九州大学学術情報リポジトリ
Kyushu University Institutional Repository

[01_01]九州大学大型計算機センター広報 : 1(1)

https://doi.org/10.15017/4843154

出版情報：九州大学大型計算機センター広報. 1 (1), pp.1-68, 1968-04. 九州大学大型計算機センター
バージョン：
権利関係：

KYUSHU UNIVERSITY

資　　料

FACOM230－60のALGOLとFORTRANについて

まえがき

大型電子計算機FACOM 230－60のソフトウエアシステムはかなり膨大な体系となる予定であるが、ななかでも利用者にとくに関心があると思われるのは、ALGOLとFORTRANであろう。富士通では現在続々とソフトウエアを開発中でALGOLやFORTRANに関してはコーディングをほゞ完了し、現在デバッグにとりかゝる段階だそうであり、従ってやがて内部仕様も一応安定し、利用者用のマニュアル（使用手引書）の出版される日もま近いことと期待される。

これまでにALGOLやFORTRANの外部仕様書案は2、3版出されている。これらは将来変更追加があるものと思われるけれども、大要はすでに固っており、従ってこれらが果してどのようなものであるのか、その大略を説明してもよい時期であると思われる。

昨1967年5月1日ALGOLおよびFORTRANの日本工業規格が制定されたが、F230－60 ALGOLはこの日本工業規格の原案水準5060、F230－60 FORTRANは同原案水準7000をもとに、さらにいくつかの機能を追加したものである。

従って、F230－60 ALGOLおよびFORTRAN の本質的な部分を知りたい方は本章末に掲げた日本規格協会発行の規格票を参照して載きたい。

なおF230－60のALGOLやFORTRANについて詳しく説明する余裕はとてもないので、ここではALGOLについては九大中央計数施設で使用されているALGOL－H，FORTRAN については東大センターで使用されているHARP について、読者は一応知っているものとして話をすすめる。そしてJISで規定されていて、HARPやALGOL－Hと異なる部分や、とくにJISに追加された部分について説明を加えるつもりである。

なお、説明の順序と内容は次のようなものである。

○ALGOL	F230－60 ALGOLとALGOL－Hとの相違点についての重点的な説明。
○F230－60 ALGOLと日本工業規格原案（水準5060）との相違点	F230－60 ALGOLがJIS原案水準5060に追加した機能と、それに加えた制限の一覧。
○FORTRAN	F230－60 FORTRANにおいてHARPにはない機能の重点的な説明。
○HARPにおいてF230－60 FORTRANと相違する部分	HARPにおける機能のなかで、F230－60 FORTRANに抵触する部分の一覧。

ALGOL

1　プログラムの記入形式

原プログラムはカードへ以下のように記入される。

欄　1　　＊がせん孔してあるときはデバッグ制御に従う文が書かれているものとみなす。他の場合は空白。

欄 2～72　原始プログラムを表1の表記法で書く。

欄 73～80　行番号

表1　区切り記号の表記法

ALGOL 記号	表記法	ALGOL 記号	表記法
go to	¦GOTOまたは¦GO¦TO	≠	¦NQまたは¦NE
if	¦IF	≡	¦EQV
then	¦THEN	⊃	¦IMP
else	¦ELSE	∧	¦AND
for	¦FOR	∨	¦ŌR
do	¦DO	¬	¦NŌT
step	¦STEP	+	+
until	¦UNTIL	−	−
while	¦WHILE	×	*
comment	¦COMMENT	/	/
begin	¦BEGIN	÷	//
end	¦END	↑	**
Boolean	¦BOOLEAN	,	,
integer	¦INTEGER	.	.
real	¦REAL	10	¦
long	¦LONG	:	..
array	¦ARRAY	;	.;
switch	¦SWITCH	:=	=
procedure	¦PROCEDURE	(	(
string	¦STRING	)	)
label	¦LABEL	[	(/または(
value	¦VALUE	]	/)または)
<	¦LSまたは¦LT	‘	¦¦
≦	¦LQまたは¦LE	’	¦¦
=	¦EQLまたは¦EQ	true	¦TRUE
≧	¦GQまたは¦GE	false	¦FALSE
>	¦GRまたは¦GT	␣	

2　型と量

一般に名前の識別には12桁まで有効である。

型には整数型、実数型、長精度の実数型、論理型の4種類ある。

整数型の量のとりうる最大値は$+2^{35}-1$すなわち34,359,738,367,最小値は-2^{35}すなわち−34,359,738,368である。

実数型の量の数値の有効桁数は10進で8桁弱、指数部は−76から+76までの範囲の整数値をとる。

長精度実数型の量の数値の有効桁数は10進16桁、指数部のとりうる範囲は実数型と同様である。数値は long 4.5, − long.$42_{10}1$ 等と表わされる。

3　式

算術式では混合演算が許される。

整数型、実数型及び長精度実数型の順に順位が高くなるものとするとき、これらに加，減，乗，除及び乗べきの演算を施こした結果の型は以下の規則に従がう。

(1)　+、−、×の場合、順位の高い被演算数の型に一致する。

(2)　／の場合、被演算数がともに整数型のとき、結果は実数型になる。他の場合は(1)と同様。

(3)　÷の場合、結果の型は整数型である。a ÷ bの計算は次の式に従がう。

|b|＜1のときERROR

|b|≧1のとき

a ÷ b＝sign（a／b）×entier（abs（entier（a＋0.5）／entier（b＋0.5））

ここでsign，entier，およびabsは標準関数である。entierの関数値は、そのパラメータを越えない最大の整数値をとる。

(4)　乗べき↑の場合、a ↑ bの演算結果は次の規則によって与えられる。bが符号のない整数であれば

a ↑ b＝a×a×……×a（b回）

として計算され、その型はaと同じ型になる。

a＞0のときは

a ↑ b＝exp（b×ln（a））,

a＝0 でb＞0ならば

a ↑ b＝0,

a＜0 でabs（entier（b＋0.5）－b）＜2^{-a}であれば、

a ↑ b＝（－1）↑entier（b＋0.5）×exp（b×ln（abs（a））

として計算され、結果の型は（2）の場合と同じになる。上記の定義に含まれない場合は、エラーメッセィジを出した後、結果を0として演算を続行する。

式の左辺の型と右辺の算術式の演算で得られた結果の型が異なっているときは、自動的に右辺の型へ変換されたのち代入される。

4　文

繰り返し文では

for I：＝A　while B do S；

が許される。これはAの値をIに代入しながら、論理式Bの値が真である間、文Sを繰り返し実行することを意味する。

型の宣言では整数型、実数型、論理型のほかに長精度実数型の宣言文、例えば

long X，Y；

がある。

配列の宣言では、添字域がプログラム実行中に動的に割当てられるいわゆる動的な配列（dynamic array）が許される。従って配列の宣言文の添字域の上限および下限を算術式で与えてもよい。例えば

array　A〔N，M＋5〕；

と書くことができる。上限及び下限の値はブロックに入るごとに一度づつ求められる。

手続きの実パラメータには式、配列名及びパラメータのない手続き名はもちろん、記号列、名札、スイッチ名及びパラメータのある手続き名も許される。しかしいかなる意味においても、手続きの回帰的な呼び出しは禁止される。

手続きには手続き文によって呼び出される手続きのほかに、関数呼び出しのできる手続きがある。後者は手続きの宣言の際に、頭に型の宣言詞をつけて定義しなければならない。

標準関数については、ＪＩＳ原案に保持することを定められたもの、及び保持することを期待されたものはすべて保持し、さらにいくつかの標準関数が追加されている。これらの標準関数でオーバーフロやアンダーフローが発生するとそれぞれの場合に応じて、ある一定の値を代入し、その旨警告して計算を続行する。

5　入出力手続き

Ｆ230－60 ALGOL の入出力手続きにはＪＩＳ原案水準６０が採用され、これにいくつかの手続きが追加されている。ＡＬＧＯＬ－Ｈに馴れた利用者にはＪＩＳの入出力手続きは非常に面倒に感じられるだろう。ＪＩＳでは書式を指定しなければならないのに対して、ＡＬＧＯＬ－Ｈでは自由書式の方針に従っているからである。

入力または出力の際には陽に、または暗黙のうちにどこから、どんな形式で、何へ読み込むのか、または何を、どんな形式で、どこへ書き出すのかを指定する必要がある。これらは論理機番（どこ）、書式（どんな形式）、項目（何）によって指定され、入力または出力の別は手続き名によって識別される。論理機番は入出力機器に対応して、処理系によってきめられた 整数値をとる算術式によって指定する。書式は入力または出力のための外部媒体上の情報の形式、およびこの形式を指定する記号列である。項目とは入力の際には読み込まれるデータの割り当てられる変数であり、出力の際には書き出される値を計算する式または記号列である。

論理機番、書式および項目は入出力手続きによって、必要なものがパラメータとして陽にまたは暗黙のうちに、かつ直接的にまたは間接的に与えられる。

例えばもっとも基本的な手続きである inlist，outlist では、まず設定（配置）手続きを呼ぶことによって書式を決め、項目指定手続きを呼ぶことによって項目が指定されたあとで、論理機番を与える算術式と、その設定手続き名及び項目指定続き名を直接のパラメータとして呼ぶことにより、間接的に書式および項目の指定がなされて、入力または出力が行われる。

項目を直接のパラメータとして指定する入出力手続きには

```
inreal ( channel, destinator )
```

等一群の手続きがある。channel は論理機番を指定し、destinator には変数名を直接与える。書式は標準書式が暗黙のうちに指定される。

以上はＪＩＳ 60に規定された入出力手続きである。このほかにＪＩＳ 60に規定されてない手続きがある。これらはＡＬＧＯＬ－Ｈの入出力手続きに似ているけれども異なった点も多い。

例えば入力手続きとして

```
read ( destinator )
```

があるがdestnator に書くことのできる変数名は１つである。

出力手続きには

```
print （E）
```

がある。Eは式である。このままでは出力は標準書式に従って、すなわち一定の桁数をもって出力される。若しその書式に不満ならばこの手続きの呼び出しの前に書式変更のための手続きを呼び出さなければならない。そのための手続きとしてrealform，longform integerform 等の手続きがあり、書式を示す記号列がその手続きのパラメータとして与えられる。

その他の出力手続きには

```
printstring ( string )
```

があり、また出力全体の配置を決める出力手続きには space （Ｅ），crlf（復帰改行），lfeed （crlf の１０個の繰り返し）及び改頁を行なうｅｊｅｃｔがある。

6　制御カード

利用者は制御カードによりかなり広範な指示を処理系に与えることができる。これらの指示のないものは標準的指示がなされたものと解釈される。

以下制御可能な項目について述べる。

○リスト制御

原始プログラムおよび目的プログラムのリストをとるかどうかを指定する。

○せん孔制御

相対番地形式の目的プログラムをカードにせん孔するかどうかを指定する。

○作業領域の制御

翻訳時に大きさの定まらない領域の大体の大きさを指定する。

○入出力機器の制御

目的プログラムを実行する際使用する入出力機器の論理機番のすべてを列記する。

○汎用記号制御

一番外側のブロックで宣言されている単純変数、配列名、手続き名、スイッチ名のうち、他のプログラム要素から参照されているものを指定する。

○未定義記号制御

翻訳をおこなうプログラムのなかで、宣言なしで使用する単純変数、配列名、手続名、スイッチ名を必要な性質の指定とともに列記する。ここで指定された名前は、このプログラム要素と同時に実行されるプログラム要素において汎用記号として指定されたものである。

○プログラムの名前

目的プログラムに名前を与える。

○サブプログラム制御

目的プログラムが、特にサブプログラムとして使用されるか、通常のプログラムとして使用されるかを指定する。

○デバッグ制御

原始プログラムの第1欄に＊記号のあるカードを有効とするか、無効とするかを指定する。またデバッグ用手続きを翻訳するかしないかを指定する。

デバッグ用手続きには制御の流れをチエックするcheckpoint，計算時の数値をみるためのsnap，snaparray，snaplist 等がある。

○精度の制御

プログラムの中で、実数型と宣言されているものが、単精度のものか長精度のものかを指定する。

○実行時のエラー制御

実行時にエラーが発生したとき、エラー表示を行なうかどうかを指定する。

7　実行時に発生するエラーの処理

目的プログラムの実行中にエラーが発生したとき、実行時のエラー制御にエラーメッセージ表示指定があれば、エラー表示がされ、値はエラーの種類に従って標準の値が代入され計算が続行される。従ってＡＬＧＯＬ－Ｈのようにzero diviser といって計算が打ち切られ、それを修正するとまた先の方で、exponent overflow といって突き返されるといったことはないが、しかし最終の結果だけを見てうまくいったと早合点するととんだことになるから注意を要する。さらにエラーが発生したとき、利用者が直接プログラムの流れに関与できるように、いくつかの手続きが用意されている。例えばオーバーフローエラーが発生したとき処理する手続きは、手続き文

ovf (procedure)

なる呼び出し形式をもった手続きの、実パラメータとして与えればよい。

F230－60 ALGOLと日本工業規格原案（水準5060との相違点

本システムが、日本工業規格原案（水準５０６０）の機能に追加した機能と、それに加えた制限の一覧を以下に示す。

追加した機能

1　宣言詞　long と長精度の実数型なる型を入れたことによって、倍精度での演算を可能にした。

2　書式記号として、５Ｆ，ＦＦＦＦＦが加えられた。

3　文字の書式として、挿入の列、反復子の使用が認められている。

4　設定手続に、page head が加えられた。

5　入出力手続きに、次のものが加えられた。

rewind, puta, geta, print, printstring, space, crlf, lfeed, eject

6　以下に示す手続きが、デバック用の手続きとして加えられる。

checkpoint, snap, snaparray, snaplist

7　実行時のエラーに対して、使用者があらかじめ処理手続きを用意できるようにした。

新しく加えられた制限

1　手続き文の実パラメタとして使用される名札は、手続き文がでてきた時点で未定義であるとき、有効な他の名前と同一であってはいけない。

2　スイッチの宣言に含まれるスイッチは、その宣言が終了したものでなくてはならない。

3　手続きの本体のなかで使用する非局所的な名前は、その宣言が完了したものでなくてはいけない。但し、本来宣言を必要としないものは除く。

4　手続き文の本体は、ＡＬＧＯＬ言語による記述以外認めない。

FORTRAN

1　プログラムの形式

プログラムのカードへの記入形式はＨＡＲＰの場合と同様であるが、そのほかに第１桁に＊記号をつけて識別するデバッグ行がある。後述のＯＰＴＩＯＮ文にＡＳＴＥＲまたは＊記号があれば、〝＊〟は空白とみなされ、なければこの行は注釈行とみなされる。

継続行は１９行まで許される。

2　データの型と性質

データには９種類の型がある。

(1)　整数型

Ｆ230－60　は1語３６ビット、うち1ビットは符号として使用するので、絶対値の最大は$2^{35}-1$すなわち 34, 359, 738, 367 である。

(2)　実数型

数値部に２７ビット、指数部に９ビット、両者とも１ビットづつ符号ビットに割当てている。従って精度は

１０進7.８桁、仮数部は８桁以下の有効数字をもち、指数部は－７６から＋７６までの整数値をとる。

(3) 倍精度実数型

精度は１０進１８.３桁、仮数部は１９桁以下の有効数字をもつ、指数部は実数型と同様。

(4) 複素数型

表現は実数型のデータの順序づけられた対の形。

(5) 倍精度複素数型

表現は倍精度実数型の順序づけられた対の形。

(6) 論理型

(7) 文字型

文字型データ及び文字型定数は nH h_1 h_2 …… h_n または' h_1 h_2 …… h_n ' で表わされる。１語に入りうる文字型データの文字の数は最大４字までである。

(8) ８進型

データは0, 1, …, ７の８種の数字の１２桁以下の列で、定数はこの列の最後にＯをつけて表わされる。

(9) ２進型

データは0,１の２種の数字の３６桁以下の列で、定数はこの列の最後にＢをつけて表わされる。

４倍精度実数型、倍長整数型はない。

3 式

原則として混合演算が許される。(脚注)

各型に順序をつけて、整数型、実数型、倍精度実数型、複素数型、倍精度複素数型とならべたとき、これらのいずれか２つに加、減、乗、除、またはべき乗の混合演算を施こしたとき得られる結果の型は，順序の後側のものになる。論理演算は論理型同志ばかりでなく整数型、実数型に対しても行うことができ、演算はビット毎に行われる。

右辺から左辺への代入の際、型の変換が自動的に行われるのはＨＡＲＰと同様である。前節でのべた２進数定数、８進数定数及び nH h_1 h_2 …… h_n の形の文字型定数および¥を頭につけた文番号は特殊型と呼ばれて式の右辺に置くことができ、左辺に整数型または実数型の変数を置くことにより、その変数に代入を行なうことができる。例えば

N＝¥100

は　　ＡＳＳＩＧＮ　　100　ＴＯ　Ｎ

と同様である。

A＝B＋4.6

C＝B＋4.6

Ｉ＝B＋4.6

の代りに

A＝C＝Ｉ＝B＋4.6

としてもよい、すなわち多重代入文が許される。

脚注　制限については「ＨＡＲＰにおいてＦ２３０－６０　ＦＯＲＴＲＡＮと相違する部分」参照。

4 宣 言

HARPにはない種類の型の宣言文にIMPLICIT文がある。

IMPLICFT LOGICAL (L)、COMPLEX(C－E)

と宣言すれば、Lを頭字にもった変数名は論理型であり、C，D，及びEを頭字にもつ変数名は複素数型であると暗黙の型宣言が行われたことになる。このようにして論理型、複素数型、倍精度実数型、倍精度複素数型の指定が容易になる。従ってプログラムの変更も容易になる。

配列の次元は最大7次元まで許される。

主プログラムにおいては、配列の宣言における宣言子添字は整定数でなければならない。すなわち動的な配列は許されてなく、翻訳時に配列の占める領域は確定しておかなければならない。しかしその領域をもつ配列が実引数として副プログラムへ渡されたとき、対応する仮引数にある配列は同領域をその領域を越えない限り、どう使おうと勝手である。次元は一致する必要はない。副プログラムの配列の要素は副プログラムで宣言した仕様に従って同領域を計算実行時に先頭から逐次コラムワイズに割当てられる。従って副プログラムにおける配列の宣言の宣言子添字は変数でもよい。

4倍精度実数型、大指数型、桁落ち計算型等の特別な数の取扱いを一挙に解決するために、特殊型宣言文が用意される。この宣言文で宣言された変数に演算が施こされたとき、その演算子に対応する演算を行なうサブルーチンが一定の方法で自動的に呼び出されるので、使用者がそれらに対応したサブルーチンをあらかじめ準備しておけば望みの演算を行なうことができる。

5 制御文

HARPにはない制御文に二分岐IF文がある。

IF(e)k_1, k_2

と表わされるが、eは論理式、k_1及びk_2は文番号で、eが真ならばk_1へ、偽ならばk_2へ制御の流れが移る。

F230－60 FORTRANでは実行中に誤りが生じても、致命的なものでない限りできるかぎり前後のつじつまを合わせて計算実行を続行させる方針である。これによって論理ミスをできるだけ多く発見させ、プログラムを走らせる回数を減少させることができる。

この方針に従って割当て型GO TO文

GO TO i,(k_1, k_2, …, k_n)

においては、iに相当する文番号がk_1, k_2, … k_nになければ警告が出されたあと、GO TO iが実行され、計算型GO TO文

GO TO (k_1, k_2, … , k_n), i

においては、i＜1のときk_1へ、 i＞nのときはk_nに制御が渡され、警告が出される。

RETURN 文には正規RETURN文と非正規RETURN文がある。非正規RETURN 文

RETURN i

はi＝0 のとき ただRETURNと解釈されるが、i≠0のときは仮引数のなかの＊の個数を左から数えてi番目に対応する実引数の文の番号をもつ文へ制御を渡す。

6 エントリー文

手続き副プログラムの途中に入口をつけるにはENTRY文を挿入すればよい。これは

ENTRY S

または

ENTRY　S（a_1，a_2，…，a_n）

と表わされる。Sは定義エントリー名で、a_1，a_2，…，a_n はＣＡＬＬ文や関数の引用で用いられる実引数に対応した仮引数であり、変数名か配列名か外部手続き名か星印“＊”である。このエントリー名を定義した点へ入るにはＣＡＬＬ文や 関数の引用でそのエントリー名を呼べばよい。これを利用すると短かい副プログラムをまとめて効率のよいプログラムを作ることができる。

例

副プログラム

```
      FUNCTION  JOE(X,Y)
   10 JOE=X+Y
      RETURN
      ENTRY  JAM
      IF(X.GT.Y)GO TO  10
   20 JOE=X-Y
      RETURN
      END
```

呼び出しプログラム

```
      ⋮
Z=A+B-JOE(3.*P,Q-1)
      ⋮
R=S+JAM(Q,2.*P)
      ⋮
```

ＤＯ文における範囲の拡張が、最深のＤＯループから許されるのはＨＡＲＰと同様であるが、ＤＯループの重なりはその場合完全入れ子の形をしていなければならない。完全入れ子とはＤＯループの重なりのうちで同一深さのループがないもののことである。

7　入出力文

入力中または出力中にデータが尽きたとき、または情報の転送にあたって誤りが発生したとき、利用者が自分で処置をとらせることができる入出力文がある。例えば、

READ（u，f，END＝n_1，ERR＝n_2）K

WRITE（u，f，END＝n_1，ERR＝n_2）K

等である。

uは入出力装置を指定する整定数か整数型の変数の引用で論理機番と呼ばれる。fは書式仕様を示すもので、ＦＯＲＭＡＴ文を指定する文番号か、その文番号を代入された変数名か、または書式を読み込んだ配列名である。Kは入出力並びである。

n_1はデータが尽きたとき、次に実行さるべき実行可能な文につけられた文の番号であり、n_2は情報の転送で誤りが発生したとき、次に実行さるべき実行可能な文につけられた文の番号である。パラメータＥＮＤ＝n_1 ＥＲＲ＝n_2の有無，選択及び順序は任意でよい。

パラメータにfのある入出力文は書式つき入出力文と呼ばれ、fのない入出力文は書式なし入出力文と呼ばれる。Ｆ２３０－６０ ＦＯＲＴＲＡＮには上記の外に幾つかの書式つき、或いは書式なし入出力文がある。

データを読み込んだり、書き出したりする変数の並びに名前をつけて、その名前を引用することによって、複雑な書式（ＦＯＲＭＡＴ）の指定なしに直接データの入出力を行なう方法がある。これは初心者には非常に便利であり、熟練者にはデバッグ用としてなくてはならないものである。

変数名及び配列名の並びに名前をつけるには一種の宣言文であるＮＡＭＥＬＩＳＴ文が使用される。

ＮＡＭＥＬＩＳＴ／x／a，b，…，c／y／d，e，…，f

この宣言文によって変数名または配列名の並びａ，ｂ，…，ｃ及びｄ，ｅ，…，ｆにそれぞれNAMELIST名ｘ及びｙがつけられる。

NAMELIST名を利用する入出力文はNAMELIST名つきREAD文またはWRITE 文と呼ばれる。NAMELIST名つきREAD文は次の形をしている。

READ（u，x，END＝n_1，ERR＝n_2）

ここでxはNAMELIST名である。この文の実行によって、uで識別される装置から値が読み込まれ、それら各値がNAMELIST名に属する各要素に、それぞれの型に従って割当てられ、各要素が定義される。

```
      DIMENSION A(10),I(5,5),L(10)
(1)   NAMELIST/NAM1/A,B,I,L
        ⋮
(2)   READ (5,NAM1)
        ⋮
```

入力データ　カードは

```
Col2 Col2
 ↓
 ␣A(3)=7. 6. 4, L=2, 2, 8 * 4. 3 ¥        ←第2枚目のカード
␣¥ NAM1␣I(2, 3)=5, J=4. 2,                ←第1枚目のカード
```

であるとする。

これが実行されると、整定数5がI（2,3）におかれ、実定数4.2は整数4に変換されてJにおかれ、整定数7は実数7.0に変換されてA(3)におかれ、次のA(4)には、実定数6.4がおかれる。整定数2,3はL(1)，L(2)に、実定数4.3は整数4に変換されて、続くL(3)，L(4)，…，L（10）におかれる。

NAMELIST名つきWRITE文には

```
WRITE(u,x,END=n1,FRR=n2)
PRINT x
PUNCH x
```

がある。

補助入出力文としては外部ファイルの位置ぎめをする、BACKSPACE文やREWIND文及び外部ファイルの境界をしるすENDFILE文がある。

入出力文には以上のほかに書式や入出力並びとの結びつきなしに、内部記憶装置のある領域と各種入出力装置との間の情報の転送を行なうBUFFER文がある。BUFFER文はHARPにおけるDRUM READ PROCEED文のように、データの転送が行われている間、プログラムは進行するので、転送中のデータが用いられる前後で、必らず各入出力装置の動作状態を検査しなければならない。

このために入出力動作状態検査文が用意されている。

入出力装置を指定する要素が含まれない入出力文にENCODE／DECODE文がある。情報はFORMAT仕様にしたがってある一つの領域から他の領域へ転送される。

8 USER文

USER文は

USER l

の形で与えられる。lは副プログラム名の並びである。この文は非基本副プログラムが基本副プログラムと同じ名前をもつとき、その性質について、何も調べないことを宣言する。

9 DEBUG文

プログラム実行時に生じた誤りの原因の発見を容易にするために、計算の実行を追跡調査する非実行文がDEBUG文であり、以下の6種類用意されている。

1 TRACE文
2 SUBTRACE文
3 SUBCHK文
4 TRANSFER文
5 DUMP文
6 TERMINATE文

これらは、原プログラムのEND行の前ならば何処にあってもよい。変数名、配列名、副プログラム名または文番号がそれぞれの文が与える条件に従って計算実行時に適当に出力される。

10 OPTION文

利用者が処理系に指示を与えるため、プログラムの先頭にOPTION文を置くことができる。この文がなければ処理系は標準的指示が与えられたとみなす。

OPTIONには次の9種類あり、利用者は適当にいくつかを取出し、コンマで区切って並べればよい。順序はどうでもよい。

1	LIST（またはNOLIST）	原プログラムを印刷する。（しない）
2	MAP（またはNOMAP）	目的プログラムの記憶場所等の詳細な情報を出力する（しない）。
3	SEQ	原プログラムのカードの順序を73～80欄をみて検査する。
4	DOUBLE	実数型変数をことごとく倍精度型変数に変える。
5	X	計算実行時に警告を出力しない。
6	ASTERまたは＊	第1桁目に＊記号のある行は空白とみなす。なければ注釈行と解釈する。
7	COMBUG	原プログラムの誤りの検査のみ行ない、目的プログラムは作り出さない。
8	PUNCH	目的プログラムをカード穿孔機に出力する。
9	NODEBUG	DEBUG文を無視する。

HARPにおいてF230－60 FORTRANと相違する部分

現在、HARPで組んでいるプログラムを将来　F230－60　FORTRAN　で使用しようという場合、そのプログラムをF230－60 FORTRANの文法に合うように書きかえねばならない。その手間をできるかぎり少くするため、あらかじめHARPの文法の中でF230－60 FORTRAN 文法に反しているものは使用しないようにするとよい。そこで現在の時点でわかる範囲で、HARP文法の中でF230－60 FORTRAN に相違しているものを列挙説明した。

なお、F230－60 FORTRANの仕様は確定したものでないので　ここに述べた事項は将来変更、追加があるものと予想される。あらかじめおことわりしておく。以下　ⒽはHARPを、ⒻはF230－60 FORTRAN意味する。

参考資料　○HITAC　5020，5020E／F　FORTRAN(HARP)
プログラムマニュアル（昭和42年9月版）
○FACOM　230−60　FORTRAN 外部仕様書

<table>
<tr><th></th><th>F230−60　FORTRAN</th><th>HARP</th></tr>
<tr><td>使用できる文字</td><td>IBMコードのみ使用可能</td><td>IBMコード、HITACHIコードが使用可能</td></tr>
<tr><td>定　　数</td><td>① 2語長整数型、4倍精度実数型はない
（以後2語長整数型、4倍精度実数型はない。）
② 文字型定数
1語にはいる文字の個数は4文字</td><td>① 2語長整数型、4倍精度実数型がある。
② 文字型定数
1語にはいる文字の個数は5文字</td></tr>
<tr><td>式</td><td>① 原則として混合演算が許される。
以下は、演算の組合せに対する評価の表である。
4則演算
第2演算子
第1演算子
<table><tr><th>+−
*／</th><th>I</th><th>R</th><th>D</th><th>C</th><th>DC</th></tr><tr><td>I</td><td>I</td><td>R</td><td>D</td><td></td><td></td></tr><tr><td>R</td><td>R</td><td>R</td><td>D</td><td>C</td><td>DC</td></tr><tr><td>D</td><td>D</td><td>D</td><td>D</td><td>C</td><td>DC</td></tr><tr><td>C</td><td></td><td>C</td><td>C</td><td>C</td><td>DC</td></tr><tr><td>DC</td><td></td><td>DC</td><td>DC</td><td>DC</td><td>DC</td></tr></table>I……整数型
R……実数型
D……倍精度実数型
C……複素数型
DC…倍精度複素数型</td><td>① 混合演算は許されない
4則演算
第2演算子
第1演算子
<table><tr><th>+−
*／</th><th>I</th><th>DI</th><th>R</th><th>D</th><th>Q</th><th>C</th><th>DC</th></tr><tr><td>I</td><td>I</td><td>DI</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>DI</td><td>DI</td><td>DI</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>R</td><td></td><td></td><td>D</td><td>D</td><td>Q</td><td>DC</td><td>DC</td></tr><tr><td>D</td><td></td><td></td><td>D</td><td>D</td><td>Q</td><td>DC</td><td>DC</td></tr><tr><td>Q</td><td></td><td></td><td>D</td><td>Q</td><td>Q</td><td>DC</td><td>DC</td></tr><tr><td>C</td><td></td><td></td><td>DC</td><td>DC</td><td>DC</td><td>DC</td><td>DC</td></tr><tr><td>DC</td><td></td><td></td><td>DC</td><td>DC</td><td>DC</td><td>DC</td><td>DC</td></tr></table>DI……2語長整数型
Q　……4倍精度実数型</td></tr>
</table>

F230−60 FORTRAN

乗べき演算

底＼指数	**	I	R	D	C	DC
	I	I				
	R	R	R	D		
	D	D	D	D		
	C	C			C	DC
	DC	DC			DC	DC

関係演算の可能な組合せ

関係演算子	I	R	D	C	DC
I	○				
R		○	○		
D		○	○		
C					
DC					

関係演算子＝(.LT.,.LE.,.EQ.,
.NE.,.GT.,.GE.)

論理演算子	I	R	D	C	DC	L
I	○	○				○
R	○	○				○
D						
C						
DC						
L	○	○				○

論理演算子＝(.AND.,.OR.,.NOT.,
.EOR.)

ビット毎の演算を行なう。

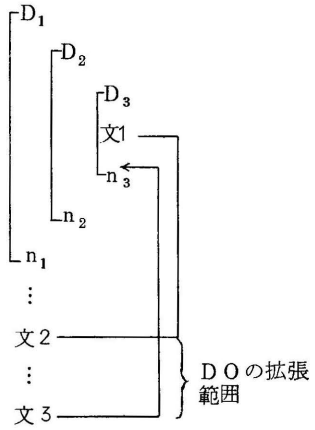
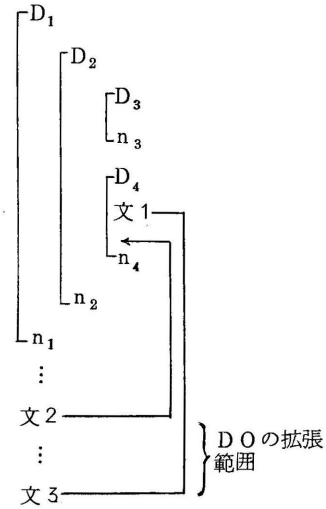
HARP

底＼指数	**	I	DI	R	D	Q	C	DC
	I	I	I					
	DI	DI	DI					
	R	D	D	D	D	D		
	D	D	D	D	D	D		
	Q	Q	Q	Q	Q	Q		
	C	DC	DC					
	DC	DC	DC					

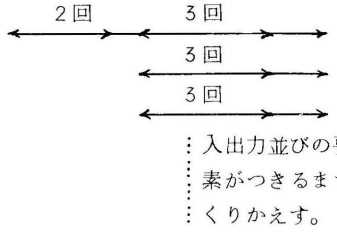
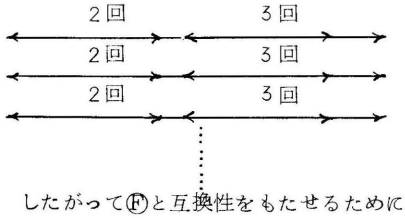
関係演算子	I	DI	R	D	Q	C	DC
I	○	○					
DI	○	○					
R			○	○	○		
D			○	○	○		
Q			○	○	○		
C							
DC							

論理演算子	I	DI	R	D	Q	C	L
I							
DI							
R							
D							
Q							
C							
DC							
L							○

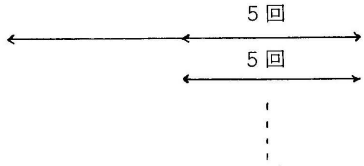
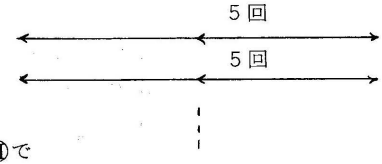
論理演算子＝(.AND.,.OR.,.NOT.)

	F230−60 FORTRAN	HARP
DO文	DO　n　i＝m_1，m_2，m_3 or DO　n　i＝m_1，m_2 ① DOの拡張範囲 DOの範囲が拡張可能なためには DOの範囲は完全入れ子をなしていなければならない。  ② 2つ以上のDO文が共通の端末文をもつ場合 この端末文の番号を、対応する一番内側のDOの範囲に、またはその拡張範囲に含まれない、GO TO文、算術IF文で使用してはならない。 例えば DO 10 I＝1，20 ⋮ IF（A／B＊＊2）10, 1, 2 ⋮ DO 10 J＝1, 5 ⋮ 10 CONTINUE は誤りである	DOの範囲は完全入れ子でなくてもよい。  GO TO文、算術IF文で使用してもよい。
入出力文 FORMAT文	① 数値の変換 A変換… 1語にはいりうる文字の数は4文字 ② 配列内の書式仕様（演算実行時につくられるFORMAT）	A変換… 1語にはいりうる文字の数は5文字

<table>
<tr><th></th><th>F230−60 FORTRAN</th><th>HARP</th></tr>
<tr><td></td><td colspan="2">書式仕様をつくる際、文字型定数の１語にはいりうる数がⒽが５個、Ⓕが４個になっていることに注意しなければならない。ⒻとⒽの互換性はない。

すなわち Ⓗで
DIMENSION IA(2)
READ(5,100)(IA(I),I=1,2)
100 FORMAT(2A5)
⋮
データは(4HABCD)と書かれている場合、プログラムをⒻへ変換するときは 100 FORMAT(2A4)としなければならない。しかしⒻへ変換したものをそのまゝ HARPで実行させると
IA(1)は (4HA␣ ,A(2)は BCD)␣ となりまちがったFORMATができる。(␣はブランクを表わす)</td></tr>
<tr><td></td><td>③ FORMATの書式制御
書式制御が書式仕様の最後の右カッコに出合うと入出力並びに要素が残っているか調べ、残っていない場合には書式制御は終了、要素が残っている場合書式制御は最後から２番目の右カッコで終っている欄記述群にもどるか、ない場合には書式仕様の先頭の左カッコにもどる。</td><td>最後の右カッコに出合うと必ず先頭の左カッコにもどる。</td></tr>
<tr><td></td><td>④ (t)は欄記述群r(t)のr＝1のときの省略形
例1
FORMAT(1H , 2(5X,E15.7),
3X,3(5X,F10.5),5X,E12.4)
の場合
 
</td><td>(t)はtの不定回反復を意味する
 

したがってⒻと互換性をもたせるためには
FORMAT((1H ,2(5X,E15.7),
3X,3(5X,F10.5),5X,
E12.4))
と書いておけばよい。</td></tr>
</table>

	F230−60 FORTRAN	HARP
	例2 FORMAT(1H ,7HNUMBER=, I4/1H0, 5(5X,E15.7)) の場合  ⑤ 補助記憶装置に関する入出力文は外部表現が異るので、Ⓕへ変換するときは書きかえねばならない。	 Ⓗで FORMAT(1H ,7HNUMBER=, I4/1H0,(5(5X,E15.7))) と書けばⒻと同じことが行われる。
宣　言　文	① プログラム本体中における宣言文の位置は次のとうりである。 宣言文 文関数定義文 } ① } ② 実行文 } DATA文は①の間のどこにあってもよい FORMATは②の間のどこにあってもよい。 ② WORD LENGTH n BITS はない。	プログラム本体中の宣言文は、その変数が使われる前ならばどこにおいてもかまわない。
文関数定義文	① 定義場所 そのプログラム単位の最初の実行文より前でかつ宣言文のあとにおかねばならない（宣言文の項参照）	そのプログラムの最初の実行文より前であればよい。
組込み関数	① 関数名 IDFIX はない ② 1つの関数に対して引数の型は1つに決められている。 関数名、引数の型、関数の型についてのⒽとの対応表はここでは省略する。	1つの関数に対して、引数の型として複数個の型が許される場合がある。

	F230−60 FORTRAN	HARP
関数副プログラム サブルーチン 副プログラム	① 仮引数 仮引数はCOMMON文　DATA文にあらわれてはならない。 ② 仮引数が配列名の場合の仮引数と実引数（配列名）の対応 実引数と仮引数の配列の大きさが異るときの対応のしかたは以下のとうりである。 呼び出しプログラム ⋮ DIMENSION A(5,5) ⋮ CALL SUB (…, A, …) 副プログラム SUBROUTINE SUB (…, B, …) DIMENSION B (2, 3) ⋮ A (1, 1) ⟷ B (1, 1) A (2, 1) ⟷ B (2, 1) A (3, 1) ⟷ B (1, 2) A (4, 1) ⟷ B (2, 2) A (5, 1) ⟷ B (1, 3) A (1, 2) ⟷ B (2, 3) A (2, 2) A (3, 2) ⋮	仮引数はCOMMON文、DATA文にあらわれてもよい。 但しCOMMON文の場合、対応する実引数もCOMMON文になければならない。 A (1, 1) ⟷ B (1, 1) A (2, 1) ⟷ B (2, 1) A (3, 1)　　＊ A (4, 1)　　＊ A (5, 1)　　＊ A (1, 2) ⟷ B (1, 2) A (2, 2) ⟷ B (2, 2) A (3, 2)　　＊ ⋮　　⋮ A (1, 3) ⟷ B (1, 3) A (2, 3) ⟷ B (2, 3) ⋮　　＊ ⋮　　⋮ ＊の部分は実引数の大きさが優先するので、仮引数の配列の大きさにかかわらず意味をもっている。 しかし、仮引数の配列名が、入出力文のリストにあらわれたときは＊の部分は意味がなくなる。

	F230−60 FORTRAN	HARP
基本外部関数	① $\sqrt[3]{X}$ （実数型、倍精度実数型、複素数型、倍精度複素数型）及びtanh(複素数型、倍精度複素数型）に関する基本外部関数はない。	
DEBUG文	外部表現が全く異るので、プログラムをⒻへ変換するときは、書きかえればならない。	

参　考

日本工業規格　電子計算機プログラム用言語

○ＪＩＳＣ6201　FORTRAN　（水準7000）　450円

○ＪＩＳＣ6212　ALGOL　（水準5000）　295円

○ＪＩＳＣ6216　ALGOLの入出力（水準６０）　335円

発行所

財団法人　日本規格協会

東京都港区赤坂４丁目１番２４号　電話(583) 8001

福岡支部

福岡市渡辺通り２丁目1街区１１号　十八銀行ビル内　電話(76) 4226