

JAERI-M

6719

放射線測定および核物質非破壊測定を目的とする  
データ集積・保存・解析のためのORACL  
プログラム集(第2集)

1976年9月

八木 秀之・竹内 紀男・五藤 博

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、複製などのお問い合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしてください。

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.

放射線測定および核物質非破壊測定を目的とするデータ  
集積・保存・解析のためのORACLプログラム集(第2集)

日本原子力研究所東海研究所原子炉工学部

八木秀之・竹内紀男<sup>+</sup>・五藤 博

(1976年9月1日 受理)

ORTEC社製GAMMA-IIIシステムを用いて、放射線測定および核物質非破壊測定を行うために必要な会話型言語ORACLによるプログラム79編を収録した。内容は、ディスク・コア・MCA・磁気テープ相互間のデータの転送、ディスク内データの編集、ピーク面積計算、平均値・標準偏差の計算、ガンマ線データ・ファイルの検索、会計データの処理、暦日の計算、その他である。また、ORACLをメタ言語とするマイクロコンピュータ開発用のサポート・システムも収録した。ORACL内蔵の関数類の使用法も総覧し、使用の便をはかった。

---

+) 日本原子力研究所アイソトープ事業部製造部

ORACL Program File  
for Acquisition, Storage and Analysis of Data  
in Radiation Measurement and  
Nondestructive Measurement of Nuclear Material  
Volume II

Hideyuki YAGI, Norio TAKEUCHI<sup>+</sup> and Hiroshi GOTOH  
Division of Reactor Engineering, Tokai, JAERI

(Received September 1, 1976)

The file contains 79 programs for radiation measurement and nondestructive measurement of nuclear material written in conversational language ORACL associated with the GAMMA-III system of ORTEC Incorporated. It deals with data transfers between disk/core/MCA/magnetic tape, edition of data in disks, calculation of the peak area, calculation of mean and standard deviation, reference to gamma-ray data files, accounting, calendar, etc. It also has a support system for micro-computer development. Usages of the built-in functions of ORACL are presented.

---

<sup>+</sup>) Div. of Radioisotope Production, Radioisotope Center, JAERI

## 目 次

( ページ数の I-2 は第 1 集の 2 ページを意味する )

1. まえがき		1
2. プログラムの用途, 使い方		2
( 紙テープまたはキーボードから読む )		
1. 紙テープをコアに読む	PTME .....	I-2
2. 紙テープをディスクに読む(1)	PTFD .....	I-2
3.           "           (2)	PTTOFD .....	I-2
4. 特殊紙テープをディスクに読む(1)	YAGIP1 .....	I-2
5.           "           (2)	YAGIPT .....	I-3
6.           "           (3)	YG5000 .....	I-3
7. 紙テープをMCAに読む(1)	PTTOMC .....	I-3
4 0.           "           (2)	PTMC .....	2
( MCA を読む )		
8. MCAをディスクに読む(1)	MCFD .....	I-3
9.           "           (2)	WRDISK .....	I-4
4 1.           "           (3)	WRDIS1 .....	2
4 2. MCAをコアに読む	MCME .....	2
( 印刷する, 紙テープに書く )		
1 0. コアを印刷する(1)	MEPT .....	I-4
4 3.           "           (2)	TABLEA .....	2
1 1. ディスクを印刷する(1)	FDPT .....	I-4
1 2.           "           (2)	FDPT1 .....	I-5
4 4.           "           (3)	FDPT2 .....	2
1 3.           "           (4)	TYPOUT .....	I-5
4 5. MCAを印刷する	MCPT .....	3
( MCA に書く )		
1 4. ディスクからMCAへ(1)	FDMC .....	I-5
1 5.           "           (2)	REDISK .....	I-5
4 6. コアからMCAへ	MEMC .....	3
( ディスク内情報の操作 )		
4 7. ディスクをフォーマットする	FORMAT .....	3
4 8. ディスクを自動コピーする	AUTOCY .....	3

1 6.	データをディスク間で移し換える	FDFD	I-6
1 7.	ディスク内データを加え合わせる	MIXF	I-6
1 8.	コメントを書換える	COMME	I-6
1 9.	データ・ファイル名を変更する	RENAMD	I-7
2 0.	ファイルを消去する(1)	SCRA	I-7
2 1.	"          (2)	SCRA1	I-7
2 2.	1ブロックあたりのデータ長を変える	TRNDA	I-8
4 9.	データ・ファイルを調査する	INFO	3
5 0.	データ・ファイルの内容を探る(1)	SEAR	4
5 1.	"          (2)	SEAR0	4
5 2.	ファイルを分割する	SEPARA	4
5 3.	データ・ファイルを修正する(1)	CORR	4
5 4.	"          (2)	CORR0	4

(コアとディスク間のやりとり)

2 3.	コアからディスクへ	MEFD	I-8
2 4.	ディスクからコアへ	FDME	I-8
2 5.	プログラムを更新する(1)	SCSA	I-9
2 6.	"          (2)	SCSA1	I-9
5 5.	データ・ファイルを更新する	SCSA2	5

(磁気テープ・ワーク)

5 6.	磁気テープ・ワークを開始する(1)	MT	5
5 7.	"          (2)	MT1	5
5 8.	磁気テープをオープンする(1)	OPEN	5
5 9.	"          (2)	OPEN2	5
6 0.	磁気テープからコアへ	READ	6
6 1.	磁気テープからMCAへ	READ1	6
6 2.	コアから磁気テープへ	WRITE	6
6 3.	MCAから磁気テープへ	WRITE1	6
6 4.	ファイルを閉じる	CLOSE	6
6 5.	ダイレクトリをみる	FILES	7
6 6.	磁気テープのラベル付け	LABELS	7
6 7.	ファイルを消去する	RID	7
6 8.	ディスクから磁気テープへ(1)	MT2	7
6 9.	"          (2)	MT3	8



## ( 会計データの処理 )

94. 整数型ファイルを作る	CRFILI .....	13
95. 文字型ファイルを作る	CRFILS .....	14
96. データを入力する(1)	WRFINA .....	14
97.       "           (2)	WRFINB .....	14
98. 品名を変更する	CHFINA .....	14
99. 請求番号をクリアする	CLFINA .....	15
100. ファイルを印刷する(1)	REFINA .....	15
101.       "           (2)	REFINB .....	15
102. グループごとに集計する	REFINC .....	15

## ( カレンダー )

103. カレンダーを作る(1)	CALEN, CALEN1, CALEN2, CALEN3, CALEN4, CALEN5 .....	15
104.       "           (2)	TDAY, TDAY1 .....	16
105.       "           (3)	YOBI, YOBI1 .....	16

## ( 基本ルーチン )

106. 空白を書込む	PADS .....	16
107. 剰余を求める	MOD .....	16
108. MCA内のデータを間引く	SCR .....	17
109. 素数を求める	PRIM, PRIM1, PRIM2 .....	17
110. 多項式を最小2乗法であてはめる	LSTSQR .....	17
111. 連立方程式の係数を読み込む	YOME .....	18
112. 連立1次方程式を解く	RENRIIT .....	18
113. 多項式を評価する	POLYNO .....	18
114. 表を書く	HYO .....	18

## ( ORACLの修正 )

115. 大きく修正する	PATCH .....	19
116. 小さく修正する	EDITT .....	19
117. ブロックトランスファー	FD2500 .....	19
118. 8進数を入力する	OCTIN .....	19

## ( その他 )

35. コア上に領域をとる(1)	BUFA .....	I-12
36.       "           (2)	BUFB .....	I-12
37. スtringの字数を数える(1)	STNU .....	I-12



38. スtringの字数を数える(2)	STNU1 .....	I-12
39. 花文字を描く(1)	FL .....	19
119. " (2)	FL2 .....	19
120. 文字列を登録する	F.ARRY .....	20
121. 数値→文字型データ	INST .....	20
122. 文字型データ→整数値	STIN .....	20
123. 文字型データ→整・実数値	STRE .....	21
124. I版をJ版に変える	MAKEJ .....	21
125. J版をI版に戻す	MAKEI .....	21
3. Micro-8 サポートシステム .....		22
M1. 紙テープソースを読み込む	PTRE .....	23
M2. 1行を64文字に整える	PAK .....	24
M3. ソース・プログラムをキーインする	ENTR .....	24
M4. ソース・ファイルを編集する	ARNG .....	25
M5. ソース・ファイルに追加する	SEDT .....	26
M6. ソーステープを作る	TRCE .....	27
M7. 指定した文字を置換える	REPL .....	28
M8. ソースリストを作る	SLST .....	28
4. 関数の用途, 使い方		
( 数学的関数 )		
F1. 平方根	SQRT .....	30
F2. 指数関数	EXP .....	30
F3. 対数関数	LOG .....	30
F4. 正弦関数	SIN .....	30
F5. 余弦関数	COS .....	30
F6. 正接関数	ATAN .....	30
F7. 絶対値	ABS .....	30
F8. 整数化	INT .....	30
F9. 符号	SIGN .....	30
( 乱数発生 )		
F10. 一様乱数	RANDOM .....	30
( 論理関数 )		
F11. 否定	NOT .....	30
F12. 論理積	AND .....	31

F 1 3. 論 理 和	OR .....	3 1
( 配列演算関数 )		
F 1 4. 配列要素の最大値	GETMAX .....	3 1
F 1 5. 配列要素の最小値	GETMIN .....	3 2
F 1 6. 配列の和	MATADD .....	3 2
F 1 7. 配列の差	MATSUB .....	3 2
F 1 8. 配列に定数を入れる	MATCON .....	3 2
F 1 9. 配列を移す	MATEQ .....	3 3
F 2 0. 逆行列と行列式	MATINV .....	3 3
F 2 1. 配列要素の最大値	MATMAX .....	3 3
F 2 2. 配列要素の最小値	MATMIN .....	3 3
F 2 3. 配列をスカラー倍する	MATSCA .....	3 3
F 2 4. 配列の積分	INTEGR .....	3 4
F 2 5. 配列の差分	DIFFER .....	3 4
( コード変換 )		
F 2 6. 符 号 化	ENCODE .....	3 4
F 2 7. 解 読	DECODE .....	3 5
F 2 8. ASCII ⇄ EBCDIC	BICHAR .....	3 5
F 2 9. 8 進 化	OCTAL .....	3 5
F 3 0. 2 進文字の入出力	FCHR .....	3 5
( スtring移換え関数 )		
F 3 1. 移し換え	MOVE\$ .....	3 6
( 作表関数 )		
F 3 2. キャリジ位置ぎめ関数	TAB .....	3 6
( ディスク操作関数 )		
F 3 3. ディスクの初期設定	INITFD .....	3 6
F 3 4. ディスクのコピー	COPYFD .....	3 7
F 3 5. ファイル作成	CREATE .....	3 7
F 3 6. ファイル結合	LINK .....	3 7
F 3 7. データ転送(1)	XFER .....	3 7
F 3 8. " (2)	TRANSF .....	3 7

(位置ぎめ関数)

F 3 9.	ディスク・ファイル位置ぎめ	SPY .....	38
F 4 0.	メモリ位置ぎめ	EXIST .....	38

(MCA関数)

F 4 1.	MCA関数	MCA .....	38
--------	-------	-----------	----

(UNIBUS関数)

F 4 2.	UNIBUS関数	UNIBUS .....	38
--------	----------	--------------	----

(磁気テープ関数)

F 4 3.	磁気テープドライバ関数	MTIO .....	38
--------	-------------	------------	----

(CAMAC関数)

F 4 4.	CAMAC関数	CAMAC .....	39
--------	---------	-------------	----

(LAMプロセサの起動)

F 4 5.	ラムプロセサ起動	CAMINT .....	39
--------	----------	--------------	----

(ROI(region of interest)関数)

F 4 6.	ROI関数	ROI .....	39
--------	-------	-----------	----

(時計の関数)

F 4 7.	時計	CLOCK .....	39
--------	----	-------------	----

5. プログラム・リスティング(アルファベット順)

41

プログラム名(含関数)	用途, 使い方のページ	リスティングのページ
ABS (関数) .....	30	
ACCUMU .....	I-9	I-14
AND (関数) .....	31	
ARNG .....	25	41
ATAN (関数) .....	30	
AUTOCY .....	3	42
BICHAR(関数) .....	35	
BUFA .....	I-12	I-14
BUFB .....	I-12	I-15
BUNCH .....	11	42
CALEN .....	15	42

CALEN1 .....	15 .....	43
CALEN2 .....	15 .....	43
CALEN3 .....	15 .....	43
CALEN4 .....	15 .....	44
CALEN5 .....	15 .....	44
CAMAC (関数) .....	39	
CAMINT ( " ) .....	39	
CHF INA .....	14 .....	45
CLF INA .....	15 .....	45
CLOCK (関数) .....	39	
CLOSE .....	6	
COMME .....	I-6	
COPYFD (関数) .....	37	
CORR .....	4 .....	46
CORR0 .....	4 .....	46
COS (関数) .....	30	
COUNT .....	8 .....	46
CREATE (関数) .....	37	
CRFILI .....	13 .....	47
CRFILS .....	14 .....	47
DECODE (関数) .....	35	
DEL .....	13 .....	47
DELAY .....	9 .....	48
DIFFER (関数) .....	34	
DISPLA .....	8 .....	48
DRHIST .....	10 .....	48
EDITT .....	19 .....	48
ENCODE (関数) .....	34	
ENTR .....	24 .....	49
ERSPHA .....	8 .....	49
ESTIMA .....	12 .....	49
EXIST (関数) .....	38	
EXP ( " ) .....	30	
FCHR ( " ) .....	35	
FD2500 .....	19 .....	50
FDFD .....	I-6	
FDMC .....	2	
FDME .....	I-8	

FDPT .....	I-4	
FDPT1 .....	I-5	
FDPT2 .....	2	50
FILES .....	7	
FL .....	19	51
FL2 .....	19	52
FORMAT .....	3	52
F.ARRY .....	20	53
G76050 .....		54
GEOEFF .....	11	55
GETMAX (関数) .....	31	
GETMIN ( " ) .....	32	
HISTO .....	10	55
HYO .....	18	56
IN .....	13	56
INFO .....	3	56
INITFD (関数) .....	36	
INST .....	20	57
INT (関数) .....	30	
INTEGR ( " ) .....	34	
LABELS .....	7	
LINE1 .....	13	57
LINE2 .....	13	58
LINK (関数) .....	37	
LOG ( " ) .....	30	
LSTSQR .....	17	58
MAKEI .....	21	58
MAKEJ .....	21	58
MARKER .....	9	59
MATADD (関数) .....	32	
MATCON (関数) .....	32	
MATEQ ( " ) .....	33	
MATINV ( " ) .....	33	
MATMAX ( " ) .....	33	
MATMIN ( " ) .....	33	
MATSCA ( " ) .....	33	
MATSUB ( " ) .....	32	
MAXMIN .....	I-11	

MCA	(関数)	38	
MCFD		I-3	
MCME		2	59
MCPT		3	59
MEA104		9	60
MEAN		I-9	
MEAN1		9	60
MEFD		I-8	
MEMC		3	61
MEPT		I-4	
MIXF		I-6	
MOD		16	61
MOVES	(関数)	36	
MT		5	
MT1		5	61
MT2		7	62
MT3		8	63
MTIO	(関数)	38	
NOT	( " )	30	
OCTAL	( " )	35	
OCTIN		19	64
OPEN		5	
OPEN2		5	64
OR	(関数)	31	
PADS		16	64
PAK		24	65
PARABO		11	66
PATCH		19	67
PEAK		10	67
PEAK1		I-11	
PEAK2		10	68
PEAREA		I-12	
POLYNO		18	68
PRIM		17	69
PRIM1		17	70
PRIM2		17	71
PTFD		I-2	
PTMC		2	71

PTME .....	I-2	
PTRE .....	23	72
PTTOFD .....	I-2	
PTTOMC .....	I-3	
RANDOM (関数) .....	30	
READ .....	6	
READ1 .....	6	72
REDISK .....	I-5	
REFINA .....	15	73
REFINB .....	15	73
REFINC .....	15	74
REMEAN .....	I-10	
RENAMD .....	I-7	
RENRI T .....	18	74
REPL .....	28	75
RID .....	7	75
ROI (関数) .....	39	
SCR .....	17	75
SCRA .....	I-7	
SCRA1 .....	I-7	
SCSA .....	I-9	
SCSA1 .....	I-9	
SCSA2 .....	5	76
SEAR .....	4	76
SEAR0 .....	4	76
SEDT .....	26	77
SEPARA .....	4	78
SIGN (関数) .....	30	
SIN ( " ) .....	30	
SLST .....	28	78
SMI (関数) .....		
SPY .....	38	
SQRT (関数) .....	30	
STAIR .....	9	79
STIN .....	20	79
STNU .....	I-12	
STNU1 .....	I-12	
STPPHA .....	8	79

STRE .....	21 .....	80
TAB (関数) .....	36	
TABLEA .....	2 .....	81
TABLEW .....	12 .....	81
TABLEX .....	12 .....	81
TABLEY .....	12 .....	82
TDAY .....	16 .....	82
TDAY1 .....	16 .....	82
TRANSF (関数) .....	37	
TRCE .....	27 .....	83
TRNDA .....	I-8	
TYP OUT .....	I-5	
UNIBUS (関数) .....	38	
W .....	12 .....	83
WR .....	13 .....	84
WRDIS1 .....	2 .....	84
WRDISK .....	I-4	
WRFINA .....	14 .....	85
WRFINB .....	14 .....	86
WRITE .....	6	
WRITE1 .....	6 .....	86
XFER (関数) .....	37	
YAGIP1 .....	I-2	
YAGIPT .....	I-3	
YG5000 .....	I-3	
YOBI .....	16 .....	87
YOBI1 .....	16 .....	87
YOME .....	18 .....	87



## 1. まえがき

著者たちは、さきにこの報告と同名の報告書（第1集と呼ぶことにする）を刊行した（JAERI-M 6499, 1976年3月）。この報告は、その後、著者たちが作成したプログラムの使い方とプログラム・リスティングを中心にまとめたものである。

第1集では、著者たちの作成したプログラム39編について述べた。この第2集では、新たに製作した67編のプログラムに、第1集に掲載したもので、その後、変更を行なったもの2編を加えて、その用途、使い方、リスティングを掲載する。

ORTEC社で製作されたもので、著者たちが入手したプログラムのうち、放射線測定および核物質の非破壊測定に必要なもの19編も合わせて集録することにした。そのうち、磁気テープ操作に関係する8編については、製作者の利益を損わないようにするために、リスティングは割愛し、用途および使い方のみを載せることにした。第2章の目次では、第1集に載せたプログラムを含めて全125項目について、内容別に分類し、索引の便をはかった。

著者たちは、ガンマ線スペクトル測定センタのために、マイクロプロセッサINTEL8080を用いたマイクロコンピュータMicro-8の開発を行なっている。このソフトウェア製作に必要なサポートシステムの一部をORACLをメタ言語として作成した。その用途、使い方、リスティングもこの報告書に収めることにした（第3章）。

現在では、ORACLについての初心者用操作説明書「ORACCL Primer」が、ORTEC社から刊行されており、操作法の学習は一段と便利になった。この第2集の編集が、あらかた終るまで、著者達は、それを利用出来る立場にいなかった。そして、この第2集に、ORACL関数類の用途および使い方を集録することにしていて、プライマには、多くの関数の使い方がくわしく書かれている。しかし、かなり多用される関数でも、そこに説明されていないものもある。したがって、この第2集では、予定どおり、すべての関数について、用途および使用法を一応掲載し、プライマに説明があるものは、その旨を記して記述を簡単にした。

第5章の目次は、関数名とプログラム名をすべて含めてアルファベット順に配列したものにし、プログラムについては、リスティング掲載ページを付した。この目次は、リスティングのページ数を索引するため、および、新たにプログラムを作る場合、その名称が、関数名およびすでに作られているプログラムの名称と衝突することを避けるため、に利用できると期待される。

著者たちが、ORACLを使用し始めたときには、その言語は第I版（Version I）であった。そして、現在では第J版に改訂されている。第J版では、まえの版になかったいくつかの関数が追加されている。このプログラム集では、第J版によることにした。したがって、第J版に含まれるすべての関数が引用されている。第I版にあった関数で、第J版では消えている関数が13個ある。これらについては、ここで名称を記載しておくことにとどめる。すなわち、FETCH, IREP, MATCH, PLOT, RECORD, REPLAC, REPLAX, SIGBC, SM, SPECS, SUBVEC, SWAPMA, XXXである。

## 2. プログラムの用途, 使い方

## 40. 紙テープをMCAに読む(2)

PTMC

(用途)

紙テープまたはキーボードより波高分析器の指定したチャンネル領域にデータを読み込む。

(使い方)

PTMC (スタートチャンネル, データ個数) とキーインする。データをキーボードより入力する場合はデータの区切りとしてCR (キャリッジリターン), LF (ラインフィード), 空白 ( ), コンマ ( , ) のいずれかを使う。データは整数形を用いる。実数形であっても小数点以下は無視される。

チャンネル数が4095を越える場合は0チャンネルへ戻る。

## 41. MCAをディスクに読む(3)

WRDIS1

(用途)

MCAの内容をコメントをつけて, 機番1のディスクに書き込む。

(使い方)

WRDIS1とキーインする。

## 42. MCAをコアに読む

MCME

(用途)

波高分析器のデータについて指定されたチャンネル領域をコアに移す。

(使い方)

MCME (スタートチャンネル, データ個数) とキーインする。データはコア上の共通領域A (整数形) に移る。

## 43. コアを印刷する(2)

TABLEA

(用途)

ある表に修正をほどこしタイプアウトする。

(使い方)

TABLEAとキーインする。

## 43. ディスクを印刷する(3)

FDPT2

(用途)

ディスクにあるコメント付データのデータ部分を紙テープにするときに用いる。

(使い方)

FDPT2(“ファイル名”, 長さ, 1トラックあたりのレコード数, “データタイプ”, 機番, 1行あたりのデータ数, 1行あけまでのデータ数)とキーインする。

#### 45. MCAを印刷する

##### MCPT

(用途)

波高分析器のデータについて指定されたチャンネル領域を印刷する。

(使い方)

MCPT(スタートチャンネル, データ個数)とキーインする。印刷されるチャンネル数が4095を越える場合は0チャンネルへ戻る。

#### 46. コアからMCAへ

##### MEMC

(用途)

コア上のデータを波高分析器の指定したチャンネル領域に読む。

(使い方)

MEMC(スタートチャンネル, データ個数)とキーインする。データはコアの共通領域A(整数形)にあらかじめ確保されていなければならない。

#### 47. ディスクをフォーマットする(供給プログラム)

##### FORMAT

(用途)

新しいディスクをFORMATする。これによってディスクが書込可能となる。

(使い方)

ディスクを機番1の場所に入れたのち, “INIT”スイッチを“UP”位置にセットし, FORMATとキーインして使い始める。機番を開いてくるので1と答える。

これが終わったあとで, CLEAR FD1とキーインするとディスクは消去されて, 全領域が書込可能となる。

#### 48. ディスクを自動コピーする(供給プログラム)

##### AUTOCY

(用途)

ディスクをコピーする。

(使い方)

AUTOCYとキーインして使用する。タイプライタが次々と指示を与えてくれる。

#### 49. データ・ファイルを調査する(供給プログラム)

##### INFO

(用途)

CREATE機能を用いて作られたデータ・ファイルの仕様を書き出すために用いられる。

(使い方)

INFOとキーインする。データファイル名および機番を開いてくるので、それらに答えれば、ファイルの型、記録のディメンジョンおよびレコード数を書き出してくる。

## 50, 51. データ・ファイルの内容を探る(1), (2)

### SEAR, SEAR0

(用途)

指定した数値または文字がファイルの何番目にあるか知りたい時に使用する。

(使い方)

SEAR(“ファイル名”, データ個数, ブロック当りのデータの長さ, “データ形式”, ディスク番号, 求める数値または文字)とキーインする。レコード番号とチャンネル数を答えてくれる。

波高分析器のデータ専用としてSEAR0(“ファイル名”, 求める数値)が用意されている。この場合データ個数=4096, ブロック当りのデータの長さ=120, データ形式=“1”, ディスク番号=1に固定されている。またデータはMCAの0チャンネルから順次ファイルされているものとする。

## 52. ファイルを分割する

### SEPARA

(用途)

ディスク内の1個のファイルにはいつている2チャンネル繰返し測定データを分割して、2つのファイルを作る。

(使い方)

SEPARAとキーインする。

## 53, 54. データ・ファイルを修正する(1), (2)

### CORR, CORR0

(用途)

データファイルの一部を修正するのに使用される。

(使い方)

入力は引数を用いる。CORR(“ファイル名”, “データ形式”, ブロック当りのデータの長さ, ディスク番号, 修正すべき最初のチャンネル番号, 修正すべき個数)とキーインする。  
○○CHANNEL=○○-----> と指定した最初のチャンネルの現在のデータを示してくれるから任意のデータをキーインする。引続いて指定したチャンネルまで続ける。

波高分析器のデータ専用としてCORR0(“ファイル名”, 最初のチャンネル, 修正個数)が用意されている。この場合データ形式は製数形, ブロック当り120個, ディスク番号は1に固定されている。なお波高分析器のデータは最初の値が0チャンネルからファイルされるのが標

準であるので一般のデータと区別して取扱うのが望ましい。

#### 55. データ・ファイルを更新する

##### SCSA2

(用途)

コア上にあるコモン・アレイを任意のディスクに移す。ファイルの修正などに用いられる。

(使い方)

入力は引数を用いる。SCSA2(“ファイル名”, ディスク番号)とキーインする。プログラムやローカルアレイを入力した場合はエラーが生じる。

#### 56. 磁気テープ・ワークを開始する(1)(供給プログラム)

##### MT

(用途)

磁気テープについての仕事を始める場合に用いる。

(使い方)

MTとキーインする。機番およびOPERATIONについて聞いてくる。機番には0と答える。OPERATIONには、READまたはWRITEと答える。つづいて、FILE SEQUENCE NUMBERを聞いてくる。そして、プログラムOPENに引きつがれる。

#### 57. 磁気テープ・ワークを開始する(2)

##### MT1

(用途)

MCAのデータを磁気テープに書き込んだり、その逆の操作をしようとするときに用いる。

(使い方)

MTと同じ。ただし、機番は聞いて来ない。

#### 58. 磁気テープをオープンする(1)(供給プログラム)

##### OPEN

(用途)

磁気テープの書き込みまたは読出し場所を探す。

(使い方)

プログラムMTおよびMT1のなかで用いられている。

#### 59. 磁気テープをオープンする(2)

##### OPEN2

(用途)

磁気テープの書き込み開始場所を探す。

(使い方)

MT2のなかで用いている。

#### 60. 磁気テープからコアへ

##### READ

(用途)

磁気テープの記録をコア上の配列に読み出す。

(使い方)

READ (配列の大きさ, 配列名) とキーインして用いる。配列はあらかじめ共通領域に1次元でとっておく必要がある。

#### 61. 磁気テープからMCAへ

##### READ1

(用途)

磁気テープに収めた波高分布データをMCAに読み込む。

(使い方)

READ1 (LENGTH) とキーインする。LENGTHは波高分布のチャンネル数。MT1のあとで用いる。

#### 62. コアから磁気テープへ

##### WRITE

(用途)

コア上の配列を磁気テープに書き込む。

(使い方)

WRITE (配列の大きさ, 配列名) の形で用いる。配列は1次元とする。プログラムOPENのあとで用いる。

#### 63. MCAから磁気テープへ

##### WRITE1

(用途)

MCA内の波高分布データを磁気テープに書き込む。

(使い方)

WRITE1 (LENGTH) の形で用いる。LENGTHは波高分布のチャンネル数である。MT2, MT3のなかで用いている。またMT1にひきつづいて用いてもよい。

#### 64. ファイルを閉じる

##### CLOSE

(用途)

オープン・ファイルを閉じる。

( 使い方 )

CLOSEとキーインする。

( 注 意 )

UNITに機番を, DEN#に3を入れておかななくてはならない。これらはCOMMON VARIABLEとすること。

( 働 き )

エンドオブファイルマークを2個つける。

#### 65. ディレクトリをみる ( 供給プログラム )

##### FILES

( 用 途 )

磁気テープのDIRECTORYをみる。

( 使い方 )

FILESとキーインする。機番を聞いてくるので, 0をキーインすると, 磁気テープに書き込まれているファイルの名前を書き出してくる。

#### 66. 磁気テープのラベル付け ( 供給プログラム )

##### LABELS

( 用 途 )

磁気テープにデータをファイルする場合, ファイル名, データの大きさ, データ形成, コメントを書き込むために用いる。

( 使い方 )

LABELSとキーインして使い始める。

#### 67. ファイルを消去する ( 供給プログラム )

##### RID

( 用 途 )

磁気テープのなかの特定のファイルを消去する。

( 使い方 )

磁気テープ機番およびファイル・シーケンス番号を開いてくるので, それらに答えればよい。

#### 68. ディスクから磁気テープへ (1)

##### MT2

( 用 途 )

ディスクに記憶させてある連続番号のファイル名のついたデータを自動的に磁気テープに移す。

( 使い方 )

MT2とキーインして使用する。ハードウェアの設定がよいかどうか, 磁気テープ上で何番

目のファイルにするか、読み込むべきディスク上のタグワード(英数字で3文字)、データ番号、データの大きさ、くり返し数を聞いてくるので、それらに答えるとよい。

3文字のタグワードに3桁のデータ番号を続けた6文字がディスク上でのファイル名である。

## 69. ディスクから磁気テープへ(2)

### MT3

(用途)

ディスクに記憶させてある波高分布データをファイル・シーケンス番号、ファイル名、データサイズのみを入力するだけで磁気テープに移すことができる。

(使い方)

MT3とキーインしたのち、キーボード上での間に答えて、上記の3データを入力すればよい。

## 70. ディスプレーする

### DISPLA

(用途)

波高分析器をDISPLAYの状態にする。

(使い方)

DISPLA(Y)をキーインする。

## 71. データを消去する

### ERSPHA

(用途)

波高分析器の内容を消去する。

(使い方)

ERSPHAとキーインする。

## 72. 計数する

### COUNT

(用途)

波高分析器をRUNの状態にする。

(使い方)

COUNTとキーインする。

## 73. PHAを止める

### STPPHA

(用途)

波高分析器を他の状態からSTOPの状態にする。



(使い方)

STPPHAとキーインする。

#### 74. ディレイを入れる

DELAY

(用途)

波高分析器をSTOP状態にしたのち、RUN状態にする場合に、1秒程度のディレイを入れると動作が確実になる。

(使い方)

COUNTの前に必要に応じてDELAYを挿入する。

#### 75. アナログ・プロッタを較正する(1)

MARKER

(用途)

MCAに100チャンネルおきに、カウント1を書き込み、1000チャンネルおきにカウント2を書き込む。

(使い方)

MARKERとキーインする。

#### 76. アナログ・プロッタを較正する(2)

STAIR

(用途)

MCAに階段状の波高分布を書き込む。

(使い方)

STAIRとキーインする。

#### 77. 平均値・標準偏差を求める(3)

MEAN1

(用途)

度数分布の形で与えられた観測データについて、平均値、分散、標準偏差を求める。

(使い方)

MEAN1 (BASE, INC, NGRP)とキーインして使う。ここに、BASEは、独立変数の下限値、INCは独立変数の刻み、NGRPは組数である。データは共通領域の並びとして与えられているものとする。

#### 78. 2チャンネル繰り返し測定データの平均と標準偏差

MEA104

(用途)

SAM-2で測定した2チャンネルくり返し測定データについて、平均値、標準偏差、相対標準偏差を求める。

(使い方)

MEA104とキーインする。データはあらかじめディスクにSAM104というファイル名で収納してあるものとする。

### 30. ヒストグラムを描く (1)

#### HISTO

(用途)

ディスク内のデータをヒストグラム状に分析し、棒状グラフを印刷する。

(使い方)

HISTO("FNAME", NDIM, NREC, BASE, INC, NSTEP, INCY)とキーインして使用する。ここにFNAMEはファイル名、NDIMは1トラックあたりのデータの長さ、NRECはデータの長さ、BASEはx座標の下限值、INCはx座標の刻み、NSTEPはx座標のステップ数、INCYはy座標の刻みである。DRHISTを内部で使用している。

### 31. ヒストグラムを描く (2)

#### DRHIST

(用途)

ヒストグラム状のデータ( $x_i, y_i$ )を印刷し、グラフを描く。

(使い方)

あらかじめ $y_i$ をNSTEPの次元をもつ整数型並びCに用意しておく。DRHIST(BASE, INC, NSTEP, INCY)とキーインする。ここでBASEはx座標の下限值、INCはx座標の刻み、NSTEPはx座標のステップ数、INCYはy座標の刻みである。

### 79. ピーク面積を求める (1)

#### PEAK

(用途)

MCAの特定ピークのピークセンターとピーク面積を求める。

(使い方)

PEAKとキーインする。下側平坦部の下限および上限チャンネルならびに上側平坦部の下限および上限チャンネルを聞いてくるので、それらに答えるとよい。

### 80. ピーク面積を求める (3)

#### PEAK2

(用途)

MCAの特定ピークの、ピークセンターとピーク面積を求める。

(使い方)

PEAK2 ( X1, X2, X3, X4 ) とキーインする。

X1 = 下側平坦部の下限チャンネル

X2 = 下側平坦部の上限チャンネル

X3 = 上側平坦部の下限チャンネル

X4 = 上側平坦部の上限チャンネル

である。

### 8.1. 2次関数バックグラウンド

#### PARABO

(用途)

ガンマ線の全エネルギーピークの両すそに2次関数をあてはめ、結果をもとのデータとともに図形的に印刷する。

(使い方)

PARABO とキーインして使い始める。

### 8.2. MCAのデータを加え合わせる

#### BUNCH

(用途)

MCAのデータを数チャンネル(任意)ずつ順番に加え合わせて新しいデータを作る。計数値が少なくて統計誤差が大きい場合、あるいはチャンネルの分割数が多過ぎる場合等に使用して目による観察を容易にする。

(使い方)

BUNCH ( NCH, SCH, DCH, SIZ ) とキーインする。ここでNCHは1チャンネルに加えられるべきチャンネル数で任意の正の整数値である。SCHは対称とするMCAの最初のチャンネル番号、DCHは加え合された新データの入るべき最初のチャンネル番号、SIZは対象とするデータのチャンネル数である。SCHとDCHは同じ値であっても差しつえないが、旧データは失われる。またMCAのフロントパネルのSTORAGE REGIONスイッチはFULLにしておくことが望ましい。

### 8.3. 幾何学的効果を求める

#### GEOEFF

(用途)

燃料棒の放射線測定について、その幾何学的効果を求める。

(使い方)

GEOEFF とキーインし、点の数、曲線の数、聞いたのち、X, Y, C の入力を催促してくるので、それらに答えるとよい。

## 84. 濃縮度を推定する

ESTIMA

(用途)

何種類かの燃料を混合して、別の何種類から燃料を作っていた場合、後の燃料の同位体比からもとの燃料の同位体比を推定する。

(使い方)

あらかじめ共通領域の1次元配列A(N), (N>10)にデータを入れておき、ESTIMAとキーインする。

## 85. ガンマ線テーブルを印刷する (1)

TABLEW

(用途)

ガンマ線テーブルを全て印刷する。

(使い方)

TABLEWとキーインする。

## 86. ガンマ線テーブルを印刷する (2)

TABLEX

(用途)

ガンマ線エネルギーとその核種名のみを印刷する。

(使い方)

TABLEXとキーインする。

## 87. ガンマ線テーブルを印刷する (3)

TABLEY

(用途)

存在が確認されたピークについて、そのエネルギー、核種名等を印刷する。

(使い方)

TABLEYとキーインする。

## 88. ガンマ線テーブルを書換える

W

(用途)

ガンマ線テーブルの内容を書き換える。

(使い方)

Wとキーインする。書き換えるべき行番号、その他を聞いてくるので、それらに答えれば、新しい内容に書き換えられる。

## 89. ガンマ線テーブルを間引く

DEL

(用途)

ガンマ線テーブルのなかから不要のデータを削除し、行間をつめる。

(使い方)

DELとキーインする。削除すべき行番号を聞いてくるので、それに答える。

## 90. ガンマ線テーブルへの追加

IN

(用途)

ガンマ線テーブルに新しいデータを挿入する。

(使い方)

INとキーインする。間に答えれば、新しいデータとして挿入される。

## 91. ガンマ線情報を印刷する(1)

LINE1

(用途)

指定した順番目にあるガンマ線について、そのエネルギー、核種名、引用文献などすべての記録されている情報を印刷する。

(使い方)

LINE1とキーインし、LINEナンバーを聞いてくるのでそれに答えるとよい。

## 92. ガンマ線情報を印刷する(2)

LINE2

(用途)

指定した順番目にあるガンマ線について、そのエネルギーおよび核種名を印刷する。

(使い方)

LINE2とキーインし、LINEナンバーを聞いてくるのでそれに答えるとよい。

## 93. 確認ガンマ線の情報書込み

WR

(用途)

特定のガンマ線について、存在が確認されたかどうかの情報を記入する。

(使い方)

WRとキーインして使い始める。

## 94. 整数型ファイルを作る

CRFILI

## (用 途)

整数形式の50行128列のファイルをディスク上にとる。

## (使い方)

CRFILIとキーインする。ファイル名をコメントを聞いてくるので入力すればよい。コメントは第1行に書込まれる。

## 95. 文字型ファイルを作る

CRFILS

## (用 途)

ストリング形式の50行512列のファイルをディスク上にとる。

## (使い方)

CRFILSとキーインする。ファイル名とコメントを聞いてくるのでそれらに答えればよい。コメントは第1行に書込まれる。

## 96. データを入力する (1)

WRFINA

## (用 途)

会計ファイルに第1行からデータを入力するとき用いる。

## (使い方)

WRFINAとキーインする。ファイル名を聞いてくるので、文字 I で終る6文字のファイル名を入力する。

## 97. データを入力する (2)

WRFINB

## (用 途)

任意の行から始めて、次々と会計データを入力するために用いる。

## (使い方)

WRFINBとキーインする。ファイル名を聞いてくるので、文字 I で終る6文字のファイル名を入力する。次に入力し始める行番号を聞いてくる。それ以後、会計データを入力する。

## 98. 品名を変更する

CHFINA

## (用 途)

すでに入力している品名の変更に用いる。

## (使い方)

CHFINAとキーインする。入力番号と新品名を聞いてくるので、それらに答えればよい。

## 99. 請求番号をクリアする

CLFINA

(用途)

まだ入力していない行における請求番号を0にする。

(使い方)

CLFINAとキーインする。請求番号を0にする最初の行番号を聞いてくるので、それに答えるとよい。その行番号以降の請求番号が0になる。

## 100. ファイルを印刷する (1)

REFINA

(用途)

会計ファイルの内容をすべて書き出す。

(使い方)

REFINAとキーインする。ファイル名を聞いてくるので、文字Iで終る6文字のファイル名を入力するとよい。

## 101. ファイルを印刷する (2)

REFINB

(用途)

会計ファイルの内容を指定した範囲で書き出す。

(使い方)

REFINBとキーインする。ファイル名と書き出すべき範囲を問に応じて答えるとよい。

## 102. グループごとに集計する

REFINC

(用途)

会計データをグループ毎に印刷し、集計する。

(使い方)

REFINCとキーインする。あとは問に答えるとよい。

## 103. カレンダーを作る (1)

CALEN, CALEN1, CALEN2, CALEN3, CALEN4, CALEN5

(用途)

カレンダーの作成に使用される。

(使い方)

入力は引数を用いる。例えばCALEN(1976)とキーインすれば西暦1976年のカレンダーを印刷する。

CALEN1(YY, ZZ)とキーインすれば、西暦YY年からZZ年までのカレンダーを順次印刷

する。CALEN2 (YY, MM) は YY 年 MM 月の暦を印刷する。CALEN3 はこれらのプログラムのサブプログラムとして使用されている。CALEN4 は 1 行に 15 日を書込んだ暦である。15 日単位で繰り返す事象等に便利である。CALEN5 は 1 行に 30 日書込んだものである。以上のプログラムは全て年号として西暦を用いている。

#### 104. カレンダーを作る (2)

##### TDAY, TDAY1

(用途)

年 = 365.2425 日として西暦年月日を日数に換算する。CALEN, DTABL などの日数計算を含むプログラムのサブプログラムとして使用される。

(使い方)

TDAY (年, 月, 日) と入力する。共通領域 TD に年月日を日数に換算した値が得られる。うるう年も正確に計算される。TDAY1 は CALEN に使用されるサブプログラムである。

#### 105. カレンダーを作る (3)

##### YOBI, YOBI1

(用途)

西暦年月日の曜日を求める。

(使い方)

入力は引数を用いる。例えば YOBI (1976, 6, 17) とキーインすれば、1976-06-17 = THURSDAY と答えてくれる。

YOBI1 は CALEN, DTABL などに用いられるサブプログラムであり、共通領域 YO に日曜～土曜日に対応して 0～7 が入る。

#### 106. 空白を書込む

##### PADS

(用途)

STRING 配列について、最初の 0 を空白に変える。

(使い方)

PADS (STR) とキーインすればよい。STR は STRING 名である。

#### 107. 剰余を求める

##### MOD

(用途)

剰余項を求める。

(使い方)

MOD (X, Y) として用いる。関数値に  $X - Z * Y$  がはいつてくる。ただし  $Z = \text{INT}(X/Y)$  である。



## 108. MCA内のデータを間引く

SCR

(用途)

MCA内のデータが規則的に余分なデータを含んでいる場合に、それを間引く。

(使い方)

SCR(REG, INIT, STEP, REPET)とキーインして用いる。REG=1の場合は領域1(0~2047チャンネル)のデータを対象とし、結果を領域2(2048~4095チャンネル)に入れる。REG=2の場合は、領域2のデータを対象とし、結果を領域1に入れる。

## 109. 素数を求める

PRIM, PRIM1, PRIM2

(用途)

素数のファイルを作るのに使用される。その演算時間を求めることにより計算機の速度の評価の手段としても使用される。

(使い方)

PRIM("ファイル名", 個数)とキーインする。ファイルは機番1のディスクに新たに作られる。個数は60000以下であること。時間が200秒経過しさらに120個の素数が求められた時点で、開始時刻からの時間, 分, 順番, 素数の値を順次印刷してくれる。例えば1200個の素数を求めるのに必要な時間は約40分である。また60000個では156時間程必要なのでプログラムの実行はしばしば途中で打切られてしまう。このために途中から継続して実行出来るようにPRIM1, PRIM2が用意されている。PRIM1("ファイル名", 旧個数, 素数, 時間, 分)とキーインする。ここでファイルはPRIMやPRIM1によってすでに作られた機番1のディスクにあるものとする。旧個数は前回の最後に印刷された120を単位とする"順番"である。素数, 時間, 分も同じく前回の最後の値を使用する。PRIM2は60001番目から120000までの素数を求めるのに使用する。入力はPRIM1と同じである。但しPRIM1によって完成されたファイル名は"SOSU"と固定されている。またエラー番号04で停止するのでその時"SOSU"ファイルを新たに書き加えるべきファイルと交換する。これは毎回行う必要がある。なお本プログラムにより求められた素数のうちで最初の1000個までは文献値で確認されている。

## 110. 多項式を最小2乗法であてはめる

LSTSQR

(用途)

N組のデータ $(x_i, y_i)$  ( $i=1, \dots, N$ )にM-1次の多項式をあてはめる場合に正規方程式の行列を作る。

(使い方)

あらかじめ、共通領域にデータ並び $X(N')$ ,  $Y(N')$ をとっておく( $N' \geq N$ )。LSTSQR(N, M)とキーインして使う。正規方程式の行列は共通領域の並び $A(M, M)$ に、右辺の

定数項は同じく  $B(M)$  に書き込まれる。並び  $A$ ,  $B$  はあらかじめ作っておかなくてもよい。

### 111. 連立方程式の係数を読み込む

#### YOME

(用途)

3元1次連立方程式の係数を読み込む。

(使い方)

YOMEとキーインする。係数を入れるための配列  $A(3, 3)$ ,  $B(3)$  は作成しておくこと。

### 112. 連立1次方程式を解く

#### RENBIT

(用途)

連立1次方程式を解く

(使い方)

あらかじめ、共通領域に2次元並び  $A(N', N'')$  および1次元並び  $B(N''')$  をとっておき、それらに連立1次方程式の係数および右辺定数項を入れておくものとする。そして RENBIT(N) とキーインすれば、並び  $B$  に解がはいって終了する。ここで  $N$  は未知数の数で、 $N'$ ,  $N''$ ,  $N'''$  は  $\geq N$  の条件を満たしておく必要がある。

### 113. 多項式を評価する

#### POLYNO

(用途)

多項式を評価する。

(使い方)

共通領域にあらかじめ1次元並び  $B(N')$  をとっておき、それに、昇巾の順に多項式の係数を入れておくものとする ( $B(1)$  には定数項)。POLYNO(N, X) とキーインすると  $X$  を変数値とする  $N$  次の多項式の値が共通領域の変数  $S$  にはいって終了する。 $N' \geq N$  を満たす必要がある。

### 114. 表を書く

#### HYO

(用途)

連立1次方程式の係数および右辺の値を印刷する。

(使い方)

HYO(N) とキーインする。 $N$  は未知数の数である。

## 115. 大きく修正する (供給プログラム)

PATCH

(用途)

ORACLに大きな修正をほどこすときに用いる。メーカーの指示以外で用いるのは危険である。

## 116. 小さく修正する (供給プログラム)

EDITT

(用途)

ORACLに小さい修正をほどこすときに用いる。メーカーの指示以外で用いるのは危険である。

## 117. ブロックトランスファー (供給プログラム)

FD2500

(用途)

ブロック・トランスファー・ルーチンと呼ばれる。ORACLの修正のとき用いる。

## 118. 8進数を入力する (供給プログラム)

OCTIN

(用途)

8進整数を10進整数に直す。ORACLの修正のとき用いる。

(使い方)

OCTINとキーインする。次に8進整数をキーインすれば、VAL8という変数に8進整数が10進整数に直ってはいってくる。

## 39, 119. 花文字を描く (1), (2)

FL, FL2

(用途)

0~9および空白より構成される任意の長さの数値を一文字が5×5の\*で示されるマトリクスで表示する。

印刷の見出しや注目個所の表示等に使用される。

(使い方)

入力は引数を用いる。FL(スペース, "数値")とキーインする。スペースは各文字間の間隔であり通常は2を用いる。数値は空白を含めた任意の長さの数字列である。テレタイプ一行に印刷できない場合は一行以上にわたって印刷される。

FL2は4桁の数値を用紙の中央に印刷したい場合に使用される。FL2(数値)の様に使用する。主にカレンダーのプログラムで用いられている。

例

\* FL(2, "1976 8 7")

```

*      *      *      *      *      *      *
*      *      *      *      *      *      *
*      *      *      *      *      *      *
*      *      *      *      *      *      *
*      *      *      *      *      *      *
*      *      *      *      *      *      *

```

## 120. 文字例を登録する

F. ARBY

(用途)

プログラムの中で使用される文字や記号のうちで一般的なものはすでに出来ていれば便利である。この様なファイルを作成するのに使用される。

(使い方)

F. ARBYとキーインすればNAME=と聞いてくるので名前を答える。名前が2つ以上の場合は、で区切る。全部の名前が必要な時はALLと答えるだけでよい。各名前はディスク0にSTRING COMMONとして確保される。1976年6月現在次の名前が登録されている。

NU1 : 0 1 2 3 4 5 6 7 8 9の文字を作る。

NU2 : 00~99の100個の文字を作る。

NU3 : 000~999の1000個の文字を作る。

WE : SUNDAY~SATURDAYの7個の文字を作る。

WE0 : SUN MON TUE WED THE FRI SATの文字を作る。

LINE : ----- (72個)を作る。

FX : 花文字FL用で、2次元で表わされる11ヶの文字を作る。

## 121. 数値→文字型データ

INST

(用途)

整数形の数値を文字形に変える。減衰表などの作表で任意の位置に数値を印刷することができる。主にサブプログラムとして使用される。

(使い方)

INST(数値)とキーインする。共通領域ISTに文字型に変換された数値が得られる。数値は整数形を対象としているが小数点付の実数でも差しつかえない。ただし小数点以下の部分は無視される。指数形の実数、符号(±)及び0~9以外の記号が入力された場合ERRORは発生しないが誤った結果が得られる。

## 122. 文字型データ→整数値

STIN

(用途)

文字型で表現された整数型のデータを元の数値に戻す。ASK命令でキーボードより入力された数値の処理などに使用される。

(使い方)

STIN(ストリング)とキーインする。共通領域SINTに整数化された数値が得られる。土と0~9以外の記号が入力されても無視され、その記号が無いものとみなされる。

### 123. 文字型データ→整・実数値

#### STRE

(用途)

文字型で表現された実数型および指数型のデータを元の数値に戻す。出力データの作表やキーボードからの入力の処理に使用される。

(使い方)

STRE(ストリング)とキーインする。共通領域R0に実数化された数値が得られる。小数点(.), 指数化記号(E), 符号(+, -), 数字(0~9)以外の記号が入力されても無視されその記号が無いものとみなされる。指数型の場合仮数部は整数型であってもさしつかえない。指数部に小数点があると小数点以下は切捨てられて整数型で指数部の計算が行われる。

### 124. I版をJ版に変える(供給プログラム)

#### MAKEJ

(用途)

第I版のORACLで書き込んだディスクを第J版のシステム・プログラムで読むために用いる。

(使い方)

第J版のシステム・プログラムのはいったディスクを機番0の場所に、第I版で書いたディスクを機番1の場所に入れ、MAKEJとキーインすると、第I版で書いたディスクを第J番で読めるように変更される。これを復元するにはMAKEIを用いる。

### 125. J版をI版に戻す(供給プログラム)

#### MAKEI

(用途)

第J版で読めるように変更された第I版で書かれたディスクを復元するために用いる。

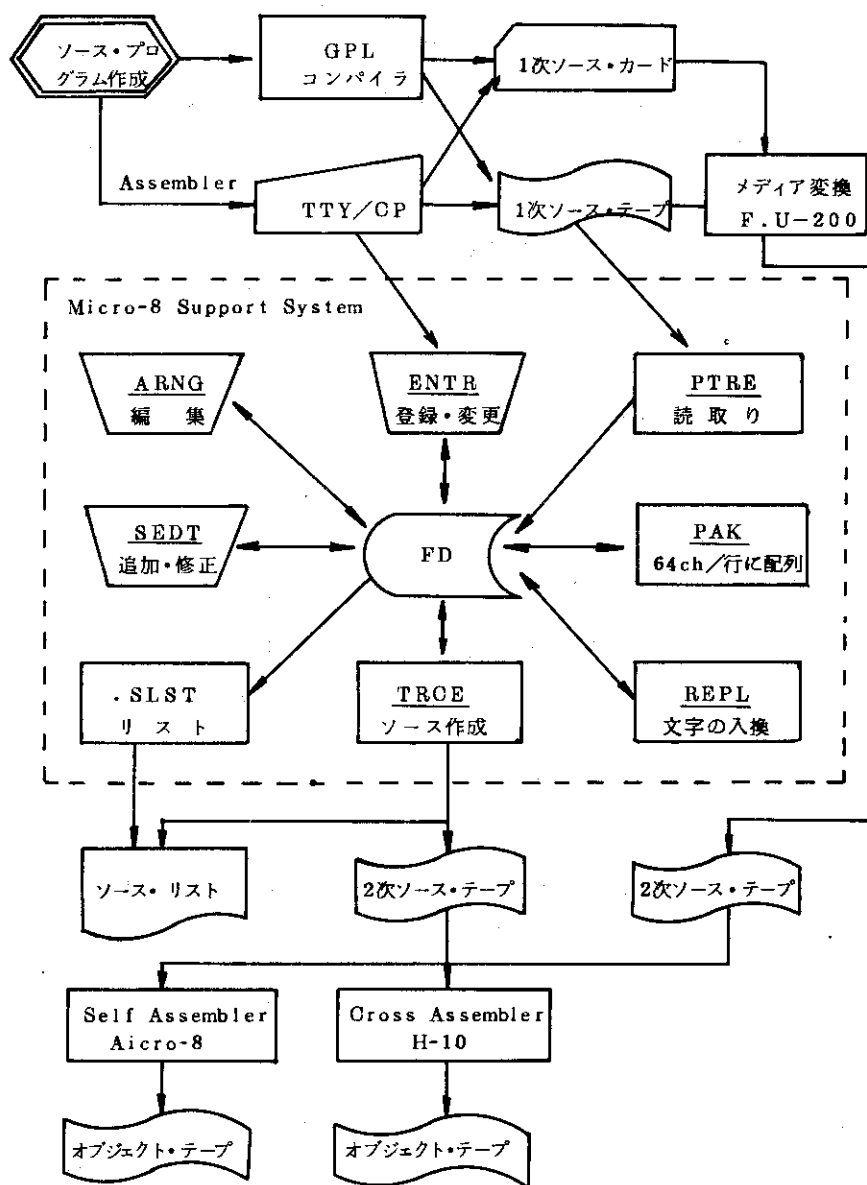
(使い方)

MAKEIとキーインすればよい。

### 3. Micro-8 サポート・システム

「ガンマ線スペクトル測定センタ」の端末システムに使用する、マイクロコンピュータ (Micro-8) のシステム・ソフトウェア製作を能率良く進めるために、ORACLを用いて、サポート・システムを作った。

第1図にマイクロコンピュータのアセンブラー・オブジェクト・テープ作成の流れ図を示す。点線で囲んだ部分がORACLによるMicro-8サポート・システムであり8個のプログラムによって構成されている。FDはデータの記録、保存用に使用されるフロッピーディスクであり、約4000行(64文字/行)のアッセンブラー・ソース・プログラムを収納できる。サポート・システムにおけるファイル形式は文字形式("S")で、1ブロック当り512文字(Byte)に固定している。1ブロックには、1行を64文字として8行収納される。またプログラムは全て対話形になっているので、特にファイルの手順を知らなくても使用できる。



第1図 Micro-8 プログラム作成手順の流れ図

## M 1. 紙テープソースを読み込む

P T R E

(用途)

紙テープ・ソースをディスクに読み込む。

(使い方)

P T R Eとキーインする。ファイル名とデータの長さ(個数)を聞いてくるので答える。ファイルはディスク1に新たに登録される。同じ名前のファイルがディスク1にすでに存在するとエラーが生じる。データの長さは一般に不明な場合が多いが大き目に予想しておく。そして紙テープの読取りが終わってから、キーボードより511個以上の任意の文字をキーインして強制打ち切り(CTRL, C)とする。ファイルの終りの方の不要な部分はARNGやSEDTによって捨てる。

```

*PTRE
FILE NAME ? M8·IO
LENGTH    ? 500

/
SMSKRG=  FFFF
SCAL01=  2000
SCAL02=  2003
SCAL03=  2006
/
/  INTERRUPT HANDLING PROCESS
/
/  .....
/  .
/  . LEVEL 0 PART 1/2
/  .
/  .   RESET START OR MONITER CALL
/  .
/  .....
/  .   ORG      0
/
SLEVL0:  NOP
          NOP
          NOP
          NOP
          DI      /  DISABLE INTERRUPT FLAG
          JMP     SMONIT /  JUMP ISTEM INITAL PROCESS
/
/  .....
/  .
/  . LEVEL 1 PART 1/2
/  .
/  .   BASIC I/O 1, 2
/  .
/  .....
/
SLEVL1:  PUSH   A
          PUSH   B
-----
*-----

```

第2図 P T R Eの使用例

## M2. 1行を64文字に整える

PAK

(用途)

紙テープソースから作ったファイル、あるいは修正したソースファイルをCRからCRまでを一行(64文字)に整列させる。

(使い方)

PAKとキーインする。新ファイル名以下4項目にわたって聞いてくるのでそれらに順次答える。

「PAKED FILE SIZE (LINE)?」には、SOURCEに使われているCRの数以上を答えればよい。実行すると、SOURCEのBLOCK NUMBERを「S. BL. #○○○」の様に打ち続いて、CR毎に1L, 2L, 3L……とLINE数を打ち8Lになると、PACKED FILEのBLOCK NUMBERを「P. BL. #○○○」と打つ。「PACKED FILE FULL!」または「SOURCE EMPTY!」で終了する。

```
*CALL PAK
*PAK
NEW FILE NAME FOR PAK      ? TEST
PACKED FILE SIZE (LINE)    ? 15
SOURCE FILE NAME           ? M8-IO
SOURCE FILE SIZE (BLOCK)   ? 4

S.BL.#    2    1L 2L 3L 4L 5L 6L 7L 8L P.BL.#    2 1L 2L
3L 4L 5L 6L 7L P.BL.#    3

PACKED FILE FULL !, PAK END !
```

## 第3図 PAKの使用例

## M3. ソース・プログラムをキーインする

ENTR

(用途)

テレタイプキーボードから入力して、アセンブラ・ソース・プログラムを作る。一行毎の変更が許される。既に登録されているSOURCE FILEを一行単位で変更する場合にも有効である。

(使い方)

ENTRとキーインする。「NEW OR OLD?」と聞いてくるので新しく登録したい時には N と答える。先に登録してあるFILEを変更したい場合には O または単に CR を押す。「FILE NAME?」, 「MAX LINE #?」に答えると“##”を打って入力待ちとなる。MAX LINE #以下のLINE #を打ってスペースを入れ続きに新しいメッセージを入力する。一行はCRによって終了する。ENTRを終了させるためには、MAX LINE #よりも大きい数を入力する。使用例を参照。

ENTRによるOLD FILEの変更は一行単位である。削除は/CRによってできるが行の挿入はできない。



```

*ENTR

NEW OR OLD ? O
OLD FILE NAME ? ROM001
MAX LINE # ? 500

##151 /
##152 /
##252 /
##251 /
##253 /
##254 /
##255 /
##256 /
##269          JZ          ·-5
##269          JZ          ·-5      /  READY
##340          JC          ·+7
##341          JNZ         ·+4
##506

```

ENTR END !!

第4図 ENTRの使用例

## M 4. ソース・ファイルを編集する

ARRNG

(用途)

ディスク上のソース・プログラムファイルから必要な部分を取り出して、新しいソースファイルに編集する。

(使い方)

ARRNGをキーインすると、「NEW OR OLD ?」と聞いてくる。新しくファイルを作る場合には N と答える。「FILE NAME」と「MAX LINE」とに答えた後、編集すべきファイルの更新したい最初の「LINE」を答える。次に「SOURCE FILE NAME」を聞いてくるがこのファイルはARRANGEするファイルと同じであってもよい。指定したLINEの内容が移された後に「CONTINUE ? Y OR N」と聞いてくるので前の続きに書き込みたい場合には Y と答える。 N または単に CR をキーインすると終了する。

ARNG

```

NEW OR OLD ? N
NEW FILE NAME TO ARENGE ? ROM001
MAX LINE # ? 500
INIT.L.# ON AREN. FILE ? 1
SOURCE FILE NAME      ? ROM002
INITIAL LINE # ON SOURCE ? 1
FINAL LINE # ON SOURCE ? 104
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
 91 92 93 94 95 96 97 98 99 100 101 102 103 104
CONTINUE ? Y OR N ? Y
SOURCE FILE NAME      ? ROM002
INITIAL LINE # ON SOURCE ? 110
FINAL LINE # ON SOURCE ? 216
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107
CONTINUE ? Y OR N ? N

ARNG END !!
*
```

## 第5図 ARNGの使用例

## M 5. ソース・ファイルを追加する

SEDT

(用途)

ソース・ファイルに行単位で任意の位置に内容を追加挿入する。

(使い方)

SEDTとキーインする。ソースファイルの名前とブロック数および挿入したい行番号を順次聞いてくるので答える。次に「##○○○」で追加したい内容を答える。CRキーを押す毎に行番号は1増加するので必要なだけ続ける。終了したい時はCTRLキーを押しながらEキーを押す。この時の行はファイルには追加されない。

ディスクのファイリングの関係で挿入行は8行単位で追加される。8の倍数に満たないで終了した場合は、その差の数だけ／の行が追加される。例えば##10行から##12行まで追加し##13行でCTRL, Eキーで終了した場合、新ソース・ファイルの1~9行は旧ファイルと同じ、10~12行は追加した内容、13~17行までは各行に／、18行以降に旧ファイルの10行目からの内容が順次移る。ディスク上のファイルのブロック数(大きさ)は必要なだけ増加している。

なお追加行数が40に達すると本プログラムは自動的に終了する。

```

* SEDT

EDIT FILE NAME ? TEST1
BLOCK SIZE      ? 5
EDIT LINE #     ? 3
**03 A
**04 BB
**05 CCC
**06 DDDD
**07 EEEEE
**08 FFFF
**09 GGG
**10 HH
**11 I
**12 E
*

```

## 第6図 SEDTの使用例

## M6. ソーステープを作る

TRCE

## (用途)

ディスクに編集された、アッセンブラー・ソース・プログラムから、ソース・リストあるいはソース・テープを作製する。

## (使い方)

TRCEとキーインする。「LIST OR TAPE ?」と聞いてくるので L または T と答える。続いて「FILE NAME ?」, 「INITIAL LINE # ?」, 「FINAL LINE # ?」に答える。ここで「TAPE」を指定してあった場合には「PTP SW ON ?」と聞いてくるので、PTPのスイッチをONにしてから Y をキーインすると、70文字のフィードを頭に付けてソース・テープが出力される。ソース・テープの最後には、アッセンブラーの終了コマンド「END」が付けられる。

```

*TRCE
/ LIST OR TAPE ? L
/ FILE NAME      ? TEST
/ INITIAL LINE # ? 1
/ FINAL LINE    * ? 16
/ ORIGIN =
ORG= /
1 /
2 SMSKRG= FFFF
3 SCAL01= 2000
4 SCAL02= 2003
5 SCAL03= 2006
6 /
7 / INTERRUPT HANDLING PROCESS
8 /
9 / *****
10 / *
11 / * LEVEL 0 PART 1/2
12 / *
13 / *      RESET START OR MONITER CALL
14 / *
15 / *****
16 /
      END

```

## 第7図 TRCEの使用例

## M7. 指定した文字を置換える

REPL

(用途)

ソース・ファイルの任意の文字を別の文字と入れ換える。

(使い方)

REPLとキーインする。ファイルの名前とブロック数を聞いてくるので答える。ブロック数はDIRECT FD1によって示されるブロック数である。つぎに訂正する文字(OLD CHARACTER)およびそれと入れ変わる文字(NEW CHARACTER)を1文字ずつ答える。

該当する文字が見つかった場合その行番号を示してくれる。一行に2個以上あるとその数だけ同じ行番号を印刷する。

```
*REPL
FILE NAME      ? TEST
BLOCK SIZE     ? 4
OLD CHARACTER  ? .
NEW CHARACTER  ? *
      9      9      9      9      9
      10
      11
      12
      13
      14
      15     15     15     15     15
*
```

第8図 REPLの使用例

## M8. ソースリストを作る

SLST

(用途)

ソース・ファイルの任意の行から指定した行までを行番号付で印刷する。

(使い方)

SLSTとキーインする。ファイル名, 印刷する始めの行番号, 終りの行番号を聞いてくるのでそれらに順次答える。

行数が64以上の時は64行毎に点線とページ数, 空白が入りA4版の綴りとして保存できる。もし巻物としてのリストのみが欲しい場合はプログラムTROEを使用すればよい。

\*SLST

FILE NAME       ? TEST  
INITIAL LINE # ? 1  
FINAL LINE #   ? 16

```
1 /  
2 SMSKRG= FFFF  
3 SCAL01= 2000  
4 SCAL02= 2003  
5 SCAL03= 2006  
6 /  
7 / INTERRUPT HANDLING PROCESS  
8 /  
9 / *****  
10 / *  
11 / * LEVEL 0 PART 1/2  
12 / *  
13 / *       RESET START OR MONITER CALL  
14 / *  
15 / *****  
16 /
```

第9図   SLSTの使用例

## 4. 関数の用途, 使い方

## F 1. 平方根

S Q R T

(用途)

平方根を求める。

(使い方)

Xを引数とするとき,  $SQRT(X)$ と書く。

## F 2~F 9

F 1に準ずる。ただし, 対数関数はeを底とするものであり, 三角関数および逆三角関数は, 弧度法による。そのほかはFORTRANと同じ用法をしてよいと思われる。

## F 10. 一様乱数

R A N D O M

(用途)

区間(0, 1)の一様乱数を発生する。

(使い方)

RANDOM(0)で, 次々と乱数が得られる。RANDOM(1)で初期設定される。

## F 11. 否定

N O T

(用途)

論理的否定関数である。

(使い方)

NOT(X)の形で使用する。Xは実または整数である。

(働き)

引数の整数部分の2進表現における論理的否定(すなわち1の補数)を10進表現に直した値を関数値とする。

例, 

TYPE NOT(3.2)
-4.0000

3.2の整数部分は3。これは2進法では000……011。

この論理的否定は 111……100。

そして, これは2進法では-4を表わす。

## F 1 2. 論理積

AND

(用途)

論理積関数である。

(使い方)

AND ( A , B ) の形で用いる。A , B は実または整変数とする。

(働き)

引数の整数部の2進表現における論理積を10進数に直した値を関数値とする。

例	TYPE AND ( 3.9 , 5.2 )
	1.0000

3.2の整数部3は2進法で 000.....011

5.2の整数部5は2進法で 000.....101

2の2数の論理積は 000.....001

これは10進法で1である。

## F 1 3. 論理和

OR

(用途)

(包含的)論理和関数である。

(使い方)

OR ( A , B ) の形で用いられる。A , B は実または整の変数とする。

(働き)

引数の整数部の2進表現における論理和を10進表現に直した値を関数値とする。

例	TYPE OR ( 3.9 , 5.2 )
	7.0000

3.2の整数部3は2進法で 000.....011

5.2の整数部5は2進法で 000.....101

2の2数の論理和は 000.....111

これは10進法で7である。

## F 1 5. 並びの最小値

## GETMAX

(用途)

並びの最大値を求める。

(使い方)

1次元並びAがある場合、GETMAX ( A ) は最大値のはいつている配列要素の引数を与

える。

#### F 1 5. 並びの最小値

##### GETMIN

(用途)

並びの最小値を求める。

(使い方)

1次元並びAがある場合、GETMIN(A)は最小値のはいつている配列要素の引数を与える。

#### F 1 6. 配列の和

##### MATADD

(用途)

配列の和を求める。

(使い方)

MATADD(A, B)の形で用いる。ただしA, Bは同次元をもつ配列とする。

(働き)

配列Aに配列和A+Bの結果がはいる。

#### F 1 7. 配列の差

##### MATSUB

(用途)

配列の差を求める。

(使い方)

MATSUB(A, B)の形で用いる。ただし, A, Bは同次元をもつ配列とする。

(働き)

配列Aに配列A-Bの結果がはいる。

#### F 1 8. 配列に定数を入れる

##### MATCON

(用途)

配列に定数を入れる。

(使い方)

入れるべき配列をAとし、定数を収めてある変数名をXとするとき、MATCON(A, X)の形で用いる。



## F 1 9. 配列を移す

M A T E Q

(用 途)

配列 B の内容を配列 A に入れる。

(使い方)

M A T E Q ( A , B ) の形で用いる。A と B は同じ次元でなければならない。

## F 2 0. 逆行列と行列式

M A T I N V

(用 途)

逆行列および行列式の値を求める。

(使い方)

実の 2 次元行列 A について、M A T I N V ( N , A ) とすれば、A は逆行列になり、関数値として 2 行列式の値が来る。N は次元数である。

## F 2 1. 配列要素の最大値

M A T M A X

(用 途)

配列要素の最大値を求める。

(使い方)

実または整の配列 A について、最大の配列要素の値を関数値として出力する。M A T M A X ( A ) の形で用いる。

## F 2 2. 配列要素の最小値

M A T M I N

(用 途)

配列要素の最小値を求める。

(使い方)

実または整の配列 A の配列要素の最小値を求める場合、M A T M I N ( A ) の形で用いると、関数値に求めるものが出力する。

## F 2 3. 配列をスカラー倍する

M A T S C A

(用 途)

配列の各要素に定数を乗じる。

(使い方)

実または整の配列 A の各要素に定数 X を乗じる場合は、M A T S C A ( A , X ) の形で用いる。

## F 2.4. 配列の積分

INTEGR

(用途)

1次元配列について、積分配列を作る。

(使い方)

1次元配列をAとするとき、INTEGR(A)の形で用いる。結果は、

```
FOR I = 2, N
  A(I) = A(I - 1) + A(I)
NEXT I
```

を実行したのと同じである。

## F 2.5. 配列の差分

DIFFER

(用途)

1次元配列について、差分をとる。

(使い方)

1次元配列をAとするとき、DIFFER(A)の形で用いる。結果は、

```
FOR I = N, -1, 2
  A(I) = A(I) - A(I - 1)
NEXT I
```

を実行したのと同じである。

## F 2.6. 符号化

ENCODE

(用途)

数式の値を指定するフォーマット・コントロールで変換してストリングに書込むために用いる。

(使い方)

ENCODE((EXPRESSION), (STRING VARIABLE) (, %(FORMAT CONTROL)))の形で用いる。フォーマット・コントロールはTYPEコマンドのそれと同じ規則によるものであり、これを指定しなければ、現に働いている出力形式で代行される。

(働き)

数式の値が計算され、ストリングに変換される(先行するゼロは圧縮され、正数にはスペース、負数にはマイナス記号がつけられる)。そして指定されたストリング変数に記憶される。

(注意)

ストリング変数が関数変換で生じるテキスト・ストリングを保持するうえで十分な大きさをもっていないければ予測できない結果が起る。つねにテキスト・ストリングに十分なテキスト・スペースが用いられるようにしておくべきである。もして形式指定よりも値が大きければ、そ

の数は“E”形式に変換される。これには15文字が必要である。

例

```
* STRING RESULT(30)
* ENCODE(3.14159*3, RESULT, %8.04)
* TYPE RESULT, " ", %8.04, 3.14159*3)
          9.4248      9.4248
```

## F 2 7. 解 読

### DECODE

(用 途)

ストリング変数に書き込まれているテキストを数値に変換する。

(使い方)

DECODE(A)の形で用いる。Aはストリング変数である。

例

```
* STRING VALUE(30)
* VALUE = " 3.14159"
* X = DECODE(VALUE)
* TYPE %8.04, X
          3.1416
*
```

## F 2 8. ASCII ⇄ EBCDIC

### BICHAR

(用 途)

ASCIIストリングをEBCDICストリングに変換する。またはその逆。

(使い方)

BICHAR(MODE, STRING)の形で用いる。MODE ≥ 0ならEBCDICからASCIIへ。MODE < 0ならASCIIからEBCDICへ。ストリングは1次元ストリングでなければならない。

## F 2 9. 8進化

### OCTAL

(用 途)

8進数を取扱う関数であると思われる。用法は不明。

## F 3 0. 2進文字の入出力

### FCHR

(用 途)

指定した入出力装置から直接に文字を入出力しようとするときに用いる。

## (使い方)

FCHR(引数)の形で用いる。引数はどんな数でもよい(正, 負または0)。引数が負の場合には, 指定した入力装置から8ビット文字を受入れ, 関数値としてその8ビット文字を10進表現で返す。引数が正の場合には, その引数の値を10進数として扱い, 指定されている出力装置に8進数として出力する。くわしくはORACLプライマをみよ。

## F 3 1 移し換え

MOVE\$

## (用途)

1つのストリング変数のある断片を他のストリング変数の断片に移すときに用いる。

## (使い方)

MOVE\$(OUT\$, OUTBEG, IN\$, INBEG, NCHARS)の形で用いる。OUT\$は出カストリング変数の名前。OUTBEGは出カストリング内のデータ転送が始まる文字位置。

IN\$は入カストリング変数の名前。INBEGは入カストリング内のデータ転送が始まる文字位置。NCHARSは転送すべき文字の数。同一ストリング内での転送も可能である。くわしくはORACLプライマをみよ。

## F 3 2. キャリジ位置きめ関数

TAB

## (用途)

出力装置のキャリジを1行のどの場所におくかをきめる関数である。

## (使い方)

TAB(X)の形で使用する。Xは次に印字すべき場所が先頭から何字目かを表わす。

## (働き)

現にキャリジがある場所がXより小さければその行のなかでXの位置まで移る。現にキャリジ・リターンとラインフィードが働きXの位置まで移る。くわしくはORACLプライマをみよ。

## F 3 3. ディスクの初期設定

INITFD

## (用途)

ディスクを初期設定またはフォーマットするために用いる。

## (使い方)

ディスク・ユニットのINITスイッチを上げてから, INITFD(機番)の形で用いる。このときWPスイッチが上げてあれば, そのディスクはソフトウェア・ライト・プロテクトの状態になる。そして, そのディスクに書込もうとするときには, そのスイッチを上げておかなければ書込めない。くわしくは, ORACLプライマをみよ。

## F 3 4. ディスクのコピー

COPYFD

(用途)

フロッピーディスクをコピーする。

(使い方)

ORACLプライマをみよ。

## F 3 5. ファイル作成

CREATE

(用途)

ファイルを作成する。

(使い方)

ORACLプライマをみよ。

## F 3 6. ファイル結合

LINK

(用途)

ディスク内データファイルの情報をコア上に書き込み, XFER関数を働かせる準備をする。

(使い方)

ORACLプライマをみよ。

## F 3 7. データ転送

XFER

(用途)

データをコアからディスクにあるいはその逆に転送する。

(使い方)

ORACLプライマをみよ。

## F 3 8. データ転送(2)

TRANSF

(用途)

データをコアからディスクへあるいはその逆に転送する。

(使い方)

ORACLプライマをみよ。

## F 3 9. ディスク・ファイル位置ぎめ

SPY

(用途)

プログラム、関数、変数がディスク上にあるかどうか、ある場合にはその位置を確かめる場合に用いる。

(使い方)

ORACLプライマをみよ。

## F 4 0. メモリ位置ぎめ

EXIST

(用途)

プログラム、関数、変数がコア上にあるかどうか、ある場合には、どの位置にあるかを確かめる場合に用いる。

(使い方)

ORACLプライマをみよ。

## F 4 1. MCA関数

MCA

(用途)

MCAとコアとの間でデータを転送する場合に用いる。

(使い方)

ORACLプライマをみよ。

## F 4 2. UNIBUS関数

UNIBUS

(用途)

周辺装置を制御したり、コア上の記憶を参照する場合に用いる。

(使い方)

ORACLプライマをみよ。

## F 4 3. 磁気テープドライバ関数

MTIO

(用途)

磁気テープに関する仕事をする場合に用いる。

(使い方)

STATS = MTIO (UNIT, MODE, DENSITY, FILEN, ARRAY) の形で用いる。

UNITはテープ機番である(現在の場合、0)。

MODEは実行する操作のコードである。

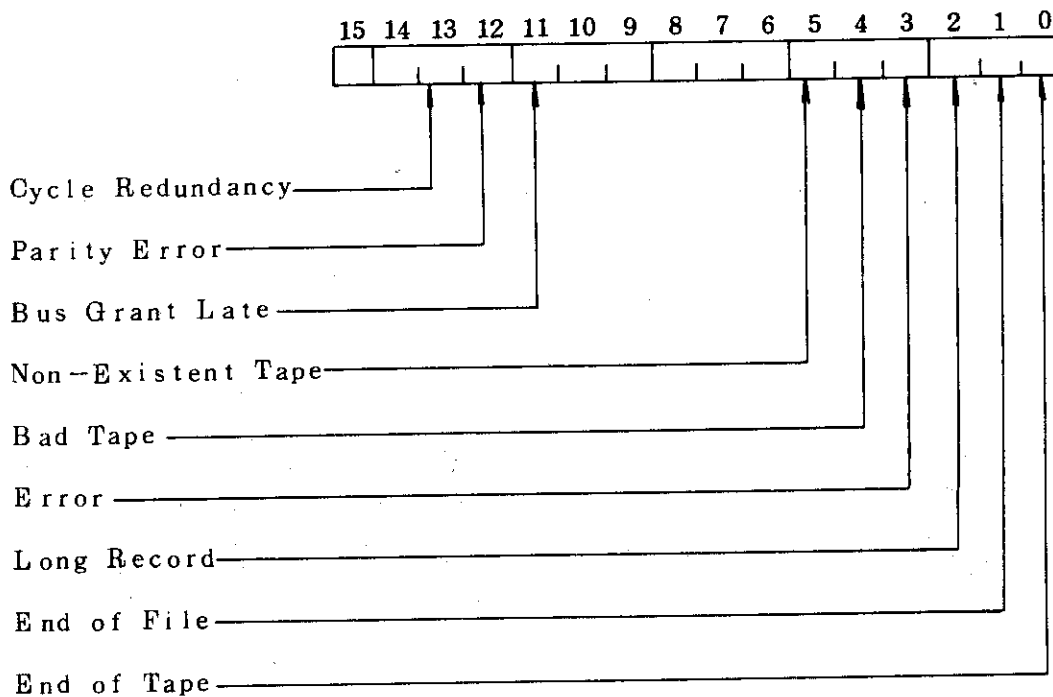
- 0 = オフライン
- 1 = 1ブロックを読む
- 2 = 1ブロックを書く
- 3 = エンドオブファイルを書く
- 4 = 前むきに FILEN 個の記録を読みとばす
- 5 = 後むきに                   "
- 6 = 1ブロックを書き、広いレコードギップをとる
- 7 = テープユニットを巻きもどす

DENSITYはテープユニット密度を表わすコードである。現在の場合、3とする(800 bpi, 9チャンネル)。FILENはスキップすべき記録の数である。

転送すべき配列 (ARRAY) がストリング配列である場合には、転送されるべきバイト数を FILEN に書かなくてはならない。こうしておけば、不適当な数のバイトを転送することが防止される。スペースについていえば、FILEN = 0 とすれば、次のエンドオブファイルにスキップされる。

ARRAYは転送されるべきブロックである。転送動作を働かさなるときには、このパラメータは何か変数であればよい。

STATUSの値には、直前に行なったテープ動作の結果を表わす一連のビットがはいっている。物理的エンドオブテープの検出はその値が1になっていることで行われる。エンドオブファイルの検出はSTATUSの値が2になっていることで行われる。バッファよりも大きな記録を転送しようとするSTATUSに4がはいる。エラーの場合には、STATUSに別のビットが立ち、どんなエラーが起きたかを知ることができる。ビットの詳細を図に示す。READまたはWRITEの間にエラーが起きた場合には、転送を完了すべく3つの試みが行われる。



## F 4 4. CAMAC関数

CAMAC

(用 途)

CAMACテストまたはデータ転送を行う。

(使い方)

ORACLプライマをみよ。

## F 4 5. ラムプロセサの起動

CAMINT

(用 途)

ORACLのLAMプロセサを起動するために用いる。

(使い方)

ORACLプライマをみよ。

## F 4 6. ROI関数

ROI

(用 途)

MCAにRegion of Interestをセットしたり、クリアしたり、あるいは調べたりするために用いる。

(使い方)

ORACLプライマをみよ。

## F 4 7. 時 計

CLOCK

(用 途)

PDP-11計算機に内蔵されている実時間時計を走らせたり、止めたり、読みとったりするために用いる。

(使い方)

CLOCK(A)の形で用いる。Aは実または整の変数または定数。

(働 き)

A &gt; 0ならば時計を走らせ、カウンタをゼロにする。

A &lt; 0ならば時計を止める。計算機が電源落ちまたは人手によって止まると時計も止まる。

A = 0にして、時計カウンタを読む。このとき時計は止まらない。値は関数値として与えられる。1カウントは電源が60Hzのとき1/60秒に、50Hzのとき1/50秒に相当する。



## 5. プログラム・リスティング

APNG

```

10 REMARK H.P.YAGY 1976-6-10
15 REMARK EDIT H.YAGY 1976-6-21
20 REMARK ARRANGEMENT FOR MICRO-8 SOURCE ON FD
90 COMMON A,B,SNAM,ANAM;ERASE A,B,SNAM,ANAM
100 STRING A(512),B(512)
110 STRING SNAM(7),ANAM(7),SLX(22),ALX(22)
120 STRING NW(5)
150 CALL LINK,XFER,CREATE,SPY
160 ASK !!, "NEW OR OLD ? ",NW
170 IF(NW(1)).NE.("N");ASK "OLD FILE NAME TO ARENGE ? ",ANAM;GOTO 230
200 ASK "NEW FILE NAME TO ARENGE ? ",ANAM
210 ASK "MAX LINE # ? ",LL;LL=INT(LL/8+1.9)
215 IF(SPY(1,ANAM)).EQ.(0);GOTO 220
216 SCRATCH FD1 @ANAM
220 CREATE(1,ANAM,"S",(512),LL)
230 LINK(1,@ANAM,ALX)
235 ASK "INIT. L.# ON AREN. FILE ? ",AA
240 AK=INT(AA/8+1.9)-1,AL=AA-(AK-1)*8,XI=(AL-1)*64+1
250 ASK "SOURCE FILE NAME ? ",SNAM
260 LINK(1,@SNAM,SLX)
270 SI=1
300 ASK "INITIAL LINE # ON SOURCE ? ",ST
310 ASK "FINAL LINE # ON SOURCE ? ",SP
320 STK=INT(ST/8+1.99),SPK=INT(SP/8+1.99)
330 STL=STK
340 SL=ST-(STK-2)*8,SI=(SL-1)*64
350 SS=1
400 XFER(1,STK,SLX,B)
440 J=0
450 FOR AI=XI,512
460 SI=SI+1,A(AI)=B(SI)
470 IF(A(AI)).NE.(142);J=J+1;IF(J).LT.(64);NEXT AI
480 J=0,AI=AL*64,SI=SL*64
490 AL=AL+1,SL=SL+1,SS=SS+1
500 IF(SL).GE.(9);SL=1,SI=0,STK=STK+1;XFER(1,STK,SLX,B)
525 TYPE %3,SS-1
530 IF(SS).GT.(SP-ST+1);XI=AI+1;GOTO 590
540 IF(AL).GE.(9);GOTO 560
550 NEXT AI
560 AL=1,AK=AK+1;XFER(-1,AK,ALX,A)
570 XI=1;GOTO 440
590 IF(AI).GE.(512);XI=1,AL=1;GOTO 600
595 FOR JI=AI+1,512;A(JI)=0;NEXT JI
600 AK=AK+1;XFER(-1,AK,ALX,A)
650 ASK !, "CONTINUE ? Y OR N ? ",SNAM
660 IF(SNAM(1)).EQ.("Y");AK=AK-1;GOTO 250
700 TYPE !!, "APNG END !!"

```

## AUTO CY

```

10 STRING DRIVE(22)
20 ASK "FD3100 OR FD2500 ? ", DRIVE
30 IF(DRIVE).EQ.("FD2500");GOTO 400
40 IF(DRIVE).NE.("FD3100");GOTO 20
50 CALL INITFP,COPYFP
60 ASK "RAISE INIT SWITCH. SEND CARRIAGE RETURN TO CONTINUE",X
70 INITFP(1);INITFP(1);COPYFP();TYPE " DO NOT REMOVE DISK YET",!;COPY
FP()
80 GOTO 600
400 CALL FORMAT,COPYFD
410 FORMAT;FORMAT;COPYFD();TYPE "DO NOT REMOVE DISK YET",!;COPYFD()
600 TYPE !!!,"COPY COMPLETE, REMOVE DISK",!!!
610 RETURN

```

## BUNCH

```

10 REMARK PREPARED BY H.P.YAGY , APR.7,1976
11 REMARK BUNCH(N CH,SOURCE CH#,DESTINATION CH#,SIZE)
20 ARG(NCH,SCH,DCH,SIZ)
30 CALL MCA
40 B=NCH*10,TY=0;TYPE %6
50 INTEGER BF(B),A(10)
80 FOR I=SCH,B,SIZ+SCH;MCA(1,I,BF(1),B);K=1
100 FOR J=1,NCH,B;FOR L=J,NCH-2+J
120 BF(J)=BF(J)+BF(L+1);NEXT L
130 A(K)=BF(J);K=K+1;NEXT J
150 MCA(-1,DCH,A(1),10);DCH=DCH+10
160 FOR II=1,10;TYPE A(II);NEXT II
190 TYPE !;TY=TY+1
200 IF(TY).GE.(10);TY=0;TYPE !
210 NEXT I

```

## CALEN

```

10 REMARK : N.TAKEUCHI 1976-02-24
12 REMARK : CALENDAR OF XX YEAR
16 REMARK : ARG(1)= YEAR(A.D.)
20 REMARK
30 ARG(YY);Y=YY
35 CALL FX,MON,WE0,LINE
40 CALL FL2,YOBI1,TDAY1,CALEN3,INST
45 COMMON LINE,WE0,MON;TYPE LINE,!!
50 FL2(YY);YY=Y
60 FOR I=1,6
70 TYPE %16,Y,MON(I)
80 TYPE %25,Y,MON(I+6),!
90 TYPE WE0,WE0,!
100 CALEN3(Y,I);TYPE !!
110 NEXT I;TYPE LINE
120 RETURN

```

## CALEN1

```

10 REMARK : N. TAKEUCHI 1976-02-27
12 REMARK : CALENDAR FOR XX--->ZZ YEAR
14 REMARK : ARG(1)= START YEAR
16 REMARK : ARG(2)= END YEAR
20 REMARK
30 ARG(MIN,MAX)
40 CALL CALEN
50 FOR I=0,INT(MAX-MIN+0.01)
60 CALEN(I+MIN);NEXT I
70 RETURN

```

## CALEN2

```

10 REMARK : N. TAKEUCHI 1976-02-24
12 REMARK : CALENDAR FOR XX YEAR ZZ MONTH
16 REMARK : ARG(1)= YEAR(A.D.)
18 REMARK : ARG(2)= MONTH
20 REMARK
30 ARG(YY,MM); DAY=1, SW=0
40 CALL MON, WE0, YOBI1, TDAY1; COMMON WE0, MON
50 TYPE !, %16, YY, MON(MM), 1, WE0, !, %3
60 YOBI1(YY,MM,1); COMMON TD, YO; L1=TD
70 TDAY1(YY,MM+1,1); COMMON TD; L1=TD-L1
80 FOR I=0, YO; TYPE " "; NEXT I
90 FOR I=YO, 6; TYPE DAY; DAY=DAY+1; NEXT I
100 TYPE !, " "; FOR K=DAY, DAY+6
110 IF(K).LE.(L1); TYPE K; NEXT K; DAY=K, SW=SW+1; GOTO 100
120 IF(SW).LE.(4); TYPE !; SW=SW+1; GOTO 120
130 RETURN

```

## CALEN3

```

10 REMARK : N. TAKEUCHI 1976-02-24
12 REMARK : MAIN SUBPROGRAM FOR CALENDAR
14 REMARK : ARG(1)= YEAR(A.D.)
16 REMARK : ARG(2)= MONTH
20 REMARK
40 ARG(YY,MM)
90 SW=0, YO1=0, L2=0, DAY2=1; TYPE %3
100 FOR K=MM, 6, MM+6; DAY=1
110 YOBI1(YY, K, 1); COMMON TD, YO; L1=TD
120 TDAY1(YY, K+1, 1); COMMON TD; L1=TD-L1
130 YO2=YO, YO=YO1, YO1=YO2, TD=L1, L1=L2, L2=TD
140 FOR I=0, YO1; TYPE " "; NEXT I
150 FOR I=YO1, 6; TYPE DAY; DAY=DAY+1; NEXT I
160 DAY1=DAY, DAY=DAY2, DAY2=DAY1; NEXT K
190 TYPE !, " "; FOR K=DAY, DAY+6
200 IF(K).LE.(L1); TYPE K; NEXT K; TYPE " "; GOTO 230
210 FOR I=1, DAY+8-K; TYPE " "; NEXT I; GOTO 230
230 DAY=DAY+7; FOR K=DAY2, DAY2+6
240 IF(K).LE.(L2); TYPE K; NEXT K
250 DAY2=DAY2+7; SW=SW+1
260 IF(SW).LT.(5); GOTO 190
270 RETURN

```

## CALEN4

```

10 REMARK : N. TAKEUCHI 1976-03-03
12 REMARK : CALENDAR OF WEEK=15 DAYS
14 REMARK
20 ARG(YY);Y=0,C=0
30 CALL TDAY,LINE,MON,ALP;COMMON LINE,MON,ALP
50 TYPE LINE,!!," "
60 FOR I=1,15;TYPE ALP(I)," ";NEXT I
70 TYPE !,%3,MON(I)
100 FOR MM=1,15;M=MM
110 IF(MM).GT.(12);M=MM-12,Y=1
120 TDAY(YY+Y,M,1);COMMON TD;L=TD
130 TDAY(YY+Y,M+1,1);COMMON TD;L=TD-L
140 FOR I=1,L;TYPE I;C=C+1
150 IF(C).EQ.(15);TYPE !," "
160 IF(C).EQ.(30);C=0;GOTO 200
170 NEXT I;NEXT MM;TYPE !,LINE;RETURN
200 IF(M).EQ.(12);M=0
210 TYPE !,MON(M+1);GOTO 170

```

## CALEN5

```

10 REMARK : N. TAKEUCHI 1976-03-03
12 REMARK : CALENDAR OF WEEK=30 DAYS
14 REMARK
30 ARG(YY);Y=YY
40 CALL INST,TDAY1,MON,LINE
50 STRING MON1(12,5),LI(62);INTEGE DA(456);MM=0,P=1
60 INST(Y);Y=YY;COMMON I ST,ISTN,LINE
70 FOR I=1,61;LI(I)=LINE(I);NEXT I;TYPE LI,!!," ",I ST
80 FOR I=ISTN+1,31;TYPE " ";NEXT I
90 INST(Y+1);Y=YY;COMMON I ST,MON;TYPE %19,Y+1,!
100 FOR M=1,15;MM=M;IF(M).GT.(12);MM=M-12
110 FOR I=1,4;MON1(MM,I)=MON(MM,I+2);NEXT I;TYPE MON1(MM);NEXT M
120 TYPE !,%3;MM=0,P=1
130 FOR MM=1,15;M=MM
140 C=0;IF(M).GT.(12);M=M-12,Y=Y+1,YY=YY+1
150 TDAY1(Y,M,1);Y=YY;COMMON TD;L=TD
160 TDAY1(Y,M+1,1);Y=YY;COMMON TD;L=TD-L
170 FOR K=P,P+L-1;C=C+1,DA(K)=C
180 IF(C).NE.(L);NEXT K
190 P=K+1;NEXT MM
210 FOR KK=1,30;FOR I=0,14;Z=I*30+KK
220 TYPE DA(Z);NEXT I;TYPE !;NEXT KK
230 TYPE !,LI,!;RETURN

```

## CHFINA

```

1  REMARK: H.GOTOH, 1976-4-16
2  REMARK: PROGRAM FOR CHANGE HINMEI IN FINANCE FILES
3  REMARK:      KEYIN "CHFINA"
7  STRING TAG(7),HIN(28),TAGG(7),YE(4)
8  TAG="FIN50I"
20 CALL TRANSF,MOVES,SPY,STOP
22 MOVES(TAGG,1,TAG,1,5);TAGG(6)="S"
25 IF(SPY(0,TAG)).EQ.(0);IF(SPY(0,TAGG)).EQ.(0);TYPE "FILES DOES NOT
EXIST !",!;STOP
40 INTEGER A(128);STRING B(512),COMMEN(128)
50 ASK "# OF CHANGE ? ",STA
52 II=INT(STA/8)
54 JJ=STA-II
70 TRANSF(0,1,TAG,II+2,A);TRANSF(0,1,TAGG,II+2,B)
72 HIN(1)=" "
75 MOVES(HIN,2,HIN,1,27)
80 MOVES(B,(JJ-1)*64+1,HIN,1,28)
90 ASK "HINMEI ? ",HIN
100 MOVES(B,(JJ-1)*64+1,HIN,1,28)
490 TRANSF(0,-1,TAG,II+2,A);TRANSF(0,-1,TAGG,II+2,B)

```

## CLFINA

```

1  REMARK: H.GOTOH, 1976-4-16
2  REMARK: PROGRAM FOR CLEAR DATA IN FINANCE FILES
3  REMARK:      KEYIN "CLFINA"
7  STRING TAG(7),HIN(28),TAGG(7),YE(4)
8  TAG="FIN50I"
20 CALL TRANSF,MOVES,SPY,STOP
22 MOVES(TAGG,1,TAG,1,5);TAGG(6)="S"
25 IF(SPY(0,TAG)).EQ.(0);IF(SPY(0,TAGG)).EQ.(0);TYPE "FILES DOES NOT
EXIST !",!;STOP
40 INTEGER A(128);STRING B(512),COMMEN(128)
50 ASK "START # OF CLEAR ? ",STA
52 II=INT(STA/8)
54 JJ=1+8*II
60 FOR I=II+2,50
70 TRANSF(0,1,TAG,I,A);TRANSF(0,1,TAGG,I,B)
80 J=1
85 IF(JJ).LT.(STA);GOTO 220
105 IF(J).GE.(114);J=1;GOTO 490
200 A(J+2)=0
220 JJ=JJ+1,J=J+16;GOTO 85
490 TRANSF(0,-1,TAG,I,A);TRANSF(0,-1,TAGG,I,B)
500 NEXT I

```

## CGPR

```

10 REMARK : N. TAKEUCHI 1976-01-28
12 REMARK : PARTIALLY CORRECT FOR FILE DATA
16 REMARK : ARG(1)= "FILE NAME"
18 REMARK : ARG(2)= "FILE TYPE"
20 REMARK : ARG(3)= DIMENSION/SECTOR
22 REMARK : ARG(4)= # OF DISK UNIT
24 REMARK : ARG(5)= INITIAL CHANNEL
26 REMARK : ARG(6)= MODIFY LENGTH
28
30 ARG(NAME, TYP, DI, EK, IN, ML)
40 CALL LINK, XFER, BUFB; BUFB(DI, TYP); COMMON B
50 STRING LX(22); LINK(EK, @NAME, LX)
60 CH=IN, N=0, K1=INT((IN/DI+1.999), K2=INT((IN+ML-1)/DI+1.999)
70 FOR K=K1, K2; XFER(1, K, LX, B); FOR I=CH, (K-1)*DI
80 TYPE I+(K-2)*DI, " CHANNEL= ", B(I), " ----> "
90 ASK B(I); N=N+1
100 IF(N).EQ.(ML); XFER(-1, K, LX, B); RETURN
110 NEXT I; XFER(-1, K, LX, B); CH=1; NEXT K
120 RETURN

```

## CORPG

```

10 REMARK : N. TAKEUCHI 1976-01-28
12 REMARK : PARTIALLY CORRECT FOR MCA FILE
16 REMARK : ARG(1)= "FILE NAME"
18 REMARK : ARG(2)= INITIAL CHANNEL
20 REMARK : ARG(3)= MODIFY LENGTH
22
30 ARG(NAME, IN, ML); DI=120, EK=1
40 CALL LINK, XFER, BUFB; BUFB(DI, "I"); COMMON B
50 STRING LX(22); LINK(EK, @NAME, LX)
60 CH=IN, N=0, K1=INT((IN+1)/DI+1.999), K2=INT((IN+ML)/DI+1.999)
70 FOR K=K1, K2; XFER(1, K, LX, B); FOR I=CH, (K-1)*DI-1
80 TYPE I+(K-2)*DI, " CHANNEL= ", B(I+1), " ----> "
90 ASK B(I+1); N=N+1
100 IF(N).EQ.(ML); XFER(-1, K, LX, B); RETURN
110 NEXT I; XFER(-1, K, LX, B); CH=0; NEXT K
120 RETURN

```

## COUNT

```

20 UNIBUS(-1, 167606, 1)

```

## CRFILI

```

1  REMARK: H. GOTOH, 1976-4-15
2  REMARK: PROGRAM FOR CREATE FINANCE FILE(INTEGER)
3  REMARK:   KEYIN "CRFILI"
7  STRING TAG(7),LX(22);INTEGER A(128)
8  ASK "FILENAME = ",TAG
10 STRING COMMEN(128);ASK "COMMENT = ",I,COMMEN
20 CALL CREATE,LINK,XFER,SPY,STOP
22 FOR I=1,128;A(I)=COMMEN(I);NEXT I
25 IF(SPY(1,TAG)).NE.(0);TYPE "FILENAME EXISTS"!;STOP
30 CREATE(0,TAG,"I",(128),50)
35 LINK(0,@TAG,LX)
37 XFER(-1,1,LX,A)
40 TYPE "FILE (" ,TAG,") CREATED !"
50 ERASE CREATE,LINK,XFER,SPY

```

## CRFILS

```

1  REMARK: H.GOTOH, 1976-4-15
2  REMARK: PROGRAM FOR CREATE FINANCE FILE(STRING)
3  REMARK:   KEYIN "CRFILS"
7  STRING TAG(7),LX(22),A(512)
8  ASK "FILENAME = ",TAG
10 ASK "COMMENT = ",I,A
20 CALL CREATE,LINK,XFER,SPY,STOP
25 IF(SPY(1,TAG)).NE.(0);TYPE "FILENAME EXISTS"!;STOP
30 CREATE(0,TAG,"S",(512),50)
35 LINK(0,@TAG,LX)
37 XFER(-1,1,LX,A)
40 TYPE "FILE (" ,TAG,") CREATED !"

```

## DEL

```

10  REMARK: DELETE A LINE FROM GAMMA-RAY TABLE
20  REMARK: PROGRAMMED BY H.GOTOH, 1976-4-7
100 COMMON FISGAM,FISGAN
200 ASK "# OF DELETE ? ",I
300 FOR J=1,99
400 FOR K=1,2;FISGAM(J,K)=FISGAM(J+1,K);NEXT K
500 FOR K=1,18;FISGAN(J,K)=FISGAN(J+1,K);NEXT K
600 NEXT J

```

DELAY

```
10 FOR I=1,500;X=1*I;NEXT I
```

DISPLA

```
20 DO UNIBUS(-1,167606,64)
```

DRHIST

```
10 REMARK PROGRAMMED BY H. GOTOH, FEB. 2, 1976
20 REMARK PROGRAM FOR DRAW HISTOGRAM
30 REMARK
70 REMARK: ARG(1)= BASE OF HISTOGRAM
80 REMARK: ARG(2)= INCREMENT OF X
90 REMARK: ARG(3)= NUMBER OF STEPS
95 REMARK: ARG(4)= INCREMENT OF Y
100 ARG(BASE, INC, NSTEP, INCY)
110 CALL TAB
120 COMMON C, NL, NG
180 TYPE !!, " COUNT RANGE CASES", !
200 NT=0; FOR I=1, NSTEP; TYPE %6, BASE+(I-1)*INC, "-", BASE+I*INC, %4, C(I)
205 TAB(INT(C(I)/INCY)+20); TYPE "*", !
210 NT=NT+C(I); NEXT I
220 NT=NT+NL+NG
230 TYPE !!, "TOTAL NUMBER OF CASES = ", NT, !
240 TYPE "NUMBER OF CASES LOWER THAN LOWER LIMIT = ", NL, !
250 TYPE "NUMBER OF CASES HIGHER THAN UPPER LIMIT = ", NG, !!
```

EDITT

```
10 STRING NAME(7)
15 CALL EXIST, TAB, OCTAL, OCTIN, UNIBUS
20 ASK "FUNCTION NAME? "NAME
30 ADDR=EXIST(NAME); IF (ADDR).EQ.(0); TYPE "NOT THERE"; RETURN
40 TAB (0); TYPE ">"; LOC=OCTIN
45 IF (LOC).LT.(0); TYPE !!; RETURN
50 LOCD=ADDR+LOC
55 TAB(15)
60 OCTAL(UNIBUS(1, LOCD)); ASK " ?", QD
70 IF (QD) 100, 40, 80
80 LOC=OCTIN; UNIBUS(-1, LOCD, LOC)
90 IF (QD).EQ.(1); GOTO 40
100 LOC=LOC+2; TAB(0); TYPE ">"; OCTAL(LOC); GOTO 50
```



ENTR

JAERI-M 6719

```
10 REMARK CODING 76-6-18 H.YAGY
20 REMARK KEYBOARD TO FD FOR MICRO-8 ASSEMBLER
100 STRING A(512),LX(22),NAME(7),NEW(7)
150 ASK !!,"NEW OR OLD ? ",NEW
160 IF(NEW(1)).EQ.("N");ASK "NEW FILE NAME ? ",NAME;GOTO 210
200 ASK "OLD FILE NAME ? ",NAME
210 ASK "MAX LINE # ? ",MAX;!!
220 KMAX=INT(MAX/8+1.9)
250 CALL LINK,CREATE,XFER
300 IF(NEW(1)).EQ.("N");CREATE(1,NAME,"S",(512),KMAX)
310 LINK(L,@NAME,LX)
320 IF(NEW(1)).NE.("N");GOTO 350
330 FOR K=1,KMAX;XFER(-1,K,LX,A);NEXT K
350 ERASE A;STRING A(512)
400 ASK "##",LL
410 IF(LL).GT.(MAX);GOTO 900
420 K=INT(LL/8+1.9)
450 XFER(1,K,LX,A)
460 L=LL-(K-2)*8
500 FOR I=(L-1)*64+1,L*64
510 ASK A(I)
520 IF(A(I)).EQ.(142);GOTO 600
530 IF(A(I)).EQ.(127);I=I-2;TYPE "\ "
550 NEXT I
560 I=I-1;IF(I).GE.(512);GOTO 610
600 FOR J=I+1,L*64;A(J)=0;NEXT J
610 XFER(-1,K,LX,A)
700 GOTO 350
900 TYPE !!," ENTR END !! ",!!
```

ERSPHA

```
20 UNIBUS(-1,167606,16)
```

ESTIMA

```
10 REMARK: H.GOTOH,1976-4-23
20 REMARK: PROGRAM FOR OBTAINING MIXING RATIO OF FUEL
100 ASK "# OF VAR ? ",NO
110 ASK "VALUE ? ",VAR
115 COMMON A,F,D,E;ERASE F,D,E
120 DIMENS COMMON F(4,3),D(4),E(4)
130 F(1,1)=.017,F(2,1)=1.997,F(3,1)=.017,F(4,1)=97.969,F(1,2)=.021,F(2,2)=2.673,F(3,2)=.002,F(4,2)=97.304
140 F(1,3)=.03,F(2,3)=3.368,F(3,3)=.009,F(4,3)=96.593
150 S=0
160 A(NO)=VAR
165 D(4)=100,E(4)=100
170 FOR I=1,3;D(4)=D(4)-A(3+I),D(I)=A(3+I),E(4)=E(4)-A(6+I),E(I)=A(6+I);NEXT I
180 FOR I=1,3
190 FOR J=1,4
200 F(J,I)=F(J,I)-A(I)*D(J)
210 F(J,I)=F(J,I)-(1-A(I))*E(J)
220 NEXT J;NEXT I
230 FOR I=1,3
240 FOR J=1,4
250 S=S+ABS(F(J,I))
260 NEXT J;NEXT I
270 TYPE %5.03;FOR I=1,9;TYPE A(I);NEXT I;TYPE %6.03,S,!
```

FD2500

```

5  REMARK - BLOCK TRANSFER ROUTINE FOR AED 2500 FLOPPY DISK
10 ARG (DRIVE, ADRES, WCOUNT, BLOCK )
20 UNIBUS(-1,164002, 0);REMARK - DISABLE DEVICE INTERRUPTS
30 UNIBUS(-1,164004,ADRES);REMARK - LOAD BUFFER ADDRESS REGISTER
40 FUN = 2*15+2*14 +2*11*DRIVE +2*5*INT(BLOCK/8)+BLOCK-8*INT(BLOCK/8)

50 XCOUNT = WCOUNT
60 IF (WCOUNT).GT.(0);FUN = FUN - 2*13;XCOUNT = -XCOUNT
70 UNIBUS (-1,164006,XCOUNT)
80 UNIBUS (-1,164000,FUN)
90 IF (UNIBUS(0,164002,128)).EQ.(0);GOTO 90
100 IF (UNIBUS(0,164002,2*15)).NE.(0);TYPE "ERROR...!";STOP
110 RETURN

```

FDPT2

```

10 REMARK : N.TAKEUCHI 1976-01-17
12 REMARK : DATA TRANSFER TO PT FROM FD
14 REMARK
16 REMARK : ARG(1)= "FILE NAME"
18 REMARK : ARG(2)= LEGTH
20 REMARK : ARG(3)= DIMENSION/SECTOR
22 REMARK : ARG(4)= "DATA TYPE" ("R","I","S")
24 REMARK : ARG(5)= # OF DISK UNIT
26 REMARK : ARG(6)= DIMENSION/HORIZONTAL
28 REMARK : ARG(7)= DIMENSION/VERTICAL
30 REMARK
40 ARG(NAME,LEG,DI,TYP,DK,HO,VE)
45 STRING LX(22)
50 CALL XFER,LINK,BUFB
60 L=INT(LEG/DI+1.99),N=0
70 LINK(DK,@NAME,LX)
80 TYPE !
85 BUFB(DI,TYP);COMMON B
87 FOR I=1,20;TYPE " " ";NEXT I;TYPE %6,!
90 FOR K=2,L;XFER(1,K,LX,B)
91 FOR I=1,DI;TYPE B(I);N=N+1
92 IF(N).GE.(LEG);GOTO 98
93 IF(N).EQ.(HO*INT(N/HO));TYPE !
94 IF(N).EQ.(VE*INT(N/VE));TYPE !
95 NEXT I;NEXT K
98 FOR I=1,20;TYPE " " ";NEXT I

```

FL

```

10 REMARK : N.TAKEUCHI 1976-01-09/1976-02-27
12 REMARK : TYPE IN FLOWER CHARACTER
16 REMARK : ARG(1)= LINE SPASE
18 REMARK : ARG(2)= "CHARACTER"
20 REMARK
30 ARG(SP, CA)
100 CALL FX;COMMON FX
110 STRING LI(73),ST(12)
120 RUN STNU(CA);COMMON STRINU
210 L1=INT((72+SP)/(5+SP));LT=INT(STRINU/L1+0.99)
220 STRING CA2(L1+1);ST="1234567890 "
230 TYPE !
240 FOR LL=1,LT
242 IF(STRINU).GT.(L1);STRINU=STRINU-L1,NN=L1;GOTO 246
244 NN=STRINU
245 FOR I=1,L1;CA2(I)=" ";NEXT I
246 FOR I=(LL-1)*L1+1,(LL-1)*L1+NN
248 CA2(I-(LL-1)*L1)=CA(I);NEXT I
250 FOR I=1,72;LI(I)=" ";NEXT I
260 FOR J=1,5;FOR K=1,L1
270 GOSUB 500
280 FOR I=1,5;LI((SP+5)*(K-1)+I)=FX(J,I,N)
290 NEXT I;NEXT K
300 GOSUB 600;NEXT J;TYPE !,;NEXT LL
310 ERASE LI,ST,FX,STRINU
320 RETURN
500 FOR II=1,9;IF(CA2(K)).EQ.(ST(II));N=II;RETURN
510 NEXT II
520 IF(CA2(K)).EQ.(ST(10));N=10;RETURN
530 REMARK : RESERVE
599 N=11;RETURN
600 SW=1;FOR I=1,72
610 IF(LI(I)).NE.(" ");SW=I
620 NEXT I;STRING LIS(SW+1)
630 FOR I=1,SW;LIS(I)=LI(I);NEXT I
640 TYPE LIS, !;ERASE LIS;RETURN

```

FL2

```

10 REMARK : N.TAKEUCHI 1976-01-09/1976-02-27
12 REMARK : SUB PROGRAM FOR CALENDAR
30 ARG(YY); SP=3
40 INST(YY); COMMON I ST, I STN, FX
45 STRING SY(I STN+3), LI(73), ST(12)
50 FOR I=1,2; SY(I)=" "; NEXT I
60 FOR I=3, I STN+2; SY(I)=I ST(I-2); NEXT I
150 STRINU=I STN+2; RENAME(SY, CA)
210 L1=INT((72+SP)/(5+SP)); LT=INT(STRINU/L1+0.99)
220 STRING CA2(L1+1); ST="1234567890 "
230 TYPE !; FOR LL=1, LT
242 IF(STRINU).GT.(L1); STRINU=STRINU-L1, NN=L1; GOTO 246
245 NN=STRINU; FOR I=1, L1; CA2(I)=" "; NEXT I
246 FOR I=(LL-1)*L1+1, (LL-1)*L1+NN
248 CA2(I-(LL-1)*L1)=CA(I); NEXT I
250 FOR I=1, 72; LI(I)=" "; NEXT I
260 FOR J=1, 5; FOR K=1, L1; GOSUB 500
280 FOR I=1, 5; LI((SP+5)*(K-1)+I)=FX(J, I, N); NEXT I; NEXT K
300 GOSUB 600; NEXT J; TYPE !!, ; NEXT LL
320 RETURN
500 FOR II=1, 9; IF(CA2(K)).EQ.(ST(II)); N=II; RETURN
510 NEXT II
520 IF(CA2(K)).EQ.(ST(10)); N=10; RETURN
530 REMARK : RESERVE
599 N=11; RETURN
600 SW=1; FOR I=1, 72
610 IF(LI(I)).NE.(" "); SW=I
620 NEXT I; STRING LIS(SW+1)
630 FOR I=1, SW; LIS(I)=LI(I); NEXT I
640 TYPE LIS, !; ERASE LIS; RETURN

```

FORMAT

```

10 ASK "WHAT UNIT NUMBER? "UNIT
20 FUN=57344+2248*UNIT
30 FOR I=0, 63
40 UNIBUS(-1, 164000, FUN+32*I)
50 IF (UNIBUS(0, 164002, 128)).EQ.(0); GOTO 50
60 NEXT I
70 TYPE !"INITIALIZATION COMPLETE...USE CLEAR COMMAND"!!

```

F.ARRY

```

10 REMARK : N.TAKEUCHI 1976-05-17
12 REMARK : MAKE COMMON ARRAY TO #0 DISK
20 STRING BUF(71),BN(20,71),NAME(7)
25 CALL SCSA2
30 ASK "NAME= ",BUF
40 K=1,SW=0;FOR I=1,70
50 IF(BUF(I)).NE(",");BN(K,I-SW)=BUF(I);NEXT I;GOTO 70
60 K=K+1,SW=1;NEXT I
70 FOR N=1,K;FOR I=1,6;NAME(I)=BN(N,I);NEXT I
99
100 REMARK : NU1(10) ; STRING COMMON
110 REMARK : 0--->9
120 IF(NAME).NE("NU1");IF(NAME).NE("ALL");GOTO 200
130 COMMON NU1;ERASE NU1;STRING COMMON NU1(11)
140 FOR I=1,10;NU1(I)=I+47;NEXT I
150 SCSA2("NU1",0);ERASE NU1
199
200 REMARK : NU2(100) : STRING COMMON
210 REMARK : 00--->99
220 IF(NAME).NE("NU2");IF(NAME).NE("ALL");GOTO 300
230 COMMON NU2;ERASE NU2;STRING COMMON NU2(100,3)
240 FOR J=0,9;FOR I=1,10;NU2(10*J+I,1)=48+J;NEXT I;NEXT J
250 FOR J=0,9;FOR I=1,10;NU2(10*J+I,2)=47+I;NEXT I;NEXT J
260 SCSA2("NU2",0);ERASE NU2
299
300 REMARK : NU3(1000) : STRING COMMON
310 REMARK : 000---->999
320 IF(NAME).NE("NU3");IF(NAME).NE("ALL");GOTO 400
330 COMMON NU3;ERASE NU3;STRING COMMON NU3(1000,4)
340 FOR K=0,9;FOR J=0,9;FOR I=1,10
350 NU3(100*K+10*J+I,1)=48+K;NEXT I;NEXT J;NEXT K
360 FOR K=0,9;FOR J=0,9;FOR I=1,10
370 NU3(100*K+10*J+I,2)=48+J;NEXT I;NEXT J;NEXT K
380 FOR K=0,9;FOR J=0,9;FOR I=1,10
390 NU3(100*K+10*J+I,3)=47+I;NEXT I;NEXT J;NEXT K
392 SCSA2("NU3",0);ERASE NU3
399
400 REMARK : WE(7) : STRING COMMON
410 REMARK : SUNDAY---->SATURDAY
420 IF(NAME).NE("WE");IF(NAME).NE("ALL");GOTO 500
430 COMMON WE;ERASE WE;STRING COMMON WE(9,10)
440 WE(1)="SUNDAY",WE(2)="MONDAY",WE(3)="TUESDAY",WE(4)="WEDNESDAY"
450 WE(5)="THURSDAY",WE(6)="FRIDAY",WE(7)="SATURDAY"
460 SCSA2("WE",0);ERASE WE
499
500 REMARK : WE0 : STRING COMMON
510 REMARK : SUN MON ---> SAT
520 IF(NAME).NE("WE0");IF(NAME).NE("ALL");GOTO 600
530 CALL WE;COMMON WE,WE0;ERASE WE0;STRING COMMON WE0(33)
540 FOR I=1,32;WE0(I)=" ";NEXT I
552 FOR K=1,7;FOR I=2,4;WE0(4*K+I)=WE(K,I-1);NEXT I;NEXT K
560 SCSA2("WE0",0);ERASE WE0
599
600 REMARK : FX(5,5,11) : STRING COMMON
601 REMARK : 0 ---> 9," "
602 IF(NAME).NE("FX");IF(NAME).NE("ALL");GOTO 700
603 COMMON FX;ERASE FX;STRING COMMON FX(5,5,11)
604 FOR K=1,11;FOR J=1,5;FOR I=1,5
605 FX(I,J,K)=128;NEXT I;NEXT J;NEXT K;GOTO 610

```

```

606 IF(I).NE.(0);GOTO 608
607 FOR N=1,5;FX(N,J,K)=132;NEXT N;RETURN
608 IF(J).NE.(0);FX(I,J,K)=132;RETURN
609 FOR N=1,5;FX(I,N,K)=132;NEXT N;RETURN
610 I=1,J=0,K=10;GOSUB 606;I=5;GOSUB 606;I=0,J=1;GOSUB 606
611 J=5;GOSUB 606
615 I=0,J=3,K=1;GOSUB 606
620 I=1,J=0,K=2;GOSUB 606;I=3;GOSUB 606;I=5;GOSUB 606
621 I=2,J=5;GOSUB 606;I=4,J=1;GOSUB 606
630 I=1,J=0,K=3;GOSUB 606;I=3;GOSUB 606;I=5;GOSUB 606
631 I=2,J=5;GOSUB 606;I=4;GOSUB 606
640 I=3,J=0,K=4;GOSUB 606;I=0,J=4;GOSUB 606
641 I=1,J=1;GOSUB 606;I=2;GOSUB 606
650 I=1,J=0,K=5;GOSUB 606;I=3;GOSUB 606;I=5;GOSUB 606
651 I=2,J=1;GOSUB 606;I=4,J=5;GOSUB 606
660 I=1,J=0,K=6;GOSUB 606;I=3;GOSUB 606;I=5;GOSUB 606
661 I=0,J=1;GOSUB 606;I=4,J=5;GOSUB 606
670 I=1,J=0,K=7;GOSUB 606;I=0,J=5;GOSUB 606;I=2,J=1;GOSUB 606
680 I=1,J=0,K=8;GOSUB 606;I=3;GOSUB 606;I=5;GOSUB 606
681 I=0,J=1;GOSUB 606;J=5;GOSUB 606
690 I=1,J=0,K=9;GOSUB 606;I=3;GOSUB 606;I=0,J=5;GOSUB 606
691 I=2,J=1;GOSUB 606
695 SCSA2("FX",0);ERASE FX
699
700 REMARK : LINE : STRING COMMON
702 REMARK : -----(72 PIECES)
710 IF(NAME).NE.("LINE");IF(NAME).NE.("ALL");GOTO 800
720 COMMON LINE;ERASE LINE;STRING COMMON LINE(73)
730 FOR I=1,72;LINE(I)="-";NEXT I
740 SCSA2("LINE",0);ERASE LINE
799
800 REMARK : MON : STRING COMMON
802 REMARK :     JAN --->     DEC
810 IF(NAME).NE.("MON");IF(NAME).NE.("ALL");GOTO 900
820 COMMON MON;ERASE MON;STRING COMMON MON(12,7)
830 MON(1)="     JAN";MON(2)="     FEB";MON(3)="     MAR"
840 MON(4)="     APR";MON(5)="     MAY";MON(6)="     JUN"
850 MON(7)="     JUL";MON(8)="     AUG";MON(9)="     SEP"
860 MON(10)="    OCT";MON(11)="    NOV";MON(12)="    DEC"
870 SCSA2("MON",0);ERASE MON
899
900 REMARK : ALP : STRING COMMON
902 REMARK : ABC--->Z
910 IF(NAME).NE.("ALP");IF(NAME).NE.("ALL");GOTO 999
920 COMMON ALP;ERASE ALP;STRING COMMON ALP(27)
930 ALP="ABCDEFGHIJKLMN O PQRSTU VWXYZ"
940 SCSA2("ALP",0);ERASE ALP
999

```

G76050

```

100 GEOEFF
110 RENRIT(4)
120 COMMON B
130 TYPE %8.04," A = ",B(1),!
140 TYPE " B1 = ",B(2),!
150 TYPE " B2 = ",B(3),!
160 TYPE " B3 = ",B(4),!!

```

## GEOEFF

```

1  REMARK: H.GOTOH,1976-5-1
2  REMARK: PROGRAM FOR GEOMETRICAL EFFECT IN FUEL ROD COUNTING
110 COMMON A,B;INTEGE I,J,M,N
120 ASK "NUMBER OF POINTS ? ",N
125 ASK " NUMBER OF CURVES ? ",M
127 FOR I=1,M+1;FOR J=1,M+1;A(I,J)=0;NEXT J;B(I)=0;NEXT I
130 FOR I=1,N
140 ASK X,Y,C,!
150 X2=X*X
160 J=INT(Y/2+1.0I),A(J+1,I)=A(J+1,I)+X2,B(J+1)=B(J+1)+C,A(I,I)=A(I,I)
+X2*X2,B(I)=B(I)+C*X2
170 A(J+1,J+1)=A(J+1,J+1)+I
210 NEXT I
220 FOR I=2,M+1;A(I,I)=A(I,I);NEXT I
280 FOR I=1,M+1;FOR J=1,M+1
290 TYPE A(I,J);NEXT J;TYPE B(I),!
300 NEXT I;TYPE !

```

## HISTO

```

10  REMARK: H.GOTOH,1976-4-21/1976-2-2
20  REMARK: PROGRAM FOR DRAW HISTOGRAM OF REPEATED COUNTING DATA
30  REMARK
40  REMARK: ARG(1)="FILE NAME"
50  REMARK: ARG(2)=DIMENS
60  REMARK: ARG(3)= NRECORDS
70  REMARK: ARG(4)= BASE OF HISTOGRAM
80  REMARK: ARG(5)= INCREMENT OF X
90  REMARK: ARG(6)= NUMBER OF STEPS
92  REMARK: ARG(7)= INCREMENT OF Y
100 ARG(FNAME,NDIM,NREC,BASE,INC,NSTEP,INCY)
110 CALL LINK,XFER,DRHIST
120 STRING COM(NDIM+1),LX(22)
125 INTEGER B(NDIM)
126 COMMON C,NL,NG;ERASE C,NL,NG
127 INTEGE COMMON C(NSTEP),NL,NG
130 LINK(1,@FNAME,LX);XFER(1,1,LX,B)
140 FOR I=1,NDIM;COM(I)=B(I);NEXT I
150 TYPE !,COM,!!
155 NL=0,NG=0;FOR I=1,NSTEP;C(I)=0;NEXT I
160 FOR I=1,NREC;FOR J=1,NDIM
165 XFER(1,I+1,LX,B)
170 IX=INT((B(J)-BASE)/INC)+1
175 IF(IX).LE.(0);NL=NL+1;GOTO 190
180 IF(IX).GT.(NSTEP);NG=NG+1;GOTO 190
185 C(IX)=C(IX)+1
190 NEXT J;NEXT I
200 DRHIST(BASE,INC,NSTEP,INCY)

```

HYO

```

100 COMMON A,B
110 ARG(N)
115 TYPE %8.03
120 FOR I=1,N
130 FOR J=1,N;TYPE A(I,J);NEXT J;TYPE B(I),!;NEXT I

```

IN

```

100 COMMON FISGAM,FISGAN
150 STRING NUC(7),REF(7),OBS(7)
200 ASK "ENERGY(KEV) ? ",ENE
250 ASK "BRANCH(<1.0) ? ",BRAN
300 ASK "NUCLIDE ? ",NUC
320 ASK "REFERENCE ? ",REF
330 ASK "OBSERVED ? ",OBS
400 FOR I=1,100
500 IF(ENE).LE.(FISGAM(I,1));GOTO 700
550 IF(FISGAM(I,1)).LE.(0);GOTO 700
600 NEXT I
700 FOR J=99,-1,I
750 IF(FISGAM(J,1)).LE.(0);GOTO 1000
800 FOR K=1,2;FISGAM(J+1,K)=FISGAM(J,K);NEXT K
900 FOR K=1,18;FISGAN(J+1,K)=FISGAN(J,K);NEXT K
1000 NEXT J
1100 FISGAM(I,1)=ENE,FISGAM(I,2)=BRAN
1200 FOR K=1,6;FISGAN(I,K)=NUC(K);NEXT K
1300 FOR K=1,6;FISGAN(I,K+6)=REF(K);NEXT K
1400 FOR K=1,6;FISGAN(I,K+12)=OBS(K);NEXT K

```

INFO

```

5 CALL LINK
10 STRING NAME(7),XLINK(22)
20 ASK "DATA FILE NAME? "NAME,"UNIT NUMBER? "UNIT
30 LINK (UNIT,@NAME,XLINK)
35 ID=XLINK(9);GOSUB 200
40 D1=XLINK(15)+256*XLINK(16)
45 D2=XLINK(17)+256*XLINK(18)
50 D3=XLINK(19)+256*XLINK(20)
60 NRCRDS=XLINK(21)+256*XLINK(22)
70 TYPE !"RECORD DIMENSIONS ARE: (",%4,D1
80 IF (D2).NE.(0);TYPE ", "D2
90 IF (D3).NE.(0);TYPE ", "D3
100 TYPE "!"
110 TYPE "AND THERE ARE"NRCRDS," RECORDS IN THE FILE"!
120 RETURN
200 TYPE "THE FILE IS "
210 IF (ID).EQ.(16);TYPE "A STRING DATA FILE";RETURN
220 IF (ID).EQ.(32);TYPE "AN INTEGER DATA FILE";RETURN
230 TYPE "A REAL DATA FILE"

```



INST

```

10 REMARK : N.TAKEUCHI 1976-02-25
12 REMARK : CONVERT STRING FROM INTEGER
16 REMARK : ARG(1)= INTEGER DATA
30 ARG(IN); S=0, SI=1; IF(IN).LT.(0); IN=ABS(IN), S=1, SI=-1
35 IN=INT(IN+1.5); C1=1, C2=1, INN=IN
40 IF(INN).GE.(10); C1=C1+1, C2=10*C2, INN=0.1*INN; GOTO 40
50 COMMON IST, ISTN; ERASE IST, ISTN
55 STRING COMMON IST(C1+1+S); INTEGER COMMON ISTN
58 IF(SI).EQ.(-1); IST(1)="-"
60 FOR I=1+S, C1+S; C2=INT(C2+0.5), C3=1
70 IF(C2*C3).LT.(IN); C3=C3+1; GOTO 70
80 IN=IN-C2*(C3-1), C2=0.1*C2
90 IST(I)=47+C3; NEXT I; ISTN=C1
100 RETURN

```

LINE1

```

100 COMMON FISGAM, FISGAN
200 ASK "# OF LINE ? ", I
300 TYPE %3, I, %7.02, FISGAM(I, 1), %8.05, FISGAM(I, 2), " "
520 FOR J=1, 6; TYPE FISGAN(I, J); NEXT J
530 TYPE " "; FOR J=7, 12; TYPE FISGAN(I, J); NEXT J
540 TYPE " "; FOR J=13, 18; TYPE FISGAN(I, J); NEXT J
550 TYPE !

```

## LINE2

```

100 COMMON FISGAM,FISGAN
200 ASK "# OF LINE ? ",I
300 TYPE %3,I,%7.02,FISGAM(I,I),"KEV("
520 FOR J=1,5;TYPE FISGAN(I,J);NEXT J;TYPE ")"
550 TYPE !

```

## LSTSQR

```

1 REMARK: H.GOTOH, 1976-5-7
2 REMARK: PROGRAM FOR LEAST SQUARES FITTING OF POLYNOMIAL
10 REMARK: LSTSQR(NUMBER OF DATA POINTS, DEGREE OF POLYNOMIAL + 1)
100 ARG(N,M)
110 COMMON X,Y,A,B;ERASE A,B;DIMENS COMMON A(M,M),B(M)
120 DIMENS U(N),V(N)
130 FOR I=1,N;U(I)=1;NEXT I
140 FOR I=1,2*M-1;S=0
150 FOR J=1,N;S=S+U(J);NEXT J
160 FOR J=1,M
170 FOR K=1,M;IF(J+K).NE.(I+1);GOTO 210
180 IF(J).GT.(M);GOTO 210
190 IF(K).GT.(M);GOTO 210
200 A(J,K)=S
210 NEXT K;NEXT J
220 FOR J=1,N;U(J)=U(J)*X(J);NEXT J;NEXT I
230 FOR I=1,N;V(I)=Y(I);NEXT I
240 FOR I=1,M;S=0;FOR J=1,N;S=S+V(J);NEXT J
250 B(I)=S;FOR J=1,N;V(J)=V(J)*X(J);NEXT J;NEXT I

```

## MAKEI

```

10 CALL FD2500,EXIST
20 STRING BUFFER(512)
30 STRING NAME(7);NAME="BUFFER",ADR=16+EXIST(NAME)
40 FD2500(1,ADR,256,2)
50 BUFFER(1)=193
60 FD2500(1,ADR,-256,2)

```

## MAKEJ

```

10 CALL FD2500,EXIST
20 STRING BUFFER(512)
30 STRING NAME(7);NAME="BUFFER",ADR=16+EXIST(NAME)
40 FD2500(1,ADR,256,2)
50 BUFFER(1)=194
60 FD2500(1,ADR,-256,2)

```

## MARKER

```

50 INTEGER A(10)
100 A(1)=1
200 FOR I=0,100,4000
300 MCA(-1,I,A(1),1)
400 NEXT I
500 A(1)=2
600 FOR I=0,1000,4000
700 MCA(-1,I,A(1),1)
800 NEXT I

```

## MCME

```

10 REMARK : N.TAKEUCHI 1976-02-06
12 REMARK : DATA TRANSFER MEMORY FROM MCA
16 REMARK : ARG(1)= INITIAL CHANNEL
18 REMARK : ARG(2)= LENGTH
20
30 ARG(IC,LEG)
40 CALL MCA,BUFA
50 BUFA(LEG,"I");COMMON A
60 MCA(1,IC,A(1),LEG)
70 RETURN

```

## MCPT

```

10 REMARK : N.TAKEUCHI 1976-02-05
12 REMARK : DATA TRANSFER PT FROM MCA
16 REMARK : ARG(1)= INITIAL CHANNEL
18 REMARK : ARG(2)= LENGTH
20
30 ARG(CH,LEG)
40 CALL MCA
50 INTEGER B(128);N=0
60 TYPE %6
100 MCA(1,IC,B(1),128)
110 FOR I=1,128;TYPE B(I);N=N+1
120 IF(N).GE.(LEG);RETURN
130 IF(N).EQ.(10*INT(N/10));TYPE !
140 IF(N).EQ.(100*INT(N/100));TYPE !
150 NEXT I;IC=IC+128;GOTO 100

```

MEAI04

```

10 REMARK: H.GOTOH, 1976-4-21
12 REMARK: MEAN AND STANDARD DEVIATION OF SAMI04
14 CALL LINK,XFER
20 STRING NAME(7),LX(22)
30 NAME="SAMI04"
40 LEG=2000
45 DI=100
50 CO=53000,C1=15000,C2=38000
60 LINK(1,@NAME,LX)
65 INTEGE A(DI)
70 R1=0,R2=0,N=0,R11=0,R22=0,R111=0,R222=0
100 FOR K=2,INT(LEG/DI+1.99)
110 XFER(1,K,LX,A)
120 FOR J=1,2,DI-1
130 M=A(J)-CO,R1=R1+M,R2=M*M+R2,N=N+1
140 M1=A(J+1)-C1,R11=R11+M1,R22=M1*M1+R22
150 M2=A(J)-A(J+1)-C2,R111=R111+M2,R222=R222+M2*M2
160 IF(N).GE.(LEG/2);GOTO 210
170 NEXT J;NEXT K
210 X1=R1/N+CO,X11=R11/N+C1,X111=R111/N+C2
220 X2=SQRT(R2/(N-1)-N*(CO-X1)*(CO-X1)/(N-1))
230 X22=SQRT(R22/(N-1)-N*(C1-X11)*(C1-X11)/(N-1))
235 X222=SQRT(R222/(N-1)-N*(C2-X111)*(C2-X111)/(N-1))
240 X3=X2/X1,X33=X22/X11,X333=X222/X111
290 TYPE "NUMBER OF DATA SETS =",N,!
300 TYPE "          MEAN VALUE =",X1,X11,X111,!
320 TYPE "STANDARD DEVIATION =",X2,X22,X222,!
340 TYPE "RELATIVE STA. DEV. =",X3,X33,X333,!!
350 STOP

```

MEAN1

```

10 REMARK: PROGRAMMED BY H. GOTOH, FEB. 2, 1976
20 REMARK: PROGRAM FOR MEAN AND STANDARD DEVIATION OF GROUPED DATA
30 REMARK
40 REMARK: ARG(1)= BASE VALUE OF INDEPENDENT VARIABLE
50 REMARK: ARG(2)= INCREMENT OF INDEPENDENT VARIABLE
60 REMARK: ARG(3)= NUMBER OF GROUPS
70 ARG(BASE,INC,NGRP)
75 COMMON C
80 FMAX=0,GMAX=0,II=20
85 CALL TAB
90 FOR I=1,NGRP
100 IF(C(I)).GT.(FMAX);FMAX=C(I),GMAX=I
110 NEXT I
120 NTOT=0,SUM=0,SUM1=0
130 FOR I=1,NGRP
140 NTOT=NTOT+C(I),SUM=SUM+(I-GMAX)*C(I),SUM1=SUM1+(I-GMAX)*2*C(I)
150 NEXT I
160 AVER=(SUM/NTOT+GMAX-.5)*INC+BASE
170 VAR=(SUM1-SUM*SUM/NTOT)*INC*INC/NTOT
180 TYPE %10.02,!, "MEAN = ";TAB(II);TYPE AVER
190 TYPE !,"VARIANCE = ";TAB(II);TYPE VAR,!
200 TYPE "STD DEVIATION = ";TAB(II);TYPE SQRT(VAR),!

```

## MEMC

```

10 REMARK : N.TAKEUCHI 1976-02-06
12 REMARK : DATA TRANSFER MCA FROM MEMORY
16 REMARK : ARG(1)= INITIAL CHANNEL
18 REMARK : ARG(2)= LENGTH
20
30 ARG(IC,LEG)
50 COMMON A
60 CALL MCA;MCA(-1,IC,A(1),LEG)
70 RETURN

```

## MOD

```

1 ARG(X,Y)
2 Z=INT(X/Y)
3 Z=X-Z*Y

```

## MT1

```

1 REMARK: 1976-3-26, H.GOTOH
20 STRING G$(7)
40 COMMON UNIT,MODE,SEQ,DEN#
60 MODE=0; DEN#=3
80 UNIT=0
100 ASK "OPERATION ? ",G$
120 IF(G$).EQ.("READ"); MODE=1; CALL READ1
140 IF(G$).EQ.("WRITE"); MODE=-1; CALL WRITE1
160 IF(MODE).EQ.(0); TYPE "ILLEGAL OPERATION!"; GOTO 100
180 ASK "FILE SEQUENCE NUMBER ? ",SEQ
200 CALL BI CHAR,MTIO,ENCODE,DECODE,AND,MOVES
220 RUN OPEN
240 RETURN

```

MT2

```

1  REMARK: 1976-3-30, H.GOTOH
2  TYPE !!,"AUTO-RECORDING IN MAG TAPE FROM DISKET",!!
3  REMARK: REVISED 1976-4-05
10  CALL SPY
20  STRING FORM(81),COMMEN(120),LX(22),D(11),Gs(7),NAMES(7),REPS(3);D=
"0123456789 "
21  COMMON A,B;ERASE A,B
22  STRING COMMON A(241),B(81)
25  INTEGER C(120),NDATA,IData,NN,I1(3),NO
30  ASK "OK OR NO ? ",REPS
35  IF (REPS).NE.("OK");STOP
40  COMMON UNIT,MODE,SEQ,DEN#
60  MODE=-1,DEN#=3
80  UNIT=0
180  ASK "INITIAL FILE SEQUENCE NUMBER ? ",SEQ
185  ASK "TAG WORD(IN JUST 3 LETTERS) ? ",NAMES
190  ASK "INITIAL # OF DATA(WITHIN 3 DIGITS) ? ",NDATA
193  ASK "DATA SIZE ? ",LEN
194  N=INT(LEN/120)
195  ASK "NUMBER OF REPETITIONS ? ",NO
200  CALL MCA,XFER,LINK,BICHAR,MTIO,ENCODE,DECODE,AND,MOVES
205  TYPE !!;RUN OPEN2
210  FOR NN=NDATA,NDATA+NO-1
212  Idata=NN,Idata=Idata-1000*INT(Idata/1000),I1(1)=INT(Idata/100),I1(
2) =INT(Idata/10)-10*I1(1),I1(3)=Idata-10*INT(Idata/10)
213  FOR J=6,-1,4;NAMES(J)=D(I1(J-3)+1);NEXT J
215  TYPE NAMES,;!;IF(SPY(1,NAMES)).EQ.(0);GOTO 480
230  LINK(1,@NAMES,LX)
235  FOR I=1,120;C(I)=D(I1);NEXT I
240  XFER(1,1,LX,C)
250  FOR I=1,120;COMMEN(I)=C(I);NEXT I;TYPE "COMMENT = ",!,COMMEN,!!
330  A(1)=" "
340  MOVES(A,2,A,1,239)
350  MOVES(A,1,NAMES,1,6)
360  ENCODE(LEN,B,%8)
370  MOVES(A,17,B,2,8)
380  PADS(FORM);MOVES(A,81,FORM,1,80)
390  BICHAR(-1,A)
400  MTIO(UNIT,2,DEN#,240,A)
410  MOVES(B,1,COMMEN,1,80)
420  PADS(COMMEN);BICHAR(-1,B)
430  MOVES(A,1,B,1,80)
432  MOVES(B,1,COMMEN,81,40)
434  BICHAR(-1,B)
436  MOVES(A,81,B,1,40)
440  A(121)=" "
450  MOVES(A,122,A,121,119)
460  MTIO(UNIT,2,DEN#,240,A)
462  FOR I=1,N;XFER(1,I+1,LX,C);MCA(-1,(I-1)*120,C(1),120);NEXT I
464  IF(LEN-N*120).EQ.(0);GOTO 470
466  XFER(1,N+2,LX,C);MCA(-1,N*120,C(1),LEN-N*120)
470  RUN WRITE1(LEN)
480  NEXT NN
490  MTIO(UNIT,3,DEN#,0,UNIT)
500  MTIO(UNIT,3,DEN#,0,UNIT)

```

MT3

```

1  REMARK: 1976-3-30, H.GOTOH
2  TYPE !!,"AUTO-RECORDING IN MAG TAPE FROM DISKET",!!
3  REMARK: REVISED 1976-4-05
10 CALL SPY
20 STRING FORM(81),COMMEN(120),LX(22),D(11),G$(7),NAMES(7),REPS(3);D=
"0123456789 "
21 COMMON A,B;ERASE A,B
22 STRING COMMON A(241),B(81)
25 INTEGER C(120),NDATA,IData,NN,I1(3),NO
30 ASK "OK OR NO ? ",REPS
35 IF (REPS).NE.("OK");STOP
40 COMMON UNIT,MODE,SEQ,DEN#
60 MODE=-1,DEN#=3
80 UNIT=0
180 ASK "FILE SEQUENCE NUMBER ? ",SEQ
185 ASK "LABEL(WITHIN 6 ALPHANUMERICAL LETTERS) ? ",NAMES
193 ASK "DATA SIZE ? ",LEN
194 N=INT(LEN/120)
200 CALL MCA,XFER,LINK,BICHAR,MTIO,ENCODE,DECODE,AND,MOVES
205 TYPE !!;RUN OPEN2
215 TYPE NAMES,!!;IF(SPY(1,NAMES)).EQ.(0);GOTO 480
230 LINK(1,@NAMES,LX)
235 FOR I=1,120;C(I)=D(11);NEXT I
240 XFER(1,1,LX,C)
250 FOR I=1,120;COMMEN(I)=C(I);NEXT I;TYPE "COMMENT = ",!,COMMEN,!!
330 A(1)=" "
340 MOVES(A,2,A,1,239)
350 MOVES(A,1,NAMES,1,6)
360 ENCODE(LEN,B,%8)
370 MOVES(A,17,B,2,8)
380 PADS(FORM);MOVES(A,81,FORM,1,80)
390 BICHAR(-1,A)
400 MTIO(UNIT,2,DEN#,240,A)
410 MOVES(B,1,COMMEN,1,80)
420 PADS(COMMEN);BICHAR(-1,B)
430 MOVES(A,1,B,1,80)
432 MOVES(B,1,COMMEN,81,40)
434 BICHAR(-1,B)
436 MOVES(A,81,B,1,40)
440 A(121)=" "
450 MOVES(A,122,A,121,119)
460 MTIO(UNIT,2,DEN#,240,A)
462 FOR I=1,N;XFER(1,I+1,LX,C);MCA(-1,(I-1)*120,C(1),120);NEXT I
464 IF(LEN-N*120).EQ.(0);GOTO 470
466 XFER(1,N+2,LX,C);MCA(-1,N*120,C(1),LEN-N*120)
470 RUN WRITE1(LEN)
490 MTIO(UNIT,3,DEN#,0,UNIT)
500 MTIO(UNIT,3,DEN#,0,UNIT)

```

## OCTIN

```
10 ASK VAL
15 VAL8=0
20 FOR I=5,-1,0
30 DIG10=INT(VAL/10+I),VAL=VAL-DIG10*10+I
35 IF (DIG10).GE.(8);TYPE "?ILLEGAL VALUE?"!"RE-INPUT: ";GOTO 10
40 VAL8=VAL8+DIG10*8+I
50 NEXT I
60 VAL8=VAL8
```

## OPEN2

```
20 COMMON UNIT,MODE,SEQ,DSIZE#,DEN#
30 CALL PADS
40 MTIO(UNIT,7,DEN#,0,SEQ)
60 J=SEQ-1
80 IF(J).LE.(0); GOTO 200
100 FOR I=1,J
120 MTIO(UNIT,4,DEN#,0,SEQ)
140 NEXT I
200 RETURN
```

## PADS

```
1 ARG (STR)
10 I=1
20 IF(STR(I)).EQ.(0); STR(I)=" "; RETURN
30 I=I+1
40 GOTO 20
```



PAK

```

9  REMARK EDIT BY H.YAGY 1976-6-13
10 REMARK : H.YAGY & N.TAKEUCHI 1976-06-03
12 REMARK : SUPPORT PROG. FOR MICRO-3
35 STRING TAG(7),TAGN(7),TYP(2),TYPN(2)
40 ASK "NEW FILE NAME FOR PAK    ? ",TAGN
42 ASK "PACKED FILE SIZE (LINE) ? ",SIZ
44 ASK "SOURCE FILE  NAME      ? ",TAG
46 ASK "SOURCE FILE SIZE (BLOCK) ? ",BL
48 TYP="S",TYPN="S"
50 STRING A(512),B(512),LX(22),LXN(22)
52 BMAX=INT(SIZ/8+1.9)
60 CALL LINK,XFER,CREATE,SPY
70 DI=512,DIN=512,DK=1,DKN=1
75 IF(SPY(1,TAGN)).EQ.(0);GOTO 30
79 SCRATCH FDI @TAGN
30 CREATE(DKN,TAGN,TYPN,(DIN),BMAX)
32 LINK(DK,@TAG,LX);LINK(DKN,@TAGN,LXN)
90 LL=2,AIN=1,J=-1,M=1,K=0
100 IF(LL).GT.(BL-1);GOTO 400
101 TYPE %3.0,!, "S.BL.#",LL," "
102 XFER(1,LL,LX,A);LL=LL+1,I=0
110 J=J+1
115 IF((M-1)*8+J).GE.(SIZ);GOTO 400
120 I=I+1,K=K+1
121 JJ=J*64+K
122 IF(I).GT.(512);J=J-1;GOTO 100
124 IF(JJ).GT.(512);J=J-1;GOTO 300
125 AIN=A(I)
126 IF(AIN).EQ.(10);K=K-1;GOTO 120
128 IF(AIN).EQ.(0);K=K-1;GOTO 120
130 B(JJ)=A(I)
140 IF(AIN).EQ.(142);K=0;TYPE %1.0,J+1,"L";GOTO 200
160 GOTO 120
200 IF(J).EQ.(7);GOTO 300
220 GOTO 110
300 M=M+1;XFER(-1,M,LXN,B);IF(M).GT.(BMAX);GOTO 350
310 ERASE B;STRING B(512)
330 J=0,K=0;TYPE %3.0," P.BL.#",M;GOTO 120
350 TYPE !!,"PACKED FILE FULL !, PAK END !",!!
360 STOP
400 M=M+1;XFER(-1,M,LXN,B)
405 TYPE %3.0," P.BL.#",M;IF(M).GE.(BMAX);GOTO 350
410 TYPE !!," SOURCE EMPTY !, PAK END !",!!

```

PARABO

```

10 REMARK: H. GOTOH, 1976-5-7
20 REMARK: PROGRAM FOR PARABOLLA FITTING TO BACKGROUND SPECTRUM
30 STRING TAG(7),LX(22),D(11),COM(9);D="01234567890"
32 STRING TY(4),TYY(4)
35 INTEGER C(120),I1(3),NN,NDATA,N,I,DATA
40 ASK !!, "TAG WORD OF FILES(IN JUST 3 LETTERS) = ", TAG
50 ASK "INITIAL # OF DATA(WITHIN 3 DIGITS) = ", NDATA
60 ASK "# OF REPETITION = ", NO
70 CALL LSTSQR,TAB,POLYNO,RENRI,T,LINK,XFER,MCA,SPY
75 ASK "DO YOU BUNCH DATA ? ", TYY
80 ASK "INITIAL CHANNEL OF RECORD = ", INIT; ASK "DATA LENGTH
   = ", LENG
85 N=INT(LENG/120)
90 ASK "LOWER LIMIT OF LOWER REGION = ", X1, "UPPER LIMIT OF LOWER REGI
ON = ", X2
100 ASK "LOWER LIMIT OF UPPER REGION = ", X3, "UPPER LIMIT OF UPPER REGI
ON = ", X4
105 ASK "DO YOU DRAW SPECTRUM ? ", TY
124 COMMON X,Y,Z,S; ERASE X,Y,Z
125 N4=X4-X1+1,N5=X4+X2-X3-X1+2,N6=X3-X2-1
126 IF(TY).EQ.("YES"); STRING W(52)
127 DIMENS COMMON X(N4),Y(N4),Z(N6)
130 FOR NN=NDATA,NDATA+NO-1
140 I,DATA=NN,I,DATA=I,DATA-1000*INT(I,DATA/1000),I1(1)=INT(I,DATA/1000),I1(
2)=INT(I,DATA/10)-10*I1(1),I1(3)=I,DATA-10*INT(I,DATA/10)
150 FOR J=6,-1,4; TAG(J)=D(I1(J-3)+1); NEXT J
155 IF(SPY(1,TAG)).EQ.(0); GOTO 235
160 LINK(1,@TAG,LX)
170 XFER(1,1,LX,C)
180 FOR I=1,8; COM(I)=C(I); NEXT I
190 IF(TY).EQ.("YES"); TYPE TAG," ", COM
200 FOR I=1,N; XFER(1,I+1,LX,C); MCA(-1,(I-1)*120+INIT,C(1),120); NEXT I
210 IF(LENG-N*120).NE.(0); XFER(1,N+2,LX,C); MCA(-1,N*120+INIT,C(1),LENG
-N*120)
212 IF(TYY).EQ.("YES"); GOSUB 600
215 MCA(1,X1,Y(1),X2-X1+1); MCA(1,X3,Y(X2-X1+2),X4-X3+1)
220 FOR I=X1,X2; X(I-X1+1)=I; NEXT I
225 FOR I=X3,X4; X(I-X3+X2-X1+2)=I; NEXT I
226 LSTSQR(N5,3)
227 RENRI(T,3)
229 MCA(1,X2+1,Z(1),X3-X2-1)
230 SS=0; FOR I=1,N6; POLYNO(3,I+X2); SS=SS+Z(I)-S; NEXT I
231 TYPE #9,SS,!
232 TAB(10); COMMON B; TYPE #8.05,"BACKGND=",B(1),"+",B(2),"*CH# +",B(3)
,"*CH# +2",!
233 IF(TY).EQ.("YES"); TYPE !; GOSUB 500
235 NEXT NN
240 TYPE "THE WORK COMPLETED"
250 STOP
500 FOR I=X1,X4; X(I-X1+1)=I; NEXT I
510 MCA(1,X1,Y(1),X4-X1+1)
515 MAX=50; FOR I=1,N4; IF(MAX).LT.(Y(I)); MAX=Y(I)
516 NEXT I
520 FOR I=1,N4
521 FOR J=1,50; W(J)=" "; NEXT J
522 K=INT(Y(I)*50/MAX); IF(K).GT.(50); K=50
523 IF(K).LT.(1); K=1
524 W(K)="*"
525 POLYNO(3,X(I)); K=INT(S*50/MAX); IF(K).GT.(50); K=50
526 IF(K).LT.(1); K=1

```

```

527 W(K)="B"
530 TYPE %4,X(I),%6,Y(I),S
540 TYPE W,!
550 NEXT I
555 TYPE !!
560 RETURN
600 FOR I=1,8;MCA(1,(I-1)*120,C(1),120)
610 FOR J=1,60;C(J)=C(2*J)+C(2*J-1);NEXT J
620 MCA(-1,(I-1)*60,C(1),60);NEXT I;RETURN

```

## PATCH

```

10 TYPE !!"PATCH PROGRAM FOR ORACL V035 J    01-AUG-75"!!
20 CALL EXIST,TAB,OCTAL,OCTIN,UNIBUS,FD2500
30 STRING BUFR(512)
40 ADR=16+EXIST("BUFR")
50 TYPE ">"
60 LOC=OCTIN
65 IF (LOC).EQ.(-1);RETURN
70 BLK=8+INT(LOC/512),OFF=ADR+LOC-INT(LOC/512)*512
80 FD2500(0,ADR,256,BLK)
90 TAB(15),OCTAL(UNIBUS(1,OFF))
100 ASK "    ",VALUE;IF (VALUE).EQ.(0);GOTO 50
105 IF (VALUE).LT.(0);GOTO 140
110 DATA=OCTIN
120 UNIBUS(-1,OFF,DATA),FD2500(0,ADR,-256,BLK)
130 IF (VALUE).NE.(2);GOTO 50
140 LOC=LOC+2;TYPE ">";OCTAL(LOC);GOTO 70

```

## PEAK

```

10 TYPE !
11 ASK "  LOWER LIMIT OF LOWER REGION=" ,K
12 ASK "  UPPER LIMIT OF LOWER REGION=" ,L
13 ASK "  LOWER LIMIT OF UPPER REGION=" ,M
14 ASK "  UPPER LIMIT OF UPPER REGION=" ,N
15 LA=L-K+1,MA=M-K+1,NA=N-K+1
17 COMMON TY,AREA
20 DIMENS  A(NA)
30 MCA(1,K,A(1),NA)
32 PA=LA,CP=A(PA)
33 FOR I=LA+1,MA;IF(A(I)).GT.(CP);PA=I,CP=A(I)
34 NEXT I;P=PA+K-1
40 CL=0;FOR I=1,LA;CL=CL+A(I);NEXT I
50 SCL=CL,A1=CL/LA,B1=-A1*(K-1)
60 FOR I=LA+1,PA;CL=CL+A(I);NEXT I
70 SCP=CL,S=(P*A(PA)-CL+B1)/(A(PA)-A1)
80 FOR I=PA+1,MA-1;CL=CL+A(I);NEXT I
90 SCM1=CL;FOR I=MA,NA;CL=CL+A(I);NEXT I
100 A3=(CL-SCM1)/(N-M+1),B3=(S+(P*A(PA)+B3-SCP)/(A(PA)
)-A3))/2
110 AREA=(A3-A1)*PB+B3-B1
115 SG=SQRT(SCM1-SCL+2*(SCL*((PB-LA-K+1)/LA)+2*(CL-SCM1)*((MA+K-2-PB)/(
N-M+1))+2)),RELSG=100*SG/AREA
120 IF(TY).EQ.("YES");TYPE %6.02;GOTO 130
125 STOP
130 TYPE !,"PEAK CENTER =",PB,%9.00
140 TYPE !,"PEAK AREA=",AREA,%5.00
150 TYPE !,"STANDARD DEVIATION=",SG,%5.02
160 TYPE !,"RELATIV ST. DEVIAT.=",RELSG,"%",!

```

## PEAK2

```

10 ARG(K,L,M,N)
12 CALL TAB
15 LA=L-K+1,MA=M-K+1,NA=N-K+1
17 COMMON TY,AREA
20 DIMENS A(NA)
30 MCA(1,K,A(1),NA)
32 PA=LA,CP=A(PA)
33 FOR I=LA+1,MA;IF(A(I)).GT.(CP);PA=I,CP=A(I)
34 NEXT I;P=PA+K-1
35 TYPE %5.!!,"CHANNEL # OF HIGHEST COUNT = ",P,!,%7;TAB(21);TYPE "C
OUNT = ",A(PA),!
40 CL=0;FOR I=1,LA;CL=CL+A(I);NEXT I
50 SCL=CL,A1=CL/LA,B1=-A1*(K-1)
60 FOR I=LA+1,PA;CL=CL+A(I);NEXT I
70 SCP=CL,S=(P*A(PA)-CL+B1)/(A(PA)-A1)
80 FOR I=PA+1,MA-1;CL=CL+A(I);NEXT I
90 SCM1=CL;FOR I=MA,NA;CL=CL+A(I);NEXT I
100 A3=(CL-SCM1)/(N-M+1),B3=SCM1-A3*(M-1),PB=(S+(P*A(PA)+B3-SCP)/(A(PA)
)-A3))/2
110 AREA=(A3-A1)*PB+B3-B1
115 SG=SQRT(SCM1-SCL+2*(SCL*((PB-LA-K+1)/LA)+2+(CL-SCM1)*((MA+K-2-PB)/
(N-M+1))+2)),RELSG=100*SG/AREA
120 TYPE %6.02,"PEAK CENTER = ",PB,!,%9,"PEAK AREA = ",AREA,!,%5,"STAN
DARD DEV = ",SG,!
125 TYPE %5.02,"REL STD DEV = ",RELSG,"%",!

```

## POLYNO

```

100 COMMON B,S
110 ARG(N,XI)
120 S=0
130 FOR I=1,N;S=S*XI+B(N-I+1);NEXT I

```

PRIM

```

17 REMARK : N.TAKEUCHI 1976-04-17
12 REMARK : 60000 PIECES OF PRIM
14 REMARK : PRIM("FILE NAME",LENGTH)
50 ARG(NAME,LEG);CLOCK(1);TC=0
60 CALL LINK,XFER,CREATE
70 INTEGE A(100),B(100);STRING LY(20)
80 L=INT(LEG/100+1.99)
90 CREATE(1,NAME,"I", (100),L);LINK(1,@NAME,LX)
100 A(1)=2,A(2)=3,A(3)=5,B(1)=2,B(2)=3,J=1,W=2,LW=1
110 J=J+4;GOSUB 200;IF(W).EQ.(0);GOTO 150
120 J=J+2;GOSUB 200;IF(W).EQ.(0);GOTO 140
130 A(W-1)=B(W-1),A(W)=B(W);GOTO 110
140 J=J+4;GOSUB 200
150 J=J+2;GOSUB 200;GOTO 140
200 P=0,LB=0
210 P=P+1,X=INT(J/A(P)+0.1)
220 IF(X).LT.(A(P));W=W+1,B(W)=J;GOTO 300
230 IF(J).EQ.(X*A(P));GOTO 310
240 IF(P).EQ.(100);P=0,LB=LB+1;XFER(1,LB,LX,A)
250 GOTO 210
300 IF(W).EQ.(100);W=0,LW=LW+1;XFER(-1,LW,LX,B);GOTO 330
310 IF(LB).EQ.(2);RETURN
320 XFER(1,0,LX,A);RETURN
330 T=CLOCK(0);IF(T).LT.(10000);GOTO 360
340 CLOCK(1);TC=TC+T,T=TC/3600
350 TYPE 02,INT(T/60)," H",T-60*INT(T/60)," M",06,100*(LW-1),J,1
360 IF(LW).NE.(L);GOTO 310
370 STOP

```

PRIM1

```

10 REMARK : N. TAKEUCHI 1976-04-17
12 REMARK : 60000 PIECES OF PRIM
14 REMARK : PRIM("FILE NAME",OLD LENGTR,PRIM, HOUR,MIN.)
20 ARG(NAME,LEG,J,H,M)
30 CALL XFER,LINK,CREATE;CLOCK(1);TYPE 26
40 INTEGER A(240),B(120);STRING LX(20)
50 IF(LEG).GE.(240);GOTO 140
60 CREATE(1,NAME,"I", (120),501);LINK(1,@NAME,LX);TC=0
70 A(1)=2,A(2)=3,A(3)=5,B(1)=2,B(2)=3,J=1,W=2,LW=1
80 J=J+4;GOSUB 200;IF(LW).EQ.(2);GOTO 120
90 J=J+2;GOSUB 200;IF(LW).EQ.(2);GOTO 110
100 A(W-1)=B(W-1),A(W)=B(W);GOTO 80
110 J=J+4;GOSUB 200;IF(LW).EQ.(3);GOTO 145
120 J=J+2;GOSUB 200;IF(LW).EQ.(3);GOTO 145
130 GOTO 110
140 LW=INT(LEG/120+1.99);LINK(1,@NAME,LX)
142 TC=H*3600+M*60,TC=TC*50
145 XFER(1,2,LX,B);FOR I=1,120;A(I)=B(I);NEXT I
150 XFER(1,3,LX,B);FOR I=1,120;A(120+I)=B(I);NEXT I
160 JJ=J-6*INT(J/6);IF(JJ).EQ.(5);GOTO 190
170 IF(JJ).NE.(1);TYPE "INPUT ERPAR: ARG(3) ?";STOP
180 J=J+4;GOSUB 200
190 J=J+2;GOSUB 200;GOTO 180
200 P=0
210 P=P+1,X=INT(J/A(P)+0.1)
220 IF(X).LT.(A(P));W=W+1,B(W)=J;GOTO 300
230 IF(J).EQ.(X*A(P));RETURN
250 GOTO 210
300 IF(W).NE.(120);RETURN
310 W=0,LW=LW+1;XFER(-1,LW,LX,B)
330 T=CLOCK(0);IF(T).LT.(10000);RETURN
340 CLOCK(1);TC=TC+T,T=TC/3000
350 TYPE INT(T/60)," H",T-60*INT(T/60)," M",120*(LW-1),J,!
360 RETURN

```

## PRIM2

```

10 REMARK : N.TAKEUCHI 1976-04-17
12 REMARK : 60000 PIECES OF PRIM FROM 60001
14 REMARK : PRIM("FILE NAME",OLD LENGTH,PRIM,HOUR,MIN)
20 ARG(NAME,LEG,J,H,M)
30 CALL XFER,LINK,CREATE;CLOCK(1);TYPE %7
40 INTEGER A(240),B(120);STRING LX(22),NA(7);NA="SOSU"
130 IF(LEG).LT.(60000);TYPE "INPUT ERROR ARG(2) ?";STOP
132 IF(LEG).NE.(120*INT(LEG/120));TYPE "INPUT ERROR ARG(3) ?";STOP
140 LW=INT((LEG-60000)/120+1.99);LINK(1,@NA,LX)
142 TC=H*3600+M*60,TC=TC*50
145 XFER(1,2,LX,B);FOR I=1,120;A(I)=B(I);NEXT I
150 XFER(1,3,LX,B);FOR I=1,120;A(I+120)=B(I);NEXT I
152 TYPE "EXCHANGE FDI DISK ---->GOTO 154 !!",ERROR
154 LINK(1,@NAME,LX)
160 JJ=J-6*INT(J/6);IF(JJ).EQ.(5);GOTO 190
170 IF(JJ).NE.(1);TYPE "INPUT ERRAR: ARG(3) ?";STOP
180 J=J+4;GOSUB 200
190 J=J+2;GOSUB 200;GOTO 180
200 R=0
210 R=R+1,X=INT(J/A(R)+0.1)
220 IF(X).LT.(A(R));W=W+1,B(W)=J;GOTO 300
230 IF(J).EQ.(X*A(R));RETURN
250 GOTO 210
300 IF(W).NE.(120);RETURN
310 W=0,LW=LW+1;XFER(-1,LW,LX,B)
330 T=CLOCK(0);IF(T).LT.(10000);RETURN
340 CLOCK(1);TC=TC+T,T=TC/3000
350 TYPE INT(T/60)," H",T-60*INT(T/60)," M",120*(LW-1)+60000,J,!
360 RETURN

```

## PTMC

```

10 REMARK : N.TAKEUCHI 1976-02-06
12 REMARK : DATA TRANSFER MCA FROM PAPER TAPE
16 REMARK ARG(1)= INNITIAL CHANNEL
18 REMARK ARG(2)= LENGTH
20
30 ARG(IC,LEG)
40 CALL XFER,LINK,MCA,STIN
50 STRING ST(10),S(2)
60 INTEGE A(128)
70 IT=1,N=0
80 COMMON SINT
100 FOR J=1,128
110 FOR I=1,10;ASK S(I);IT=S(I)
120 IF(IT).EQ.(10);GOTO 200
130 IF(IT).EQ.(142);GOTO 200
140 IF(S(1)).EQ.(" ");GOTO 200
142 IF(S(1)).EQ.(",");GOTO 200
150 ST(I)=S(I);NEXT I
200 I=10;NEXT I
210 STIN(ST);A(J)=SINT,ST(1)=0,N=N+1
220 IF(N).LT.(LEG);NEXT J;MCA(-1,IC,A(1),128);IC=IC+128;GOTO 100
230 MCA(-1,IC,A(1),J)

```

## PTRE

```

10 REMARK : N.TAKEUCHI 1976-06-06
12 REMARK : READ DATA FROM PAPER TAPE
13
20 STRING NAME(7),LX(22)
30 ASK "FILE NAME ? ",NAME
40 ASK "LENGTH ? ",LEG
50 CALL XFER,LINK,CREATE
60 L=INT(LEG/512+1.99),N=0
70 CREATE(1,NAME,"S",(512),L);LINK(1,@NAME,LX)
90 FOR K=2,L;STRING B(512);FOR I=1,512
92 ASK B(I);N=N+1
94 IF(N).GE.(LEG);XFER(-1,K,LX,B);GOTO 99
96 NEXT I
98 XFER(-1,K,LX,B);FRASE B;NEXT K
99 RETURN

```

## READI

```

1 REMARK: H.GOTOH, 1976-3-30
10 ARG(SIZE0)
20 STRING A(241),B(81)
30 INTEGER ARRAY(120)
40 COMMON UNIT,DEN#
80 SIZE=120
90 FOR L=1,120,SIZE0
100 FOR I=1,30,SIZE
110 RETRY=0
120 X=MTIO(UNIT,1,DEN#,240,A)
140 IF(AND(X,1)).NE.(0); TYPE "END OF TAPE"!; RETURN
160 IF(AND(X,2)).NE.(0); TYPE "END OF FILE"!; RETURN
170 IF(AND(X,-256)).NE.(0); GOTO 400
180 BICHAR(1,A)
200 K=29
220 IF(SIZE0-L-I).LT.(30); K=SIZE0-L-I
240 FOR J=0,K
260 MOVES(B,1,A,J*8+1,8)
280 ARRAY(I+J)=DECODE(B)
300 NEXT J
320 NEXT I
322 LEN=120
323 IF(SIZE0-L+1).LT.(120);LEN=SIZE0-L+1
325 MCA(-1,L-1,ARRAY(1),LEN)
330 NEXT L
340 RETURN
400 RETRY=RETRY+1
420 IF(RETRY).GT.(3); TYPE "PERMANANT ERROR"!; RETURN
440 X=MTIO(UNIT,5,DEN#,1,I)
460 GOTO 120

```



## REFINA

```

1  REMARK: H.GOTOH, 1976-4-15
2  REMARK: PROGRAM FOR READ DATA FROM FINANCE FILES
3  REMARK:   KEYIN "REFINA"
7  STRING TAG(7),HIN(28),TAGG(7),YE(4)
8  ASK "FILENAME = ",TAG
20 CALL TRANSF,MOVES,TAB,SPY,STOP
22 MOVES(TAGG,1,TAG,1,5);TAGG(6)="S"
25 IF(SPY(0,TAG)).EQ.(0);IF(SPY(0,TAGG)).EQ.(0);TYPE "FILES DOES NOT
EXIST !",!;STOP
40 INTEGER A(128);STRING B(512),COMMEN(128)
56 TRANSF(0,1,TAG,1,A)
57 FOR I=1,128;COMMEN(I)=A(I);NEXT I
58 TYPE "COMMENT = ",!,COMMEN,!
59 JJ=0
60 FOR I=2,50
70 TRANSF(0,1,TAG,I,A);TRANSF(0,1,TAGG,I,B)
80 FOR J=1,16,113
90 JJ=JJ+1;IF(A(J+2)).EQ.(0);STOP
100 TYPE %3,JJ,A(J+2)
110 TAB(10);FOR K=1,28;TYPE B(K+(J-1)*4);NEXT K
120 TAB(40);TYPE %10,A(J+8),%3,A(J+10),!
130 NEXT J
140 NEXT I

```

## REFINE

```

1  REMARK: H.GOTOH, 1976-4-15
2  REMARK: PROGRAM FOR READ DATA FROM FINANCE FILES
3  REMARK:   KEYIN "REFINE"
7  STRING TAG(7),HIN(28),TAGG(7),YE(4)
8  ASK "FILENAME = ",TAG
20 CALL TRANSF,MOVES,TAB,SPY,STOP
22 MOVES(TAGG,1,TAG,1,5);TAGG(6)="S"
25 IF(SPY(0,TAG)).EQ.(0);IF(SPY(0,TAGG)).EQ.(0);TYPE "FILES DOES NOT
EXIST !",!;STOP
40 INTEGER A(128);STRING B(512),COMMEN(128)
50 ASK "START # OF LINE ? ",MS
52 ASK "STOP # OF LINE ? ",NS
54 IF(NS).GT.(392);NS=392
56 TRANSF(0,1,TAG,1,A)
57 FOR I=1,128;COMMEN(I)=A(I);NEXT I
58 TYPE "COMMENT = ",!,COMMEN,!
60 I=INT(MS/8-.1)+2,J=MS-8*(I-2)
70 TRANSF(0,1,TAG,I,A);TRANSF(0,1,TAGG,I,B)
100 TYPE %3,MS,%4,A((J-1)*16+3)
110 TAB(10);FOR K=1,28;TYPE B(K+(J-1)*64);NEXT K
120 TAB(40);TYPE %8,A((J-1)*16+9),%3,A((J-1)*16+11),!
130 IF(MS).GE.(NS);TYPE !!!;STOP
140 IF(J).GE.(8);I=I+1,J=1,MS=MS+1;GOTO 70
150 J=J+1,MS=MS+1;GOTO 100

```

## REFINC

```

1  REMARK: H.GOTOH, 1976-4-19
2  REMARK: PROGRAM FOR TOTALIZING FINANCIAL DATA IN GROUP WISE
3  REMARK:   KEYIN "REFINC"
7  STRING TAG(7),HIN(28),TAGG(7),YE(4)
8  ASK "FILENAME = ",TAG
20 CALL TRANSF,MOVES,TAB,SPY,STOP
22 MOVES(TAGG,1,TAG,1,5);TAGG(6)="S"
25 IF(SPY(0,TAG)).EQ.(0);IF(SPY(0,TAGG)).EQ.(0);TYPE "FILES DOES NOT
EXIST !",!;STOP
40 INTEGER A(128);STRING B(512),COMMEN(128)
50 ASK "START # OF LINE ? ",MMS
52 ASK "STOP # OF LINE ? ",NS
54 IF(NS).GT.(392);NS=392
56 TRANSF(0,1,TAG,1,A)
57 FOR I=1,128;COMMEN(I)=A(I);NEXT I
58 TYPE "COMMENT = ",!,COMMEN,!
59 TYPE !,"SEIKYU      HINMEI";TAB(40);TYPE "SHI SHUTSU  GROUP",!, "
#",!
60 II=INT(MMS/8-.1)+2,JJ=MMS-8*(II-2)
62 SS=0
65 FOR L=0,4
66 S=0,I=II,MS=MMS,J=JJ
70 TRANSF(0,1,TAG,I,A);TRANSF(0,1,TAGG,I,B)
90 IF(A((J-1)*16+11)).NE.(L);GOTO 130
95 S=S+A((J-1)*16+9)
100 TYPE %4,A((J-1)*16+3)
110 TAB(10);FOR K=1,28;TYPE B(K+(J-1)*64);NEXT K
120 TAB(40);TYPE %8,A((J-1)*16+9),%3,A((J-1)*16+11),!
130 IF(MS).GE.(NS);GOTO 155
140 IF(J).GE.(8);I=I+1,J=1,MS=MS+1;GOTO 70
150 J=J+1,MS=MS+1;GOTO 90
155 TAB(20);TYPE %2,"GROUP ",L," TOTAL ",%8,S," YEN",!
157 SS=SS+S
160 NEXT L
170 TAB(20);TYPE "LABORATORY TOTAL      ",%8,SS," YEN",!

```

## RENBIT

```

1  REMARK: H.GOTOH, 1971-12-5
2  REMARK: PROGRAM FOR SOLVING SIMULTANEOUS LINEAR EQUATIONS
100 ARG(N)
110 DIMENS C(N)
120 COMMON A,B
130 FOR K=1,N
140 IF(K).GE.(N);GOTO 500
150 FOR J=K+1,N;X=A(J,K),X=ABS(X),Y=A(K,K),Y=ABS(Y)
155 IF(X).LE.(Y);GOTO 400
160 D=B(K),B(K)=B(J),B(J)=D
170 FOR I=1,N;C(I)=A(K,I),A(K,I)=A(J,I),A(J,I)=C(I);NEXT I
400 NEXT J
500 Y=A(K,K);FOR J=1,N;IF(J).EQ.(K);GOTO 600
510 X=A(J,K);FOR I=1,N;A(J,I)=A(J,I)-A(K,I)*X/Y;NEXT I
520 B(J)=B(J)-B(K)*X/Y
600 NEXT J;FOR I=1,N;A(K,I)=A(K,I)/Y;NEXT I
610 B(K)=B(K)/Y;NEXT K

```

## REPL

```

10 REMARK : N.TAKEUCHI & H.YAGY 1976-06-20
12 REMARK : REPLACE ONE CHARACTER
13
20 STRING NAME(7),OS(2),NS(2),LX(22),B(512)
30 CALL XFER,LINK
40 ASK "FILE NAME ? ",NAME
50 ASK "BLOCK SIZE ? ",BL
60 ASK "OLD CHARACTER ? ",OS
70 ASK "NEW CHARACTER ? ",NS
80 TYPE %4;SW=0,IT=0;LINK(1,@NAME,LX)
100 FOR K=2,BL-1;J=1;XFER(1,K,LX,B)
120 FOR I=J,512;FOR J=I,I+63;IT=B(J)
122 IF(IT).EQ.(0);J=I+63;GOTO 140
130 IF(B(J)).EQ.(OS);B(J)=NS;SW=1;TYPE 8*(K-2)+INT(I/64+0.99)
140 NEXT J;I=J-1
150 IF(SW).EQ.(1);SW=0;TYPE !
160 NEXT I;XFER(-1,K,LX,B)
170 NEXT K;RETURN

```

## PID

```

20 STRING A(241),B(81)
40 CALL BICHAR,MTIO,MOVES,AND
60 ASK "UNIT NUMBER ? ",U
80 DEN#=3; J=0
90 MTIO(U,7,DEN#,0,J)
100 ASK "FILE SEQUENCE NUMBER TO BE DELETED ? ",N
120 N=N-1
140 IF(N).EQ.(0); GOTO 220
160 FOR I=1,N
180 MTIO(U,4,DEN#,0,N)
200 NEXT I
220 MTIO(U,1,DEN#,240,A)
240 A(1)=128; REMARK--128 IS EBCDIC SPACE
250 J=J+1
260 MTIO(U,5,DEN#,1,N)
280 MTIO(U,2,DEN#,240,A)
300 RETURN

```

## SCR

```

10 REMARK: H. GOTOH, 1976-3-18
20 REMARK: SCRATCH MCA DATA IN STEP WISE
30 REMARK: SCR(ORIGINAL REGION,INIT CH,STEP SIZE,# OF REPETITION)
40 ARG(REG,INIT,STEP,REPET)
45 REG1=0,REG2=2048
48 IF(REG).EQ.(2);REG1=2048,REG2=0
50 DIMENS A(STEP)
55 CALL MCA;STEPP=STEP-1
60 FOR I=1,REPET
70 MCA(1,REG1+(I-1)*STEP+INIT,A(1),STEP)
80 MCA(-1,REG2+(I-1)*STEPP+INIT,A(2),STEPP)
90 NEXT I

```

## SCSA2

```

10 REMARK : N.TAKEUCHI 1976-01-09/1976-03-03
12 REMARK : SCRATCH TO COMMON FILE AND SAVE TO ONE FROM CORE
14 REMARK : ARG(1)= "FILE NAME"
16 REMARK : ARG(2)= # OF DISK UNIT
18
20 CALL SPY
30 ARG(TAG,DK);COMMON @TAG
55 IF(DK).EQ.(0);GOTO 60
56 IF(DK).EQ.(1);GOTO 60
57 TYPE "# ERROR : ",%3,DK,"??",!;STOP
60 IF(SPY(DK,TAG)).EQ.(0);GOTO 90
70 IF(DK).EQ.(0);SCRATCH FD0 @TAG;GOTO 90
80 SCRATCH FD1 @TAG
90 IF(DK).EQ.(0);SAVE FD0 @TAG;RETURN
100 SAVE FD1 @TAG;RETURN
110 RETURN

```

## SEAR

```

10 REMARK : N.TAKEUCHI 1976-05-17
12 REMARK : SEARCH THE FILE DATA
14 REMARK : ARG("FILE NAME",LENGTH,DIMENSION/SECTOR,"TYPE",DISK #,X)
20
30 ARG(NAME,LEG,DI,TYP,DK,X)
40 CALL LINK,XFER,BUFA
50 STRING LX(22);BUFA(DI,TYP);COMMON A
60 LINK(DK,@NAME,LX);L=INT(LEG/DI+1.99)
70 FOR K=2,L;XFER(1,K,LX,A);FOR I=1,DI
80 IF(A(I)).NE.(X);NEXT I;NEXT K;TYPE "DO NOT FOUND !!!";RETURN
90 TYPE " RECORD # =",%3,K," LENGTH = ",%5,DI*(K-2)+1
100 RETURN

```

## SEAR0

```

10 REMARK : N.TAKEUCHI 1976-05-17
12 REMARK : SEARCH THE MCA DATA
14 REMARK : ARG("FILE NAME",X)
20
30 ARG(NAME,X);LEG=4096,DI=120,DK=1
40 CALL LINK,XFER,BUFA
50 STRING LX(22),TYP(2);TYP="I";BUFA(DI,TYP);COMMON A
60 LINK(DK,@NAME,LX);L=INT(LEG/DI+1.99)
70 FOR K=2,L;XFER(1,K,LX,A);FOR I=1,DI
80 IF(A(I)).NE.(X);NEXT I;NEXT K;TYPE "DO NOT FOUND !!!";RETURN
90 TYPE "#",%4,120*(K-2)+1-1," CHANNEL= ",%6,A(1)
100 RETURN

```

SEDIT

```

12 REMARK : N.TAKEUCHI 1976-06-22
12 REMARK : EDIT FOR MICRO-8
20
30 STRING NAME(7),LX(22),LX2(22),LXN(22),A(512),B(512)
32 CALL XFER,LINK,CREATE,SPY,NU2;COMMON NU2
34 INTEGE BL,II,IL,LN,IT;TYPE !,24
40 ASK "EDIT FILE NAME ? ",NAME
50 ASK "BLOCK SIZE ? ",BL
60 ASK "EDIT LINE # ? ",IL
68 IF(SPY(1,"QAZ")).NE.(0);SCRATCH FDI QAZ
70 CREATE(1,"QAZ","S",(512),BL+4)
80 LINK(1,@NAME,LX);LINK(1,QAZ,LXN)
90 K=INT((IL-1)/8)+1,BN=IL-8*(K-1),I=64*(BN-1)+1,KO=K+1,LN=0
100 IF(K).LE.(1);GOTO 120
110 FOR J=1,K;XFER(1,J,LX,A);XFER(-1,J,LXN,A);NEXT J
120 K=K+1;XFER(1,K,LX,A)
130 FOR J=1,512;M=IL+LN;TYPE "###"
140 IF(M).GE.(100);MM=INT(M/100),M=M-100*MM;TYPE NU2(MM+1)
142 TYPE NU2(M+1)," "
150 FOR I=J,J+63;ASK A(I);IT=A(I);IF(IT).EQ.(5);GOTO 300
160 IF(IT).NE.(142);NEXT I;A(I-2)=142,A(I-1)=0;GOTO 200
170 IF(I).EQ.(J+63);A(I-1)=142,A(I)=0;GOTO 190
180 A(I+1)=0
190 I=J+63;NEXT I
200 J=I-1;LN=LN+1;IF(LN).GE.(40);GOTO 200
210 NEXT J;XFER(-1,K,LXN,A);K=K+1,I=1;GOTO 130
220 TYPE !," *** SEDIT CLOSED FOR BUFFER OVER FLOW ***",!
300 JJ=INT(I/64);I=512;NEXT I;J=512;NEXT J
310 IF(JJ).EQ.(0);GOTO 600
320 IF(BN).EQ.(1);GOTO 620
350 IF(BN-JJ-1)360,400,450
360 FOR I=64*JJ+1,64,449;A(I)=131,A(I+1)=142,A(I+2)=0
362 NEXT I;XFER(-1,K,LXN,A);K=K+1
370 XFER(1,KO,LX,A);FOR I=1,64,64*(BN-2)+1
380 A(I)=131,A(I+1)=142,A(I+2)=0;NEXT I;GOTO 470
400 XFER(1,KO,LX,B);FOR I=64*JJ+1,512;A(I)=B(I);NEXT I
410 XFER(-1,K,LXN,A);GOTO 430
450 FOR I=64*JJ+1,64,64*(BN-2)+1;A(I)=131,A(I+1)=142,A(I+2)=0;NEXT I
460 XFER(1,KO,LX,B);FOR I=64*(BN-1)+1,512;A(I)=B(I);NEXT I
470 XFER(-1,K,LXN,A)
480 KO=KO+1,K=K+1,LL=K-KO,L=BL-1+LL
490 FOR I=K,L;XFER(1,I-LL,LX,A);XFER(-1,I,LXN,A);NEXT I
500 SCRATCH FDI @NAME;CREATE(1,NAME,"S",(512),L);LINK(1,@NAME,LX2)
510 FOR I=1,L;XFER(1,I,LXN,A);XFER(-1,I,LX2,A);NEXT I
520 SCRATCH FDI QAZ
530 RETURN
600 IF(BN).EQ.(1);XFER(1,KO,LX,A);GOTO 470
610 GOTO 370
620 FOR I=64*JJ+1,512;A(I)=0;NEXT I;KO=KO-1;GOTO 470

```

## SEPARA

```

10 REMARK: H.GOTOH, 1976-4-21
20 REMARK: PROGRAM FOR SEPARATION OF A FILE INTO 2 FILES
30 REMARK
40 REMARK:
50 STRING TAG(7), COM(100), TAG1(7), TAG2(7), LX(22), LX1(22), LX2(22)
60 INTEGE A(100), B(100), C(100)
70 TAG="SAM104", TAG1="SA1104", TAG2="SA2104"
80 CALL CREATE, LINK, XFER
90 CREATE(1, TAG1, "I", (100), 11); CREATE(1, TAG2, "I", (100), 11)
100 LINK(1, @TAG, LX); LINK(1, @TAG1, LX1); LINK(1, @TAG2, LX2)
110 XFER(1, 1, LX, A); XFER(-1, 1, LX1, A); XFER(-1, 1, LX2, A)
120 FOR I=1, 100; COM(I)=A(I); NEXT I
130 TYPE "COMMENT = ", COM, !
140 FOR I=1, 10
150 XFER(1, 2*I, LX, A)
160 FOR J=1, 50
170 B(J)=A(2*J-1), C(J)=A(2*J); NEXT J
180 XFER(1, 2*I+1, LX, A)
190 FOR J=1, 50; B(J+50)=A(2*J-1), C(J+50)=A(2*J); NEXT J
200 XFER(-1, I+1, LX1, B); XFER(-1, I+1, LX2, C)
210 NEXT I

```

## SLST

```

10 REMARK : N.TAKEUCHI 1976-06-22
12 REMARK : SOURCE LIST FOR MICRO-8
20
30 STRING NAME(7), LX(22), A(512), B(65)
32 INTEGE IL, FL, IT, I, PAGE
34 CALL XFER, LINK, LINE, NU2; COMMON LINE, NU2
40 ASK !, "FILE NAME ? ", NAME
50 ASK "INITIAL LINE # ? ", IL
60 ASK "FINAL LINE # ? ", FL
70 K=INT((IL-1)/8)+1, BS=IL-8*(K-1)
80 LINK(1, @NAME, LX); TYPE !, %4
90 IL2=IL, PAGE=INT((IL-1)/64)
92 IF(FL-IL).GE.(63); GOSUB 200
100 K=K+1; XFER(1, K, LX, A)
110 FOR J=BS, 8; JJ=64*(J-1); IT=A(JJ+1)
120 IF(IT).EQ.(0); B(1)=131, B(2)=142, B(3)=0; GOTO 170
130 FOR I=JJ+1, 64*J; B(I-JJ)=A(I); IT=A(I)
140 IF(IT).EQ.(142); B(I-JJ+1)=0; I=64*J; NEXT I; GOTO 170
150 IF(IT).NE.(0); NEXT I; I=I-1
160 B(I-JJ)=142, B(I-JJ+1)=0
170 TYPE IL, " ", B; IL=IL+1
180 IF(IL).LE.(FL); NEXT J; GOSUB 200; BS=1; GOTO 100
182 IF(FL-IL2).LT.(63); RETURN
184 I=IL-IL2; IF(I).NE.(64*INT(I/64)); TYPE !; IL=IL+1; GOTO 184
186 TYPE !!, LINE; STOP
200 I=IL-IL2
210 IF(I).NE.(64*INT(I/64)); RETURN
220 PAGE=PAGE+1
230 TYPE !!, LINE, NAME, " ---- ", NU2(PAGE+1), !!!
240 RETURN

```

## STAIR

```

100 INTEGER A(100)
200 INIT=2048
300 FOR I=1,100;A(I)=0;NEXT I
400 MCA(-1,INIT,A(1),100)
500 K=1
600 FOR J=1,6
700 FOR I=1,100;A(I)=K;NEXT I
800 MCA(-1,INIT+100*J,A(1),100)
900 K=K*10
1000 NEXT J

```

## STIN

```

10 REMARK : N.TAKEUCHI 1976-05-04
12 REMARK : CONVERT INTEGER FROM STRING
16 REMARK : ARG(String DATA)
20 ARG(ST)
30 CALL STNU,NUI;COMMON STRINU,NUI,SINT
40 ERASE SINT;INTEGER COMMON SINT
50 STNU(ST);C=1,P=0,SINT=0,N=STRINU
60 FOR I=1,N;IF(ST(I)).NE("-");NEXT I;SI=1;GOTO 90
70 SI=-1,P=1,I=N;NEXT I
80 FOR I=1,P;ST(I)="0";NEXT I;P=0
90 STRING STS(N+1);FOR I=1,N;FOR J=1,10
100 IF(ST(I)).EQ.(NUI(J));J=10;NEXT J;GOTO 120
110 NEXT J;P=P+1;NEXT I;N=N-P,P=N;GOTO 130
120 STS(I-P)=ST(I);NEXT I;N=N-P,P=N
130 IF(N).GT.(1);C=C*10,N=N-1;GOTO 130
140 FOR I=1,P;FOR J=48,57
150 IF(STS(I)).NE.(J);NEXT J;J=48
160 SINT=INT(SINT+C*(J-48)+0.9),C=C/10;NEXT I
170 SINT=SINT*SI
180 RETURN

```

## STPPHA

```

10 UNIBUS(-1,167606,1)

```

STRE

```

100 REMARK : N.TAKEUCHI 1976-05-04
110 REMARK : CONVERT PEAL FROM STRING
120 REMARK : ARG(STRING DATA)
130 ARG(ST)
140 CALL STNU,STIN,NUI
150 COMMON SINT,STRINU,NUI,R0;ERASE R0;COMMON R0
160 STNU(ST);N=STRINU
170 E.=0,S.=0,D.=0,SI=1,P1=0
180 FOR I=1,N;FOR J=1,10;IF(ST(I)).EQ.(NUI(J));J=10;NEXT J;GOTO 150
190 NEXT J;IF(ST(I)).EQ.( ".");D.=1;GOTO 150
200 IF(ST(I)).EQ.( "-");SI=-1
210 P1=P1+1;NEXT I
220 I=N;NEXT I;STRING BN(N-P1+1);BN(1)=ST(P1+1)
230 IF(N).EQ.(1);GOTO 330
240 FOR I=P1+2,N;FOR J=1,10;IF(ST(I)).EQ.(NUI(J));J=10;NEXT J;GOTO 320
250 NEXT J;IF(ST(I)).EQ.( "E");GOTO 220
260 IF(ST(I)).EQ.( "-");GOTO 260
270 IF(ST(I)).EQ.( ".");GOTO 220
280 P1=P1+1;NEXT I;GOTO 300
290 IF(E.).EQ.(0);GOTO 240
300 P1=P1+N-I-1;I=N;NEXT I;GOTO 310
310 IF(D.).EQ.(0);E.=I-P1;GOTO 300
320 GOTO 210
330 IF(E.).EQ.(0);GOTO 210
340 IF(S.).EQ.(1);GOTO 210
350 S.=1;GOTO 300
360 IF(E.).EQ.(0);E.=I-P1;GOTO 300
370 GOTO 210
380 BN(I-P1)=ST(I);NEXT I
390 NN=N-P1,N=NN;STRING BN1(N),BN2(N),BN3(N)
400 IF(E.).NE.(0);NN=E.-1
410 IF(D.+1).EQ.(E.);NN=D.-1,BN(D.)=0,L.=0
420 IF(D.).EQ.(0);GOTO 440
430 FOR I=D.+1,NN;BN1(I-L.)=BN(I);NEXT I;STIN(BN1)
440 P1=D.-NN;GOSUB 500;P0=SINT*R0
450 IF(D.).EQ.(1);SINT=0;GOTO 380
460 FOR I=1,D.-1;BN2(I)=BN(I);NEXT I;STIN(BN2)
470 R0=R0+SINT
480 IF(E.).EQ.(0);GOTO 430
490 IF(E.+S.).GE.(N);GOTO 430
500 FOR I=E.+1,N;BN3(I-E.)=BN(I);NEXT I
510 STIN(BN3);P1=SINT;GOSUB 500;R0=R0*R0
520 R0=R0*SI;RETURN
530 IF(E.).EQ.(0);ERASE R0;INTEGE COMMON R0;STIN(BN);R0=SINT;GOTO 430
540 R0=0,D.=E.;GOTO 360
550 R2=1;IF(P1).EQ.(0);RETURN
560 IF(P1).LE.(0);GOTO 600
570 IF(R1).GE.(48);R1=R1-48,R2=R2*.1E49;GOTO 520
580 IF(R1).GE.(8);R1=R1-8,R2=R2*.1E9;GOTO 530
590 IF(R1).GT.(0);R1=R1-1,R2=R2*10;GOTO 540
600 IF(P1).LT.(0);P1=P1+1,R0=R0/10;GOTO 550
610 RETURN
620 IF(P1).LE.(-48);P1=P1+48,R2=R2/.1E49;GOTO 600
630 IF(P1).LE.(-8);P1=P1+8,R2=R2*.1E-7;GOTO 610
640 GOTO 540

```



## TABLEA

```

80 COMMON B
90 FOR J=1,3
95 TYPE %10.3,"Y = ",J*2-2,!
100 FOR I=-5,1,5
110 TYPE I,I*I*B(I)+B(J+1),!
120 NEXT I
130 NEXT J

```

## TABLEU

```

10 REMARK: H.GOTOH, 1976-4-6
20 REMARK: OUTPUT PROGRAM FOR GAMMA-RAY TABLE
30 TYPE !!!,"TABLE OF GAMMA-RAYS FROM FISSILE MATERIAL",!!
40 TYPE " # ENERGY BRANCH NUCLIDE REFERE OBSERVED",!
45 CALL TAB
50 TAB(8);TYPE "(KEV) RATIO";TAB(37);TYPE "NCE OR NOT",!!
100 COMMON FISGAM,FISGAN
200 FOR I=1,100
300 IF(FISGAM(I,1)).LE.(0);STOP
400 TYPE %3.1,%7.02,FISGAM(I,1),%8.05,FISGAM(I,2)," "
520 FOR J=1,6;TYPE FISGAN(I,J);NEXT J
530 TYPE " ";FOR J=7,12;TYPE FISGAN(I,J);NEXT J
540 TYPE " ";FOR J=13,18;TYPE FISGAN(I,J);NEXT J
550 TYPE !
600 NEXT I

```

## TABLEX

```

10 REMARK: H.GOTOH, 1976-4-8
20 REMARK:
30 TYPE !!!,"TABLE OF GAMMA-RAYS FROM FISSILE MATERIAL",!!
40 TYPE " # ENERGY BRANCH NUCLIDE REFERE OBSERVED",!
45 CALL TAB
50 TAB(8);TYPE "(KEV) RATIO";TAB(37);TYPE "NCE OR NOT",!!
100 COMMON FISGAM,FISGAN
200 FOR I=1,100
300 IF(FISGAM(I,1)).LE.(0);STOP
400 TYPE %3.1,%7.02,FISGAM(I,1),"KEV("
520 FOR J=1,6;TYPE FISGAN(I,J);NEXT J
540 TYPE ")")
550 TYPE !
600 NEXT I

```

## TABLEY

```

10 REMARK: H.GOTOH, 1976-4-6/4-12
20 REMARK: OUTPUT PROGRAM FOR GAMMA-RAY TABLE(OMITS NOT OBSERVED)
30 TYPE ! ! !, "TABLE OF GAMMA-RAYS FROM FISSILE MATERIAL", ! !
40 TYPE " # ENERGY BRANCH NUCLIDE REFERE OBSERVED", !
45 CALL TAB
47 STRING TAG(5)
50 TAB(8); TYPE "(KEV) RATIO"; TAB(37); TYPE "NCE OR NOT", ! !
100 COMMON FISGAM, FISGAN
150 K=1
200 FOR I=1, 100
210 FOR J=1, 5; TAG(J)=FISGAN(I, 12+J); NEXT J
220 IF(TAG).EQ.(" "); GOTO 600
300 IF(FISGAM(I, 1)).LE.(0); STOP
400 TYPE %3, K, %7.02, FISGAN(I, 1), %8.05, FISGAN(I, 2), " "
520 FOR J=1, 6; TYPE FISGAN(I, J); NEXT J
530 TYPE " "; FOR J=7, 12; TYPE FISGAN(I, J); NEXT J
540 TYPE " "; FOR J=13, 18; TYPE FISGAN(I, J); NEXT J
550 TYPE !
590 K=K+1
600 NEXT I
610 TYPE ! ! ! !

```

## TDAY

```

10 REMARK : N.TAKEUCHI 1976-02-24
12 REMARK : NUMBER OF TOTAL DAYS FOR DATE
16 REMARK : ARG(1)= YEAR(A.D.)
18 REMARK : ARG(2)= MONTH
20 REMARK : ARG(3)= DAY
22 REMARK * L
30 ARG(YY, MM, DD); Y=YY
40 COMMON TD ; ERASE TD ; INTEGE COMMON TD
50 CO=INT(1/MM+0.5)
60 Y=YY-CO; M=MM+12*CO
70 TD=INT(365.25*Y)-INT(Y/100)+INT(Y/400)+INT(30.6*M+1.61)+DD
80 RETURN

```

## TDAY1

```

10 REMARK : N.TAKEUCHI 1976-02-24
12 REMARK : NUMBER OF TOTAL DAYS FOR DATE
16 REMARK : ARG(1)= YEAR(A.D.)
18 REMARK : ARG(2)= MONTH
20 REMARK : ARG(3)= DAY
22 REMARK * L
30 ARG(YY, MM, DD); Y=YY
40 COMMON TD ; ERASE TD ; INTEGE COMMON TD
50 Y=YY-400*INT(YY/400)+400; CO=INT(1/MM+0.5)
60 Y=YY-CO; M=MM+12*CO
70 TD=INT(365.25*Y)-INT(Y/100)+INT(Y/400)+INT(30.6*M+1.61)+DD
80 RETURN

```

TRCE

JAERI-M 6719

```
5 REMARK 76-6-29 MODIFY "TRCE" H.P.YAGY
10 REMARK H.P.YAGY 1976-6-8
20 REMARK SOURCE LIST, SOURCE TAPE, OPTION WRITE FOR MICRO-8 ASSEMBLER
30 REMARK
100 STRING A(512), B(64)
110 STRING SW(6), S(6), NAME(7), LX(22)
120
140 TYPE Z4, !!
150 ASK "/ LIST OR TAPE ?", S
160 ASK "/ FILE NAME ? ", NAME
170 ASK "/ INITIAL LINE # ? ", STK
180 ASK "/ FINAL LINE # ? ", SPK
230 LINK(1, @NAME, LX)
250 IF(S(1)).EQ.("L"); GOTO 290
260 ASK "/ PTP SW ON ? ", SW, !!
265 IF(SW(1)).NE.("Y"); GOTO 260
267 B(1)=0
270 FOR I=1, 70; TYPE B(1); NEXT I
290 L=STK-1
500 K=INT(STK/8+1.9), LL=STK, I=(LL-(K-2)*8-1)*64
510 GOTO 550
530 I=0; IF(S(1)).EQ.("L"); TYPE !
540 K=K+1
550 XFER(1, K, LX, A)
560 GOTO 590
580 LL=LL+1, I=(LL-(K-2)*8-1)*64
590 IF(LL).GT.(SPK); GOTO 900
600 IF(INT(LL/8-0.01)).GT.(K-2); GOTO 530
610 IF(S(1)).EQ.("L"); L=L+1; TYPE L, " "
660 IF(A(I+1)).EQ.(0); TYPE "/, !; GOTO 580
670 TYPE A(I+1)
680 FOR M=2, 64
690 J=M+I
695 IF(A(J)).EQ.(0); GOTO 810
710 IF(A(J)).EQ.(142); GOTO 810
720 TYPE A(J)
800 NEXT M
810 TYPE !
820 GOTO 580
900 TYPE "          END", !
950 FOR I=1, 30; TYPE " "; NEXT I; TYPE !
```

W

```
5 REMARK: H.GOTOH, 1976-4-6
10 REMARK: REWRITING OF TABLE OF GAMMA ENERGY
50 STRING TAG(7), REF(7), OBS(7)
100 COMMON FISGAM, FISGAN
200 ASK "# OF LINE ? ", X
250 ASK "ENERGY(KEV) ? ", ENE
270 ASK "BRANCH(<1.0) ? ", BRAN
300 ASK "NAME OF NUCLIDE ?", TAG
310 ASK "REFERENCE ? ", REF
320 ASK "OBSERVED ? ", OBS
350 FISGAM(X, 1)=ENE, FISGAM(X, 2)=BRAN
400 FOR I=1, 6
500 FISGAN(X, I)=TAG(I)
600 NEXT I
700 FOR I=1, 6; FISGAN(X, I+6)=REF(I); NEXT I
800 FOR I=1, 6; FISGAN(X, I+12)=OBS(I); NEXT I
```

WR

```

100 COMMON FISGAN
150 STRING OBS(6)
200 ASK "# OF LINE ? ",I
300 ASK "OBSERVED ? ",OBS
400 FOR K=1,5;FISGAN(I,12+K)=OBS(K);NEXT K

```

WRDIS1

```

1 REMARK PREPARED BY H. GOTOH, DEC. 11, 1975
2 REMARK PROGRAM FOR WRITE DATA FROM MCA TO DISK
3 REMARK KEYIN "WRDIS1"
6 REMARK: REVISED BY H. GOTOH, FEB. 20, 1976
7 STRING TAG(7),YE(4)
8 ASK "FILENAME = ",TAG
9 ASK "NEW COMMENT ? ",YE
10 IF(YE).EQ.("YES");COMMON COMMEN;ERASE COMMEN;STRING COMMON COMMEN(
120);ASK "COMMENT = ",!,COMMEN;GOTO 12
11 COMMON COMMEN
12 ASK "INITIAL CHANNEL = ",INIT,"LENGTH = ",LENG
14 N=INT(LENG/120)
15 NN=N+2;IF(LENG-120*N).EQ.(0);NN=NN-1
20 CALL CREATE,XFER,LINK,MCA,SPY,STOP
25 IF(SPY(1,TAG)).NE.(0);TYPE "FILENAME EXISTS!";STOP
30 CREATE(1,TAG,"I",(120),NN)
40 INTEGER A(120)
53 STRING LX(22)
55 LINK(1,@TAG,LX)
57 FOR I=1,120;A(I)=COMMEN(I);NEXT I
58 XFER(-1,1,LX,A)
60 FOR I=1,N
70 MCA(1,(I-1)*120+INIT,A(1),120)
80 XFER(-1,I+1,LX,A)
85 NEXT I
87 IF(LENG-120*N).EQ.(0);GOTO 100
88 FOR I=1,120;A(I)=0;NEXT I
90 MCA(1,N*120+INIT,A(1),LENG-120*N)
95 XFER(-1,NN,LX,A)
100 ERASE CREATE,XFER,LINK,SPY

```

## WRFINA

```

1  REMARK: H.GOTOH, 1976-4-15
2  REMARK: PROGRAM FOR WRITE DATA INTO FINANCE FILES
3  REMARK:   KEYIN "WRFINA"
7  STRING TAG(7),HIN(28),TAGG(7),YE(4)
8  ASK "FILENAME = ",TAG
10 FOR I=1,5;TAGG(I)=TAG(I);NEXT I;TAGG(6)="S"
20 CALL CREATE,TRANSF,SPY,STOP
25 IF(SPY(0,TAG)).EQ.(0);IF(SPY(0,TAGG)).EQ.(0);TYPE "FILES DOES NOT
EXIST !",!;STOP
40 INTEGER A(128);STRING B(512),COMMEN(128)
56 TRANSF(0,1,TAG,1,A)
57 FOR I=1,128;COMMEN(I)=A(I);NEXT I
58 TYPE "COMMENT = ",!,COMMEN,!
60 FOR I=2,50
70 TRANSF(0,1,TAG,I,A);TRANSF(0,1,TAGG,I,B)
80 FOR J=1,16,113
90 IF(A(J+2)).EQ.(0);GOTO 105
100 NEXT J
105 TYPE J, !;IF(J).GE.(114);GOTO 490
107 ASK "MORE INPUT(YES OR NO) ? ",YE
108 IF(YE).EQ.("NO");GOTO 490
110 ASK "SEIKYU # ?",SEI
120 ASK "HINMEI ? ",HIN
150 ASK "SHISHUTSU ? ",SHI
160 ASK "GROUP ? ",GRO
200 A(J+2)=SEI,A(J+10)=GRO,A(J+8)=SHI
210 FOR K=1,28;B(K+(J-1)*4)=HIN(K);NEXT K
220 J=J+16;GOTO 105
490 TRANSF(0,-1,TAG,I,A);TRANSF(0,-1,TAGG,I,B)
495 IF(YE).EQ.("NO");STOP
500 NEXT I

```

## WRFINB

```

1  REMARK: H.GOTOH, 1976-4-15
2  REMARK: PROGRAM FOR WRITE DATA INTO FINANCE FILES
3  REMARK:   KEYIN "WRFINB"
7  STRING TAG(7),HIN(28),TAGG(7),YE(4)
8  TAG="FIN501"
10 FOR I=1,5;TAGG(I)=TAG(I);NEXT I;TAGG(6)="S"
20 CALL TRANSF,MOVES,SPY,STOP
25 IF(SPY(0,TAG)).EQ.(0);IF(SPY(0,TAGG)).EQ.(0);TYPE "FILES DOES NOT
EXIST !",!;STOP
40 INTEGER A(128);STRING B(512),COMMEN(128)
50 ASK "START # ? ",STA
52 II=INT(STA/8)
54 JJ=8*II+1
60 FOR I=II+2,50
70 TRANSF(0,1,TAG,I,A);TRANSF(0,1,TAGG,I,B)
80 J=I
85 IF(JJ).LT.(STA);GOTO 220
105 TYPE JJ,!;IF(J).GE.(114);J=1;GOTO 490
107 ASK "MORE INPUT(YES OR NO) ? ",YE
108 IF(YE).EQ.("NO");GOTO 490
109 HIN(1)=" ";MOVES(HIN,2,HIN,1,27)
110 ASK "SEIKYU # ? ",SEI
120 ASK "HINMEI ? ",HIN
150 ASK "SHISHUTSU ? ",SHI
160 ASK "GROUP ? ",GRO
200 A(J+2)=SEI,A(J+10)=GRO,A(J+8)=SHI
210 FOR K=1,28;B(K+(J-1)*4)=HIN(K);NEXT K
220 J=J+16,JJ=JJ+1;GOTO 85
490 TRANSF(0,-1,TAG,I,A);TRANSF(0,-1,TAGG,I,B)
495 IF(YE).EQ.("NO");STOP
500 NEXT I

```

## WRITE1

```

1  REMARK: H.GOTOH, 1976-3-26
10 ARG(SIZE0)
20 COMMON UNIT,DEN#
30 CALL MCA,CLOSE
40 STRING A(241),B(81)
50 INTEGER ARRAY(120)
55 SIZE=120
60 FOR L=1,120,SIZE0
65 IF(SIZE0-L).LT.(120);MCA(1,L-1,ARRAY(1),SIZE0-L);GOTO 100
70 MCA(1,L-1,ARRAY(1),120)
100 FOR I=1,30,SIZE
110 A(1)=" ";MOVES(A,2,A,1,239)
120 K=29
140 IF(SIZE0-L-1).LT.(30);K=SIZE0-L-1
160 FOR J=0,K
180 ENCODE(ARRAY(I+J),B,%7)
200 MOVES(A,J*8+1,B,1,8)
220 NEXT J
230 BICHAR(-1,A)
240 MTIO(UNIT,2,DEN#,240,A)
260 NEXT I
270 NEXT L

```

## YOBI

```

10 REMARK : N.TAKEUCHI 1976-02-24
12 REMARK : WEEK OF THE TODAY
14 REMARK : ARG(1)= YEAR(A.D.)
16 REMARK : ARG(2)= MONTH
18 REMARK : ARG(3)= DAY
20 REMARK
30 ARG(YY,MM,DD);Y=YY,M=MM,D=DD
40 CALL TDAY1,INST,WE,NU
50 INST(Y);Y=YY;TDAY1(Y,M,D);COMMON IST,WE,NU,TD
60 YO=TD-7*INT(TD/7)
70 TYPE %5,Y,"-",NU(M+1),"-",NU(D+1),"= ",WE(YO+1)
80 RETURN

```

## YOBI 1

```

10 REMARK : N.TAKEUCHI 1976-02-24
12 REMARK : WEEK OF THE TODAY
14 REMARK : ARG(1)= YEAR(A.D.)
16 REMARK : ARG(2)= MONTH
18 REMARK : ARG(3)= DAY
20 REMARK
30 ARG(YY,MM,DD)
40 TDAY1(YY,MM,DD)
50 COMMON TD,YO;ERASE YO;INTEGE COMMON YO
60 YO=TD-7*INT(TD/7)

```

## YOME

```

100 COMMON A,B
110 DIMENS C(3)
120 FOR I=1,3
130 ASK C(1),C(2),C(3),B(I),!
140 FOR J=1,3;A(I,J)=C(J);NEXT J
150 NEXT I

```