUT,605) (INAME(I),I=1,2)          UPD17560
25     16 CONTINUE
26     WRITE(OUTPUT,606) (INAME(I),I=1,2)          UPD17570
27     17 CONTINUE
28     WRITE(OUTPUT,607) (INAME(I),I=1,2)          UPD17580
29     18 CONTINUE
30     WRITE(OUTPUT,608) (INAME(I),I=1,2)          UPD17590
31     19 CONTINUE
32     WRITE(OUTPUT,609) (INAME(I),I=1,2)          UPD17600
33     20 CONTINUE
34     RETURN
35     600 FORMAT(1X,2A4,' DECK STRUCTURE HAS BEEN CHANGED ')
36     601 FORMAT(' *** ',2A4,' IS NOT A VALID DECK NAME ***')
37     602 FORMAT(' *** DO/DONT IDENT ',2A4,' IS NOT YANKED/YANKED/NULL DO/DOUPD17840
38     1ONT ***')
39     603 FORMAT(' *** DUPLICATE DECK ',2A4,' NEWPL ILLEGAL ***')
40     604 FORMAT(' *** DUPLICATE FILE NAME OF ',2A4,' JOB ABORTED ***')
41     605 FORMAT(' *** DUPLICATE IDENT CHENGED TO ',2A4,' ***')
42     606 FORMAT(' *** FILENAME OF ',2A4,' IS TOO LONG; UPDATE ABORTED ***') UPD17850
43     607 FORMAT(' *** IDENT LONGER THAN EIGHT CHARACTERS ',2A4,' *** ')
44     608 FORMAT(' *** UNKNOWN IDENTIFIER NAME ',2A4,' *** ')
45     609 FORMAT(' *** YANK, SELYANK, OR YANKDECK IDENT ',2A4,' NOT KNOWN **UPD17920
46     1* 1)
47     ENTRY    ERROR1 (IOP  )          UPD17940
48     IF(IOP.LE.0,OR,IOP.GT.50) STOP ' ERROR1 '
49     IF(IOP.LT.42) NER =NER+1
50     GOTO ( 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, UPD17950
51     1        40, 41, 42, 43, 44, 45, 46, 47, 48, 49, UPD17960
52     2        50, 51, 52, 53, 54, 55, 56, 57, 58, 59, UPD17970
53     3        60, 61, 62, 63, 64, 65, 66, 67, 68, 69, UPD17980
54     4        70, 71, 72, 73, 74, 75, 76, 77, 78, 79) ,IOP UPD18000
55     30 CONTINUE
56     WRITE(OUTPUT,500)          UPD18010
57     31 CONTINUE
58     WRITE(OUTPUT,501)          UPD18020
59     32 CONTINUE
60     WRITE(OUTPUT,502)          UPD18030
61     33 CONTINUE
62     WRITE(OUTPUT,503)          UPD18040
63     34 CONTINUE
64     WRITE(OUTPUT,504)          UPD18050
65     35 CONTINUE
66     WRITE(OUTPUT,505)          UPD18060
67     36 CONTINUE
68     WRITE(OUTPUT,506)          UPD18070
69     37 CONTINUE
70     WRITE(OUTPUT,507)          UPD18080
71     38 CONTINUE
72     WRITE(OUTPUT,508)          UPD18090
73     39 CONTINUE
74     WRITE(OUTPUT,509)          UPD18100
75     40 CONTINUE
76     WRITE(OUTPUT,510)          UPD18110
77     41 CONTINUE
78     WRITE(OUTPUT,511)          UPD18120
79     42 CONTINUE
80     WRITE(OUTPUT,512)          UPD18130
81     43 CONTINUE
82     WRITE(OUTPUT,513)          UPD18140
83     44 CONTINUE
84     WRITE(OUTPUT,514)          UPD18150
85     45 CONTINUE
86     WRITE(OUTPUT,515)          UPD18160
87     46 CONTINUE

```

```

105      WRITE(OUTPUT,516)          UPD18510
106      GOTO 80                 UPD18520
107      47 CONTINUE              UPD18530
108      WRITE(OUTPUT,517)          UPD18540
109      GOTO 80                 UPD18550
110      48 CONTINUE              UPD18560
111      WRITE(OUTPUT,518)          UPD18570
112      GOTO 80                 UPD18580
113      49 CONTINUE              UPD18590
114      WRITE(OUTPUT,519)          UPD18600
115      GOTO 80                 UPD18610
116      50 CONTINUE              UPD18620
117      WRITE(OUTPUT,520)          UPD18630
118      GOTO 80                 UPD18640
119      51 CONTINUE              UPD18650
120      WRITE(OUTPUT,521)          UPD18660
121      GOTO 80                 UPD18670
122      52 CONTINUE              UPD18680
123      WRITE(OUTPUT,522)          UPD18690
124      GOTO 80                 UPD18700
125      53 CONTINUE              UPD18710
126      WRITE(OUTPUT,523)          UPD18720
127      GOTO 80                 UPD18730
128      54 CONTINUE              UPD18740
129      WRITE(OUTPUT,524)          UPD18750
130      GOTO 80                 UPD18760
131      55 CONTINUE              UPD18770
132      WRITE(OUTPUT,525)          UPD18780
133      GOTO 80                 UPD18790
134      56 CONTINUE              UPD18800
135      WRITE(OUTPUT,526)          UPD18810
136      GOTO 80                 UPD18820
137      57 CONTINUE              UPD18830
138      WRITE(OUTPUT,527)          UPD18840
139      GOTO 80                 UPD18850
140      58 CONTINUE              UPD18860
141      WRITE(OUTPUT,528)          UPD18870
142      GOTO 80                 UPD18880
143      59 CONTINUE              UPD18890
144      WRITE(OUTPUT,529)          UPD18900
145      GOTO 80                 UPD18910
146      60 CONTINUE              UPD18920
147      WRITE(OUTPUT,530)          UPD18930
148      GOTO 80                 UPD18940
149      61 CONTINUE              UPD18950
150      WRITE(OUTPUT,531)          UPD18960
151      GOTO 80                 UPD18970
152      62 CONTINUE              UPD18980
153      WRITE(OUTPUT,532)          UPD18990
154      GOTO 80                 UPD19000
155      63 CONTINUE              UPD19010
156      WRITE(OUTPUT,533)          UPD19020
157      GOTO 80                 UPD19030
158      64 CONTINUE              UPD19040
159      WRITE(OUTPUT,534)          UPD19050
160      GOTO 80                 UPD19060
161      65 CONTINUE              UPD19070
162      WRITE(OUTPUT,535)          UPD19080
163      GOTO 80                 UPD19090
164      66 CONTINUE              UPD19100
165      WRITE(OUTPUT,536)          UPD19110
166      GOTO 80                 UPD19120
167      67 CONTINUE              UPD19130
168      WRITE(OUTPUT,537)          UPD19140
169      GOTO 80                 UPD19150
170      68 CONTINUE              UPD19160
171      WRITE(OUTPUT,538)          UPD19170
172      GOTO 80                 UPD19180
173      69 CONTINUE              UPD19190
174      WRITE(OUTPUT,539)          UPD19200
175      GOTO 80                 UPD19210
176      70 CONTINUE              UPD19220
177      WRITE(OUTPUT,540)          UPD19230
178      GOTO 80                 UPD19240
179      71 CONTINUE              UPD19250
180      WRITE(OUTPUT,541)          UPD19260
181      GOTO 80                 UPD19270
182      72 CONTINUE              UPD19280
183      WRITE(OUTPUT,542)          UPD19290
184      GOTO 80                 UPD19300
185      73 CONTINUE              UPD19310
186      WRITE(OUTPUT,543)          UPD19320
187      GOTO 80                 UPD19330
188      74 CONTINUE              UPD19340
189      WRITE(OUTPUT,544)          UPD19350
190      GOTO 80                 UPD19360
191      75 CONTINUE              UPD19370
192      WRITE(OUTPUT,545)          UPD19380
193      GOTO 80                 UPD19390
194      76 CONTINUE              UPD19400
195      WRITE(OUTPUT,546)          UPD19410
196      GOTO 80                 UPD19420
197      77 CONTINUE              UPD19430
198      WRITE(OUTPUT,547)          UPD19440
199      GOTO 80                 UPD19450
200      78 CONTINUE              UPD19460
201      WRITE(OUTPUT,548)          UPD19470
202      GOTO 80                 UPD19480
203      79 CONTINUE              UPD19490
204      WRITE(OUTPUT,549)          UPD19500
205      GOTO 80                 UPD19510
206      80 CONTINUE              UPD19520
207      500 FORMAT(' *** ADDFILE FIRST CARD MUST BE DECK OR COMDECK *** ') UPD19530
208      501 FORMAT(' *** ADDFILE CARD INVALID ON REMOTE FILE *** ') UPD19540
209      502 FORMAT(' *** BAD ORDER ON YANK DIRECTIVE *** ') UPD19550
210      503 FORMAT(' *** CARD NUMBER ZERO OR INVALID CHARACTER IN NUMERIC FILE *** ') UPD19560
211      *LD *** ')
212      504 FORMAT(' *** CONTROL CARD INVALID OR MISSING *** ') UPD19580
213      505 FORMAT(' *** COPY TO EXTERNAL FILE NOT ALLOWED WHEN READING FROM UPD19590
*ALTERNATE INPUT UNIT *** ')
214      506 FORMAT(' *** DECK NAME ON ABOVE CARD NOT LAST DECLARED DECK *** ') UPD19610
* )
215      507 FORMAT(' *** DECK SPECIFIED ON MOVE OR COPY CARD NOT ON OLDPL, CAUPD19630
*RD WILL BE IGNORED *** ') UPD19640

```

```

215 508 FORMAT( ' *** DUPLICATE IDENT NAME *** ' ) UPD19650
216 509 FORMAT( ' *** DUPLICATE IDENT NAME IN ADDFILE *** ' ) UPD19660
217 510 FORMAT( ' *** ERROR *** NOT ALL MODS WERE PROCESSED *** ' ) UPD19670
218 511 FORMAT( ' *** FILE NAME ON ABOVE CARD GREATER THAN SEVEN CHARACTER UPD19680
     **** ' ) UPD19690
219 512 FORMAT( ' *** IDENT CARD MISSING NO NEWPL REQUESTED, DEFAULT IDENT UPD19700
     *IFER OF NO. ID. USED *** ' ) UPD19710
220 513 FORMAT( ' *** IDENTIFIERS SEPARATED BY PERIOD IN WRONG ORDER *** ' ) UPD19720
     * )
221 514 FORMAT( ' *** ILLEGAL CONTROL CARD IN ADDFILE *** ' ) UPD19740
222 515 FORMAT( ' *** INVALID NUMERIC FIELD *** ' ) UPD19750
223 516 FORMAT( ' *** LENGTH ERROR ON OLDPL, UNUSABLE OLDPL OR HARDWARE ERUPD19760
     *ROR, *** ' ) UPD19770
224 517 FORMAT( ' *** LIST BELOW ARE ALL IDENT NAMES WHICH WERE CHANGED DUUPD19780
     *RING THE MERGE *** ' ) UPD19790
225 518 FORMAT( ' *** ALL YANK, SELYANK, YANKECK, AND CALL CARDS AFFECTEDUPD19800
     * HAVE BEEN CHANGED *** ' ) UPD19810
226 519 FORMAT( ' *** NEW IDENT ON CHANGE CARD IS ALREADY KNOWN *** ' ) UPD19820
227 520 FORMAT( ' *** NO ACTIVE CARDS WERE FOUND WITHIN THE COPY RANGE, NUUPD19830
     *LL COPY *** ' ) UPD19840
228 521 FORMAT( ' *** NO DECK NAME ON DECK CARD *** ' ) UPD19850
229 522 FORMAT( ' *** NULL ADDFILE *** ' ) UPD19860
230 523 FORMAT( ' *** NULL DECK NAME *** ' ) UPD19870
231 524 FORMAT( ' *** NULL IDENT *** ' ) UPD19880
232 525 FORMAT( ' *** OLDPPL READ ERROR-POSSIBLE LUST DATA AFTER FOLLOWING CUPD19890
     *ARD *** ' ) UPD19900
233 526 FORMAT( ' *** AND BEFORE THE FOLLOWING CARD *** ' ) UPD19910
234 527 FORMAT( ' *** OUTPUT LINE LIMIT EXCEEDED, LIST OPTIONS 3 AND 4 DEFUPD19920
     *EATED, *** ' ) UPD19930
235 528 FORMAT( ' *** PREMATURE END OF RECORD ON OLD PROGRAM LIBRARY *** ' ) UPD19940
     * )
236 529 FORMAT( ' *** THE ABOVE CALLED COMMON DECK WAS NOT FOUND *** ' ) UPD19960
237 530 FORMAT( ' *** THE ABOVE CARD AFFECTS A DECK OTHER THAN THE DECLAREUPD19970
     *D DECK *** ' ) UPD19980
238 531 FORMAT( ' *** THE ABOVE CARD IS ILLEGAL DURING A CREATION RUN *** UPD19990
     * ')
239 532 FORMAT( ' *** THE ABOVE CONTROL CARD IS ILLEGAL AFTER A DECK HAS BUPD20010
     *EEN DECLARED *** ' ) UPD20020
240 533 FORMAT( ' *** THE ABOVE LISTED CARDS CANNOT EXIST IN THE YANK DECKUPD20030
     * AND HAVE BEEN PURGED DURING EDITING *** ' ) UPD20040
241 534 FORMAT( ' *** THE ABOVE OPERATION IS NOT LEGAL WHEN REFERENCING THUPD20050
     *E YANK DECK *** ' ) UPD20060
242 535 FORMAT( ' *** THE ABOVE SPECIFIED CARD WAS NOT ENCOUNTERED *** ' ) UPD20070
243 536 FORMAT( ' *** THE INITIAL CARD OF THE COPY RANGE WAS NOT FOUND. NUUPD20080
     *LL COPY *** ' ) UPD20090
244 537 FORMAT( ' *** THE TERMINAL CARD SPECIFIED ABOVE WAS NOT ENCOUNTEREDUPD20110
     *D *** ' ) UPD20110
245 538 FORMAT( ' *** IT MAY EXIST IN A DECK NOT MENTIONED ON A COMPILE CAUPD20120
     *RD *** ' ) UPD20130
246 539 FORMAT( ' *** THE TERMINAL CARD OF THE COPY RANGE WAS NOT FOUND. CUPD20140
     *OPY ENDS AT END OF SPECIFIED DECK *** ' ) UPD20150
247 540 FORMAT( ' *** TOO MANY CHBS--INCREASE L. CHB *** ' ) UPD20160
248 541 FORMAT( ' *** UNBALANCES TEXT/ENDTEXT CARDS, LAST ENDTEXT CARD IGNUPD20170
     *ORED *** ' ) UPD20180
249 542 FORMAT( ' *** WARNING *** OLDPPL CHECKSUM ERROR *** ' ) UPD20190
250 543 FORMAT(15,' ERRORS IN UPDATE INPUT ' ) UPD20200
251 544 FORMAT(15,' DECLARE ERRORS ' ) UPD20210
252 545 FORMAT(15,' FATAL ERRORS ' ) UPD20220
253 546 FORMAT(15,' NONFATAL ERRORS ' ) UPD20230
254 547 FORMAT(15,' UPDATE ERRORS, JOB ABORTED ' ) UPD20240
255 548 FORMAT(15,' 15,1 ERRORS IN INPUT,NEWPL,COMPILE,SOURCE SUPPRESSEDUPD20250
     1D, *** ' ) UPD20260
256 549 FORMAT(//)
257     RETURN
258     END

```

```

*DECK FTMMAIN
1 ELEMENT FTMMAIN UPD20300
2 COMMON INPUT OUTPUT OLDPL NEWPL COMPILE SOURCE UPD20310
3 1 IO ICOMP ITEXT IXOLD IXNEW IX UPD20320
4 2 MASTER F D IFILE NUM IUPDAT UPD20330
5 INTEGER INPUT OUTPUT OLDPL NEWPL COMPILE SOURCE UPD20340
6 LOGICAL F D IFILE NUM IUPDAT UPD20350
7 NAMELIST /UPDATES/ UPD20360
8 1 INPUT OUTPUT OLDPL NEWPL COMPILE SOURCE UPD20370
9 2 MASTER F D IFILE NUM IUPDAT UPD20380
10 COMMENT UPDATE FILE SETTING UPD20390
11 6000 CONTINUE UPD20400
12 INPUT = 5 UPD20410
13 OUTPUT = 6 UPD20420
14 OLDPL = 0 UPD20430
15 NEWPL = 0 UPD20440
16 COMPILE = 0 UPD20450
17 SOURCE = 0 UPD20460
18 IO = 8 UPD20470
19 ICOMP = 98 UPD20480
20 ITEXT = 99 UPD20490
21 MASTER = 1H* UPD20500
22 F = .FALSE. UPD20510
23 D = .FALSE. UPD20520
24 IFILE = 0 UPD20530
25 NUM = 0 UPD20540
26 IUPDAT = 0 UPD20550
27 COMMENT INPUT UPDATE PARAMETER UPD20560
28 READ (5,UPDATE-END=7000) UPD20570
29 COMMENT OUTPUT UPDATE PARAMETER UPD20580
30 WRITE (6,UPDATE) UPD20590
31 COMMENT COPY INPUT TEXT TO FILE ITEXT UPD20600
32 CALL COTEXT UPD20610
33 CALL OPENIO UPD20620
34 GO TO (1000,2000),IUPDAT UPD20630
35 COMMENT UPDATE CREATION RUN UPD20640
36 1000 CONTINUE UPD20650
37 CALL TABSET UPD20660
38 CALL CREATE UPD20670
39 GO TO3000 UPD20680
40 2000 CONTINUE UPD20690
41 COMMENT UPDATE CORRECTION RUN UPD20700
42 CALL TABSET UPD20710
43 CALL CORREC UPD20720
44 3000 CONTINUE UPD20730
45 COMMENT GENERATE COMPILE FILE UPD20740
46 CALL COMP UPD20750
47 CALL TABSET UPD20760
48 GO TO6000 UPD20770
49 7000 CONTINUE UPD20780
50 STOP UPD20790
51 END UPD20800
52
53 *DECK GETSET UPD20820
54 SUBROUTINE GETSET UPD20830
55 DIMENSION IDENT ( 1 ),JDENT ( 1 ) UPD20840
56 DIMENSION IMAT ( 3 ),AMAT ( 3 ) UPD20850
57 DIMENSION NUMBER ( 2 ) UPD20860
58 INTEGER AMAT UPD20870
59 DATA IMAT /'( )', '( )', '( )'/
60 DATA AMAT /'( )', '( )', '( )'/
61 C *** JREC IS NUMBER OF CHARACTER PACKED BCD TO UNPACKED BCD UPD20880
62 ENTRY GETBCD ( IDENT *JREC *JDENT ) UPD20890
63 DECODE(80,8000,IDENT) (JDENT(J),J=1,JREC) UPD20900
64 RETURN UPD20930
65 C *** JREC IS NUMBER OF CHARACTER INTEGER TO UNPACKED BCD UPD20940
66 ENTRY GETNUM ( IDENT *JREC *JDENT ) UPD20950
67 IF(JREC.LT.0) GO TO1020 UPD20960
68 DO 1000 JREC=1,8 UPD20970
69 LREC = 10**JREC UPD20980
70 IF(IDENT(1).LT.LREC)GO TO1010 UPD20990
71 1000 CONTINUE UPD21000
72 JREC = 8 UPD21010
73 1010 CONTINUE UPD21020
74 LREC = JREC UPD21030
75 GO TO1030 UPD21040
76 1020 CONTINUE UPD21050
77 LREC = IABS(JREC) UPD21060
78 1030 CONTINUE UPD21070
79 IF(LREC.GT.8) LREC=8 UPD21080
80 ENCODE(4,8002,IMAT(2)) LREC UPD21090
81 ENCODE(4,8002,AMAT(2)) LREC UPD21100
82 ENCODE(8,IMAT,NUMBER) IDENT(1) UPD21110
83 DECODE(8,AMAT,NUMBER) (JDENT(J),J=1,LREC) UPD21120
84 RETURN UPD21130
85 C *** IREC IS NUMBER OF WORDS UNPACKED BCD TO PACKED BCD UPD21140
86 ENTRY SETBCD ( IDENT *IREC *JDENT ) UPD21150
87 ENCODE(80,8000,JDENT) (JDENT(I),I=1,IREC) UPD21160
88 RETURN UPD21170
89 C *** IREC IS NUMBER OF WORDS UNPACKED BCD TO NUMBER UPD21180
90 ENTRY SETNUM ( IDENT *IREC *JDENT ) UPD21190
91 IF(IREC.LE.0) GO TO2010 UPD21200
92 ENCODE(4,8002,AMAT(2)) IREC UPD21210
93 ENCODE(4,8002,IMAT(2)) IREC UPD21220
94 ENCODE(8,AMAT,NUMBER) (JDENT(I),I=1,IREC) UPD21230
95 DECODE(8,IMAT,NUMBER) JDENT(1) UPD21240
96 RETURN UPD21250
97 2010 CONTINUE UPD21260
98 JDENT(1)= 0 UPD21270
99 RETURN UPD21280
100 8000 FORMAT(80A1) UPD21290
101 8002 FORMAT(14) UPD21300
102 END UPD21310

```

```

*DECK INDEX
1. SUBROUTINE INDEX (KRE ,KND ,KRE1 ,KND1 ) UPD21320
2. KRE1 = KRE+KND UPD21330
3. IF(KRE1.LE.0) KRE1=KRE UPD21340
4. IF(KRE1.LE.0) STOP 1000 UPD21350
5. KND1 = KRE1/1041 UPD21360
6. IF(MOD(KRE1,10).EQ.0) KND1=KND1-1 UPD21370
7. RETURN UPD21380
8. END UPD21390
UPD21400

*DECK OPENIO
1. SUBROUTINE OPENIO UPD21410
2. DEFINEFILE 1(1000+1280,L,IX) UPD21420
3. DEFINEFILE 2(1000+1280,L,IX) UPD21430
4. DEFINEFILE 3(1000+1280,L,IX) UPD21440
5. DEFINEFILE 4(1000+1280,L,IX) UPD21450
6. DEFINEFILE 8(1000+1280,L,IX) UPD21460
7. RETURN UPD21470
8. END UPD21480
UPD21490

*DECK RESETH
1. SUBROUTINE RESETH (HOL1 ,IHOL ,HOL2 ) UPD21500
2. DIMENSION HOL1 ( 1 ) UPD21510
3. INTEGER HOL1 ,HOL2 UPD21520
4. DO 1000 II=1,IMCL UPD21530
5. HOL1(II)= HOL2 UPD21540
6. 1000 CONTINUE UPD21550
7. RETURN UPD21560
8. END UPD21570
UPD21580

*DECK RESETI
1. SUBROUTINE RESETI (INT1 ,INT 1) UPD21590
2. DIMENSION INT1 ( 1 ) UPD21600
3. INTEGER INT1 ,INT2 UPD21610
4. DO 1000 II=1,INT UPD21620
5. INT1(II)= INT2 UPD21630
6. 1000 CONTINUE UPD21640
7. RETURN UPD21650
8. END UPD21660
UPD21670

*DECK TABLES
1. SUBROUTINE TABLES UPD21680
2. COMMON INPUT ,OUTPUT ,OLDPL ,NEWPL ,COMPILE ,SOURCE ,IX UPD21690
3. 1 IO ,ICOMP ,ITEXT ,IXOLD ,IXNEW ,IX ,IUPDAT UPD21700
4. 2 MASTER ,F ,D ,FILE ,NUM ,IUPDAT UPD21710
5. INTEGER INPUT ,OUTPUT ,OLDPL ,NEWPL ,COMPILE ,SOURCE UPD21720
6. LOGICAL F ,D ,FILE ,NUM ,IUPDAT UPD21730
7. DIMENSION XTAB ,KTAB ( 32+ 10)*KTABJ ( 32+ 10) UPD21740
8. COMMON/TABK/XTAB ( 900*KNAME ( 90 ),KDENT ( 90 ),KYAK ,KYAK ,KCOM ,LNAME ,LNAME ,LDENT ,LDENT ,LDE ,LDE ,JNDE ,JNDE ) UPD21750
9. DATA INDE /0 / UPD21760
10. C *** CLEAR TABLES UPD21770
11. ENTRY CLEAR ,GETBCD (14HYANK***) ,9 ,LDENT ) UPD21780
12. CALL GETBCD (14HYANK*** ,9 ,LDENT ) UPD21790
13. IF = 1 UPD21800
14. KSEQ = 0 UPD21810
15. KREC = 101 UPD21820
16. KIND = 101 UPD21830
17. KNUM = 0 UPD21840
18. KYAK = -1 UPD21850
19. KCOM = 0 UPD21860
20. KDUM = 0 UPD21870
21. GO TO 1000 UPD21880
22. 1000 CONTINUE UPD21890
23. CALL RESETH (XTAB ,80 ,+6H ) UPD21900
24. CALL RESETH (KNAME ,9 ,+6H ) UPD21910
25. CALL RESETH (KDENT ,9 ,+6H ) UPD21920
26. DO 1100 K=1,100 UPD21930
27. KREC = K UPD21940
28. KND0 = 0 UPD21950
29. CALL INDEX (KREC ,KND0 ,KRE1 ,KND1 ) UPD21960
30. KREE = MOD(KRE1,10) UPD21970
31. IF(KREE.EQ.0) KREE=10 UPD21980
32. IF(KREE.NE.1) GO TO 1200 UPD21990
33. CALL RESETI (XTABJ ,320 ,0 ) UPD22000
34. IF(KREC.NE.1) GO TO 1200 UPD22010
35. CALL COPYH (LDENT ,1 ,KDENT ,1 ,+8 ,+8 ) UPD22020
36. CALL COPYH (LDENT ,1 ,KNAME ,1 ,+8 ,+8 ) UPD22030
37. GO TO 1300 UPD22040
38. 1200 CONTINUE UPD22050
39. CALL RESETH (KDENT ,8 ,+6H ) UPD22060
40. KREC = 101 UPD22070
41. KIND = 0 UPD22080
42. KYAK = 0 UPD22090
43. 1300 CONTINUE UPD22100
44. ENCODE(128+8001+KTABJ(1+KREE)) (KTAB(K80),K80=1,80) , UPD22110
1. (KNAME(K80),K8=1,8) , UPD22120
1. (KDENT(K80),K8=1,8) , UPD22130
1. KSE0,KREC+KIND,KNUM, UPD22140
1. KYAK,KCOM,KDUM, UPD22150
45. IF(KREE.NE.10) GO TO 1100 UPD22160
46. IX = KND1 UPD22170
UPD22180
UPD22190
UPD22200
UPD22210
UPD22220
UPD22230
UPD22240

```

```

47      WRITE (IO'IX)    ((KTABJ(K32,K10),K32=1,32),K10=1,10)    UPD22250
48      1100 CONTINUE
49      RETURN
50      C *** WRITE TABLES
51      ENTRY   WRITET (KRE   ,KND   )    UPD22260
52      IF      = 2    UPD22270
53      GO TO 6000  UPD22280
54      KSEQ   = LSEQ  UPD22290
55      KREC   = LREC  UPD22300
56      KIND   = LIND  UPD22310
57      KNUM   = LNUM  UPD22320
58      KTYP   = LTYP  UPD22330
59      KYAK   = LYAK  UPD22340
60      KCOM   = LCOM  UPD22350
61      KDUM   = LDUM  UPD22360
62      IF(KRE1,LE,100) GO TO 2010  UPD22370
63      IF(KRE1,GT,100) GO TO 2030  UPD22380
64      2000 CONTINUE
65      ENCODE(128,8001,KTABJ(1,KREE))((KTAB(K80),K80=1,80),    UPD22390
66      1          (KNAME(K8),K8=1,8),    UPD22400
67      1          (KDENT(K8),K8=1,8),    UPD22410
68      1          KSEQ,KREC,KIND,KNUM,    UPD22420
69      1          KTYP,KYAK,KCOM,KDUM    UPD22430
70      2010 CONTINUE
71      ENCODE(128,8001,KTAB(1,KREE))((KTAB(K80),K80=1,80),    UPD22440
72      1          (KNAME(K8),K8=1,8),    UPD22450
73      1          (KDENT(K8),K8=1,8),    UPD22460
74      1          KSEQ,KREC,KIND,KNUM,    UPD22470
75      1          KTYP,KYAK,KCOM,KDUM    UPD22480
76      IF(KREE,NE,10) RETURN
77      #WRITE (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)    UPD22490
78      RETURN
79      2030 CONTINUE
80      ENCODE(128,8001,KTAB(1,KREE))((KTAB(K80),K80=1,80),    UPD22500
81      1          (KNAME(K8),K8=1,8),    UPD22510
82      1          (KDENT(K8),K8=1,8),    UPD22520
83      1          KSEQ,KREC,KIND,KNUM,    UPD22530
84      1          KTYP,KYAK,KCOM,KDUM    UPD22540
85      IF(KREE,EQ,0) RETURN
86      #WRITE (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)    UPD22550
87      RETURN
88      C *** READ TABLES
89      ENTRY   READT (KRE   ,KND   )    UPD22560
90      IF      = 3    UPD22570
91      GO TO 6000  UPD22580
92      3000 CONTINUE
93      CALL   INDEX  (KRE   ,KND   ,KRE1 ,KND1 )    UPD22590
94      KREE   = MOD(KRE1,10)    UPD22600
95      IF(KREE,EQ,0) KREE=10    UPD22610
96      IF(KRE1,LE,100) GO TO 3010  UPD22620
97      IF(KRE1,GT,100) GO TO 3030  UPD22630
98      3010 CONTINUE
99      IF(JNDE,EQ,0) GO TO 3015  UPD22640
100     IF(KND1,NE,JNDE) GO TO 3020  UPD22650
101     IX   = JNDE
102     IX   = JNDE
103     WRITE (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)    UPD22660
104     3015 CONTINUE
105     JNDE   = KND1
106     IX   = KND1
107     READ (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)    UPD22670
108     3020 CONTINUE
109     IF(IF,EQ,2) GO TO 2000  UPD22680
110     DECODE(128,8001,KTABJ(1,KREE))((KTAB(K80),K80=1,80),    UPD22690
111     1          (KNAME(K8),K8=1,8),    UPD22700
112     1          (KDENT(K8),K8=1,8),    UPD22710
113     1          KSEQ@+KREC,KIND,KNUM,    UPD22720
114     1          KTYP,KYAK,KCOM,KDUM    UPD22730
115     RETURN
116     6000 CONTINUE
117     LSEQ   = KSEQ  UPD22740
118     LREC   = KREC  UPD22750
119     LIND   = KIND  UPD22760
120     LNUM   = KNUM  UPD22770
121     LTYP   = KTYP  UPD22780
122     LYAK   = KYAK  UPD22790
123     LCOM   = KCOM  UPD22800
124     LDUM   = KDUM  UPD22810
125     GO TO (1000+3000+3000)+IF    UPD22820
126     8001 FORMAT(B0A1,8A1,8A1,1D4)    UPD22830
127     END

```

```

1 *DECK TABSET          UPD23170
2   SUBROUTINE TABSET          UPD23160
3     COMMON INPUT  ,OUTPUT ,OLDPL ,NEWPL ,COMPILE ,SOURCE ,UPD23190
4       IO    ,ICOMP ,ITEXT ,IXOLD ,IXNEW ,IX ,UPD23200
5       MASTER ,F    ,D    ,FILE ,NUM ,IUPDAT ,UPD23210
6     INTEGER INPUT  ,OUTPUT ,OLDPL ,NEWPL ,COMPILE ,SOURCE ,UPD23220
7     LOGICAL F    ,D    ,UPD23230
8     DIMENSION KTABI ( 32, 10),KTABJ ( 32, 10)
9     COMMON/TABK/KTAB ( 90),KNAME ( 90),KDENT ( 90),UPD23240
10    KSEGU ( 90),UPD23250
11    COMMON/TABT/KSEQ ( 90),KREC ,KIND ,KNUM ,KTYP ,KYAK ,UPD23260
12    KCOM ,KDUM ,UPD23270
13    COMMON/TABM/MTAB ( 100),MNAME ( 9,100),MDENT ( 9,100),UPD23280
14    MSEQ ( 100),MREC ( 100),MIND ( 100),UPD23290
15    MNUM ( 100),MTYP ( 100),MYAK ( 100),UPD23300
16    MCOM ( 100),MDUM ( 100),UPD23310
17    COMMENT UPDATE CREATION RUN
18    C *** MAIN TABLE CLEAR
19    CALL  CLART          UPD23320
20    GO TO (1000,2000,3000),IUPDAT          UPD23330
21    C *** COPY MAIN TABLE TO MTAB ARRAY
22    1000 CONTINUE          UPD23340
23    LIND = 0              UPD23350
24    DO 1100 K=1,10          UPD23360
25    IX = K              UPD23370
26    READ (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)          UPD23380
27    DO 1200 KREE=1,10          UPD23390
28    M100 = (K-1)*10+KREE          UPD23400
29    DECODE(128,8000,KTABJ(1,KREE))((MNAME(MB,M100),MB=1,8),
30      (MDENT(MB,M100),MB=1,8),          UPD23410
31      MSE(M100),MREC(M100),MIND(M100),          UPD23420
32      MNUM(M100),MTYP(M100),MYAK(M100),          UPD23430
33      MCOM(M100),VNUM(M100),          UPD23440
34      MCOD(M100)=0          UPD23450
35      MDUM(M100)=0          UPD23460
36      IF(IUPDAT,EQ.1) GO TO1200          UPD23470
37      IF(LIND.LE.MIND(M100)) LIND=MIND(M100)          UPD23480
38    1200 CONTINUE          UPD23490
39    1100 CONTINUE          UPD23500
40    CALL  COPYH (MDENT ,1 ,MNAME ,1 ,8)          UPD23510
41    IF(IUPDAT,EQ.2) GO TO2200          UPD23520
42    RETURN          UPD23530
43    COMMENT UPDATE CORRECTION RUN
44    2000 CONTINUE          UPD23540
45    IF(OLDPL,EQ.0) STOP          UPD23550
46    DO 2100 K=1,10          UPD23560
47    IXOLD = K              UPD23570
48    READ (OLDPL,IXOLD) ((KTABJ(K32,K10),K32=1,32),K10=1,10)          UPD23580
49    IX = K              UPD23590
50    WRITE (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)          UPD23600
51    2100 CONTINUE          UPD23610
52    GO TO1000          UPD23620
53    COMMENT
54    2200 CONTINUE          UPD23630
55    LREC = 0              UPD23640
56    CALL  INDEX (LREC ,LIND ,KRE1 ,KND1 )          UPD23650
57    DO 2300 K=1,KND1          UPD23660
58    IXOLD = K              UPD23670
59    READ (OLDPL,IXOLD) ((KTABJ(K32,K10),K32=1,32),K10=1,10)          UPD23680
60    IX = K              UPD23690
61    WRITE (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)          UPD23700
62    2300 CONTINUE          UPD23710
63    RETURN          UPD23720
64    COMMENT
65    3000 CONTINUE          UPD23730
66    IF(NEWPL,EQ.0) GO TO5000          UPD23740
67    CALL  RESETH (KTAB ,90 ,6H )          UPD23750
68    LIND = 0              UPD23760
69    DO 3100 M=1,100          UPD23770
70    CALL  COPYH (MNAME(1,M),1 ,KNAME ,1 ,9)          UPD23780
71    CALL  COPYH (MDENT(1,M)+1 ,XDENT ,1 ,9)          UPD23790
72    KSEQ = MSE(M)          UPD23800
73    KREC = MREC(M)          UPD23810
74    KIND = MIND(M)          UPD23820
75    KNUM = MNUM(M)          UPD23830
76    KTYP = MTYP(M)          UPD23840
77    KYAK = MYAK(M)          UPD23850
78    KCOM = MCOM(M)          UPD23860
79    KDUM = MDUM(M)          UPD23870
80    KRE = M              UPD23880
81    KND = 0              UPD23890
82    CALL  WRITET (KRE ,KND )          UPD23900
83    IF(LIND.LE.MIND(M)) LIND=MIND(M)          UPD23910
84    3010 CONTINUE          UPD23920
85    LREC = 0              UPD23930
86    CALL  INDEX (LREC ,LIND ,KRE1 ,KND1 )          UPD23940
87    DO 3100 K=1,KND1          UPD23950
88    IX = K              UPD23960
89    READ (IO'IX) ((KTABJ(K32,K10),K32=1,32),K10=1,10)          UPD23970
90    IXNEW = K              UPD23980
91    WRITE (NEWPL,IXNEW) ((KTABJ(K32,K10),K32=1,32),K10=1,10)          UPD23990
92    3100 CONTINUE          UPD24000
93    5000 CONTINUE          UPD24010
94    RETURN          UPD24020
95    8000 FORMAT(80X,8A1,8A1,10A4)          UPD24030
96    END          UPD24040

```