

JAERI-M

6499

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

1976年3月

竹内 紀男・八木 秀之・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

日本原子力研究所東海研究所
竹内紀男・八木秀之・五藤博
(1976年2月20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39編を収録した。内容は、ディスク、コア、MCA間のデータの転送、ピーク面積計算、平均値・標準偏差の計算、くり返し測定、結果の印刷を主題としている。

An ORACL Program File
for Acquisition, Storage and Analysis of Data
in Radiation Measurement and
in Non-Destructive Measurement of Nuclear Material

Norio TAKEUCHI, Hideyuki YAGI and Hiroshi GOTOH

Tokai Research Establishment, JAERI

(Received February 20, 1976)

This program file contains 39 programs useful for radiation measurement and non-destructive measurement of nuclear material written in the conversational language ORACL associated with the GAMM-III system of ORTEC Incorporated. The main part of the file deals with data transfer between disks, core and MCA, calculation of peak area, calculation of mean and standard deviation, repeated counting and print of results.

目 次

1. ま え が き		1
2. データ転送プログラムの相互関係		1
3. プログラムの名称, 用途, 使い方		2
(紙テープを読む)		
1. 紙テープをコアに読む	P T M E	2
2. 紙テープをディスクに読む(1)	P T F D	2
3. " (2)	P T T O F D	2
4. 特殊紙テープをディスクに読む(1)	Y A G I P 1	2
5. " (2)	Y A G I P T	3
6. " (3)	Y G 5 0 0 0	3
7. 紙テープをMCAに読む	P T T O M C	3
(MCAを読む)		
8. MCAをディスクに読む(1)	M C F D	3
9. " (2)	W R D I S K	4
(印刷する, 紙テープに書く)		
10. コアを印刷する	M E P T	4
11. ディスクを印刷する(1)	F D P T	4
12. " (2)	F D P T 1	5
13. " (3)	T Y P O U T	5
(MCAに書く)		
14. ディスクからMCAへ(1)	F D M C	5
15. " (2)	R E D I S K	5
(ディスク内データの操作)		
16. ディスクを移し換える	F D F D	6
17. ディスクを加え合わせる	M I X F	6
18. コメントを書き換える	C O M M E	6
19. ファイル名を変更する	R E N A M D	7
20. ディスクを消去する(1)	S C R A	7
21. " (2)	S C R A 1	7
22. 1ブロックあたりのデータ長を変える	T R N D A	8
(コアとディスク間のやりとり)		
23. コアからディスクへ	M E F D	8
24. ディスクからコアへ	F D M E	8
25. プログラムを更新する(1)	S C S A	9
26. " (2)	S C S A 1	9

(測定)

27. くり返し測定する ACCUMU 9

(データ解析)

28. 平均値・標準偏差を求める(1) MEAN 9

29. " (2) REMEAN 10

30. ヒストグラムを描く(1) HISTO 10

31. " (2) DRHIST 10

32. 最大・最小値を探す MAXMIN 11

33. ピーク面積を求める(1) PEAK1 11

34. " (2) PEAREA 12

(その他)

35. コア上に領域をとる(1) BUFA 12

36. " (2) BUFB 12

37. スtringの字数を数える(1) STNU 12

38. " (2) STNU1 12

39. 花文字を描く FL 13

4. プログラム・リスティング (アルファベット順) (使い方の順序)

ACCUMU 27 14

BUFA 35 14

BUFB 36 15

COMME 18 15

DRHIST 31 15

FDFD 16 16

FDMC 14 16

FDME 24 17

FDPT 11 17

FDPT1 12 18

FL 39 18

HISTO 30 20

MAXMIN 32 20

MCFD 8 21

MEAN 28 21

MEFD 23 22

MEPT 10 22

MIXF 17 23

PEAK1 33 23

PEAREA 34 24

PTFD 2 24

PTME	1	25
PTTOFD	3	25
PTTOMC	7	26
REDISK	15	26
REMEAN	29	27
RENAMD	19	27
SCRA	20	28
SCRA1	21	28
SCSA	25	28
SCSA1	26	29
STNU	37	29
STNU1	38	29
TRNDA	22	30
TYPOUT	13	30
WRDISK	9	31
YG5000	6	31
YAGIP1	4	32
YAGIPT	5	33

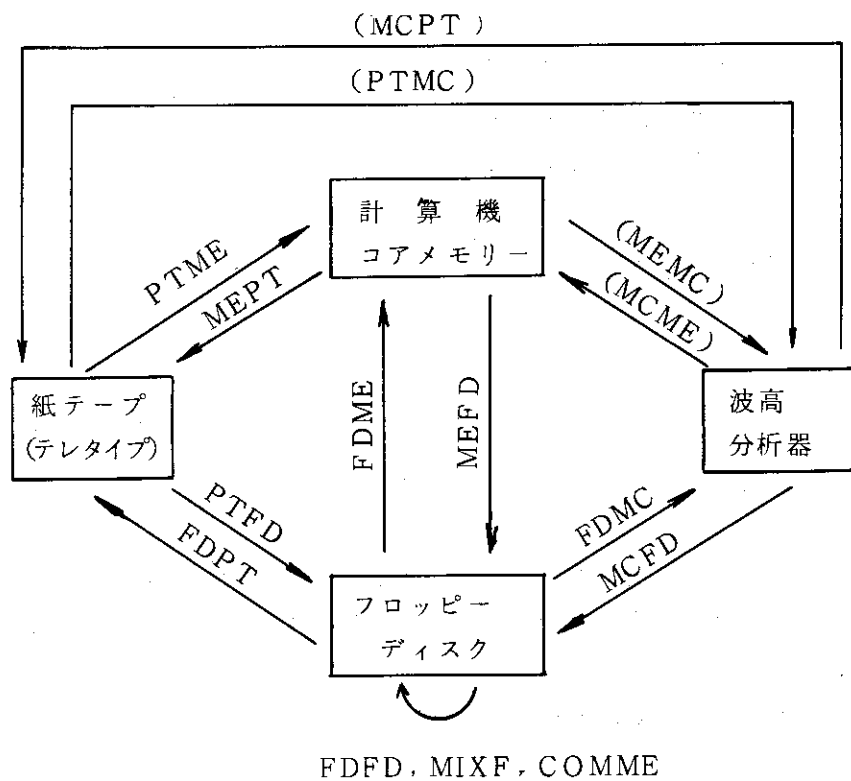
1. ま え が き

日本原子力研究所 R I 製造部検査課および原子炉工学部原子炉計測研究室は共同でORTEC社製GAMMA-IIIシステムを放射線測定および核物質破壊測定に使用するうえで役立つプログラムを製作している。このプログラム集には、当初（1975年11月）から、翌年2月までに製作したものの中で、ひきつづき使用する可能性のあるものを収録することにした。

この第1集では、データの転送に関するものが大半を占めることになった。それは、データの貯蔵に便なファイル形式と解析操作に便なファイル形式は必ずしも一致しないためである。

2. データ転送プログラムの相互関係

データ転送に関するプログラムの種類が多いので、系統的に作成したものについて相互関係を第1図に示す。図中、カッコ（ ）で囲んだプログラムはまだできていない。テレタイプで印刷だけをとりたい場合には、紙テープ穿孔スイッチをオフにしておけばよい。



第1図 データ転送プログラムの相互関係

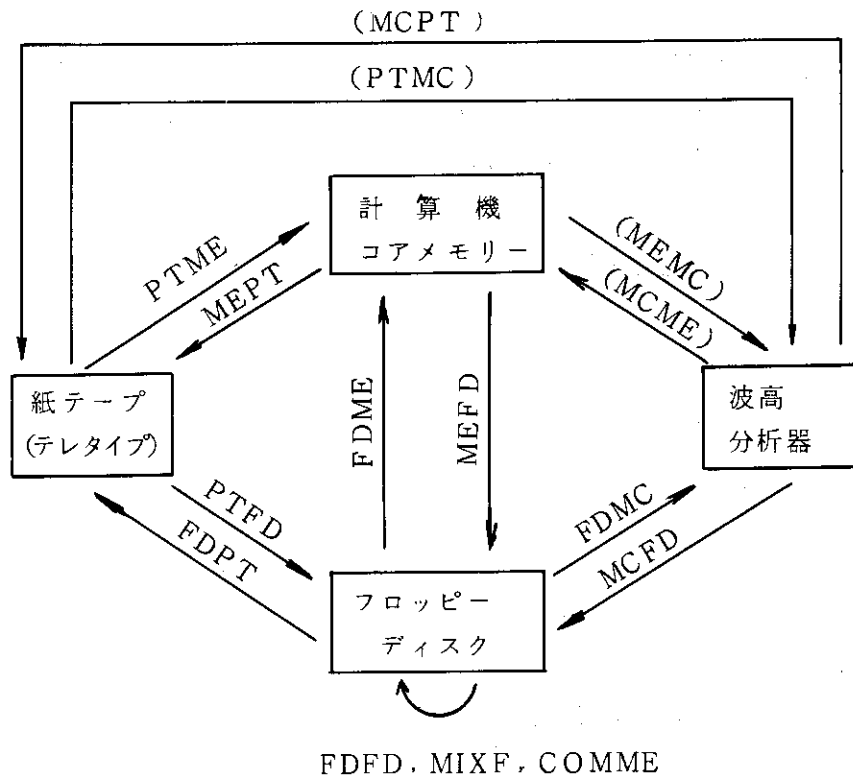
1. ま え が き

日本原子力研究所 R I 製造部検査課および原子炉工学部原子炉計測研究室は共同でORTEC社製GAMMA-IIIシステムを放射線測定および核物質破壊測定に使用するうえで役立つプログラムを製作している。このプログラム集には、当初(1975年11月)から、翌年2月までに製作したものの中で、ひきつづき使用する可能性のあるものを収録することにした。

この第1集では、データの転送に関するものが大半を占めることになった。それは、データの貯蔵に便なファイル形式と解析操作に便なファイル形式は必ずしも一致しないためである。

2. データ転送プログラムの相互関係

データ転送に関するプログラムの種類が多いので、系統的に作成したものについて相互関係を第1図に示す。図中、カッコ()で囲んだプログラムはまだできていない。テレタイプで印刷だけをとりたい場合には、紙テープ穿孔スイッチをオフにしておけばよい。



第1図 データ転送プログラムの相互関係

3. プログラムの名称，用途，使い方

1. 紙テープをコアに読む

PTME

(用途)

紙テープのデータを計算機のコアに読む。

(使い方)

入力は引数を用いる。PTME (データ個数, "データ形式")とキーインする。共通領域Aに紙テープの内容が移るのでAを用いてデータの処理ができる。紙テープのリードスイッチをOFFにすれば、テレタイプの手キーボードよりデータを入れることができる。

2. 紙テープをディスクに読む(1)

PTFD

(用途)

紙テープのデータをディスクにファイルする。FDPTにより穿孔された紙テープを再びディスクに戻すのに用いる。

(使い方)

入力は引数を用いる。PTFD ("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)とキーインする。紙テープのリードスイッチをOFFにすれば、テレタイプの手キーボードよりデータを入れることもできる。

3. 紙テープをディスクに読む(2)

PTTOFD

(用途)

指定した長さの紙テープデータにコメントをつけて、整数型のデータとして、ディスクに書き込む。

(使い方)

PTTOFDとキーインする。データの長さ、ディスク1ブロックあたりのデータの長さ、ファイル名、コメントを聞いてくるので、それに答えたのち、紙テープを読み込ませる。

4. 特殊紙テープをディスクに読む(1)

YAGIP1

(用途)

10個ごとにキャリジ・リターンが2個はいった紙テープデータをディスクに読む。

(使い方)

YAGIP1とキーインして使い始める。データの長さ、1ブロックあたりのデータの長さ、ファイル名、コメントを聞いてくる。

5. 特殊紙テープをディスクに読む(2)YAGIPT

(用途)

紙テープの形になっている12桁の整数型データをディスクに読み込む。

(使い方)

YAGIPTとキーインして使い始める。データの長さ、1ブロックあたりのデータの長さ、ファイル名、コメントを聞いてくる。

6. 特殊紙テープをディスクに読む(3)YG5000

(用途)

指定した長さの紙テープデータを1ブロックあたり120の長さで、整数型としてディスクに書き込み、同じデータをMCAに書き込んで、データを観察することができる。

(使い方)

PTTTFDに準ずる。

7. 紙テープをMCAに読む(2)PTTOMC

(用途)

指定した長さの紙テープデータをMCAの指定したチャンネル領域に読み込む。

(使い方)

PTTOMC (INIT, LENG)とキーインするとよい。INITは読みこまれる領域の最初のチャンネル数、LENGはデータの長さである。

8. MCAをディスクに読む(1)MCFD

(用途)

波高分析器のデータをディスクに移す。

(使い方)

入力は引数を用いる。MCFD (“ファイル名”, データ個数, ブロック当りのデータの長さ, ディスク番号)とキーインする。新ファイルはディスクにすでに存在するファイルと同じ名前であってはならない。データの個数は4096以内で任意の値が使用できる。

9. MCAをディスクに読む(2)

WRDISK

(用途)

MCA内の指定した領域のデータを整数型のデータとして、ディスク内にコメントをつけて書き込む。

(使い方)

あらかじめSTRING COMMON TAG (7), COMMEN (120) をとり, TAG にファイル名を6文字以内の英数字で, COMMENには120文字以内の英数字でコメントを書き込んでおく。

WRDISK (INIT, LENG)とキーインすると書き込みが完了する。ここで, INITは読みとるべきMCA領域の最初のチャンネル, LENGは領域のチャンネル数である。ディスクには1ブロックに120個の数値が書き込まれる。データは第2ブロック以降に入れられる。コメントは第1ブロックに書き込まれる。

10. コアを印刷する

MEPT

(用途)

計算器のコアの内容を紙テープに打ち出す。

(使い方)

入力は引数を用いる。MEPT (データ個数, テレタイプ1行当りのデータ数, テレタイプ1ブロック当りのデータ数)とキーインする。データはコアの共通領域Aにあらかじめ確保されていなければならない。紙テープの穿孔とテレタイプの印刷が同時に行われる。紙テープが不要の時は穿孔スイッチをOFFにしておく。

11. ディスクを印刷する(1)

FDPT

(用途)

ディスク上のファイルの内容を紙テープに打ち出す。

(使い方)

入力は引数を用いる。FDPT (“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, テレタイプ1行当りのデータ数, テレタイプ1ブロック当りのデータ数)とキーインする。第1セクターのコメントも数値で印刷される。FDPT1を使用すればコメントは文字で印刷される。紙テープが不要の時は穿孔スイッチをOFFにしておく。

12. ディスクを印刷する(2)FDPTI

(用途)

ディスク内にあるコメント付データについて、コメントとデータを頭から読みだして印刷する。

(使い方)

FDPTI("NAME", LEG, DI, "TYP", DK, H \bar{O} , VE)とタイプインすればよい。ここでNAMEはファイル名, LEG は読み出すべきデータの長さ (\leq 書き込んであるデータの長さ), DI は1ブロックあたりのデータの長さ, TYPは整数 (I), 実数 (R), ストリング (S) の区別, DKはディスク機番, H \bar{O} は印刷1行あたりのデータの長さ, VEは印刷に1行のスペースを入れるまでのデータの長さ (VEはH \bar{O} の整数倍) である。

13. ディスクを印刷する(3)TYPOUT

(用途)

ディスク内にあるコメント付整数型データをコメントとともに印刷する。

(使い方)

TYPOUT (LENGTH) とキーインする。LENGTHはデータの長さである。1ブロックあたりのデータの長さは120と固定している。

14. ディスクからMCAへ(1)FDMC

(用途)

ディスク上のデータを波高分析器に入れブラウン管に表示する。

(使い方)

入力は引数を用いる。FDMC ("ファイル名", データ個数, ブロック当りのデータの長さ, ディスク番号) とキーインする。波高分析器にはあらかじめ全チャンネルを0にセットし, CRTモードにしておくとディスクからの転送状態がよくわかる。もし4096ケ以上のデータがある場合は0チャンネルから書き換えられる。第1ブロックはコメントのため転送されない。

15. ディスクからMCAへ(2)REDISK

(用途)

ディスク内にあるコメント付整数型データを波高分析器に読み込む。

(使い方)

REDISK (INIT, LENG) とキーインする。INITは最初のデータのはいるべきチャンネル数, LENGはデータの長さである。

16. ディスクを移し換える

FDFD

(用 途)

ディスク上にあるデータファイルと同じ内容のファイルを任意のディスクに新しく作る。主にファイルを別のディスクに移すのに使用される。

(使い方)

入力は引数を用いる。FDFD (データ個数, ブロック当りのデータの長さ, “データ形式”, “ファイル名”, ディスク番号, “新ファイル名”, 新ディスク番号) とキーインする。

ディスク番号と新ディスク番号が同じ場合, ファイル名と新ファイル名が同じであってはならない。データ個数はファイルにある全データ数以内ならば任意の値が許される。もしそれ以上の値を入力した場合はエラーが表示されるがファイルにあるデータは完全に移される。

17. ディスクを加え合わせる

MIXF

(用 途)

データ形式, ブロック当りのデータの長さ, データ数が全て同じようないくつかのファイルについて, 各ファイルの同じ順番のデータを全て加え別に新しいファイルを作る。波高分析器のスペクトルの合成などに使用される。

(使い方)

MIXF とキーインする。データ個数 (LENGTH), ブロック当りデータの長さ, データ形式, 処理するファイルの個数を聞いてくるので順番にキーボードより答える。次にファイルの名前とそのディスク番号を聞いてくるので処理するファイル名を順次答える。最後に新ファイル名とディスク番号を同様に答える。

新ファイル名はすでにディスク上に存在するファイル名と一致してはならない。ミックスされるファイルのデータ個数が一致していない場合は, その中の最少値以下の値を使用しなければならない。それ以上の値を入力するとエラーが表示されるが最少値までのデータは完全に処理される。

18. コメントを書き変える

COMME

(用 途)

データファイルの第1セクターはコメント用に使用されている。このコメントの内容を印刷し, 必要に応じてコメントを書き変えることができる。

(使い方)

入力は引数を用いる。COMME (“ファイル名”, “データの形式”, ブロック当りのデータの長さ, ディスク番号) とキーインするとコメントの内容を印刷する。最後に DO YOU WANT TO WRITE NEW COMMENT? と聞いてくるので YES, NO で答える。YES の場合 WRITE TO NEW COMMENT WITHIN ○○ LETTERS と指示してくるから, 新しいコメントをキーインする。古いコメントは全部消される。

19. ファイル名を変更する

RENAMD

(用途)

ディスク内にある1ブロックあたり120個の整数型ファイルのファイル名を変更する。

(使い方)

RENAMD とキーインする。旧ファイル名, 新ファイル名, ブロック数を聞いてくるので, それらに応えるとよい。

20, 21. ディスクを消去する(1)(2)

SCRA, SCRA1

(用途)

フロッピーディスクには整数型で約6.0K語相当のプログラム, データ, バリアブル, ストリングなどが混在できる。ディスク上の不必要なものだけを消去するのに使用する。

(使い方)

SCRAとキーインするとNAME =と印刷するので機番0のディスク上にある消去したいプログラム, データ等をキーインする。何ヶでも続けることができる。もしまちがってディスク上に無い名前を入力した時は○○○FLE NOT EXIST ON #0 DISK と答えるから引続いて正しい名前を入力する。終了したならば数字(任意)をキーインすると計算機は待期状態に戻る。

SCRA1は機番1のディスクの内容を消去するのに使用する。

2.2. 1ブロックあたりのデータ長さを変える

TRNDA

(用 途)

ディスク上にあるファイルデータを、任意のデータ形式で任意のブロック当りのデータの長さに変換し別の新しいファイルを作る。

(使い方)

入力は引数を用いる。TRNDA (データ個数, “ファイル名”, ブロック当りのデータ長さ, “データ形式”, ディスク番号, “新ファイル名”, 新ファイルのブロック当りのデータの長さ, “新ファイルのデータ形式”, 新ファイルのディスク番号) とキーインする。データ個数が4Kの場合約200秒で完了し計算機は待期状態に戻る。

ディスク番号と新ディスク番号が同じ場合, ファイル名と新ファイル名が同じであってはならない。データ個数はファイルにある全データ数以内ならば任意の値が許される。もしそれ以上の値を入力した場合はエラーが表示されるがファイルにあるデータは完全に変換される。

データ形式とブロック当りのデータの長さが新, 旧ファイルで同じ場合はプログラムFDFDを使用する方が処理時間が早い。

2.3. コアからディスクへ

MEFD

(用 途)

計算機のコアの内容をディスクに移す。

(使い方)

入力は引数を用いる。MEFD (“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) とキーインする。データはコアの共通領域Aにあらかじめ確保されていないなければならない。

2.4. ディスクからコアへ

FDME

(用 途)

ディスク上のファイルのデータを計算機のコアに移す。

(使い方)

入力は引数を用いる。FDME (“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) とキーインする。共通領域Aにディスクの内容が移るのでAを用いてデータの処理ができる。

25, 26. プログラムを更新する (1), (2)

SCSA, SCSA 1

(用 途)

機番 0 のディスクにある指定されたプログラムを消去し、コア上の同じ名前のプログラムをディスクに移す。プログラムのデバッキングの時に使用する。

(使い方)

SCSA とキーインすると NAME = と印刷するので機番 0 に移したいコアにあるプログラムの名前をキーインする。何ヶでも続けることができる。終了したならば数字 (任意) をキーインすると計算機は待期状態に戻る。

SCSA 1 は機番 1 のディスクに移したい場合に使用する。

27. くり返し測定する

ACCUMU

(用 途)

同じ試料の波高分布を指定した個数だけくり返し測定するためのもの。波高分布データを指定したチャンネル区間について各個ごとにファイル名を新たにディスク内に記憶させることができる。指定した 1 個のピークについて最大計数をもつチャンネル数とその計数、ピーク中心の精密値、ピーク面積、およびその誤差を計算し印刷させることができる。

(使い方)

ACCUMU とキーインして使い始める。

28. 平均値・標準偏差を求める (1)

MEAN

(用 途)

ディスク上にあるファイルデータの平均値、標準偏差、相対標準偏差を求めるのに使用される。

(使い方)

入力は引数を用いる。MEAN("ファイル名", データ個数, ブロック当りのデータの長さ, ディスク番号, 定数) とキーインする。

平均値、標準偏差、相対標準偏差の順に印刷される。データの個数はそのファイルの全データ数よりも少なければ任意の値が使用できる。定数はデータのバラツキが少なくて標準偏差の値が小さい時に使用する。一般には平均値に近い区切りの良い値を用いる。バラツキが大きい場合は 0 でも差しつかえない。

印刷の書式はデータの種類により異なるので MEAN の使用前に TYPE %00.00 とキーインしておく。6 桁の整数形のデータの場合 TYPE %6 でよい。

29. 平均値・標準偏差を求める(2)

REMEAN

(用途)

ディスク上にあるファイルデータを任意の大きさのグループに分割しそれぞれの平均値, 標準偏差, 相対標準偏差を求め, 順次印刷する。最後にグループ毎の平均値の平均値, 標準偏差, 相対標準偏差を印刷する。データの時間的な変動をみる場合等に使用される。

(使い方)

入力は引数を用いる。REMEAN("ファイル名", データ個数, ブロック当りのデータ長さ, ディスク番号, グループ当りのデータ個数, 定数)とキーインする。データの全個数とグループ数との関係で最後のグループにデータの数が不足する様な時はそのグループは無視される。このプログラムは6桁の整数形のデータに合わせて印刷の書式が決められている。

30. ヒストグラムを描く(1)

HISTO

(用途)

ディスク内の同種データをヒストグラム状に解析し, 棒状グラフを印刷する。

(使い方)

HISTO(FNAME, NDIM, NREC, BASE, INC, NSTEP)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座標の最下限値, INCはx座標の核み, NSTEPはx座標のステップ数である。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変数 C に用意しておく。DRHIST(BASE, INC, NSTEP)とキーインする。ここでBASEはx座標の最低値, INCはx座標の核み, NSTEPはx座標のステップ数である。なお, y座標は1ステップが10となっている。

32. 最大・最小値を探す

MAXMIN

(用途)

ディスク内のデータの最大値と最小値を探し、それらの値を印刷する。

(使い方)

MAXMIN ("FNAME", NDIM, NREC)とキーインする。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 はデータの長さである。

33. ピーク面積を求める

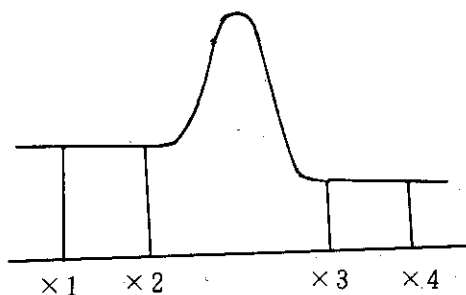
PTAKI

(用途)

シングルピークの精密な中心値と面積および面積の標準偏差と相対標準偏差を求め印刷する。データはMCA内のものを対象とする。

(使い方)

PEAK1 (X1, X2, X3, X4)とキーインすると結果を得る。X1, X2, X3, X4には第2図のように、低エネルギー側および高エネルギー側の平坦部の下限および上限チャンネル数である。



第2図 波高分布

34. ピーク面積を求める(2)PEAREA

(用途)

同一試料の波高分布をくり返し測定して得られたディスク内のデータについて特定ピークの中心値およびピーク面積を求めて印刷し、ピーク面積については、PEAKというファイル名でディスク内に記録する。

(使い方)

PEAREAとキーインする。データのファイル名の共通する最初の3英数字をキーインし、ついでファイル名の最後の3数字となるべき続き番号の初期値をキーインする。くり返し数、ディスク内データの最初のチャンネル数、データの長さをキーインする。最後に、ピークの下側の平坦部の下限・上限値、上側の平坦部の下限・上限値および結果を印刷するか否かの答えをキーインする。

35, 36. コア上に領域をとる(1), (2)BUFA, BUFB

(用途)

コア上に実数形、整数形、文字形のいずれかで任意の大きさの共通領域Aを確保する。主にプログラムの中でのサブプログラムとして使用される。

(使い方)

入力は引数を用いる。メインプログラム中でRUN BUFA(データ個数, "データ形式"); COMMON Aを実行することにより共通領域Aを自由に使用することができる。

BUFBは共通領域Bが確保される。

37, 38. スtringの字数を数える(1), (2)STNU, STNU1

(用途)

任意のstringの文字の個数を求める。メインプログラム中でのサブプログラムとして使用される。

(使い方)

入力は引数を用いる。STNU("文字")とキーインする。共通領域である整数形の変数STRINUにその文字の字数が入る。サブプログラムとして使用する場合はRUN STNU(STRING変数); COMMON STRINUの様に使用する。

STNU1の場合は○○○=○○LETTERSと印刷してくれる。

39. 花文字を描く

FL

(用途)

0～9の各数値と空白とから構成される任意の長さの数値を図3に示した様に5×5のマトリクスで表示する。

タイプ印刷の見出しや注目個所の指示等に使用される。

```
*FL(2,"51 03 17")
```

```
***** *          ***** ***** *          *****
*        *          *        *          *          *        *
***** *          *        *          *          *          *
*        *          *        *          *          *          *
***** *          ***** *****          *          *
```

第3図 花文字FLの使用例

(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0～9よりなる任意の長さの数字列である。テラ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。

このプログラムはサブプログラムとしてY, T, Uを専用に使っている。数字の形の変更や、英字や新たな記号の追加はこれらのサブプログラムの修正により簡単に行うことができる。

4. プログラム・リスティング

ACCUMU

```

10 REMARK: PROGRAM FOR REPEATED MEASUREMENT OF 4096 CH DATA
12 REMARK: PREPARED BY H. GOTOH, NOV. 19, 1975
14 REMARK: REVISED BY H. GOTOH, JAN. 5, 1976
20 CALL DISPLA,ERSPHA,COUNT,DELAY,MCA
180 COMMON TAG,COMMEN,TY;ERASE TAG,COMMEN,TY
185 INTEGER I,NDATA,IDATA,NN,TIME(3),TIM(3),IX,NO,I1(3)
190 STRING COMMON TAG(7),COMMEN(120),TY(4)
195 STRING PE(4),DI(4),D(11),COM(110);D="01234567890"
200 ASK !,"TAG WORD(IN JUST 3 LETTERS) = ",TAG
210 ASK "INITIAL # OF DATA WITHIN 3 DIGITS) = ",NDATA;ASK "DO YOU WRIT
E IN DISK ? ",DI
211 IF(DI).EQ.("YES");ASK "INITIAL CHANNEL OF RECORD = ",ICH;ASK "DATA
LENGTH = ",LENG
212 IF(DI).EQ.("YES");ASK "WRITE COMMENT WITHIN 110 LETTERS : ",!,COM
213 ASK "DO YOU WANT THE AREA OF A SPECIFIED PEAK ? ",PE
214 IF(PE).EQ.("YES");ASK "LOWER LIMIT OF LOWER REGION = ",X1,"UPPER L
IMIT OF LOWER REGION = ",X2
215 IF(PE).EQ.("YES");ASK "LOWER LIMIT OF UPPER REGION = ",X3,"UPPER L
IMIT OF UPPER REGION = ",X4
216 ASK "# OF REPETITION = ",NO,"DO YOU TYPE LABEL DATE START.... ? ",
TY
220 FOR I=1,110;COMMEN(10+I)=COM(I);NEXT I
230 COMMEN(3)="",COMMEN(6)="",COMMEN(9)=" ",COMMEN(12)=" "
310 IF(PE).EQ.("YES");CALL PEAK1
320 IF(DI).EQ.("YES");CALL WRDISK
380 CLOCK(1)
385 ASK "WHAT DAY OF THE MONTH TODAY = ",TIME(1)
390 ASK "WHAT TIME IS IT NOW(HH,MM) = ",TIME(2);ASK TIME(3)
400 T0=CLOCK(0)
408 IF(TY).EQ.("YES");TYPE !," MAX COUNT",!
410 IF(TY).EQ.("YES");TYPE " LABEL DATE START CHNL COUNT CENTER
AREA SGM REL.SGM",!
510 FOR NN=NDATA,NDATA+NO-1
512 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)
513 FOR J=6,-1,4;TAG(J)=D(I1(J-3)+1);NEXT J
515 DISPLA;ERSPHA;DELAY;COUNT
517 IF(UNIBUS(0,167606,1)).EQ.(0);GOTO 515
520 T1=CLOCK(0)
525 TM=INT(TIME(3)+(T1-T0)/3000)
526 TH=TIME(2)+INT(TM/60)
527 TIM(3)=TM-60*INT(TM/60)
528 TIM(1)=TIME(1)+INT(TH/24),TIM(2)=TH-24*INT(TH/24)
530 IF(TY).EQ.("YES");TYPE %2.00,TAG," ",TIM(1),TIM(2),":",TIM(3)
532 FOR I=1,3;IX=INT(TIM(1)/10),COMMEN(3*I-2)=D(IX+1)
534 IX=TIM(1)-10*IX,COMMEN(3*I-1)=D(IX+1);NEXT I
540 IF(UNIBUS(0,167606,1)).NE.(0);GOTO 540
550 IF(PE).EQ.("YES");PEAK1(X1,X2,X3,X4)
560 IF(DI).EQ.("YES");WRDISK(ICH,LENG)
908 NEXT NN
910 TYPE "THE WORK COMPLETED"

```

BUFA

```

10 REMARK : N.TAKEUCHI 1976-01-17
12 REMARK : SUB PROGRAM
14 REMARK
16 REMARK : ARG(1)= LEGTH
18 REMARK : ARG(2)= "DATA TYPE" ("R","I","S")
20 REMARK
30 COMMON A;ERASE A
40 ARG(LEG,TYP)
50 IF(TYP).EQ.("R");DIMENS COMMON A(LEG);GOTO 90
60 IF(TYP).EQ.("I");INTEGE COMMON A(LEG);GOTO 90
70 IF(TYP).EQ.("S");STRING COMMON A(LEG);GOTO 90
80 TYPE "ERROR OF DATA TYPE";STOP
90 RETURN

```

BUFB

```

10 REMARK : N.TAKEUCHI 1976-01-17
12 REMARK : SUB PROGRAM
14 REMARK
16 REMARK : ARG(1)= LEGTH
18 REMARK : ARG(2)= "DATA TYPE" ("R","I","S")
20 REMARK
30 COMMON B;ERASE B
40 ARG(LEG,TYP)
50 IF(TYP).EQ.("R");DIMENS COMMON B(LEG);GOTO 90
60 IF(TYP).EQ.("I");INTEGE COMMON B(LEG);GOTO 90
70 IF(TYP).EQ.("S");STRING COMMON B(LEG);GOTO 90
80 TYPE "ERROR OF DATA TYPE";STOP
90 RETURN

```

COMME

```

10 REMARK : N.TAKEUCHI 1976-01-09
12 REMARK : READ/WRITE FOR COMMENT OF DATA FILE
14 REMARK
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
30 ARG(NAME,TYP,DI,DK)
40 CALL XFER,LINK
50 STRING COM(DI+1),LX(22),SW(4)
60 RUN BUFB(DI,TYP);COMMON B
100 LINK(DK,@NAME,LX);XFER(1,1,LX,B)
110 FOR I=1,DI;COM(I)=B(I);NEXT I
140 TYPE !,COM,!!
150 ASK "DO YOU WANT TO WRITE NEW COMMENT ? = ",SW
160 IF(SW).NE.("YES");GOTO 210
165 TYPE %3
170 TYPE "WRITE TO NEW COMMENT WITHIN",DI," LETTERS",!!
180 ASK COM
190 FOR I=1,DI;B(I)=COM(I);NEXT I
200 XFER(-1,1,LX,B)
210 ERASE XFER,LINK,COM,LX,SW
220 RETURN

```

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
90 REMARK: ARG(3)= NUMBER OF STEPS
100 ARG(BASE,INC,NSTEP)
110 CALL TAB
120 COMMON C
180 TYPE !," COUNT RANGE CASES",!
190 NL=0,NG=0
200 NT=0;FOR I=1,NSTEP;TYPE %6,BASE+(I-1)*INC,"-",BASE+I*INC,%4,C(I)
205 TAB(INT(C(I)/10)+20);TYPE "%",!
210 NT=NT+C(I);NEXT I
220 NT=NT+NL+NG
230 TYPE !,"TOTAL NUMBER = ",NT,!,"NUMBER OF LOWER LIMIT = ",NL,!,"NUMBER OF HIGHER LIMIT = ",NG,!!

```

FDFD

```

10 REMARK : N.TAKEUCHI 1976-01-24/1976-02-04
12 REMARK : DATA TRANSFER TO DISK FROM DISK
14 REMARK
16 REMARK : ARG(1)= LEGTH
18 REMARK : ARG(2)= DIMENSION/SECTOR
20 REMARK : ARG(3)= "FILE TYPE" ("R","I","S")
22 REMARK : ARG(4)= SOURCE : "FILE NAME"
24 REMARK : ARG(5)= # OF DISK UNIT
26 REMARK : ARG(6)= OBJECT : "FILE NAME"
28 REMARK : ARG(7)= # OF DISK UNIT
30 REMARK
40 ARG(LEG,DI,TYP,TAG,DK,TAGN,DKN)
50 CALL LINK,XFER,CREATE
60 STRING LX(22),LXN(22)
70 RUN BUFB(DI,TYP);L=INT(LEG/DI+1.99)
80 CREATE(DKN,TAGN,TYP,(DI),L)
90 LINK(DK,@TAG,LX);LINK(DKN,@TAGN,LXN)
95 COMMON B
100 FOR K=1,L
110 XFER(1,K,LX,B);XFER(-1,K,LXN,B)
120 NEXT K
140 ERASE CREATE,LINK,XFER,LX,LXN,B
150 RETURN

```

FDMC

```

10 REMARK : N.TAKEUCHI 1976-02-06
12 REMARK : DATA TRANSFER TO MCA 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)= # OF DISK UNIT
24 REMARK
40 ARG(NAME,LEG,DI,DK)
45 STRING LX(22)
50 CALL XFER,LINK,MCA
60 L=INT(LEG/DI+1.99),N=0
70 LINK(DK,@NAME,LX)
80 INTEGE B(DI)
90 FOR K=2,L;XFER(1,K,LX,B)
100 IF(N+DI).LT.(LEG);MCA(-1,N,B(1),DI);N=N+DI;NEXT K
120 DI=LEG-N;INTEGE BB(DI)
130 FOR I=1,DI;BB(I)=B(I);NEXT I
140 MCA(-1,N,BB(1),DI)
150 ERASE XFER,LINK,MCA,B,BB,LX
160 RETURN

```


FDME

```

10 REMARK : N.TAKEUCHI 1976-01-17
12 REMARK : DATA TRANSFER TO MEMORY 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
30 STRING LX(22)
40 ARG(NAME,LEG,DI,TYP,DK)
50 CALL XFER,LINK
60 L=INT(LEG/DI+1.99),N=0
70 LINK(DK,@NAME,LX)
80 RUN BUFA(LEG+DI,TYP);RUN BUFB(DI,TYP);COMMON A,B
90 FOR K=1,L;XFER(1,K,LX,B)
92 FOR I=1,DI;A((K-1)*DI+I)=B(I),N=N+1
94 IF(N).GE.(LEG);GOTO 98
96 NEXT I;NEXT K
98 RETURN

```

FDPT

```

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
90 FOR K=1,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 RETURN

```

```

10 REMARK : N.TAKEUCHI 1976-01-17
12 REMARK: READ COMMENT AND DATA IN FLOPPY DISK
14 REMARK: MODIFIED BY H. GOTOH, JAN. 28, 1976
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),COMM(DI)
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 K=1,L;XFER(1,K,LX,B)
88 IF(K).GT.(1);GOTO 91
89 FOR I=1,DI;COMM(I)=B(I);TYPE COMM(I);NEXT I;TYPE !!
90 IF(K).EQ.(1);GOTO 96
91 FOR I=1,DI;TYPE B(I);N=N+1
92 IF(N).GE.(LEG);GOTO 96
93 IF(N).EQ.(HO*INT(N/HO));TYPE !
94 IF(N).EQ.(VE*INT(N/VE));TYPE !
95 NEXT I
96 NEXT K
97 TYPE !!!!
98 RETURN

```

FL

```

10 REMARK : N.TAKEUCHI 1976-01-09
12 REMARK : TYPE IN FLOWER CHARACTER
14 REMARK
16 REMARK : ARG(1)= LINE SPASE
18 REMARK : ARG(2)= "CHARACTER"
20 REMARK
30 ARG(SP,CA)
110 COMMON FX;ERASE FX
120 INTEGER COMMON FX(5,5,11)
130 STRING LI(73),ST(12)
150 RUN STNU(CA);COMMON STRINU
160 RUN Y
210 LI=INT((72+SP)/(5+SP));LT=INT(STRINU/LI+0.99)
220 STRING CA2(LI+1);ST="1234567890 "
230 TYPE !!
240 FOR LL=1,LT
242 IF(STRINU).GT.(LI);STRINU=STRINU-LI,NN=LI;GOTO 246
244 NN=STRINU
245 FOR I=1,LI;CA2(I)=" ";NEXT I
246 FOR I=(LL-1)*LI+1,(LL-1)*LI+NN
248 CA2(I-(LL-1)*LI)=CA(I);NEXT I
250 FOR I=1,72;LI(I)=" ";NEXT I
260 FOR J=1,5;FOR K=1,LI
270 GOSUB 500
280 FOR I=1,5;LI((SP+5)*(K-1)+I)=FX(J,I,N)
290 NEXT I;NEXT K
300 TYPE LI;NEXT J;TYPE !!;NEXT LL
310 ERASE FX,LI,ST,STRINU,SP,CA
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
600 N=11;RETURN

```

Y

```

10 REMARK : SUB PROGRAM FOR FL
20 COMMON FX
30 CALL T,U
80 FOR N=1,11;FOR J=1,5;FOR I=1,5
90 FX(I,J,N)=128;NEXT I;NEXT J;NEXT N
100 T(1,0,10);T(5,0,10);T(0,1,10);T(0,5,10)
101 T(0,3,1)
102 T(1,0,2);T(3,0,2);T(5,0,2);U(2,5,2);U(4,1,2)
103 T(1,0,3);T(3,0,3);T(5,0,3);U(2,5,3);U(4,5,3)
104 T(3,0,4);T(0,4,4);U(1,1,4);U(2,1,4)
105 T(1,0,5);T(3,0,5);T(5,0,5);U(2,1,5);U(4,5,5)
106 T(1,0,6);T(3,0,6);T(5,0,6);T(0,1,6);U(4,5,6)
107 T(1,0,7);T(0,5,7);U(2,1,7)
108 T(1,0,8);T(3,0,8);T(5,0,8);T(0,1,8);T(0,5,8)
109 T(1,0,9);T(3,0,9);T(0,5,9);U(2,1,9)
110 REMARK : RESERVE
180 ERASE T,U
190 RETURN

```

T

```

10 REMARK : SUB PROGRAM FOR FL,Y
20 COMMON FX
30 ARG(I,J,N)
40 IF(I).EQ.(0);GOTO 70
50 FOR K=1,5;FX(I,K,N)=132;NEXT K
60 GOTO 80
70 FOR K=1,5;FX(K,J,N)=132;NEXT K
80 RETURN

```

U

```

10 REMARK : SUB PROGRAM FOR FL,Y
20 COMMON FX
30 ARG(I,J,N)
40 FX(I,J,N)=132
50 RETURN

```

HISTO

```

10  REMARK PROGRAMMED BY H. GOTOH, FEB. 2, 1976
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
90  REMARK: ARG(6)= NUMBER OF STEPS
100 ARG(FNAME,NDIM,NREC,BASE,INC,NSTEP)
110 CALL LINK,XFER,DRHIST
120 STRING COM(NDIM+1),LX(22)
125 INTEGER B(NDIM)
127 INTEGE COMMON C(NSTEP)
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)

```

MAXMIN

```

10  REMARK PROGRAMMED BY H. GOTOH, FEB. 2, 1976
20  REMARK PROGRAM FOR SEARCH MAX AND MIN VALUES IN FLOPPY DATA
30  REMARK
40  REMARK: ARG(1)="FILE NAME"
50  REMARK: ARG(2)=DIMENS
60  REMARK: ARG(3)= NRECORDS
100 ARG(FNAME,NDIM,NREC)
110 CALL LINK,XFER
120 STRING COM(NDIM+1),LX(22)
125 INTEGER B(NDIM)
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 XFER(1,2,LX,B);VX=B(1),VN=B(1)
160 FOR I=1,NREC;FOR J=1,NDIM
165 XFER(1,I+1,LX,B)
170 IF(B(J)).GT.(VX);VX=B(J)
180 IF(B(J)).LT.(VN);VN=B(J)
190 NEXT J;NEXT I
200 TYPE %8,!, "MAX VALUE = ",VX,!, "MIN VALUE = ",VN

```

MCFD

```

10 REMARK : N.TAKEUCHI 1976-02-06
12 REMARK : DATA TRANSFER TO FD FROM MCA
14 REMARK
16 REMARK : ARG(1)= "FILE NAME"
18 REMARK : ARG(2)= LEGTH
20 REMARK : ARG(3)= DIMENSION/SECTOR
22 REMARK : ARG(4)= # OF DISK UNIT
24 REMARK
40 ARG(NAME,LEG,DI,DK)
45 STRING LX(22)
50 CALL XFER,LINK,MCA,CREATE
60 L=INT(LEG/DI+1.99),N=0
70 CREATE(DK,NAME,"I", (DI),L);LINK(DK,@NAME,LX)
80 INTEGE B(DI)
90 FOR K=2,L;MCA(1,N,B(1),DI)
100 IF(N+DI).LT.(LEG);XFER(-1,K,LX,B);N=N+DI;NEXT K
110 IF(N+DI).EQ.(LEG);GOTO 140
130 FOR I=N+DI-LEG+1,DI;B(I)=0;NEXT I
140 XFER(-1,K,LX,B)
150 ERASE XFER,LINK,MCA,B,CREATE,LX
160 RETURN

```

MEAN

```

10 REMARK : N.TAKEUCHI 1975-12-24/1976-01-08
12 REMARK : CALUCLATION OF MEAN,STANDARD DEVIATION
14 REMARK
16 REMARK : ARG(1)= "FILE NAME"
18 REMARK : ARG(2)= LEGTH
20 REMARK : ARG(3)= DIMENSION/SECTOR
22 REMARK : ARG(4)= # OF DISK UNIT
24 REMARK : ARG(5)= CONSTANT
26 REMARK
30 ARG(NAME,LEG,DI,DK,CO)
40 CALL LINK,XFER,STOP
50 STRING LX(22)
60 INTEGER A(DI)
70 LINK(DK,@NAME,LX)
80 R1=0,R2=0,N=0
100 FOR K=2,INT(LEG/DI+1.99)
110 XFER(1,K,LX,A)
120 FOR J=1,DI
130 M=A(J)-CO,R1=R1+M,R2=M*M+R2,N=N+1
160 IF(N).GE.(LEG);GOTO 180
170 NEXT J;NEXT K
180 REMARK : TYPE !," TOTAL NUMBER OF DATA =" ,N,!
190 REMARK : TYPE " TOTAL OF C(I)-CO =" ,R1,!
200 REMARK : TYPE " TOTAL OF (C(I)-CO)^2 =" ,R2,!
210 X1=R1/N+CO
220 X2=SQRT(R2/(N-1)-N*(CO-X1)*(CO-X1)/(N-1))
240 X3=X2/X1
300 TYPE !," MEAN VALUE =" ,X1,!
320 TYPE " STANDARD DEVIATION =" ,X2,!
340 TYPE " RELATIVE STA. DEV. =" ,X3
350 RETURN

```

MEFD

```

10 REMARK : N.TAKEUCHI 1976-01-17
12 REMARK : DATA TRANSFER TO FD FROM MEMORY
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
30 STRING LX(22)
40 ARG(NAME,LEG,DI,TYP,DK)
50 CALL XFER,LINK,CREATE
60 L=INT(LEG/DI+1.99),N=0
70 CREATE(DK,NAME,TYP,(DI),L);LINK(DK,@NAME,LX)
80 RUN BUFB(DI,TYP);COMMON A,B
90 FOR K=1,L;FOR I=1,DI
92 B(I)=A((K-1)*DI+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);NEXT K
99 RETURN

```

MEPT

```

10 REMARK : N.TAKEUCHI 1976-01-17/1976-02-06
12 REMARK : DATA TRANSFER TO PT FROM MEMORY
14 REMARK
16 REMARK : ARG(1)= LEGTH
18 REMARK : ARG(2)= DIMENSION/HORIZONTAL
20 REMARK : ARG(3)= DIMENSION/VERTICAL
22 REMARK
30 COMMON A
40 ARG(LEG,HO,VE)
50 TYPE !
60 FOR I=1,LEG;TYPE A(I)
70 IF(I).EQ.(HO*INT(I/HO));TYPE !
75 IF(I).EQ.(VE*INT(I/VE));TYPE !
80 NEXT I;TYPE !
90 RETURN

```

MIXF

```

10 REMARK : N.TAKEUCHI 1976-01-24
12 REMARK : MIX FOR DATA FILES ON DISK
14 REMARK : NO ARGUMENT
18 STRING TYP(2),BUFF(7)
20 ASK "          LEGTH= ",LEG
22 ASK "    DIMENSION/SECTOR= ",DI
24 ASK "          FILE TYPE= ",TYP
26 ASK "    NUMBER OF FILES= ",NU
30 STRING NAME(NU+1,7),LX1(22),LX2(22)
32 INTEGE DK(NU+1)
34 L=INT(LEG/DI+1.99)
40 TYPE "-----";TYPE %2
42 FOR I=1,NU
44 TYPE !,I,"          FILE NAME= "
46 ASK NAME(I)
48 ASK "    # OF DISK UNIT= ",DK(I)
50 NEXT I
52 TYPE !,"-----",!
54 ASK "    NEW FILE NAME= ",NAME(I)
56 ASK "    # OF DISK UNIT= ",DK(I)
57 TYPE !;TYPE %6
60 CALL CREATE,XFER,LINK,BUFA,BUFB,STOP
62 BUFA(DI,TYP);BUFB(DI,TYP);COMMON A,B
63 FOR F=1,7;BUFF(F)=NAME(I,F);NEXT F
64 CREATE(DK(I),BUFF,TYP,(DI),L)
66 LINK(1,@BUFF,LX1)
70 FOR K=2,L
72 FOR I=1,DI;A(I)=0;NEXT I
74 FOR I=1,NU;GOSUB 90;NEXT I
76 XFER(-1,K,LX1,A);NEXT K
80 RETURN ;STOP
90 FOR F=1,7;BUFF(F)=NAME(I,F);NEXT F
94 LINK(DK(I),@BUFF,LX2)
96 XFER(1,K,LX2,B)
98 FOR II=1,DI;A(II)=A(II)+B(II);NEXT II
99 RETURN

```

PEAK1

```

10 ARG(K,L,M,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
35 IF(TY).EQ.("YES");TYPE %5.00,P,%7.00,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 IF(TY).EQ.("YES");TYPE %6.02,PB,%9.00,AREA,%5.00,SG,%5.02,RELSG,"%",!

```

PEAREA

```

10 REMARK: PROGRAMMED BY H. GOTOH, 1976-1-5
20 REMARK: PROGRAM FOR STATISTICAL ANALYSIS OF PEAK AREA OF REPEATED
MEASUREMENT
30 STRING TAG(7),LX(22),D(11),COM(9);D="01234567890"
32 COMMON TY;ERASE TY;STRING COMMON TY(4)
33 COMMON B;ERASE B;INTEGER COMMON B(100)
34 COMMON AREA;ERASE AREA;COMMON AREA
35 INTEGER A(100),I1(3),NN,NDATA,N,IDATA
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 PEAK1,LINK,XFER,MCA,CREATE,SPY
80 ASK "INITIAL CHANNEL OF RECORD = ",INIT;ASK "DATA LENGTH
= ",LENG
85 N=INT(LENG/100)
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 TYPE LABEL DATE START..... ? ",TY
110 IF(TY).EQ.("YES");TYPE 1," MAX COUNT",!
120 IF(TY).EQ.("YES");TYPE " LABEL DATE START CHNL COUNT CENTER
APEA SGM REL.SGM",!
130 FOR NN=NDATA,NDATA+NO-1
140 IDATA=NN, IDATA=IDATA-1000*INT(IDATA/1000), I1(1)=INT(IDATA/1000), I1(
2)=INT(IDATA/100)-10*I1(1), I1(3)=IDATA-10*INT(IDATA/100)
150 FOR J=6,-1,4;TAG(J)=D(I1(J-3)+1);NEXT J
160 LINK(1,@TAG,LX)
170 XFER(1,1,LX,A)
180 FOR I=1,8;COM(I)=A(I);NEXT I
190 IF(TY).EQ.("YES");TYPE TAG," ",COM
200 FOR I=1,N;XFER(1,I+1,LX,A);MCA(-1,(I-1)*120+INIT,A(1),120);NEXT I
210 IF(LENG-N*120).NE.(0);XFER(1,N+2,LX,A);MCA(-1,N:120+INIT,A(1),LENG
-N*120)
220 PEAK1(X1,X2,X3,X4)
225 B(NN)=AREA
230 NEXT NN
235 TAG="PEAK";IF(SPY(1,TAG)).NE.(0);SCRATCH FDI @TAG
236 CREATE(1,TAG,"I", (100),2);LINK(1,@TAG,LX);XFER(-1,2,LX,B)
240 TYPE "THE WORK COMPLETED"

```

PTFD

```

10 REMARK : N.TAKEUCHI 1976-01-17
12 REMARK : DATA TRANSFER TO FD FROM MEMORY
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
30 ARG(NAME,LEG,DI,TYP,DK)
40 STRING LX(22);TYPE !
50 CALL XFER,LINK,CREATE,BUFB
60 L=INT(LEG/DI+1.99),N=0
70 CREATE(DK,NAME,TYP,(DI),L);LINK(DK,@NAME,LX)
80 BUFB(DI,TYP);COMMON B
90 FOR K=1,L;FOR I=1,DI
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);NEXT K
99 RETURN

```


PTME

```

10 REMARK : N.TAKEUCHI 1976-01-17/1976-02-06
12 REMARK : DATA TRANSFER TO MEMORY FROM PT
14 REMARK
16 REMARK : ARG(1)= LEGTH
18 REMARK : ARG(2)= "DATA TYPE" ("R","I","S")
20 REMARK
30 ARG(LEG,TYP)
40 TYPE !
50 RUN BUFA(LEG,TYP);COMMON A
60 FOR I=1,LEG;ASK A(I);NEXT I
70 TYPE !
80 RETURN

```

PTTOFD

```

1 REMARK: PREPARED BY H. GOTOH, DEC. 18, 1975
2 REMARK: PROGRAM FOR READ DATA FROM PAPER TAPE TO FLOPPY DISK
3 REMARK: MODIFIED BY H. GOTOH, JAN. 28, 1976
5 CALL SPY,CREATE,XFER,LINK
10 ASK "LENGTH OF DATA = ",LEN
12 ASK "DIMENS/SECTOR = ",NN
15 N=INT(LEN/NN)
17 STRING CH(3),TAG(7)
20 ASK "NAME OF FILE(IN JUST 6 LETTERS) = ",TAG
30 ASK "DO YOU WANT TO CHANGE COMMENT ? ",CH
35 COMMON COMMEN
40 IF(CH).EQ.("NO");GOTO 100
50 ERASE COMMEN
60 STRING COMMON COMMEN(NN)
70 ASK "COMMENT = ",!,COMMEN
100 IF(SPY(1,TAG)).NE.(0);TYPE "FILE NAME EXISTS !",!;STOP
110 CREATE(1,TAG,"I",(NN),N+2)
120 INTEGER A(NN)
130 STRING LX(22)
140 LINK(1,@TAG,LX)
145 IF(N).EQ.(0);GOTO 200
150 FOR I=1,N
160 FOR J=1,NN;ASK A(J);NEXT J
185 XFER(-1,I+1,LX,A)
190 NEXT I
200 IF(LEN-NN*N).EQ.(0);GOTO 300
205 FOR J=1,NN;A(J)=0;NEXT J
210 FOR J=1,LEN-NN*N;ASK A(J);NEXT J
240 XFER(-1,N+2,LX,A)
300 FOR I=1,NN;A(I)=COMMEN(I);NEXT I
310 XFER(-1,1,LX,A)
320 ERASE CREATE,XFER,LINK,SPY

```

PTTOMC

```

1 REMARK PREPARED BY H. GOTOH, DEC. 11, 1975
2 REMARK PROGRAM FOR READ DATA FROM PAPER TAPE TO MCA
3 REMARK      PTTOMC(INITIAL CHANNEL #, LENGTH)
10 ARG(INIT,LENG)
20 NN=INT(LENG/10)
30 DIMENS A(10)
40 CALL MCA
50 FOR I=1,NN
60 FOR J=1,10
70 ASK A(J);NEXT J
80 MCA(-1,10*(I-1)+INIT,A(1),10)
90 NEXT I
100 FOR J=1,LENG-10*NN
110 ASK A(J);NEXT J
120 MCA(-1,10*NN+INIT,A(1),LENG-10*NN)
130 TYPE "END",!
140 GOTO 140

```

REDISK

```

1 REMARK PREPARED BY H. GOTOH, DEC. 11, 1975
2 REMARK PROGRAM FOR READ DATA FROM DISK TO MCA
3 REMARK      REDISK(INITIAL CHANNEL #, LENGTH)
5 ARG(INIT,LENG)
10 N=INT(LENG/120)
15 STRING TAG(7)
17 ASK "FILENAME = ",TAG
20 INTEGER A(120)
30 CALL MCA,XFER,LINK
35 STRING LX(22),COMMEN(120)
37 LINK(1,0TAG,LX)
38 XFER(1,1,LX,A)
39 FOR I=1,120;COMMEN(I)=A(I);NEXT I;TYPE "COMMENT = ",COMMEN,!
45 FOR I=1,N
50 XFER(1,I+1,LX,A)
60 MCA(-1,(I-1)*120+INIT,A(1),120)
70 NEXT I
75 IF (LENG-N*120).EQ.(0);GOTO 150
80 XFER(1,N+2,LX,A)
100 MCA(-1,N*120+INIT,A(1),LENG-N*120)
150 EFASE XFER,LINK

```

REMEAN

```

10 REMARK : N.TAKEUCHI 1975-12-25/1976-01-07
12 REMARK : CALUCLATION OF MEAN,STNDARD DEVIATION AT REPEAT
16 REMARK
18 REMARK : ARG(1)= "FILE NAME"
20 REMARK : ARG(2)= LEGTH
22 REMARK : ARG(3)= DIMENSION/SECTOR
24 REMARK : ARG(4)= # OF DISK UNIT
26 REMARK : ARG(5)= DIMENSION/BLOCK
28 REMARK : ARG(6)= CONSTANT
30 REMARK
40 ARG(NAME,LEG,DI,DK,DIN,CO)
42 CALL LINK,XFER,STOP
44 STRING LX(22)
46 INTEGE A(DI);LINK(DK,@NAME,LX)
50 N=INT(LEG/DIN);TYPE !!
60 R1=0,R1=0,R11=0,R22=0,GA=0,GB=0,H=0,K=1
100 K=K+1;XFER(1,K,LX,A)
110 SW=1;IF(DI-GA-DIN+GB)120,150,160
120 FOR I=GA+1,DI
130 A(I)=A(I)-CO,R1=R1+A(I),R2=R2+A(I)*A(I);NEXT I
140 GB=GB+DI-GA,GA=0;GOTO 100
150 SW=0
160 FOR I=GA+1,GA+DIN-GB
170 A(I)=A(I)-CO,R1=R1+A(I),R2=R2+A(I)*A(I);NEXT I
180 X1=R1/DIN,X2=SQRT(R2/(DIN-1)-DIN*X1*X1/(DIN-1))
190 TYPE %4
200 TYPE H*DIN+1," ----", (H+1)*DIN," "
210 TYPE %8.5
220 TYPE X1+CO," ",X2," ",X2/(X1+CO),!
230 R11=R11+X1,R22=R22+X1*X1
240 R1=0,R2=0,H=H+1
250 IF(H).GE.(N);GOTO 300
260 GA=GA+DIN-GB,GB=0
270 IF(SW).EQ.(1);GOTO 110
280 GA=0;GOTO 100
300 IF(H).LE.(1);GOTO 360
310 X1=R11/H+CO,X2=SQRT(R22/(H-1)-H*(X1-CO)*(X1-CO)/(H-1))
320 TYPE %4;TYPE !
330 TYPE 1," ----",H," "
340 TYPE %8.5
350 TYPE X1," ",X2," ",X2/X1
360 RETURN

```

RENAMD

```

10 REMARK: ORIGINATED BY H. GOTOH, JAN. 4, 1976
20 REMARK: REVISED BY
30 REMARK: PROGRAM FOR RENAME FILEIN DISK
50 STRING OLD(7),NEW(7),LX(22),LXX(22)
60 ASK "OLD NAME = ",OLD
70 ASK "NEW NAME = ",NEW
80 INTEGER A(120)
90 CALL XFER,LINK,CREATE,SPY,STOP
100 IF(SPY(1,OLD)).EQ.(0);TYPE "OLD NAME DOES NOT EXIST !";STOP
110 IF(SPY(1,NEW)).NE.(0);TYPE "NEW NAME ALREADY EXISTS !";STOP
120 ASK "NUMBER OF TRACKS = ",NN
130 CREATE(1,NEW,"I",(120),NN-1)
140 LINK(1,@OLD,LX);LINK(1,@NEW,LXX)
150 FOR I=1,NN-1
160 XFER(1,I,LX,A);XFER(-1,I,LXX,A)
170 NEXT I

```

SCRA

```

10 REMARK : N.TAKEUCHI 1976-01-09/1976-02-03
12 REMARK : DELETE THE CONTENTS ON #0 DISK
14 REMARK
16 REMARK : NO ARGUMENT
18 REMARK
20 CALL SPY;STRING ST(12);ST="0123456789 "
30 STRING TAG(7)
40 ASK "NAME = ",TAG
45 FOR I=1,11
50 IF(TAG(I)).EQ.(ST(I));ERASE SPY,TAG;STOP
55 NEXT I
60 IF(SPY(0,TAG)).EQ.(0);GOTO 90
70 SCRATCH FD0 @TAG
80 ERASE TAG;GOTO 30
90 TYPE TAG," FILE NOT EXIST ON #0 DISK",!;GOTO 80

```

SCRA1

```

10 REMARK : N.TAKEUCHI 1976-01-09/1976-02-03
12 REMARK : DELETE THE CONTENTS ON #1 DISK
14 REMARK
16 REMARK : NO ARGUMENT
18 REMARK
20 CALL SPY;STRING ST(12);ST="0123456789 "
30 STRING TAG(7)
40 ASK "NAME = ",TAG
45 FOR I=1,11
50 IF(TAG(I)).EQ.(ST(I));ERASE SPY,TAG;STOP
55 NEXT I
60 IF(SPY(1,TAG)).EQ.(0);GOTO 90
70 SCRATCH FD1 @TAG
80 ERASE TAG;GOTO 30
90 TYPE TAG," FILE NOT EXIST ON #1 DISK",!;GOTO 80

```

SCSA

```

10 REMARK : N.TAKEUCHI 1976-01-09/1976-02-03
12 REMARK : SCRATCH TO FILE ON #0 DISK AND SAVE TO ONE FROM CORE
14 REMARK
16 REMARK : NO ARGUMENT
18 REMARK
20 CALL SPY;STRING ST(12);ST="0123456789 "
30 STRING TAG(7)
40 ASK "NAME= ",TAG
50 FOR I=1,11;IF(TAG(I)).EQ.(ST(I));ERASE SPY,TAG;STOP
55 NEXT I
60 IF(SPY(0,TAG)).EQ.(0);GOTO 80
70 SCRATCH FD0 @TAG
80 SAVE FD0 @TAG
90 ERASE TAG;GOTO 30

```

SCSA1

```

10 REMARK : N.TAKEUCHI 1976-01-09/1976-02-03
12 REMARK : SCRATCH TO FILE ON #1 DISK AND SAVE TO ONE FROM CORE
14 REMARK
16 REMARK : NO ARGUMENT
18 REMARK
20 CALL SPY;STRING ST(12);ST="0123456789 "
30 STRING TAG(7)
40 ASK "NAME= ",TAG
50 FOR I=1,11;IF(TAG(I)).EQ.(ST(I));ERASE SPY,TAG;STOP
55 NEXT I
60 IF(SPY(1,TAG)).EQ.(0);GOTO 80
70 SCRATCH FDI @TAG
80 SAVE FDI @TAG
90 ERASE TAG;GOTO 30

```

STNU

```

10 REMARK : N.TAKEUCHI 1976-01-27
12 REMARK : CALUCLATION TO CHARACTER NUMBERS OF ARGUMENT STRING
14 REMARK
16 REMARK : ARG(1)= "STRING"
18 REMARK
30 ARG(SN);I=0,A=0
40 I=I+1;A=SN(I);IF(A).NE.(0);GOTO 40
50 COMMON STRINU;ERASE STRINU
60 INTEGE COMMON STRINU
70 STRINU=I-1
80 RETURN

```

STNU1

```

10 REMARK : N.TAKEUCHI 1976-01-27
12 REMARK : CALUCLATION TO CHARACTER NUMBERS OF ARGUMENT STRING
14 REMARK
16 REMARK : ARG(1)= "STRING"
18 REMARK
30 ARG(SN);I=0,A=0
40 I=I+1;A=SN(I);IF(A).NE.(0);GOTO 40
50 COMMON STRINU;ERASE STRINU
60 INTEGE COMMON STRINU
70 STRINU=I-1
75 TYPE %2;TYPE SN,"=",STRINU," LETTERS",!
80 RETURN

```

TRNDA

```

10 REMARK : N.TAKEUCHI 1975-12-24 /1976-02-04
12 REMARK : DATA TRANSFER WITH MODIFYING ARRAY,TYPE ON DISK TO DISK
14 REMARK
16 REMARK : ARG(1)= LEGTH (TOTAL # OF TRANSFER DATA)
18 REMARK : ARG(2)= SOURCE : "FILE NAME"
20 REMARK : ARG(3)= DIMENSION/SECTOR
22 REMARK : ARG(4)= "FILE TYPE" ("R","I","S")
24 REMARK : ARG(5)= # OF DISK UNIT
26 REMARK : ARG(6)= OBJECT : "FILE NAME"
28 REMARK : ARG(7)= DIMENSION/SECTOR
30 REMARK : ARG(8)= "FILE TYPE" ("R","I","S")
32 REMARK : ARG(9)= # OF DISK UNIT
34 REMARK
40 ARG(LEG, TAG, DI, TYP, DK, TAGN, DIN, TYPN, DKN)
50 CALL LINK, XFER, CREATE
60 STRING LX(22), LXN(22)
70 RUN BUFA(DI, TYP); RUN BUFB(DIN, TYPN); COMMON A, B
75 L=INT(LEG/DIN+1.99)
80 CREATE(DKN, TAGN, TYPN, (DIN), L)
90 LINK(DK, @TAG, LX); LINK(DKN, @TAGN, LXN)
95 GA=0, GB=0, N=0, H=1, K=1
100 K=K+1; XFER(1, K, LX, A)
110 SW=1; IF(DI-GA-DIN+GB)120, 150, 160
120 FOR I=GA+1, DI
130 B(I-GA+GB)=A(I), N=N+1; IF(N).GE.(LEG); GOTO 300
140 NEXT I
145 GB=GB+DI-GA, GA=0; GOTO 100
150 SW=0
160 FOR I=GA+1, GA+DIN-GB
170 B(I-GA+GB)=A(I), N=N+1; IF(N).GE.(LEG); GOTO 300
180 NEXT I
190 H=H+1; XFER(-1, H, LXN, B); GA=GA+DIN-GB, GB=0
200 IF(SW).EQ.(1); GOTO 110
210 GA=0; GOTO 100
300 XFER(-1, H+1, LXN, B)
310 ERASE LINK, XFER, LX, LXN, CREATE
320 RETURN

```

TYP OUT

```

1 REMARK: PREPARED BY H. GOTOH, JAN.13,1976
2 REMARK: PROGRAM FOR TYPE OUT DATA IN DISK
3 REMARK: TYP OUT(LENGTH)
5 ARG(LENG)
15 STRING TAG(7)
17 ASK "FILENAME = ", TAG
20 INTEGER A(120), M, N, I, J, K
25 N=INT(LENG/120); TYPE %6.00
30 CALL XFER, LINK
35 STRING LX(22), COMMEN(120)
37 LINK(1, @TAG, LX)
38 XFER(1, 1, LX, A)
39 FOR I=1, 120; COMMEN(I)=A(I); NEXT I; TYPE "COMMENT = ", COMMEN, !
45 FOR I=1, N
50 XFER(1, I+1, LX, A)
60 FOR J=1, 12; FOR K=1, 10; TYPE A(K+(J-1)*10); NEXT K; TYPE !; NEXT J
70 NEXT I
75 IF (LENG-N*120).EQ.(0); GOTO 150
80 XFER(1, N+2, LX, A)
100 M=LENG-N*120; IF(INT(M/10)).NE.(0); FOR J=1, INT(M/10); FOR K=1, 10; TYP
E A(K+(J-1)*10); NEXT K; TYPE !; NEXT J
110 M=LENG-120*N-10*INT(M/10); IF(M).NE.(0); FOR K=1, M; TYPE A(10*K); NE
XT K
150 ERASE XFER, LINK

```

WRDISK

```

1  REMARK PREPARED BY H. GOTOH, DEC. 11, 1975
2  REMARK PROGRAM FOR WRITE DATA FROM MCA TO DISK
3  REMARK      WRDISK(INITIAL CHANNEL.#, LENGTH)
4  REMARK PREPARE STRING COMMON TAG(7),COMMEN(120)
5  ARG(INIT,LENG)
6  REMARK: REVISED BY H. GOTOH, DEC. 25,1975
8  N=INT(LENG/120)
10 COMMON TAG,COMMEN
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

```

YG5000

```

1  REMARK: PREPARED BY H. GOTOH, DEC. 19, 1975
2  REMARK: PROGRAM FOR READ DATA OF 5000 TIMES REPETITION FROM PAPER
TAPE TO FLOPPY DISK
5  CALL SPY,CREATE,XFER,LINK
10 ASK "LENGTH OF DATA = ",LEN
15 N=INT(LEN/120)
17 STRING CH(3),TAG(7)
20 ASK "NAME OF FILE(IN JUST 6 LETTERS) = ",TAG
30 ASK "DO YOU WANT TO CHANGE COMMENT ? ",CH
35 COMMON COMMEN
40 IF(CH).EQ.("NO");GOTO 100
50 ERASE COMMEN
60 STRING COMMON COMMEN(120)
70 ASK "COMMENT = ",!,COMMEN
100 IF(SPY(1,TAG)).NE.(0);TYPE "FILE NAME EXISTS !",!;STOP
110 CREATE(1,TAG,"I",(120),N+2)
120 INTEGER A(120)
130 STRING LX(22)
140 LINK(1,@TAG,LX)
142 FOR I=1,120;A(I)=COMMEN(I);NEXT I
143 XFER(-1,1,LX,A)
145 IF(N).EQ.(0);GOTO 200
150 FOR I=1,N
160 FOR J=1,24;FOR K=1,5;ASK X,A(K+5*(J-1));NEXT K;ASK X;NEXT J
185 XFER(-1,I+1,LX,A)
187 MCA(-1,120*(I-1),A(1),120)
190 NEXT I
200 IF(LEN-120*N).EQ.(0);GOTO 320
205 FOR J=1,120;A(J)=0;NEXT J
210 FOR J=1,INT((LEN-120*N)/5)
220 FOR K=1,5;ASK X,A(K+5*(J-1));NEXT K;ASK X;NEXT J
225 IF(LEN-120*N-5*J).EQ.(0);GOTO 240
230 FOR K=5*J+1,LEN-120*N;ASK X,A(K);NEXT K
240 XFER(-1,N+2,LX,A)
250 MCA(-1,120*N,A(1),LEN-120*N)
320 ERASE CREATE,XFER,LINK,SPY

```

YAGIP1

```

2  REMARK: PROGRAM FOR YAGI'S PAPER TAPE INTO FLOPPY DISK
3  REMARK: PREPARED BY H. GOTOH, JAN. 28, 1976
5  CALL SPY, CREATE, XFER, LINK
10 ASK "LENGTH OF DATA = ", LEN
12 ASK "DIMENS/SECTOR = ", NN
15 N=INT(LEN/NN)
17 STRING CH(3), TAG(7)
20 ASK "NAME OF FILE(IN JUST 6 LETTERS) = ", TAG
30 ASK "DO YOU WANT TO CHANGE COMMENT ? ", CH
35 COMMON COMMEN
40 IF(CH).EQ.("NO");GOTO 100
50 ERASE COMMEN
60 STRING COMMON COMMEN(NN)
70 ASK "COMMENT = ", I, COMMEN
100 IF(SPY(1, TAG)).NE.(0);TYPE "FILE NAME EXISTS !", I;STOP
110 CREATE(1, TAG, "I", (NN), N+2)
120 INTEGER A(NN), B(2)
130 STRING LX(22)
140 LINK(1, TAG, LX)
142 FOR I=1, NN; A(I)=COMMEN(I); NEXT I
143 XFER(-1, I, LX, A)
145 IF(N).EQ.(0);GOTO 200
150 FOR I=1, N
155 FOR L=1, NN/10
160 FOR J=1, 10
170 ASK A(J+(L-1)*10); NEXT J
180 ASK B(1)
182 NEXT L
185 XFER(-1, I+1, LX, A)
190 NEXT I
200 IF(LEN-NN*N).EQ.(0);GOTO 320
205 FOR J=1, NN; A(J)=0; NEXT J
210 FOR J=1, LEN-NN*N; ASK A(J); NEXT J
240 XFER(-1, N+2, LX, A)
320 ERASE CREATE, XFER, LINK, SPY

```


YAGIPT

```

2 REMARK: PROGRAM FOR YAGI'S PAPER TAPE INTO FLOPPY DISK
3 REMARK: PREPARED BY H. GOTOH, JAN. 28, 1976
5 CALL SPY,CREATE,XFER,LINK
10 ASK "LENGTH OF DATA = ",LEN
12 ASK "DIMENS/SECTOR = ",NN
15 N=INT(LEN/NN)
17 STRING CH(3),TAG(7)
20 ASK "NAME OF FILE(IN JUST 6 LETTERS) = ",TAG
30 ASK "DO YOU WANT TO CHANGE COMMENT ? ",CH
35 COMMON COMMEN
40 IF(CH).EQ.("NO");GOTO 100
50 ERASE COMMEN
60 STRING COMMON COMMEN(NN)
70 ASK "COMMENT = ",!,COMMEN
100 IF(SPY(1,TAG)).NE.(0);TYPE "FILE NAME EXISTS !",!;STOP
110 CREATE(1,TAG,"I",(NN),N+2)
120 INTEGER A(NN),B(2)
130 STRING LX(22)
140 LINK(1,@TAG,LX)
142 FOR I=1,NN;A(I)=COMMEN(I);NEXT I
143 XFER(-1,1,LX,A)
145 IF(N).EQ.(0);GOTO 200
150 FOR I=1,N
155 FOR L=1,NN/5
160 FOR J=1,5
165 FOR K=1,2;ASK B(K);NEXT K
170 A(J+5*(L-1))=B(1)*1000000+B(2)
175 NEXT J
180 ASK B(1)
182 NEXT L
185 XFER(-1,I+1,LX,A)
190 NEXT I
200 IF(LEN-NN*N).EQ.(0);GOTO 320
205 FOR J=1,NN;A(J)=0;NEXT J
210 FOR J=1,LEN-NN*N;ASK A(J);NEXT J
240 XFER(-1,N+2,LX,A)
320 ERASE CREATE,XFER,LINK,SPY

```