

41 - 5

電 子 計 算

技 術 管 理

土 木 研 究 所 資 料

第 175 号

電 算 速 報

第 5 号

昭和 4 1 年 5 月

建設省土木研究所

電 算 速 報 5 号 目 次

§ 1. 電算運営資料

- (1) ALGOLマニュアル..... 1
- (2) NEW ALGOL-H文法詳論.....27

§ 2. プログラム紹介

- (1) attachされているプロセデュアについて.....47
- (2) 現地観測データの分類, 集計, 分散分析のプログラム.....51

§ 3. 電算室よりのお知らせ

- (1) 計算機使用状況およびO. D. O契約について..... 60
- (2) プログラマー追加名簿(その3).....76
- (3) 操作指図書・時間外申込書.....84

電算速報第5号正誤表

頁	行	誤	正
1	-4~	4 宣言子 Boolean..... Procedure..... ('H')..... string..... 5 指定子	4 宣言子 Boolean..... procedure..... ('H')..... string..... 5 指定子
9	+8	も機検は無視する。	も機検は無視する。
"	+9	は無視する。	は無視する。
"	-13	印刷様式指定は' 'でつつんで	印刷様式の指定は' 'でつつんで
10	-3	と宣告された	と宣言された
14	-10	for I:=1 step 1	for I:=1 step 1
"	-3	書き込まれる	書き込まれる
17	-12	procedure, label	, label を削除
"	-10	のタイプが異なっていて	のタイプが異なっていて
21	-3	real procedure	real procedure
23	+4	又は;さん孔	又は;をさん孔
24	+15	RPAOB(AB);	SPACE(AB);
"	-8	2. デソミタ	2. デリミタ
29	-8, -7	<十進数値>	<10進数値>
30	+12, 13, 16	<添字変数>	<添字付変数>
"	+17	また添字が	また添字式が
34	-10	::=go to <DE>	::=go to <DE>
35	+9	<スイッチ名>	<スイッチ名> <手続名>
"	+12	<手続名> <A P部> <L> <手続命令>	<手続名> <A P部> 以下削除
36	+6	つながりを でくった	つながりを' 'でくった
41	-5	REWIND(1);	REWIND(1);
44	+5	<Yリスト>;	<Yリスト>; <明細指定部> <明細指定子> <Yリスト>;
"	+15~16	! <ブロック> <複合命令>	削除
"	-12	手続頭には	'H' では手続頭部に
"	-11	FPリストに	'H' ではFPリストに
"	-7	arrayは	'H' ではarrayは
45	-7	読み込まれる。デー	読み込まれる。データ
47	+1	attachされているプロシエアについて	attachされているプロシエアについて
54	-9	for K:=1 step	for K:=1 step
"	-13	for I:=1 step	for I:=1 step
55	+1	if A<I then A:=;	if A<I then A:=I;
"	-7	, AN(3), AN(4);	, AN(3), AN(4);
56	-1	eud	eud;
57	+4	PRINT('2B' 6D:=ZD, D(I, J);	PRINT('2B 6D:=ZD, D(I, J);
59	-6	PRINT('2/B'	PRINT('21B
"	-10	for I:=1 step	for I:=1 step
"	-8	I, B, 2/,'	I, B, 2/,'
60	-5	表-4に	別表に
62	+9, 10	構築	構築

ALGOL マニュアル

** 中 村 慶 一

はじめに

ALGOLの文法概要や解説は電算速報などにも、いろいろとのせてきたが、講習終了者などから日常プログラムを書き下すときのマニュアルとしては使いにくいという話が出たので、この向きのものを準備してみた。多少厳密でない表現を用いるのでNEW ALGOL-H文法詳論の項ナンバーを(詳論 3.2.5)などと引用する。

また、NEW ALGOL-HとALGOL-60との相違点はそのつど触れるほか、他の代表的なコンパイラと共に巻末にまとめて相違点の説明をつけ加えておく。

1° 基本記号, 名前, 数値, スtring

1.1 基本記号 (第 2)

- | | | | | | | | | | | | | | | | | | | | |
|----------|---|---|--------------------------------------|---|----------------------------|--------------------------------------|-----------------------------------|--------|-----------------------------------|---------------------|--------|-----------------------------------|---------------------------|---------------------------|--------------------|-----------------------|--------|--------------------|--|
| 基本記号 | { | 1. 文 字.....AからZまでとaからzまでの52文字
('H' ではAからZまでの大文字に限る) | | | | | | | | | | | | | | | | | |
| | | 2. 数 字.....0から9までの10個 | | | | | | | | | | | | | | | | | |
| | | 3. 論 理 値.....trueとfalseの2個
(logical value) ('H' ではTRUE, FALSEの2個) | | | | | | | | | | | | | | | | | |
| | | 4. 演 算 子 | | | | | | | | | | | | | | | | | |
| | | <table border="0"> <tr> <td>1. 算術演算子</td> <td>+ - × ÷ +↑</td> <td>('H' では×の代りに*, 以下*を用いる。)</td> </tr> <tr> <td>2. 関係演算子</td> <td>< ≤ = ≥ > ≠</td> <td></td> </tr> <tr> <td>3. 論理演算子</td> <td>≡ ∩ ∨ ∧ ¬</td> <td></td> </tr> <tr> <td>4. 逐次演算子</td> <td>go to if then else for do</td> <td>('H' ではこのほかにon codeがある)</td> </tr> </table> | 1. 算術演算子 | + - × ÷ +↑ | ('H' では×の代りに*, 以下*を用いる。) | 2. 関係演算子 | < ≤ = ≥ > ≠ | | 3. 論理演算子 | ≡ ∩ ∨ ∧ ¬ | | 4. 逐次演算子 | go to if then else for do | ('H' ではこのほかにon codeがある) | | | | | |
| 1. 算術演算子 | + - × ÷ +↑ | ('H' では×の代りに*, 以下*を用いる。) | | | | | | | | | | | | | | | | | |
| 2. 関係演算子 | < ≤ = ≥ > ≠ | | | | | | | | | | | | | | | | | | |
| 3. 論理演算子 | ≡ ∩ ∨ ∧ ¬ | | | | | | | | | | | | | | | | | | |
| 4. 逐次演算子 | go to if then else for do | ('H' ではこのほかにon codeがある) | | | | | | | | | | | | | | | | | |
| | | 4. 限 定 詞
(delimiter) | | | | | | | | | | | | | | | | | |
| | | <table border="0"> <tr> <td>1. 演算子</td> <td> <table border="0"> <tr> <td>1. 区切り</td> <td>{ , . ; = step until while comment }</td> <td>('H' ではこのほかにhaltがあり, の代りに#を用いる)</td> </tr> <tr> <td>2. 括弧</td> <td>() [] ' ' ' ' begin end</td> <td>('H' にはattachがある)</td> </tr> <tr> <td>3. 宣言子</td> <td>Boolean integer real array switch</td> <td></td> </tr> <tr> <td>4. 宣言子</td> <td>procedure</td> <td>('H' にはexternalがある)</td> </tr> <tr> <td>5. 指定子</td> <td>string label value</td> <td></td> </tr> </table> </td> </tr> </table> | 1. 演算子 | <table border="0"> <tr> <td>1. 区切り</td> <td>{ , . ; = step until while comment }</td> <td>('H' ではこのほかにhaltがあり, の代りに#を用いる)</td> </tr> <tr> <td>2. 括弧</td> <td>() [] ' ' ' ' begin end</td> <td>('H' にはattachがある)</td> </tr> <tr> <td>3. 宣言子</td> <td>Boolean integer real array switch</td> <td></td> </tr> <tr> <td>4. 宣言子</td> <td>procedure</td> <td>('H' にはexternalがある)</td> </tr> <tr> <td>5. 指定子</td> <td>string label value</td> <td></td> </tr> </table> | 1. 区切り | { , . ; = step until while comment } | ('H' ではこのほかにhaltがあり, の代りに#を用いる) | 2. 括弧 | () [] ' ' ' ' begin end | ('H' にはattachがある) | 3. 宣言子 | Boolean integer real array switch | | 4. 宣言子 | procedure | ('H' にはexternalがある) | 5. 指定子 | string label value | |
| 1. 演算子 | <table border="0"> <tr> <td>1. 区切り</td> <td>{ , . ; = step until while comment }</td> <td>('H' ではこのほかにhaltがあり, の代りに#を用いる)</td> </tr> <tr> <td>2. 括弧</td> <td>() [] ' ' ' ' begin end</td> <td>('H' にはattachがある)</td> </tr> <tr> <td>3. 宣言子</td> <td>Boolean integer real array switch</td> <td></td> </tr> <tr> <td>4. 宣言子</td> <td>procedure</td> <td>('H' にはexternalがある)</td> </tr> <tr> <td>5. 指定子</td> <td>string label value</td> <td></td> </tr> </table> | 1. 区切り | { , . ; = step until while comment } | ('H' ではこのほかにhaltがあり, の代りに#を用いる) | 2. 括弧 | () [] ' ' ' ' begin end | ('H' にはattachがある) | 3. 宣言子 | Boolean integer real array switch | | 4. 宣言子 | procedure | ('H' にはexternalがある) | 5. 指定子 | string label value | | | | |
| 1. 区切り | { , . ; = step until while comment } | ('H' ではこのほかにhaltがあり, の代りに#を用いる) | | | | | | | | | | | | | | | | | |
| 2. 括弧 | () [] ' ' ' ' begin end | ('H' にはattachがある) | | | | | | | | | | | | | | | | | |
| 3. 宣言子 | Boolean integer real array switch | | | | | | | | | | | | | | | | | | |
| 4. 宣言子 | procedure | ('H' にはexternalがある) | | | | | | | | | | | | | | | | | |
| 5. 指定子 | string label value | | | | | | | | | | | | | | | | | | |

** 技術管理室長 工博

1.2 略 号

以下説明の簡略化のため次の略号を使用する。

- (A) E 算術式 (arithmetic expression)
結果として1つの算術値に帰着する表現
- (A) P 実パラメタ (actual parameter)
手続命令に用いるパラメタ
- (B) E ブール式 (Boolean expression)
結果として1つの論理値に帰着する表現
- (D) 宣言 (declaration)
そのブロック内で使用する名前種類の種類等について先頭の
beginの後最初の命令までの間で行なう定義
- (D) E 指示式 (designational expression)
結果として1つの名札に帰着する表現
- (F) P 形式パラメタ (formal parameter)
手続宣言に用いるパラメタ
- (F) S FOR命令 (for statement)
コントロール変数を変化させて計算を実行する命令
- (I) 名前 (identifier)
変数、関数指示子、手続、スイッチ、名札に対し他と区別
するためにつけられる目印。
- (L) 名札 (label)
命令、ブロック、複合命令につける行先指示の目印し。
- (S) 命令 (statement) 計算機に1つの操作を実行させる単位
- (S) A 単純算術式 (simple arithmetic expression)
条件付でない算術式
- (S) B 単純ブール式 (simple Boolean expression)
条件付でないブール式
- (S) D 単純指示式 (simple designational expression)
条件付でない指示式
- (S) V 単純変数 (simple variable)
添字付でない変数

SE 添字式 (subscript expression) 添字として用いる式

US 無条件命令 (unconditional statement)

基本命令, ブロック, 及び複合命令

UI 無符号整数 (unsigned integer)

V 変数 (variable)

またパンチャの誤読をさけるため I 17, O 0 D, 5 S, V V, O (の区別をはっきりさせる必要がある。

またスペース改行は機械が無視する。

2.2 標準関数, 換算関数

abs sign sin cos sqrt arctan ln exp entier (換算関数その値をこさぬ最大の整数をとる) その後 () で算術式をつつむ。real integer の何れでもよいが abs signを除いてすべて real の答となる。

absは () の中のタイプと同じ絶対値, sign は integer の結果で正のとき+1, 0のとき0, 負のとき-1となる。('H'では大文字を用いるまたOLD-Hとの関連からFLOAT, FIXも認める)。

2.3 算術式

1. 結果として1つの算術値に帰着する表現であって次のものがある。

- | | | | |
|---------------------------------|---|--|---|
| 算術式
(A E) | { | 1. 算術定数 | -2.8 3.5 ₁₀ -2 ₁₀ -3 5 |
| | | 2. 算術変数 | AI BQ[I, J, K] |
| | | 3. 算術関係指示子 | real procedure または integer procedureとして宣言されたものを用いる。
TAN(X) |
| | | 4. 条件付算術式を丸カッコに包んだもの | (if A=B then 3 else 4*O+A(R)) |
| | | 5. 1~4の前に+または-をつけたもの。すでに+または-のついているものはカッコにつつむ。 | |
| | | 6. 1~4ならびに5をカッコにつつんだものを+-*/÷↑で結んだもの。
A*B-(O+D)*E*(if A=B then L else F) | |
| 単純算術式
(SA) | | | |
| 条件付算術式
if BE then SA else AE | | | |

2. 演算の実行順序は次のようになる。

1. ()
2. ↑
3. * / ÷
4. + -

3. 算術演算子の両側は integer 又は real で混合した場合はその結果は real となる。ただし、÷の両辺は integer に限り、その結果は integer となると共に絶対値の切り捨て結果に符号を付したものとなり $5 \div 2$ は 2 である。

4. 添字付変数はその所在をつきとめるのに時間がかかるので、式中に何回も現われるときは一旦他の単純変数に受ける方が演算時間が短くなる。

5. 条件式は if then else が揃っている必要があり、then の後は単純式に限る。もし then の次に条件式がくるときは then (if ……)
else とし、() でくくって単純式としなければならない。

(‘H’ では else の前後のもののタイプは同じものでなければならない。

またコンパイラの都合でカッコの直後の文字数字は無視するので $(A+B)C + D$ などと誤って書くと $(A+B) + D$ の結果を出すので注意を要する)

6. $A \uparrow I$ は $I > 0$ なら $A * A * \dots * A$ (I 回)

$I = 0$ で $A \neq 0$ なら A と同じタイプの 1

$A = 0$ なら誤り

$I < 0$ で $A \neq 0$ (A : real) なら $1 / (A * \dots * A)$

(A : integer) なら 0

$A \uparrow B$ は $A > 0$ なら $\exp(B * \ln(A))$ で real

$A \leq 0$ なら $0 \uparrow 0 = 1.0$ 以外は誤り

7. 算術演算子を / などと連続して用いてはならない。また条件式を演算子で結んで用いるときはカッコに入れて単純式として用いる。

2.4 ブール式 (Boolean expression BE)

1. 結果として 1 つの論理値に帰着する表現で次のものがある。

- ブール式 (BE)
- 単純ブール式 (SB)
 1. 論理定数 true および false ('H'ではTRUEおよびFALSE)
 2. 論理変数 Booleanとして宣言された単純変数またはBoolean arrayとして宣言された配列の要素
 3. 論理関数指示子 Boolean procedureとして宣言されたものを用いる。
 4. 条件付ブール式を丸カッコで包んだもの
 5. 関 係 $\overline{A} \overline{B} \wedge \overline{A} \overline{B}$
 ρ としては $> \geq = \leq < \neq$ がある。
 6. 1～5の前に $\neg$ をつけたもの $\neg A = B$
 7. 1～5および6をカッコにつづんだものを $\wedge \vee \equiv \supset$ で結んだもの。
 - 条件付ブール式
 if BE then SB else BE

従って演算の実行順序は次のようなものになる。

1. ()
2. 算術演算子
3. 関係演算子
4. $\neg$
5. $\wedge$
6. $\vee$
7. $\supset$
8. $\equiv$

2. $\neg \wedge \vee \supset \equiv$ は次の結果をもたらす。

	結 果
$\neg A$	Aがtrueのときfalse, Aがfalseのときtrue
$A \wedge B$	A, B共にtrueのときのみtrue
$A \vee B$	A, B共にfalseのときのみfalse
$A \equiv B$	A, B共にtrueのとき, または共にfalseのときのみtrue
$A \supset B$	A true かつ B falseのときのみfalse

3. if if if $A=B$ then $O1 \neq D$ else $O2=Q$ then $S>R$ else $S=45$ then $P1 \leq Q1$ else $AB=X48 \wedge (P+Q)*R < 365$
 は1つのブール式である。

4. 関係演算子の左右両辺のタイプが異なるときは real として比較される。
 5. 論理演算子は $\vee \wedge$ などと連続して用いてはならない。

条件式を演算記号で結んで用いるときはカッコに入れて単純式にして用いる。

2.5 指示子 (designational expression $\overline{DE}$)

1. 結果として1つの名札に帰着する表現であって次のものがある。

指示式 ($\overline{DE}$)	{	単純指示式 ($\overline{SD}$)	{	1. 名札 命令, ブロック, 複合命令に行先指示のためにつけられる目印しであって名前と無符号整数に限られ: で区切られる。 ('H' では無符号整数の名札を認めない)
		2. スイッチ指示子 $PS[I]$ 3. 条件は指示式にカッコをつけたもの (if $A \neq B$ then $L1$ else $L2$)		
		条件付指示式 if $\overline{BE}$ then $\overline{SD}$ else $\overline{DE}$		

2. スイッチ指示子の添字式が real のときは4捨5入される。スイッチとして宣言された範囲をこえるときは空命令扱いとする。
 ('H' ではエラー扱いとする)

3° 節 clause

1. 節には次の2つがある

節 {	IF 節	if $\overline{BE}$ then
	FOR 節	for $\overline{V} := \langle \text{for list} \rangle$ do

いずれもそのままではなく IF 節は式と命令の一部, FOR 節は命令の一部として用いられる。

2. IF 節を式の中で用いるときは

if $\overline{BE}$ then 単純式 else 式

と then の後を単純式とし if then else の3つは完全に含まれていなければならない。

3. IF 節を命令に用いるときは

if	$\overline{BE}$	then	$\overline{US}$	;
if	$\overline{BE}$	then	$\overline{US}$	else $\overline{S}$;
if	$\overline{BE}$	then	$\overline{FS}$	;

の3種の使い方があり何れも then の後は条件命令であってはならないが、else を含まぬものもあり得る。勿論その場合 $\overline{BE}$ が false であれば次の命令が実行される。

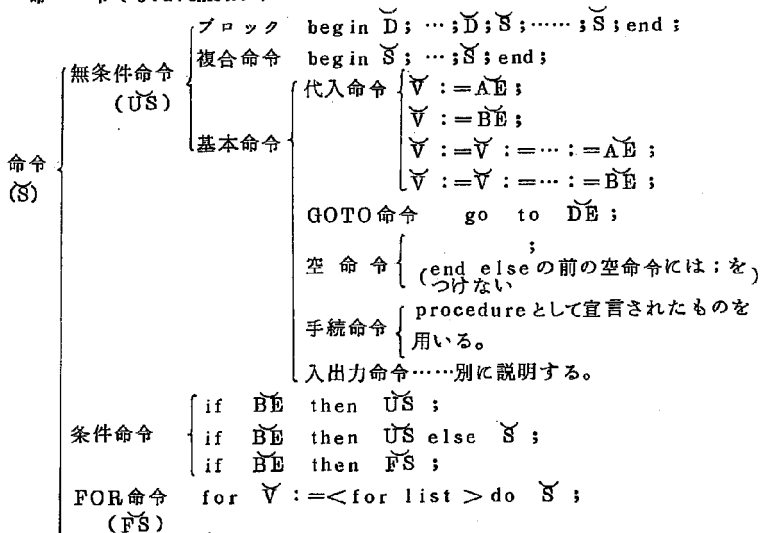
4. FOR節の<for list>は

- | | |
|----|--|
| 1. | $\overline{AE}$ |
| 2. | $\overline{AE}$ step $\overline{AE}$ until $\overline{AE}$ |
| 3. | $\overline{AE}$ while $\overline{BE}$ |
| 4. | 1~3をカンマでつないだもの |

の4種がある。

do の後は1つの $\overline{S}$ にしか for の効力は及ばないから2つ以上の命令を実行させなければ begin end でくくられなければならない。

4° 命令 (statement)



4.1 ブロック，複合命令

ブロックと複合命令の差違は宣言の有無だけである。立札もその立てられた場所において宣言されたとみなされる。

宣言はそれを含む最もせまい `begin` から `end` までの間についてしか意味を持たず，その外側において宣言された名前は内側で同じ名前が宣言されていない限り有効である。同じ名前が内側でも宣言されているときは内側では内側で宣言された意味の方が優先する。立札についても外に飛び出すことは可能であるが内側には `begin` から入って行くより仕方がない。セミコロンは同種の宣言と命令の区切りとして使用され `end` の前ではいらないが，つい `end` の後に不用であるかのように錯覚して誤りをおかすことが多いので，`end` の前に常に空命令があると考えて例で示したように `end` の前にもセミコロンを入れるくせをつけておいた方がよい。

4.2 基本命令

代入命令で，左右辺のタイプが異なるときは自動的に変換する。特に左辺 `integer` 右辺 `real` のときは4捨5入を行なう。

また左辺が何重かになった多重代入命令が許されるが左辺の中では同じタイプでなければならない。

4.3 条件命令

`then` の後は無条件命令又は `FOR` 命令に限る。したがって，条件命令が `then` の後に来るときは `begin end` でくくって無条件命令に直す必要がある。

4.4 FOR命令

- たとえば1から1きざみで20まで23，25後は5きざみで100まで，以降倍々にして10000まで A と A^2 を印字させるには

```
for A := 1 step 1 until 20, 23, 25 step 5 until
100, I*2 while A ≤ 10000 do
PRINT ( 2R / , A, A↑2 );
```

とすればよい。

- `until` の後は `0.1 step 0.01 until 5 do` などとしたときは丁度5になったとき2進変換の都合で実行しない場合があり得るので `5.002` などときざみの $1/4 \sim 1/5$ を加えて（負のきざみなら引いて）おく方が

よい。

3. FOR命令の途中で外にとび出したときは、コントロール変数はその飛び出す直前の値を保持しているがFOR命令が完全に終了したときは、何の値がコントロール変数に入っているか不明である。
4. またfor命令の途中に外から飛び込むことはできない。

4.5 コメント

begin 及び ; の直後で comment と書けば次の ; がくるまで何を書いても機械は無理する。end の後は ; end else がくるまで何を書いても機械は無理する。

(‘H’では procedure 宣言の heading 部に comment を書くことは許さない)

5° 入出力命令

ALGOL-60 には入出力命令に関する何等の規定もない。従って ‘H’ の規定を示す。

5.1 印刷様式指定 (§ 4.8.1)

印刷の PRINT, テープパンチの PUNCH, 磁気テープ関係の MTRREAD 及び MTWRITE はすべて印刷様式指定とその中に入れるべき情報のあるときはそのリストとが必要である。印刷様式指定は でつつんで行なわれる。このフォーマットストリングの中では特殊な記号が使用されるのでそれについて説明する。

5.1.1 記号

- | | |
|---|-----------------------------|
| A | アルファタイプ 1 桁の印刷 |
| B | ブランクの印刷 |
| C | カンマの印刷 |
| D | 数字 1 ケタの印刷 |
| F | ブーリアンタイプ 1 字の印刷 |
| I | 所定のフォーマットで integer タイプの情報印刷 |
| L | 所定のフォーマットでブーリアンタイプの情報印刷 |
| P | ブーリアンタイプを 0 又は 1 で印刷 |
| R | 所定のフォーマットで real タイプの情報印刷 |

S 文字数字等1字の印刷
 T 以降切りすでの指示(印字面には出ない)
 V 印字なしの小数点位置表示(印字面には出ない)
 X レプリケータ(その後の記号等のくり返し指示)
 Z 数字1ケタの印刷(先頭からの無効ゼロ消去)
 . 小数点印刷
 + 正負記号印刷
 - 負のときのみ記号印刷
 10 10 印刷
 ↑ 一頁おくれ(LP), ⑩を入れよ(MT)
 / 以上の情報を印刷し1行おくれ(LP)
 I R Gを入れよ(MT)
 ; 今迄の情報を行おくりなしに印刷せよ(LP)
 {...} フォーマットストリング中に更に' 'につつまれたときはその……をそのまま印刷

このうちA, B, D, F, P, S, Z, ↑, / および()につつまれたものの前に無符号整数又はXをおいて、くり返し回数を指定できる。ただしXには後述のREP X命令によってあらかじめ値を与えてなければならない。

またA, B等前述のものはAAAなど重ねて用いても差し支えない。

5.1.2 アルファタイプの印刷

既にinteger K ;と宣告された変数に=TOKYOと読み込まれているか又はK:=EQUIV('TOKYO');と代入された内容を印刷するには

```

PRINT('5A/', K);

```

とすればよい。1語は7字までであるからYOKOHAMA など8文字のときは2語に分ける。スペースは#で下段記号も含む。

5.1.3 実数型の印刷

real P ;と宣告された変数で 1.652×10^3 の内容になっているものを有効数字4ケタで印刷したければ

```

PRINT('D.3D10+ZD/', P);

```

勿論10のべきを含まぬ形で印刷するなら165.2となる筈であるから

```
PRINT ('3D.D/', P);
```

としてもよいが小数点以上3ケタ以上もあり得る場合にはゼロサプレスして(5ケタ以上はあり得ないとし負のときは符号も入れ)

```
PRINT ('-4DZ.D/', P);
```

とする。

5.1.4 整数型の印刷

integer R ;とされた変数を有効5ケタで印刷したいときは

```
PRINT ('-4ZD/', R);
```

となる。5Zとすると0のときは全く何も印字しないので、1ケタはDタイプにしておかなければならない。ZはDの後や・の後に使用できないことは勿論である。(10の後ではZは使用してよい)

5.1.5 ブーリアンタイプの印刷

ブーリアンとしてPが宣言されていれば

```
PRINT ('5F/', R);
```

でその内容に従って印字される。

フォーマット	TRUE	FALSE
P	1	0
F	T	F
2F	TR	FA
5F	TRUE	FALSE
8F	TRUE	FALSE

5.1.6 ノンフォーマット

integer タイプKを所定のフォーマットで印字したいときは

```
PRINT ('I/', K);
```

でreal タイプはR, Boolean タイプはLを用いる。

IとRは後のリストのタイプを変換し、たとえばreal Pに対しフォーマットでIを指定したときは四捨五入される。

I BB-ZZZZZZZZZZZZZD (13ケタ)

R BB-.DDDDDDDDDD,0+DD (9ケタ)

L BBFFFFFF

5.1.7 整列マーク

／↑；は印刷の配列指示に用いられ次の意味を持つ

	LP	MT	HTP
／	印刷し1行改行	IRG	無視
↑	新しい頁の行頭へ	①	①
；	改行しないで印刷	エラー	エラー

5.1.8 フォーマットストリング

フォーマットストリング中では後のリストに対応するDZ, 10, QTVの組み合わせ, A, FP等, IRL等は1つつの組を形式し, これらはカンマで区切られる。これに対しBS↑／；等および'...'でつづまれたストリングはこの何れかの組の中に入れてもカンマで切ってもよい。

たとえば

AX = (Aの内容Rタイプ) 空白8字 N = (MAの内容整数2ケタ)
を印刷したければ

```
PRINT ("AX='R, 8B 'N='ZD/ ", A, MA);
```

となる。

5.2 読込命令(④ 4.8.3)

A(I), K, B(I), Lをこの順にテープより読み込みたいければ

```
READ(A(I), K, B(I), L);
```

とする。読込リストはいくらつづいてもカンマで切ればよい。データテープ上のタイプがreal integerの何れにかかわらずリストにあげられた変数のタイプに従って自動的に変換される。

読込命令とデータテープとは必ずその順序が対応してなければならない。

また配列がreal A(1:4, 1:5) ; などと宣言されておりデータテープ上行なりに20個のデータが宣言された範囲通りに並んでいれば

```
READ(A);
```

と変数だけでなく配列名を書いてもよいが, 宣言された範囲よりもデータ数が少ないときは for命令によるか, 又は次の配列読込命令を用いる。

5.3 配列読込命令(④ 4.8.4)

20行30列の配列Aに対し

A₁₂ A₁₃ A₁₄ A₂₂ A₂₃ A₂₄

と並んでいるデータテープより読み込むには

READRRAY (A, 1, 2, 2, 4);

とすればよい。すなわち配列名の次に行列等の下限上限の順にカンマで区切って指定する。

3次元以上も限界リストを拡大すればよい。

5.4 印刷命令 (Ⓖ 4.8.2)

PRINT ('<フォーマット>', <リスト>);

でよい。フォーマットとリストとはカンマで区切り、リストはフォーマットと対応した個数だけカンマで切って並べる。

勿論リスト不用のときはフォーマットだけでよい。たとえば

uuNUMBERuuRESULTuuNUMBERuuRESULT

という字を行頭に印刷したければ

PRINT ('2(' ##NUMBER##RESULT')/');

でよい。

5.5 パンチ命令 (Ⓖ 4.8.5)

パンチテープを作りたければ

PUNCH (<フォーマット>', <リスト>);

とする。

フォーマットはR, I, LおよびAタイプに限り、R I L以外の形のときはこの標準形に変換される。

またAタイプで指定すると、たとえばinteger KにTOKYOというアルファタイプの情報が入っていれば

PUNCH ('5A', K);

によって=TOKYO、と=が前についてパンチされる。また情報の終りには自動的に、が入れられる。

5.6 PUT, GET命令 (Ⓖ 4.8.6)

同じプログラムの中で1時ある情報を外部メモリーに出し又後これを内部に移して使用したいときに用いる。格納目印(名札)としては無符号整数を用い、同じ名札を指定してPUTしたときは新しい情報に書き換えられ、容量が前のものより大きければ自動的に他の場所にも分けて書かれる。

PUT (<無符号整数>, <リスト>);

GET (<無符号整数>, <リスト>);

リストとしては変数, 配列名が用いられる。

5.7 MT入出力命令 (④ 4.8.7)

情報を外部メモリーに出して他のプログラムで使用しないときに用いる。
この場合はユニットとして土研では1又は2 (又はこれを入れた単純変数)
指定し, ④をたよりに次の方法で使用する。

(1) 書き込み (④ 4.8.7)

たとえば real タイプの配列 P をユニット 1 に 1 から N まで書き出したければ, まづ巻きもとして 20 単位以上消去した後④を1つ記入してから書き込む。

```
REWIND (1, 20);  
MTWRITE (1, '↑');  
for I := 1 step 1 until N do  
    MTWRITE (1, 'R', P(I));  
MTWRITE (1, '✓');
```

(2) 読み込み (④ 4.8.7)

上の例の配列を A に読み込んで使用したいときには巻きもとした後
(このときは消去しなくてもよい), ④をさがしてから読み出す。

```
REWIND (1);  
TMSEARCH (1, 1);  
for I := 1 step 1 until N do  
    MTREAD (1, 'R', A(I));
```

(3) REWIND 命令 (④ 4.8.9)

これは前にユニット後に消去ブロック数を書く, 一般に 20 以上をとる。

読み出しのときはユニットだけを指定して良い。

(4) TMSEARCH 命令 (④ 4.8.8)

MT上では '↑' が出てくるたびに1つづつ④が書き込まれるのでこの数を頼りにテープ上の位置を確定する。前にユニット後に何番目の④であるかを指定する。負を指定したときはその個数巻き送って最後に1つ巻

きもどる。

5.8 特殊な代入命令

(1) くり返し数代入命令 (Ⓔ 4.8.10)

N個のブランクの右にK個*を打ちかければ

REPX (N, K);

PRINT ('X' BX ('*') /');

とREPXによりフォーマットのXに対しリスト順に値を代入することができる。新しく代入されるまで一旦入った値は保持される。

(2) アルファ代入命令 (Ⓔ 4.8.11)

TOKYOという字を integer Pに代入するときは

P := EQUIV ('TOKYO');

とする。スペースは#を用い1語は7文字以下である。

6° 宣言 (declaration D) (Ⓔ 5)

宣 言	タイプ宣言	real, integer, Boolean
	配列宣言	{ (real) array integer array Boolean array
	手続宣言	{ 手続 procedure 関数指示子 { real procedure integer procedure Boolean procedure
	スイッチ宣言	switch

‘H’ にはこの他に

イクスターナル宣言 external がある。

6.1 タイプ宣言 (Ⓔ 5.1)

宣言子の後に単純変数をカンマで区切って書く。

一段落でセミコロンを入れる。

6.2 配列宣言 (Ⓔ 5.2)

1. 宣言子の後に名前を書き [] の中に添字の限界を行について下限上限, 列について下限上限の順に書く。

何次元でもかまわない。

2. また同じ限界なら名前だけをならべてもよい。

```
array A, B[1:10], PQ[1:5, -8:4];
```

```
Boolean array O[0:4];
```

real array のときは単に array としてもよい。上下限は何でもよいが
(下限) $\leq$ (上限) でなければならぬ。

(‘H’では上下限は整数のナンバーに限る。)

6.3 スイッチ宣言 (⊙ 5.3)

一種の名札の配列で、この要素かまたは他のスイッチ指示子であってもよい。

```
switch Q := L1, L2, if A=B then D else LD, RS, S[I];
```

この場合 L1, L2, D, LD, RS は名札, S[I] は他のスイッチリストの要素である。(‘H’ではそれ以前に宣言されていなければならない)

6.4 イクスターナル宣言 (⊙ 5.5)

1. ‘H’では既に attach によって機械語に翻訳された procedure をその
ジョブネームとしての名前を頼りに引用することができる。そのために
は使用しようする既に attach されているジョブネームのものを、
external の後にカンマでつないで宣言すればよい。

```
external PRINTF, RENRITU;
```

2. 関数指示式は

```
external real TAN;
```

```
external integer SUM;
```

等と後にタイプを付して宣言する。

3. external procedure の中で external 宣言することはできない。

4. これに対応する attach 命令は

```
<ジョブネーム>① attach <手続宣言>; end ; とすればよい。
```

end の前の ; はこの場合に限りどうしても必要である。

6.5 手続宣言 (⊙ 5.4)

1. 手続宣言は

```
procedure 手続名(FP部);
```

```
<value 指定部>;
```

```
<明細指定部>;
```

(heading部)

<手続本体> ;

の形をとる。(関数指示子宣言はprocedureの前にタイプを入れる)
形式的パラメタのないときは、手続名の後に手続本体が来るし、value
指定部のない場合もあり得る。

('H'ではFP部にあるパラメタはすべて明細指定されなければならない
為まだarrayはvalue指定できない)

2. 関数指示子は1価関数に限り、宣言中で答を自分自身の名前に代入する。そしてその答のタイプをprocedureの前にreal procedure などとつける。使用は算術式で行なう。
3. 実行に当ってプログラム本体より値を貰ってくるパラメタはvalue指定しておいた方がよい。value指定してないパラメタはすべて名前でよばれる(call by name)。この手続の結果答が得られるものはvalue指定できない。
4. 明細指定部は、明細指定子の後にパラメタがカンマでならべられるが、明細指定子としては次のものがある。

string, real, integer, Boolean, array, real array,
integer array, Boolean array, label, switch, procedure, real procedure, integer procedure, Boolean
procedure, label,

('H'ではAPとFPのタイプ種類は合致しなければならないがvalue指定されたときはinteger realのタイプが異なってもよい、またheading部にcommentを書くことを禁止する。)

APには勿論式を書くことができる。

('H'では八進表現で書かれた機械語をprocedure本体中で書くことを認める。このときはcodeの後に8進14ケタ単位の機械語を書いたものである。)

7° データ(第7)

1. 'H'ではデータとしてreal, integer, Booleanおよびalphaタイプの情報を書くことを認める。

何れもREAD命令に対応しカンマで区切られ、最後にカンマはセミコロ

ンの来たものである。

2. BooleanはTRUE又はFALSE, alphaは=TOKYO, などと7文字以内のものの前に=を付し, 読み込まれる変数はintegerに限る。7文字以上のときは最後の7文字が読み込まれ, 7文字以下のときは以下ブラントとなる。
3. データタイプがinteger, realタイプの何れであっても読み込まれる変数のタイプに従って自動変換される。
4. データタイプの読込一時中止は /①
プログラムタイプの読込一時中止は halt ;①
である。
5. データテープ上, , となっているときは0が読み込まれる。
6. 同じ値を何回も続けて読ませるときはたとえば, 3.5を6回くり返すなら, 6 * 3.5, としてもよい。

8° 各種コンパイラのALGOL 60との相違点

8.1 OKITAC 5090-H用 NEW ALGOL-H

(1) 文法上の制限

1. 名前に小文字を認めない。また6文字まで判別する。
2. 名札に無符号整数を認めない。
3. 標準関数, 論理値換算関数は大文字とする。
4. 配列宣言の上下限は整数のナンバーに限る。
5. Xの代りに*, Lの代りに#を使用する。
6. realは $10^{\pm 76}$ (1部 ± 77), integerは12ケタ (1部13ケタ) までとする。
7. 条件付算術式のelseの両側の式は同じタイプに限る。
8. FOR命令のuntilの後は2進換算の都合上完全に実行しないことがある。
9. 手続宣言の形式パラメタはすべて明細指定されなければならない。
10. 手続宣言でarrayはvalue指定できない。
11. ownを認めない。
12. 手続宣言のheading部にcommentを書くことを許さない。

13. 手続宣言および手続命令において、パラメタデリミタは、しか認めない。
14. 手続の call by name のパラメタは型が一致していなければならない。
15. 算術式のカッコの直後の文字数字は無視する。

(2) 文法上の拡張

1. アルファモードの代入を EQUIV により、読込を先頭の = によって integer の変数に対し認める。またアルファモードの印字をフォーマット中の A によって認める。また 7A 以上を指定したときは 7 文字をこえる数だけブランクが入り、その後に先頭から 7 文字入る。
2. attach によって procedure をマシンコードに変換したものを MT に格納し、external 宣言によってそのジョブネームを頻りに自由に使用することを認める。
3. code によって procedure 中で機械語との接続を認める。
4. データタイプの読込停止を halt ; ① により、データタイプの読込停止を / ① により認める。
5. プログラムには必ずジョブネーム ① を付し、これにより 2 座席の割り込みを許す。

(3) 入出力命令

1. 読込 READ ($\tilde{V}$, $\tilde{V}$ ) ;
READARRAY ($\tilde{I}$, L₁, U₁, L₂, U₂) ;
2. 印刷 PRINT ('フォーマット', リスト) ;
フォーマット中 A (アルファ), B (ブランク), O (カンマ),
D (数字), F (ブーリアン), I (ノンフォーマットの整数),
L (ノンフォーマットのブーリアン), R (ノンフォーマットの実数),
S (文字数字), X (レプリケータ), Z (ゼロサプレス数字),
+ (正負記号), - (負の時のみの印字), . , 10 , ,
'.....' (.....そのものの印字), / (印刷実行改行), ; (改行なしの印刷実行)
3. パンチ PUNCH ('フォーマット', リスト) ;
フォーマットは R I L A に限りこれ以外のものはすべて自動的にノ

ンフォーマットとし、数値の間にはカンマを入れてしまう。アルファ
タイプのもは=が前に入る。

3. MT関係

同一プログラム内 PUT (レーベル, リスト);

GET (レーベル, リスト);

レーベルは符号なし整数

プログラム間 MTWRITE (ユニット, 'フォーマット',
リスト);

MTREAD (ユニット, 'フォーマット',
リスト);

ただし一連の独立して読みたいMTWRITEの前に

REWIND (ユニット, ナンバー);

MTWRITE (ユニット, '↑');

を入れ後にMTWRITE (ユニット, '／');を入れる。

MTREADの前に

REWIND (ユニット);

TMSEARCH (ユニット, ナンバー);

を入れる。引きつづいて読むときはこの2つは不用である。

REWINDのナンバーはイレーズ単位数で20以上程度,

TMSEARCHのナンバーはWRITE時の↑の数である。

4. フォーマット中のくり返し制御

たとえばN個のブランクの右にK個*を打ちたいときは

REPX (N, K);

PRINT ('XBX ('*') / ');

とする。

5. データテープ

READ命令に対応しカンマで区切る。アルファタイプのもは=
を先頭とする。最後は、又は;とする。、となったときは0が読み
込まれ、タイプはREAD命令のリストに対応し自動変換される。

8.2 OKITAO 5090H用(OLD) ALGOL-H

(I) 文法上の制限

1. 名前に小文字を認めない。
2. 名札に無符号整数を認めない。
3. 標準関数は大文字とする。
4. 換算関数はFLOAT, FIXとしentierを認めないほかFIXは切りすてとする。
5. $\times$ の代りに*, $\cup$ の代りに#を使用する。
6. realは $10^{\pm 78}$ (1部±77), integerは12ケタ(1部13ケタ)までとする。
7. 混合演算を認めない。
8. FOR命令のuntilの後は2進換算の都合上完全に実行しないことがある。
9. 手続宣言の形式パラメタはすべて明細指定されなければならない。
10. 手続宣言でarrayはvalue指定できない。
11. ownを認めない。
12. 手続命令の実パラメタはvalue指定してない限り式を書くことを許さない。
13. 手続宣言および手続命令においてパラメタデリミタは、しか認めない。
14. 関数指示子を認めない。
15. スイッチ宣言の要素は名札に限る。
16. $\equiv$ を認めない。
17. FOR命令のwhileエレメントを認めない。
18. FOR命令のコントロール変数は単純変数に限る。
19. 代入命令の左右辺の型は同じものでなければならない。
20. 配列, スイッチの添字式は整数の結果を与えるものでなければならない。
21. 配列宣言の上下限は整数のナンバーに限る。
22. 関係の左右辺の型は同じものでなければならない。
23. 明細指定子にstring, real procedure, integer procedure, Boolean procedure, labelを認めないほかprocedureと明細指定されたものはパラメタなしのものに限る。

24. 条件付算術式でelseの左右のA_Eのタイプは一致しなければならない。

25. FOR命令のstep untilエレメントのA_Eのタイプは一致しなければならない。

(2) 文法上の拡張

1. プログラムテープの読込停止をhalt ; ① ②データテープの読込停止を/① ②によって認める。

(3) 入出力命令

1. 読込 READ (V, V, ……);
データテープと型が一致しなければならない。

2. 印刷 PRINT (n, V);
real 及び integer の有効ケタ数をnとする。

PRINT (n₁, n₂, V);

小数点以上 及び 以下のケタ数指定

ORLF (n); 印刷実行しn行改行

SPACE (n); ブランクn個

PRINT ('……'); ……を印刷

3. MT関係

PUT (レーベル, Y);

GET (レーベル, Y);

レーベルは無符号整数とする。

8.3 FACOM 230-30 用ALGOL (231共通)

(1) 文法上の制限

1. 換算関数
2. +の代りにdiv, $\geq$, $\leq$ の代りに $\geq$, $\leq$ を, $\wedge$, $\vee$, $\neg$, $\equiv$ の代りにand or not imp eqv $\times$ の代りに*を用いる。
3. realは $10^{\pm 50}$ までとし, 指定しないときはinteger 4ケタreal 10ケタとする。
4. 手続宣言の形式パラメタはすべて明細指定を要する。
5. 手続宣言でarrayはvalue 指定できない。
6. ブーリアンの読込を認めない。

7. スペースは判別するので、あけるべき所は必ずスペースをとらなければならない。
8. 名前は6字まで判別する。
9. 最後のendの後10以上のスペース又は;さん孔すること。
10. 手続のFPとAPの型種類は一致しなければならない。
11. 立札は名前に限る。

(2) 文法上の拡張

1. 先頭のbeginの後でcomment reallength := 15, integer-length := 8;などと指数指定ができる。

(3) 入出力命令

1. 読込Readreal (V);

Readinteger (V);

1つずつの読込命令に限り、タイプは合っていないなければならない。またブーリアンの読込を認めない。

Readarray (Y);

2. 印刷

Print (AE); realのノンフォーマット印刷

Printinteger (AE); integerのノンフォーマット印刷

Printfix (AE, n₁, n₂); realを小数点上h, ケタ
下n₂ケタで印刷

Printstring ('.....'); を印刷

Space(n); ブランクn個

ORLF(n); 印刷実行しn回改行

ORLF; " 1回 "

Printx(AE, n) 整数を有効nケタで印刷

Printreal(AE, n) 実数を有効nケタで印刷

8.4 FAOOM 230-50用 ALGOL

(1) 文法上の制限

1. ownを認めない。
2. 名札は名前に限る。
3. 名前の有効長は12文字とする。

4. 実際に評価してみなければならぬ $\tilde{A}\tilde{B}$ のタイプはすべて real とする。
 5. call by name のパラメタは integer と明細指定されなければすべて real として扱う。
 6. $\tilde{F}\tilde{P}$ と $\tilde{A}\tilde{P}$ のタイプは一致しなければならない。
 7. call by value のパラメタはすべて明細指定必要
- (2) 文法上の拡張
1. 標準関数に LOG, TANH, RANDOM を追加する。
- (3) 入出力命令
1. 読込
 $\text{READ}(\tilde{V}, \tilde{V} \dots \dots)$; 変数の型に合わせて読込
 2. 印刷
 $\text{PRINT}(\tilde{A}\tilde{B})$; $\tilde{A}\tilde{B}$ の型に合わせて 1 個印刷
 $\text{PRINTSTRING}(' \dots ')$; ... を印刷
 $\text{RSPACE}(\tilde{A}\tilde{E})$;
 CRLF ; 1 回復帰改行
 $\text{OUTPUT}(200, \text{'フォーマット'}, \text{'リスト'})$;
 フォーマット中で, Z, D, +, -, A, F, I, R, L, S,
 $\uparrow$, V, / は NEW ALGOL-H と同じであるがスペースはじと
 する。

8.5 TOSBAC 用 KT-ALGOL

(1) 文法上の制限

1. 名前に小文字を認めない。
2. デソミタは 'BEGIN' と大文字を ' ' で囲んで示す。
3. $\geq > = < \leq \neq$ の代りに 'GEQ' 'GRT' 'EQL' 等で示す。
4. $\uparrow$ の代りに % を用いる。
5. $\cdot$ の代りに # を用いる。
6. own array 宣言の上下限は整数のナンバーに限る。
7. 立札は名前に限る。
8. call by value の $\tilde{F}\tilde{P}$ はすべて明細指定必要。
9. call by name の形式パラメタで integer のみ明細指定すれば他

の算術変数は real 扱いとなる。

(2) 文法上の拡張

(3) 入出力命令

1. 読込

READ (V, V);

2. 印刷

WRITE (n, リスト, フォーマット);

8.5 ODO3600 用 ALGOL

(1) 文法上の制限

1. 名前は 64 文字まで有効
2. array は value 指定できない。
3. call by name のパラメタで算術値のものは, integer と指定されなければすべて real 扱いとなる。
4. Boolean は boolean とする。
5. real は 10 ± 308 まで integer は 14 ケタまでとする。
6. 名前は小文字を認めない。

(2) 文法上の拡張

1. external 宣言を認める。
2. PRINT 命令のリストに for expression ともしうべきものを認める。
3. WRITE 命令のリストに conditional string expression ともしうべきものを認める。

(3) 入出力命令

1. 読込

READ (V, V); 変数の型に合わせて読込

2. 印刷

WRITE ('...'); ...を印刷
WRITE (if D $\neq$ 0 then 'TRUE' else 'FALSE');
PRINT (A, B, A, B); 'ンフォーマット' で印刷
PRINT (for I := 1 step 1 until 10 do A(I));
OUTPUT (61, 'フォーマット', リスト);

フォーマット中の文字はフォートランと似ておりHはキャラクタ, Eは指数部を含む real, Fは小数点のみを含む real となる。

NEW ALGOL-H 文法詳論

** 中 村 慶 一

はじめに

電算速報 633 に NEW ALGOL-H の文法概要を紹介したが, Backus notation による厳密な文法を要求される向きが多いので多少くだいて御紹介しておく

1^a 一般的説明

以下の文章を次の記号を用いる

記 号	意 味
-----	-----

:: =	(左辺) とは (右辺) のようなものである 又は
------	------------------------------

①	磁気テープの特殊信号として用いるテープマーク ALGOL の単語であることを示す
---	---

アンダライン

<.....> というもの

また次の定義を下しておく

<空> :: =

これで<空>とは何もないことを意味するものと定義したことになる。

以下 ALGOL-60 と変っている部分に ●印をつける。

また記述の簡略化のために次の略号を用いる。

A E	算術式 (arithmetic expression)
A P	実パラメタ (actual parameter)
B E	ブール式 (Boolean expression)
D	宣言 (declaration)
D E	指示式 (designational expression)
F P	形式パラメタ (formal parameter)
F S	for 命令 (for statement)
I	名前 (identifier)
L	名札 (label)

** 技術管理室長 工博

$\overline{S}$	命令 (statement)
$\overline{SA}$	単純算術式 (simple arithmetic expression)
$\overline{SB}$	単純ブール式 (simple Boolean expression)
$\overline{SD}$	単純指示式 (simple designational expression)
$\overline{SV}$	単純変数 (simple variable)
$\overline{ST}$	ストリング (string)
$\overline{US}$	無条件命令 (unconditional statement)
$\overline{UI}$	無符号整数 (unsigned integer)
$\overline{V}$	変数 (variable)

2° 基本記号 (basic symbols)

<基本記号> ::= <文字> | <数字> | <論理値> | <限定詞>

2.1 文字 (letters)

- <文字> ::= A | B | C | D | E | F | G | H | I | J | K | L | M |
N | O | P | Q | R | S | T | U | V | W | X | Y | Z
ALGOL-60には小文字もあるがNEW-Hでは大文字に限る(もちろん限定詞としては小文字も使用できる)。

2.2.1 数字 (digits)

<数字> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

2.2.2 論理値 (logical values)

- <論理値> ::= TRUE | FALSE
ALGOL-60ではtrue falseである。

2. 限定詞 (delimiters)

<限定詞> ::= <演算子> | <区切り子> | <括弧> | <宣言詞> |
<指定詞>

<演算詞> ::= <算術演算詞> | <関係演算詞> | <論理演算詞> |
<逐次演算詞>

- <算術演算詞> ::= + | - | * | / | ÷ | ↑
ALGOL-60では掛算が×であるが大文字のエックスとの混乱を避けるため*を用いる。
<関係演算詞> ::= < | ≤ | ≥ | > | ≠

<論理演算詞> ::= $\equiv$ | $\supset$ | $\vee$ | $\wedge$ | $\neg$

- <逐次演算詞> ::= go to | if | on | then | else | for |
do | code

ALGOL-60にはon codeはない

- <区切り子> ::= $=$, | $.$ | $_{10}$ | $:$ | $:$ | $:$ | $:=$ | step | until |
while | comment | $\#$ | halt

ALGOL-60ではスペースは $\#$ ではなく $\sqcap$ で示し、またhaltはない。

- <括弧> ::= (|) | (|) | ' | ' | begin | end | attach

ALGOL-60にはattachはない。

- <宣言子> ::= Boolean | integer | real | array | switch |
procedure | external

ALGOL-60にはexternalはない。またownがある。

<指定子> ::= string | label | value

2.4 名前 (identifier, $\tilde{I}$ と略)

<名前: $\tilde{I}$ と略> ::= <文字> | < $\tilde{I}$ > <文字> | < $\tilde{I}$ > <数字>

先頭の6字だけを判別する。従ってMATRIXAとMATRIXBは区別されない。A12Bは良いが45Dは不可

2.5 数値 (numbers)

<無符号整数: $\tilde{U}I$ と略> ::= <数字> | < $\tilde{U}I$ > <数字>

<整数> ::= < $\tilde{U}I$ > | +< $\tilde{U}I$ > | -< $\tilde{U}I$ >

<小数部> ::= $\cdot$ < $\tilde{U}I$ >

<指数部> ::= $_{10}$ <整数>

<十進数値> ::= < $\tilde{U}I$ > | <小数部> | < $\tilde{U}I$ > <小数部>

<無符号数値> ::= <十進数値> | <指数部> | <十進数値> <指数部>

<数値> ::= <無符号数値> | +<無符号数値> | -<無符号数値>

従って-5 -4. 4.6 $_{10}$ -2 $_{10}$ -2 -4.6 $_{10}$ 5 などはずべて数値であるが、3. 4. $_{10}$ 3 4. $_{10}$ 2.5 4.9.2 $_{10}$ などは不可。又取扱う数値は整数で12ケタまで実数で10 $^{\pm 75}$ までにしておく。

2.6 スtring (string, $\tilde{S}T$ と略)

<プロパー $\tilde{S}T$ > ::= <'> 以外の基本記号の任意の並び>

$$\begin{aligned} \langle \text{オープン} . \check{S} \check{T} \rangle &::= \langle \text{プロパー} . \check{S} \check{T} \rangle \mid \langle \text{オープン} . \check{S} \check{T} \rangle \mid \\ &\quad \langle \text{オープン} . \check{S} \check{T} \rangle \langle \text{オープン} . \check{S} \check{T} \rangle \\ \langle \check{S} \check{T} \rangle &::= \langle \text{オープン} \check{S} \check{T} \rangle \end{aligned}$$

3° 式 (expressions)

$$\langle \text{式} \rangle ::= \langle \check{A} \check{E} \rangle \mid \langle \check{B} \check{E} \rangle \mid \langle \check{D} \check{E} \rangle$$

3.1 変数 (variables, $\check{V}$ と略)

$$\langle \text{変数名} \rangle ::= \langle \check{I} \rangle$$

$$\langle \check{S} \check{V} \rangle ::= \langle \text{変数名} \rangle$$

$$\langle \text{添字式} \rangle ::= \langle \check{A} \check{E} \rangle$$

$$\langle \text{添字リスト} \rangle ::= \langle \text{添字式} \rangle \mid \langle \text{添字リスト} \rangle \langle \text{添字式} \rangle$$

$$\langle \text{配列名} \rangle ::= \langle \check{I} \rangle$$

$$\langle \text{添字変数} \rangle ::= \langle \text{配列名} \rangle [\langle \text{添字リスト} \rangle]$$

$$\langle \text{変数} \rangle ::= \langle \check{S} \check{V} \rangle \mid \langle \text{添字変数} \rangle$$

従って AB 045 などは単純変数 ($\check{S} \check{V}$) であり、

$P(3)$ $RA(4K)$ $PQR(I, J, K)$

$STU(\text{if } A=B \text{ then } M \text{ else } N, J)$ などは添字変数である。

また添字が real のときは四捨五入される。

3.2.1 関数指示子 (function designator)

$$\langle \text{手続名} \rangle ::= \langle \check{I} \rangle$$

$$\langle \check{A} \check{P} \rangle ::= \langle \check{S} \check{T} \rangle \mid \langle \text{式} \rangle \mid \langle \text{配列名} \rangle \mid \langle \text{スイッチ名} \rangle \mid$$

$$\langle \text{手続名} \rangle$$

● $\langle \check{A} \check{P} \text{ リスト} \rangle ::= \langle \check{A} \check{P} \rangle \mid \langle \check{A} \check{P} \text{ リスト} \rangle, \langle \check{A} \check{P} \rangle$

$$\langle \check{A} \check{P} \text{ 部} \rangle ::= \langle \text{空} \rangle \mid (\langle \check{A} \check{P} \text{ リスト} \rangle)$$

$$\langle \text{関数指示子} \rangle ::= \langle \text{手続名} \rangle \langle \check{A} \check{P} \text{ 部} \rangle$$

従って $TAN(A)$ $SIMPLEX(N, A, B)$ RAN などは関数指示子であり得る。

ALGOL-60では $\check{A} \check{P}$ リストに、の外にパラメタ・デリミタ) <文字>: (を認める。

● 3.2.4 標準関数 (standard function)

ABS $SIGN$ SIN COS $SQRT$ $ARCTAN$ LN

EXP

ALGOL-60では小文字である。

● 3.2.5 換算関数 (transfer function)

ENTIER FLOAT FIX

ALGOL-60ではentierでそれを越さぬ最大の整数をとる。従ってENTIER(-8.6)は-9である。

FLOAT FIXはOLD-Hとの接続のために残した。

3.3 算術式 (arithmetic expression, $\tilde{A}\tilde{E}$ と略)

<加減演算子> ::= + | -

<乗除演算子> ::= * | / | +

<プライマリ> ::= <無符号数値> | < $\tilde{V}$ > | ($\tilde{A}\tilde{E}$) |

<関数指示子>

<ファクタ> ::= <プライマリ> | <ファクタ> ↑ <プライマリ>

<ターム> ::= <ファクタ> | <ターム> <乗除演算子> <ターム>

< $\tilde{S}\tilde{A}$ > ::= <ターム> | <加減演算子> <ターム> |

< $\tilde{S}\tilde{A}$ > <加減演算子> <ターム>

<IFクローズ> ::= if < $\tilde{B}\tilde{E}$ > then | on < $\tilde{U}\tilde{I}$ > then

< $\tilde{A}\tilde{E}$ > ::= < $\tilde{S}\tilde{A}$ > | <IFクローズ> < $\tilde{S}\tilde{A}$ > else < $\tilde{A}\tilde{E}$ >

従って演算順序は下の順で同順位の場合は式の左側のものが優先する。

1 ()

2 ↑

3 * / +

4 + -

また+は整数割算で $5 \div 2 = 2$ である。

Iをintegerタイプ, Rをrealタイプとしたとき

$A \uparrow I$ は $I > 0$ なら $A * A * \dots * A$ (i回)

$I = 0$ で $A \neq 0$ なら Aと同じタイプの1

$A = 0$ なら 誤り

$I < 0$ で $A \neq 0$ (A : real) なら $1/(A * \dots * A)$

(A : integer) なら 0

$A \uparrow R$ は $A > 0$ なら $EXP(R * LN(A))$ で real

$A \leq 0$ なら $0 \uparrow 0 = 1.0$ 以外は誤り

条件は算術式で else の左右辺は同じタイプでなければ誤りとする。

またコンパイラの都合で)の直後の文字数字は無視するので $(A+B)$

$* C + D$ を $(A+B)C + D$ と書いたときエラーチェックを行なわ

ないで $(A+B)+C$ の結果を出すので注意する必要がある。

もちろん+の両側は integer に限るがその他の場合もし混合して居ればすべて real の結果となる。

3.4 ブール式 (Boolean expression, $\widetilde{B\bar{E}}$ と略)

<関係演算子> ::= < $\leq$ $=$ $\geq$ $>$ $\neq$

<関係> ::= < $\widetilde{S\bar{A}}$ > <関係演算子> < $\widetilde{S\bar{A}}$ >

<ブール・プライマリ> ::= <論理値> | < $\widetilde{V}$ > | <関係> | (< $\widetilde{B\bar{E}}$ >)

<ブール・セコンダリ> ::= <ブール・プライマリ> |

$\neg$ <ブール・プライマリ>

<ブール・ファクタ> ::= <ブール・セコンダリ> |

<ブール・セコンダリ> $\wedge$ <ブール・セコン

ダリ>

<ブール・ターム> ::= <ブール・ファクタ> |

<ブール・ファクタ> $\vee$ <ブール・ファクタ>

<含意> ::= <ブール・ターム> $\supset$ <ブール・ターム>

< $\widetilde{S\bar{B}}$ > ::= <含意> | <含意> $\equiv$ <含意>

< $\widetilde{B\bar{E}}$ > ::= < $\widetilde{S\bar{B}}$ > | <IFクローズ> < $\widetilde{S\bar{B}}$ > else < $\widetilde{B\bar{E}}$ >

従って演算順序は

1. ()
2. 算術演算子
3. 関係演算子
4. $\neg$
5. $\wedge$
6. $\vee$
7. $\supset$
8. $\equiv$

である。

$\tilde{B}E$ は結果として1つの論理値に帰着する表現である。

また

```
if if if A=B then C1≠D else C2=Q then S>R
else S=45 then P1≤Q1 else AB=X48 ∧ (P+Q)
*R<365
```

は1つのブール式である。

関係演算子の両側のタイプが異なるときは real として比較される。

3.5 指示式 (designational expression, $\tilde{D}E$ と略)

$\langle \tilde{L} \rangle :: = \langle \tilde{I} \rangle$

$\langle \text{スイッチ名} \rangle :: = \langle \tilde{I} \rangle$

$\langle \text{スイッチ指示子} \rangle :: = \langle \text{スイッチ名} \rangle [\langle \text{添字式} \rangle]$

$\langle \tilde{S}D \rangle :: = \langle \tilde{L} \rangle | \langle \text{スイッチ指示子} \rangle | (\langle \tilde{D}E \rangle)$

$\langle \tilde{D}E \rangle :: = \langle \tilde{S}D \rangle | \langle \text{IFクローズ} \rangle \langle \tilde{S}D \rangle \text{ else } \langle \tilde{D}E \rangle$

ALGO L-60では名札として無符号整数を認める。

$\tilde{D}E$ は結果として1つの名札に帰着する表現であり

```
if P=Q then ( if R≠S then L1 else L2 )
else if P>300 then OPEN else END1
```

は1つの $\tilde{D}E$ である。

4° 命令 (statement, $\tilde{S}$ と略)

4.1 複合命令 (compound $\tilde{S}$) とブロック (block)

$\langle \text{名札なし基本命令} \rangle :: = \langle \text{代入命令} \rangle | \langle \text{GO TO命令} \rangle |$

$\langle \text{空命令} \rangle | \langle \text{手続命令} \rangle | \langle \text{入出力命令} \rangle$

$\langle \text{基本命令} \rangle :: = \langle \text{名札なし基本命令} \rangle | \langle \tilde{L} \rangle : \langle \text{基本命令} \rangle$

$\langle \text{無条件命令} \rangle :: = \langle \text{基本命令} \rangle | \langle \text{複合命令} \rangle | \langle \text{ブロック} \rangle$

$\langle \text{命令} \rangle :: = \langle \text{無条件命令} \rangle | \langle \text{条件命令} \rangle | \langle \text{FOR命令} \rangle$

$\langle \text{複合尾部} \rangle :: = \langle \text{命令} \rangle \text{ end} | \langle \text{命令} \rangle ; \langle \text{複合尾部} \rangle$

$\langle \text{ブロック頭部} \rangle :: = \text{begin} \langle \text{宣言} \rangle | \langle \text{ブロック頭部} \rangle ; \langle \text{宣言} \rangle$

$\langle \text{名札なし複合命令} \rangle :: = \text{begin} \langle \text{複合尾部} \rangle$

$\langle \text{複合命令} \rangle :: = \langle \text{名札なし複合命令} \rangle | \langle \tilde{L} \rangle : \langle \text{複合命令} \rangle$

$\langle \text{名札なしブロック} \rangle ::= \langle \text{ブロック頭部} \rangle ; \langle \text{複合尾部} \rangle$
 $\langle \text{ブロック} \rangle ::= \langle \text{名札なしブロック} \rangle | \langle \tilde{L} \rangle : \langle \text{ブロック} \rangle$
 $\langle \text{プログラム} \rangle ::= \langle \text{名札なしブロック} \rangle | \langle \text{名札なし複合命令} \rangle$

● $\langle \text{ジョブネーム} \rangle ::= \langle \tilde{I} \rangle \textcircled{T}$

● $\langle \text{NEW ALGOL-H プログラム} \rangle ::= \langle \text{ジョブネーム} \rangle$
 $\quad \quad \quad \langle \text{プログラム} \rangle \textcircled{T} | \langle \text{ジョブネーム} \rangle$
 $\quad \quad \quad \langle \text{アタッチ} \tilde{S} \rangle \textcircled{T}$

ブロックは $\text{begin } \tilde{D} ; \dots ; \tilde{D} ; \tilde{S} ; \dots ; \tilde{S} \text{ end}$

複合命令は $\text{begin } \tilde{S} ; \dots ; \tilde{S} \text{ end}$

であるがプログラムミス (end の後のセミコロンを落すミス) を防止するために end の前に常に空命令をおき、また end の後も常にセミコロンを附加して

$\text{begin } \tilde{D} ; \dots ; \tilde{D} ; \tilde{S} ; \dots ; \tilde{S} ; \text{ end ;}$

$\text{begin } \tilde{S} ; \dots ; \tilde{S} ; \text{ end ;}$

と書く習慣をつけた方がよい。

NEW ALGOL H のプログラムには常にジョブネームが必要であって、これによってモニタは割込んだ仕事の状態を判別する。

4.2 代入命令 (assignment $\tilde{S}$)

$\langle \text{左辺} \rangle ::= \langle \tilde{V} \rangle : = | \langle \text{手続名} \rangle : =$

$\langle \text{左辺リスト} \rangle ::= \langle \text{左辺} \rangle | \langle \text{左辺リスト} \rangle \langle \text{左辺} \rangle$

$\langle \text{代入命令} \rangle ::= \langle \text{左辺リスト} \rangle \langle \tilde{A E} \rangle | \langle \text{左辺リスト} \rangle \langle \tilde{R E} \rangle$

4.3 GO TO 命令 (go to $\tilde{S}$)

$\langle \text{GO TO 命令} \rangle ::= \underline{\text{go to}} \langle \tilde{D E} \rangle$

4.4 空命令 (dummy $\tilde{S}$)

$\langle \text{空命令} \rangle ::= \langle \text{空} \rangle$

4.5 条件命令 (conditional $\tilde{S}$)

$\langle \text{IF クローズ} \rangle ::= \underline{\text{if}} \langle \tilde{B E} \rangle \underline{\text{then}} | \underline{\text{on}} \langle \tilde{U I} \rangle \underline{\text{then}}$

$\langle \text{無条件命令} \rangle ::= \langle \text{基本命令} \rangle | \langle \text{複合命令} \rangle | \langle \text{ブロック} \rangle$

$\langle \text{IF 命令} \rangle ::= \langle \text{IF クローズ} \rangle \langle \text{無条件命令} \rangle$

$\langle \text{条件命令} \rangle ::= \langle \text{IF 命令} \rangle | \langle \text{IF 命令} \rangle \underline{\text{else}} \langle \text{命令} \rangle |$

$\quad \quad \quad \langle \text{IF 命令} \rangle \underline{\text{else}} \langle \text{FOR 命令} \rangle |$

$\quad \quad \quad \langle \tilde{L} \rangle : \langle \text{条件命令} \rangle$

4.6 FOR命令 (for S)

$\langle \text{FORリスト要素} \rangle ::= \langle \tilde{A} \tilde{E} \rangle \mid \langle \tilde{A} \tilde{E} \rangle \text{step} \langle \tilde{A} \tilde{E} \rangle \text{until}$
 $\langle \tilde{A} \tilde{E} \rangle \mid \langle \tilde{A} \tilde{E} \rangle \text{while} \langle \tilde{B} \tilde{E} \rangle$
 $\langle \text{FORリスト} \rangle ::= \langle \text{FORリスト要素} \rangle \mid$
 $\langle \text{FORリスト} \rangle, \langle \text{FORリスト要素} \rangle$
 $\langle \text{FORクローズ} \rangle ::= \text{for} \langle \tilde{V} \rangle : \langle \text{FORリスト} \rangle \text{do}$
 $\langle \text{FOR命令} \rangle ::= \langle \text{FORクローズ} \rangle \langle \tilde{S} \rangle \mid \langle \tilde{L} \rangle : \langle \text{FOR命令} \rangle$

4.7 手続命令 (procedure S)

$\langle \tilde{A} \tilde{P} \rangle ::= \langle \tilde{S} \tilde{T} \rangle \mid \langle \text{式} \rangle \mid \langle \text{配列名} \rangle \mid \langle \text{スイッチ名} \rangle$
 $\langle \tilde{A} \tilde{P} \text{リスト} \rangle ::= \langle \tilde{A} \tilde{P} \rangle \mid \langle \tilde{A} \tilde{P} \text{リスト} \rangle, \langle \tilde{A} \tilde{P} \rangle$
 $\langle \tilde{A} \tilde{P} \text{部} \rangle ::= \langle \text{空} \rangle \mid (\langle \tilde{A} \tilde{P} \text{部} \rangle)$
 $\langle \text{手続命令} \rangle ::= \langle \text{手続名} \rangle \mid \langle \tilde{A} \tilde{P} \text{部} \rangle \mid \langle \tilde{L} \rangle : \langle \text{手続命令} \rangle$
 ALGOL-60ではA Pリストに、の外に) <文字> : (というバ
 ラメタデリミタを認める。

● 4.8 入出力命令 (input or output S)

4.8.1 フォーマット スtring (format string)

4.8.1.1 基本要素 (basic components)

$\langle \text{レプリケータ} \rangle ::= \langle \tilde{U} \tilde{I} \rangle \mid X$
 $\langle \text{挿入部} \rangle ::= B \mid \langle \text{レプリケータ} \rangle B \mid \langle \tilde{S} \tilde{T} \rangle$
 $\langle \text{挿入列} \rangle ::= \langle \text{空} \rangle \mid \langle \text{挿入列} \rangle \langle \text{挿入部} \rangle \mid$
 $\langle \text{レプリケータ} \rangle \langle \text{挿入列} \rangle$
 $\langle Z \rangle ::= Z \mid \langle \text{レプリケータ} \rangle Z \mid \langle \text{挿入列} \rangle O \mid$
 $\langle \text{レプリケータ} \rangle Z \langle \text{挿入列} \rangle O$
 $\langle Z \text{部} \rangle ::= \langle Z \rangle \mid \langle Z \text{部} \rangle \langle Z \rangle$
 $\langle D \rangle ::= D \mid \langle \text{レプリケータ} \rangle D \mid D \langle \text{挿入列} \rangle O \mid$
 $\langle \text{レプリケータ} \rangle D \langle \text{挿入列} \rangle O$
 $\langle D \text{部} \rangle ::= \langle D \rangle \mid \langle D \text{部} \rangle \langle D \rangle$
 $\langle \text{符号部} \rangle ::= \langle \text{空} \rangle \mid \langle \text{挿入列} \rangle + \mid \langle \text{挿入列} \rangle -$
 $\langle \text{整数部} \rangle ::= \langle Z \text{部} \rangle \mid \langle D \text{部} \rangle \mid \langle Z \text{部} \rangle \langle D \text{部} \rangle$
 $\langle T \text{部} \rangle ::= \langle \text{空} \rangle \mid T$
 Dは1けたの数値を印刷する (digit)

Zは無効の0のとき印刷しない。(従って・の後やDの後で用いてはならない)(zero suppression)

Cはカンマの印刷(comma) Tは打ち切りでDで切れたときは四捨五入されるがTを最後のDの次に入れると切りすてを行なう。+は符号の印刷 -は負のときのみ符号を印刷する。挿入部でBは空白(blank)でこのほか任意の文字のつながりを でくくったストリング(string S T)がみとめられるほかZ, D, Bはその前のレプリケータによってくり返しを命ずることもできる。レプリケータは原則として無符号整数であるが、場合によってR E P XによってXにある値を代入しておいてX回のくり返しを命ずることもできる。

4.8.12 ナンバーフォーマット

<U I フォーマット> ::= <挿入列> <整数部>
<小数部フォーマット> ::= <挿入列> <D部> <T部> |
V <挿入列> <D部> <T部>
<指数部フォーマット> ::= , <符号部> <U I フォーマット>
<10進数値フォーマット> ::= <U I フォーマット> <T部> |
<挿入列> <小数部フォーマット> |
<無符号整数フォーマット> <小数部フォーマット>
<ナンバーフォーマット> ::= <符号部> <10進数値フォーマット> |
<符号部> <10進数値フォーマット>
<指数部フォーマット> |
<レプリケータ> (<ナンバーフォーマット>)

4.8.13 ブール フォーマット

<F> ::= F | <F> F
<ブール部要素> ::= P | <F>
<ブール部> ::= <ブール部要素> | <レプリケータ> <ブール部> |
<挿入列> <ブール部>
<ブールフォーマット> ::= <挿入列> <ブール部> |
<レプリケータ> (<ブール部>)

Pおよび5Fの印刷様式は次の通り

	TRUE	FALSE
P	1	0
F	T	F
2F	TR	FA
5F	TRUE	FALSE
8F	TRUE	FALSE

4.8.1.4 ノンフォーマット

<ノンフォーマット要素> ::= I | R | L

<ノンフォーマット部> ::= <ノンフォーマット要素> |

<レプリケータ> <ノンフォーマット要素>

<ノンフォーマット> ::= <ノンフォーマット部> |

<挿入列> <ノンフォーマット部> |

<ノンフォーマット> <挿入列> |

<レプリケータ> (<ノンフォーマット>)

IとRはフォーマットに従いがいタイプの異なるときは変換されrealタイプにIフォーマットが指定されたときは四捨五入される。

I BB-ZZZZZZZZZZZZZD (13ケタ)

R BB-DDDDDDDDDD, + DD (9ケタ)

L BBFFFFFF

4.8.1.5 スtringフォーマット

<S> ::= S | <レプリケータ> S | <挿入列> <S> |

<S> <挿入列>

<S部> ::= <S> | <S部> S

<Stringフォーマット> ::= <S部> | <レプリケータ> (<Stringフォーマット>)

String1字に対応してフォーマット部にSを書く。

もちろんフォーマット部に挿入列としてストリングを書くこともできる。

4.8.1.6 アルファフォーマット

$\langle A \rangle ::= A \mid \langle \text{レプリケータ} \rangle A \mid \langle \text{挿入列} \rangle \langle A \rangle$
 $\langle A \text{ 部} \rangle ::= \langle A \rangle \mid \langle A \text{ 部} \rangle \langle A \rangle$
 $\langle \text{アルファフォーマット} \rangle ::= \langle A \text{ 部} \rangle \mid \langle \text{レプリケータ} \rangle (\langle \text{アルファフォーマット} \rangle)$

アルファタイプの情報は integer の変数に格納されているがこの一字に対応しフォーマット部ではAを用いる。

4.8.1.7 整列マーク

$\langle \text{整列マーク要素} \rangle ::= / \mid \uparrow \mid ;$
 $\langle \text{整列マーク部} \rangle ::= \langle \text{整列マーク要素} \rangle \mid \langle \text{レプリケータ} \rangle \langle \text{整列マーク要素} \rangle \mid \langle \text{整列マーク要素} \rangle \langle \text{整列マーク部} \rangle$
 $\langle \text{整列マーク} \rangle ::= \langle \text{整列マーク部} \rangle \mid \langle \text{レプリケータ} \rangle (\langle \text{整列マーク部} \rangle) ;$

	ラインプリンタ (LP)	磁気テープ (MT)	高速テープパンチ (HTP)
/	印刷し1行改行	IRG	無視
↑	新しい頁の行頭へ	TM	TM
;	改行しないで印刷	エラー	エラー

4.8.1.8 フォーマットストリング

$\langle \text{フォーマット アイテム} \rangle ::= \langle \text{ナンバーフォーマット} \rangle \mid \langle \text{ストリングフォーマット} \rangle \mid \langle \text{アルファフォーマット} \rangle \mid \langle \text{ブールフォーマット} \rangle \mid \langle \text{ノンフォーマット} \rangle \mid \langle \text{ST} \rangle$
 $\langle \text{フォーマット アイテム} \rangle \langle \text{整列マーク} \rangle \mid \langle \text{整列マーク} \rangle \langle \text{フォーマット アイテム} \rangle$
 $\langle \text{フォーマット プライマリ} \rangle ::= \langle \text{フォーマット アイテム} \rangle \mid \langle \text{挿入列} \rangle \langle \text{整列マーク} \rangle$

$\langle \text{レプリケータ} \rangle (\langle \text{フォーマットセコンダリ} \rangle) |$
 $(\langle \text{フォーマットセコンダリ} \rangle)$
 $\langle \text{フォーマットセコンダリ} \rangle :: = \langle \text{フォーマットプライマリ} \rangle |$
 $\langle \text{フォーマットセコンダリ} \rangle, \langle \text{フォーマットプライマリ} \rangle$
 $\langle \text{フォーマットストリング} \rangle :: = \langle \text{フォーマットセコンダリ} \rangle$

4.8.2 印刷命令

$\langle \text{印刷リスト要素} \rangle :: = \langle \tilde{A} \tilde{E} \rangle | \langle \tilde{B} \tilde{E} \rangle | \langle \tilde{S} \tilde{T} \rangle$
 $\langle \text{印刷リスト} \rangle :: = \langle \text{印刷リスト要素} \rangle |$
 $\langle \text{印刷リスト} \rangle, \langle \text{印刷リスト要素} \rangle$
 $\langle \text{印刷リスト部} \rangle :: = \langle \text{空} \rangle | , \langle \text{印刷リスト} \rangle$
 $\langle \text{印刷命令} \rangle :: = \text{PRINT} (\langle \text{フォーマットストリング} \rangle \langle \text{印刷リスト部} \rangle)$
 従って real P, Q integer J をノンフォーマットで印刷するなら
 $\text{PRINT} (' 2 R, I / ', P, Q, J);$
 となる。

4.8.3 読込命令

$\langle \text{読込リスト要素} \rangle :: = \langle \tilde{V} \rangle | \langle \text{配列名} \rangle$
 $\langle \text{読込リスト} \rangle :: = \langle \text{読込リスト要素} \rangle |$
 $\langle \text{読込リスト} \rangle, \langle \text{読込リスト要素} \rangle$
 $\langle \text{読込命令} \rangle :: = \text{READ} (\langle \text{読込リスト} \rangle)$
 従って array B (1 : 5) と real A とを B₁ B₂ B₃ B₄ B₅ A の
 順に書いてあるデータテープ上より読込むなら
 $\text{READ} (B, A);$
 でよい。配列宣言と異なった個数の配列読込みは次の配列読込命令によ
 る。

4.8.4 配列読込命令

$\langle \text{下限式} \rangle :: = \langle \tilde{A} \tilde{E} \rangle$
 $\langle \text{上限式} \rangle :: = \langle \tilde{A} \tilde{E} \rangle$
 $\langle \text{限界リスト要素} \rangle :: = \langle \text{下限式} \rangle, \langle \text{上限式} \rangle$
 $\langle \text{限界リスト} \rangle :: = \langle \text{限界リスト要素} \rangle |$
 $\langle \text{限界リスト要素} \rangle, \langle \text{限界リスト} \rangle$

<配列読込リスト> ::= <配列名>, <限界リスト>

<配列読込命令> ::= READARRAY (<配列読込リスト>)

従って20行30列の配列Aに対しA₃₂ A₃₃ A₃₄ A₄₂ A₄₃
A₄₄ とある2行3列の要素を対応する番号のところに読み込むには
READARRAと(A, 3, 4, 2, 4);

とする。3次元以上もたゞ限界リストを拡大すればよい。

4.8.5 パンチ命令

<パンチリスト要素> ::= <A E> | <B E>

<パンチリスト> ::= <パンチリスト要素> |

<パンチリスト>, <パンチリスト要素>

<パンチリスト部> ::= <空> |, <パンチリスト部>

<パンチ命令> ::= PUNCH (<フォーマットストリング>

<パンチリスト部>)

次の人力データを作るため高速テープパンチに情報を出すもので情報はすべてR, I, LおよびAタイプで出される。また情報の終りにはすべてカンマが自動的に入れられる。またAタイプで指定するとテープ上では、たとえば integer KにTOKYOというアルファタイプの情報が入っているとすれば

PUNCH ('5A', K);

よって =TOKYO, とパンチアウトされる。

4.8.6 PUT GET 命令

<名札部> ::= <UI>

<PGリスト要素> ::= <A E> | <B E> | <配列名>

<PGリスト> ::= <PGリスト要素> |

<PGリスト>, <PGリスト要素>

<PUT命令> ::= PUT (<名札部>, <PGリスト>)

<GET命令> ::= GET (<名札部>, <PGリスト>)

<同じプログラムの中で一時ある情報を外部メモリに出してその番地を使用した後で持ち帰って計算を続行するとき用いる。目印としては無符号整数を用い同じ番号により大きな配列をPUTすると自動的に前の部分が一杯になった後は他の場所に格納する。>

配列Aを名札部3から全部格納したいなら

```
PUT (3, A);
```

これと同じ次元の配列Bに入れて使用したいなら

```
GET (3, B);
```

である。

4.8.7 MT入出力命令

```
<ユニット> ::= <SV> | <UI>
```

```
<MTリスト要素> ::= <AE> | <BE> | <ST>
```

```
<MTリスト> ::= <MTリスト要素> |
```

```
<MTリスト>, <MTリスト要素>
```

```
<MTリスト部> ::= <空> |, <MTリスト>
```

```
<MT書込命令> ::= MTWRITE (<ユニット>,  
                           <フォーマットストリング> <MTリスト部>)
```

```
<MT読出命令> ::= MTRREAD (<ユニット>,  
                           <フォーマットストリング> <MTリスト部>)
```

情報をMTに出してとっておき別のプログラムで使用するときに用いる。

ユニットは土研の場合1か2を使用する。これを単純変数に入れておいて指定することもできる。MT書込の前には次の巻きもどし命令によって20単位以上消去した後TMを記入しておく必要がある。たとえばRタイプで配列Pを1からNまで書き込むには

```
REWIND (1, 20);
```

```
MTWRITE (1, '↑');
```

```
for I := 1 step 1 until N do
```

```
    MTWRITE (1, 'R', P[I]);
```

とする。

これを別のプログラムのAに代入して使用するには

```
REWIND (1);
```

```
TMSEARCH (1, 1);
```

```
for I := 1 step 1 until N do
```

```
    MTRREAD (1, 'R', A[I]);
```

とする。

4.8.8 TMサーチ命令

<TMサーチ命令> ::= TMS E A R O H (<unit>, <整数>)

MT上の書き込み場所はテープマークをもとにして求める。

↑ が出てくるたびに1つつつTMが書き込まれるので整数のところに5を指定すれば5番目のTMの次から読み始める。整数には負を指定してよい。このときは巻き送って指定された個数を数え最後に一つだけもどる。

4.8.9 巻きもどし命令

<巻きもどしリスト> ::= <U I>

<巻きもどしリスト部> ::= <空> | , <巻きもどしリスト>

<巻きもどし命令> ::= R E W I N D (<ユニット><巻きもどしリスト部>)

リスト部の数の単位だけ消去を行なう。リスト部が空の場合は巻きもどしのみ。

4.8.10 くり返し数代入命令

<くり返し数リスト> ::= <A E> | <くり返し数リスト>, <A E>

<くり返し数代入命令> ::= R E P X (<くり返し数リスト>)

たとえばN個のブランクの右にK個*を打ちたかったら

```
REP X (N, K);  
PRINT ('XBX(' * ') /');
```

とする。

4.8.11 アルファ代入命令

<アルファ代入命令> ::= E Q U I V (<ストリング>)

T O K Y O という字を integer P に代入するときは

```
P := E Q U I V ('T O K Y X');
```

である。スペース#を用い1語は7文字以下である。

5° 宣言 (declaration, D̃と略)

- <D̃> ::= <タイプD̃> | <配列D̃> | <スイッチD̃> | <手続D̃> |
 <イクスターナルD̃>

5.1 タイプ宣言

<タイプリスト> ::= <S Ṽ> | <S Ṽ>, <タイプリスト>

$\langle \text{タイプ} \rangle ::= \text{real} \mid \text{integer} \mid \text{Boolean}$

$\langle \text{タイプ} \tilde{D} \rangle ::= \langle \text{タイプ} \rangle \langle \text{タイプリスト} \rangle$

5.2 配列宣言

● $\langle \text{下限} \rangle ::= \langle \text{整数} \rangle$

● $\langle \text{上限} \rangle ::= \langle \text{整数} \rangle$

$\langle \text{限界対} \rangle ::= \langle \text{下限} \rangle : \langle \text{上限} \rangle$

$\langle \text{限界対リスト} \rangle ::= \langle \text{限界対} \rangle \mid \langle \text{限界対リスト} \rangle, \langle \text{限界対} \rangle$

$\langle \text{配列セグメント} \rangle ::= \langle \text{配列名} \rangle [\langle \text{限界対リスト} \rangle]$

$\langle \text{配列名} \rangle, \langle \text{配列セグメント} \rangle$

$\langle \text{配列リスト} \rangle ::= \langle \text{配列セグメント} \rangle \mid$

$\langle \text{配列リスト} \rangle, \langle \text{配列セグメント} \rangle$

$\langle \text{配列} \tilde{D} \rangle ::= \text{array} \langle \text{配列リスト} \rangle \mid \langle \text{タイプ array} \langle \text{配列リスト} \rangle$

実数配列は array と書いても real array と書いてもよい。例を示すと

array A, B [1 : 10], PQ [1 : 5, -8 : 4] ;

Boolean array O [0 : 4] ;

下限 ≤ 上限 でなければならない。

ALGOL-60 では下限および上限に算術式を認めている。

5.3 スイッチ宣言

$\langle \text{スイッチリスト} \rangle ::= \langle \tilde{DE} \rangle \mid \langle \text{スイッチリスト} \rangle, \langle \tilde{DE} \rangle$

$\langle \text{スイッチ} \tilde{D} \rangle ::= \text{switch} \langle \text{スイッチ名} \rangle : \langle \text{スイッチリスト} \rangle$

スイッチは一種の名札の配列である。例を示すと

switch Q : L1, L2, if A=B then D else LD, RS,
S [I] ;

この場合 L1, L2, D, LD, RS は名札 S [I] はそれ以前に宣言されたスイッチリストの要素である。

5.4 手続宣言

$\langle \tilde{FP} \rangle ::= \langle \tilde{I} \rangle$

$\langle \tilde{FP} \text{リスト} \rangle ::= \langle \tilde{FP} \rangle \mid \langle \tilde{FP} \text{リスト} \rangle, \langle \tilde{FP} \rangle$

$\langle \tilde{FP} \text{部} \rangle ::= \langle \text{空} \rangle \mid (\langle \tilde{FP} \text{リスト} \rangle)$

$\langle \tilde{I} \text{リスト} \rangle ::= \langle \tilde{I} \rangle \mid \langle \tilde{I} \text{リスト} \rangle, \langle \tilde{I} \rangle$

$\langle \text{VALUE部} \rangle ::= \text{value } \langle \text{Iリスト} \rangle ; | \langle \text{空} \rangle$
 $\langle \text{明細指定子} \rangle ::= \text{string} | \langle \text{タイプ} \rangle | \text{array} |$
 $\quad \langle \text{タイプ} \rangle \text{ array} | \text{switch} | \text{procedure} |$
 $\quad \langle \text{タイプ} \rangle \text{ procedure}$
 $\langle \text{明細指定部} \rangle ::= \langle \text{空} \rangle | \langle \text{明細指定子} \rangle \langle \text{Iリスト} \rangle ;$
 $\langle \text{手続頭部} \rangle ::= \langle \text{手続名} \rangle \langle \text{FP部} \rangle ; \langle \text{VALUE部} \rangle$
 $\quad \langle \text{明細指定部} \rangle$
● $\langle \text{八進数} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7$
● $\langle \text{マシン語} \rangle ::= \langle \text{八進数} \rangle \langle \text{八進数} \rangle \langle \text{八進数} \rangle \langle \text{八進数} \rangle$
 $\quad \langle \text{八進数} \rangle \langle \text{八進数} \rangle \langle \text{八進数} \rangle \langle \text{八進数} \rangle$
 $\quad \langle \text{八進数} \rangle \langle \text{八進数} \rangle \langle \text{八進数} \rangle \langle \text{八進数} \rangle$
 $\quad \langle \text{八進数} \rangle \langle \text{八進数} \rangle$
● $\langle \text{マシンプログラム} \rangle ::= \langle \text{マシン語} \rangle |$
 $\quad \langle \text{マシン語} \rangle \langle \text{マシンプログラム} \rangle$
● $\langle \text{手続本体} \rangle ::= \langle \text{命令} \rangle | \text{code } \langle \text{マシンプログラム} \rangle |$
 $\quad \langle \text{ブロック} \rangle | \langle \text{複合命令} \rangle$
 $\langle \text{手続 } \bar{D} \rangle ::= \text{procedure } \langle \text{手続頭部} \rangle \langle \text{手続本体} \rangle |$
 $\quad \langle \text{タイプ} \rangle \text{ procedure } \langle \text{手続頭部} \rangle \langle \text{手続本体} \rangle$

手続頭部には comment を書けない。

FP リストにあげたものは、すべて明細指定を要する。value 指定されたものは値と呼ばれ、その他のものは名前と呼ばれる。FP, AP の対応するものの型、種類は合致しなければならないが value 指定されたときは型が変わってもよい。

array は value 指定できない。

● 5.5 イクスターナル宣言

$\langle \text{イクスターナル } \bar{D} \rangle ::= \text{external } \langle \text{Iリスト} \rangle |$
 $\quad \text{external } \langle \text{タイプ} \rangle \langle \text{Iリスト} \rangle$

次の attach 命令によって機械語となった手続が格納されているとき、そのジョブネームを頼りに引用することを宣言するもので関数指示子のときは external <タイプ> の宣言子を用いる。

● 5.6 特殊命令

<アタッチ命令> ::= attach <手続 D> ; end

<停止命令> ::= halt ; (T)

● 5.7 データ

<任意のストリング> ::= <下段を含む任意の字の列>

<データリスト要素> ::= <数値> | <論理値> | <任意のストリング>

<データリスト> ::= <データリスト要素> |

<U I> * <データリスト要素>

<データリスト>, <データリスト要素> |

<データリスト>, <空>

<データ> ::= <データリスト>, (T) | <データリスト>; (T) |

<データリスト> / (T)

データテープは読込命令に合わせて数値、論理値又は=の後の文字数字を読み込むために作成する。

データはすべて、で区切られ、となったときは0が読み込まれる。データテープの読込停止は/(T) による。

アルファ読込みで7文字以上のときは最後の7文字が、7文字以上のときは以下ブランクが読み込まれる。またアルファタイプ of の印刷で7A以上を指定したときは、最初に7文字をこえる分だけブランクが入りその後指定されたものが頭から7文字出てくる。

3.5という数値を6回続けて読み込むときは、6 * 3.5としてよい。

[1] attach されているプロデュアについて

標 題	REN RITSU (A, O, X, N, CASE)	
概 説	連立方程式(一次)を解く。 パラメタ A: データ用 real array O: 逆マトリクスの array X: 解を格納する array N: 連立方程式の元数 CASE: 演算指示専用カウンター (注) 詳細は電算速報 2号参照	
標 題	PRI RA 2 (M, N, A)	
概 説	二次元の実数タイプの印刷を行なう。 パラメタ M: 行数 N: 列数 A: 印刷を行なう array の名前。	但し、有効4ケタとし1行に組を入れ自動的に改行し、1行おわれば2行おくる。また表頭、表側を(n)という形で印刷する。
標 題	PRI IA 2 (M, N, A)	
概 説	二次元の整数タイプの印刷を行なう。 パラメタ M: 列数 N: 行数 A: 印刷を行なう array の名前。	但し、有効4ケタとし1行に組を入れ自動的に改行し、1行おわれば2行おくる。また表頭、表側を(n)という形で印刷する。
標 題	CRLF (N)	
概 説	N回復帰改行を行なう。 パラメタ N: 復帰改行を行なわせる数。 Nが0の場合を認める。(改行なしの印字を行ない、次の情報は同じ行の先頭から入る。)	
標 題	SPACE (N)	
概 説	N個スペースをとる パラメタ	

標 題	SPACE(N)
概 説	N: スペースをとらせる数。 Nが0の場合を認める。0なら何もしない。
標 題	PRINTI(N, AE)
概 説	整数タイプの印刷を行なう。 パラメタ N: 印刷を行なわせる個数。 AE: 印刷を行なわせるものの名前。 $\boxed{\text{I}} \pm \underbrace{\text{X} \cdots \text{X}}_{\text{Nケタ}} \boxed{\text{I}} \text{ (N+3ケタ)}$
標 題	PRINTR(N, AE)
概 説	実数タイプの印刷を行なう。 パラメタ N: 有効ケタ数 AE: 印刷したい算術式 $\boxed{\text{R}} \pm \underbrace{\text{X} \cdots \text{X}}_{\text{Nケタ}} \cdots \text{X}_{10} \pm \text{X} \cdots \boxed{\text{R}} \text{ (N+8ケタ)}$
標 題	PRINTF(S, AE)
概 説	実数を小数点以上 n_1 ケタ以下 n_2 ケタで印刷 $\boxed{\text{F}} \pm \underbrace{\text{X} \cdots \text{X}}_{n_1} \cdot \underbrace{\text{X} \cdots \text{X}}_{n_2} \boxed{\text{F}}$ S: $n_1 \cdot n_2$ の形で書く AE: 印刷したい算術式
標 題	RA2PUT(NAME, M, N, P)
概 説	二次元・実数タイプの配列にPの値を代入する手続 パラメタ NAME: 書き込む array の名前 M: 行数 N: 列数 P: 代入したい数値 } value 指定してある。 0を代入するときなどに利用できる。
標 題	RA1PUT(MAME, N, P)
概 説	一次元・実数タイプの配列にPの値を代入する手続 パラメタ

標 題	RA1PUT (NAME, N, P)
概 説	NAME: 書き込む array の名前 N: 個数 P: 代入したい数値 } value 指定してある。
標 題	IA2PUT (NAME, M, N, P)
概 説	二次元・整数タイプの配列に P の値を代入する手続 パラメータ NAME: 書き込む array の名前 M: 行数 N: 列数 P: 代入したい数値 } value 指定してある。
標 題	ACOS (X)
概 説	$\cos^{-1} x$ の値を求める関数指示子 パラメータ X: 変数 (注) 電算速報第3号を参照
標 題	PLOT (X, Y, RITU, PO, S, O)
概 説	データをグラフ形式にプロットする手続 パラメータ X: X軸方向の値 Y: Y軸方向の値 RITU: Y軸の縮尺の倍率 PO: Y軸の原点の位置を指定する S: 手続を呼ぶ時の数値で最初は1, 2回目以降は2 O: リザーブパラメータ (注) 詳細は電算速報第3号参照
標 題	JACOBI (A, S, EVE, N, PHO)
概 説	n 次の実対称行列Aの固有値, および固有ベクトルを求める手続 パラメータ A: 行列の名前

標 題	JACOBI (A, S, EVE, N, PHO)
概 説	S : 行列の要素を格納する。 EVE : 固有値が格納される。 N : 行列の次数 PHO : 収束判定値 (注) 詳細は電算速報 3 号参照
標 題	HOOS (X)
概 説	coshx を求める関数指示子 パラメータ X : 変数 (注) 詳細は電算速報 3 号参照
標 題	HSIN (X)
概 説	sin hx を求める関数指示子 パラメータ X : 変数 (注) 詳細は電算速報 3 号参照
標 題	IAPUT (NAME, N, P)
概 説	一次元整数タイプの配列に P の値を代入する手続 パラメータ NAME : 書き込む array の名前 N : 偶数 P : 代入したい数値 } value 指定してある

現地観測データの分類, 集計, 分散分析のプログラム

** 村 野 義 郎

表-1のように, Q 個の特性値からなる現地観測データ (DATA) が P 組ある。そのうちから任意の $L1$ 番目と $L2$ 番目の特性値の階級にしたがって, $L3$ 番目と $L4$ 番目にある測定値 $DATA(L3)$ と $DATA(L4)$ を2元に分類集計し, かつ要因 $DATA(L1)$ に対して $DATA(L3)/DATA(L4)$ の級間変動と級内変動の比 F を計算する。

表-1 測定データ

特性番号 組番号	1	2		Q
1				
2				
...				
P	DATA(1)	DATA(2)		DATA(Q)

(1) 分類

$AS(1)$: P 個ある $DATA(L1)$ のうちの最小値

$AS(2)$: $DATA(L1)$ の階級幅

$AS(3)$: P 個ある $DATA(L2)$ のうちの最小値

$AS(4)$: $DATA(L2)$ の階級幅

とし, ある行の $DATA(L3)$ および $DATA(L4)$ の属する階級 (I, J) を, 同じ行にある $DATA(L1)$ と $DATA(L2)$ の階級にしたがって決定する。ただし, I は20以下, J は16以下とする。

そして

$N(I, J)$: ある階級 (I, J) に属するデータの総数.

$Z1(I, J)$: " $DATA(L3)$ の合計.

$Z2(I, J)$: " $DATA(L4)$ の合計.

** 砂防研究室

$Z_{12}(I, J) = Z_1(I, J) / Z_2(I, J)$
を計算し、表-2に示す分割表を作成する。

表-2 分割表

$\begin{array}{c} J \\ \backslash \\ I \end{array}$	1	2		MJ
1				
2				
⋮				
MI			$N(I, J)$ $Z_1(I, J) = \sum DATA(L3)$ $Z_2(I, J) = \sum DATA(L4)$ $Z_{12}(I, J) = Z_1(I, J) / Z_2(I, J)$	

もしDATA(L4)が不要なときにはL4を1000に指定する。そうすれば
 $Z_2(I, J) = N(I, J)$
 になり、したがって

$$Z_{12}(I, J) = Z_1(I, J) / N(I, J)$$

で、平均値が求められることになっている。

特性の数Qが多ければ、4個の特性の組合せ数も非常に多くなるから、組合せの識別記号(AN)を分割表および分散分析結果の頭部にプリントしておく必要がある。このプログラムでは4種類の記号、たとえば、

AN(1): 組合せ番号

AN(2): DATA(L1)とDATA(L2)の特性名、

AN(3): DATA(L3)とDATA(L4)の特性名、

AN(4): 他にも同種の現地測定データがある場合の、データ識別記号を用いることになっている。

(2) 分散分析

表-2の分割表において、 $TN(I)$ を行 I に属する標本数とし、

$$TX(I) = \sum_j Z_{12}(I, J)$$

$$TXX(I) = \sum_j (Z_{12}(I, J))^2$$

$$SN = \sum_j TN(I)$$

$$SX = \sum_j TX(I)$$

$$SXX = \sum_I TXX(I)$$

$$TM(I) = TX(I) / TN(I)$$

を計算する。そして水準数を M ，級間変動を BV ，級内変動を WV ，分散比を F として、

$$BV = \sum_I \frac{TX(I)^2}{TN(I)} - \frac{SX^2}{SN}$$

$$WV = SXX - \sum_I \frac{TX(I)^2}{TN(I)}$$

$$F = \frac{BV}{M-1} / \frac{WV}{SN-M}$$

の計算を行なう。この分散分析は各水準間の標本数が異なる場合に用いる平均値均一性の検定である。

(3) 分類方法の指示

パラメーターのうち特性の組合せを指示する L_1, L_2, L_3, L_4, AN ， AS の値は actual parameter ならびにプログラム本体の中で代入してもよいが、多数の組合せについて以上の分類、計算を行いたい場合には、組合せを指示するテープ (instruction tape) を作成し、データーテープより先に読み込ませる。そうすればデーターテープおよびプログラムテープには手を加えずに、任意の組合せについて計算を行なうことができる。

```

procedure ASSTEST (P, Q, L1, L2, L4, AN, AS, DATA, F);
    comment, Assortment of data according to two characteristics, and test for
    uniformity of mean value can be attained with this program.;

value P, Q, L1, L2, L3, L4;
integer P, L1, L2, L3, L4;
array DATA, AS;
integer array AN;
real F;

begin
    integer I, J, MI, MJ;
    array Z1, Z2, Z12 [1:20, 1:16], XC [1:20];
    integer array N [1:20, 1:16];
    begin comment, This block is for assortment.;
        integer K, A, B;
        for I:=1 step 1 until 20 do
            for J:=1 step 1 until 16 do
                Z1 [I, J] := Z2 [I, J] := N [I, J] := -1;
        A:=B:=0;
        for K:=1 step 1 until P do
            begin
                READ ARRAY (DATA, I, Q);
                I:=FIX4((DATA [L1] - AS [1])/AS [2]+1);
                J:=FIX4((DATA [L2] - AS [3])/AS [4]+1);
                N [I, J] := N [I, J] +1;
                Z1 [I, J] := Z1 [I, J] +DATA [L3];
                if L4=1000 then Z2 [I, J] := N [I, J] else
                    Z2 [I, J] := Z2 [I, J] +DATA [L4];
            end
        end
    end
end

```

```
        if A<I then A:=;
        if B<J then B:=J;
    end;
    MI:=A;
    MJ:=B;
    for I:=1 step 1 until MI do
        for J:=1 step 1 until MJ do
            if N[I,J]=-1 then Z12[I,J]:=-1 else
                begin
                    Z1[I,J]:=Z1[I,J]+1;
                    Z2[I,J]:=Z2[I,J]+1;
                    N[I,J]:=N[I,J]+1;
                    if Z2[I,J]#0 then Z12[I,J]:=Z1[I,J]/Z2[I,J] else
                        Z12[I,J]:=0;
                end;
            end;
        end;
    end;
    begin comment This block is for assorted data printing.;
        integer U, V, K, L,
        array D[1:20, 1:20];
        PRINT ('10B7A, 6B7A, 3B7A, 5B7A, 2/', AN[1], AN[2], AN[3], AN[4];
        PRINT ('15B 'ASSORTMENT' 3/');
        PRINT ('5B'XC'3B'YC');
        for J:=1 step 1 until 8 do
            PRINT ('5B2ZD.4D', AS[3]+(J-1)*AS[4]);
        PRINT ('2/');
    I:=1;
```

```
Q2: XC[I]:=AS[1]+(I-1)*AS[2];
L:=U:=1;
if MJ≤8 then V:=MJ else V:=8;
Q1: for J:=U step 1 until V do
begin
  if L=1 then
begin
  D[I,J]:=N[I,J];
  K:=EQUIV('N');
end else
  if L=2 then
begin
  D[I,J]:=Z1[I,J];
  K:=EQUIV('Z1');
end else
  if L=3 then
begin
  D[I,J]:=Z2[I,J];
  K:=EQUIV('Z2');
end else
  if L=4 then
begin
  D[I,J]:=Z12[I,J];
  K:=EQUIV('Z12');
end;
end;
```

```
PRINT ('2ZD. 4D, 2B3A', XC[I], K);
for J:=U step 1 until V do
  if D[I, J]=-1 then PRINT ('2B7('-') 4B') else
    PRINT ('2B: 6D10-ZD', D[I, J]);
PRINT ('/');
L:=L+1;
if L=5^MJ=V then
begin
  I:=I+1;
  if I>MI then go to Q3 else
    PRINT ('3/');
  go to Q2;
end else
if L=5^MJ>V then
begin
  U:=9; V:=MJ;
  PRINT ('/');
  L:=1;
end;
go to Q1;
Q3: PRINT ('5/');
end;
begin comment, This block is for sum up and test for uniformity of means.;
  real SX, SXX, SMXX, BV, WV;
  integer SN, M;
  array TM, TX, TXX(1:20);
  integer array TN(1:20);
```

```

SN:=M:=SX:=SXX:=SMXX:=0;
for I:=1 step 1 until MI do
begin
  TN(I):=TX(I):=TXX(I):=0;
  for J:=1 step 1 until MI do
    if Z12(I,J) # -1 then
    begin
      TX(I):=TX(I)+Z12(I,J);
      TXX(I):=TXX(I)+Z12(I,J)*2;
      TN(I):=TN(I)+1;
    end;
    SX:=SX+TX(I);
    SXX:=SXX+TXX(I);
    SN:=SN+TN(I);
    if TN(I) # 0 then M:=M+1;
  end;
for I:=1 step 1 until MI do
  if TN(I) = 0 then TM(I):=0 else
  begin
    TM(I):=TX(I)/TN(I);
    SMXX:=SMXX+TX(I)*2/TN(I);
  end;
BV:=(SMXX-SX)/2/SN;
WV:=(SXX-SMXX);
F:=(BV/(M-1))/(WV/(SN-M));
end;

```

begin comment, This block is for printing of sum and test.;

PRINT ('10B7A, 6B7A, 3B7A, 5B7A, 2/', AN{1}, AN{2}, AN{3}, AN{4});

PRINT ('15B'SUM'3/');

PRINT ('25B'1'9B'XC'15B'TX'14B'TN'15B'TM'2/');

for l:=1 step 1 until MI do

PRINT ('10BI,3B2ZD. 4D, R, I, R, 2/', I, XC{I}, TX{I}, TN{I}, TM
{I});

PRINT ('2/B'TOTAL' 11BR, I, R, 5/', SX, SN, (SX/SN));

PRINT ('15B'TEST'3/');

PRINT ('5B'BV='R, 3B'M-1='ZD, 5B'WV='R, 3B'N-M= ZD, 5B'F='R, /'
, BV, (M-1), WV, (SN-M), F);

end;

end;

§ 3. 電算室よりのお知らせ

(1) 計算機使用状況およびC. D. C. 契約について

昭和40年3月3日より駒込において運転を開始した、OKITAC 5090-Hは順調な使用実績を示して、40年7月頃より夜間運転の要請が増加したため、長時間演算のものはOKITAC 5090-Hによるチェック(文法及び簡単な答のチェック)を終了した紙テープそのままをより大型の計算機に移す計画を実行し、既にCDC-3600に対する処置を完了した。一方、長時間演算による文法チェック及び短時間計算の待機時間が増加したので、41年1月より文法制限が緩い上に、二重割込を許すNEW ALGOL-Hへの移行を行なった。

また、大型計算による記憶容量の不足に対して41年度中に、今の12K長語を16K長語に変更する予定である。

次に40年度計算機使用状況をまとめた、月別使用時間を表-1に、研究室別使用時間を表-2に、日数・件数・時間・日平均時間の関係を図-1に、河川系研究室、道路系研究室、その他との関係を図-2より図-6までに、処理時間別件数を図-7より図-9までに、個人別件数を図-10に示した。

以上が土木研究所における電子計算機の使用状況であるが、前に述べたODC-3600型電子計算機を設置している日本原子力産業会議に対し、表-4に示す単価(計算時間はOKITAC5090-Hの約1/40程度)により計算を依頼出来る。

これに対する演算申込みは計算請求書(図-11)、計算仕様書(図-12)のみで足りる様に契約済みである。

表-1

稼働総数表及び月別利用率

月	日数	件数	月総時間	1日平均時間	計算時間	通電時間	保守時間	ダウン時間	R1 (%)	R2 (%)
					A (分)	B (分)	C (分)	D (分)	(A+C)/B	(A+C+D)/B
4	15	73	2107	140						
5	22	73	2585	118						
6	21	110	3448	164						
7	18	176	11891	661						
8	22	265	8122	369						
9	21	208	9304	443	9304	11499	1002	30	82.63	82.89
10	23	235	14102	613	14102	18936	972	170	72.60	80.50
11	20	266	10328	516	10328	14457	972	340	78.16	80.51
12	24	226	15247	635	15247	20934	829	130	76.79	77.41
1	17	161	8152	480	8152	11897	804	540	75.28	79.82
2	20	211	12717	636	12717	16488	853	210	82.50	83.58
3	21	173	10103	481	10103	15454	857	30	70.92	71.11
合計	244	2177	108106	5256	79953	109665	6289	1450	78.64	79.96
月平均	20.3	181.4	9008.8	438	11142.19	15666.4	898.4	207.1	78.64	79.96

表-2

研究室別使用時間及び件数

	研 究 室 名	使 用 時 間	件 数
1	技 術 管 理 室	28,677	750
2	道 路 研 究 室	21,665	304
3	ダム構造 "	9,439	201
4	水 文 "	7,052	88
5	トンネル "	6,140	68
6	河 川 "	5,726	132
7	橋 梁 調 査 室	5,362	110
8	橋 梁 研 究 室	5,099	53
9	海 岸 "	3,761	67
10	ダム水理 "	3,700	127
11	研 修	3,454	17
12	砂 防 研 究 室	1,380	47
13	関 東 地 方 建 設 局	1,331	9
14	鹿 島 水 理 試 験 所	1,283	22
15	中 国 地 方 建 設 局	855	17
16	土 質 研 究 室	639	25
17	近 畿 地 方 建 設 局	465	6
18	本 省	414	20
19	下 水 道 研 究 室	196	13
20	鋪 装 "	189	7
21	機 械 "	140	2
22	構 造 "	131	7
23	基 礎 調 査 室	127	6
24	施 工 研 究 室	114	7
25	地すべり "	42	4
	その他	721	12
	合 計	108,102	2,121

図-1 稼動総数図

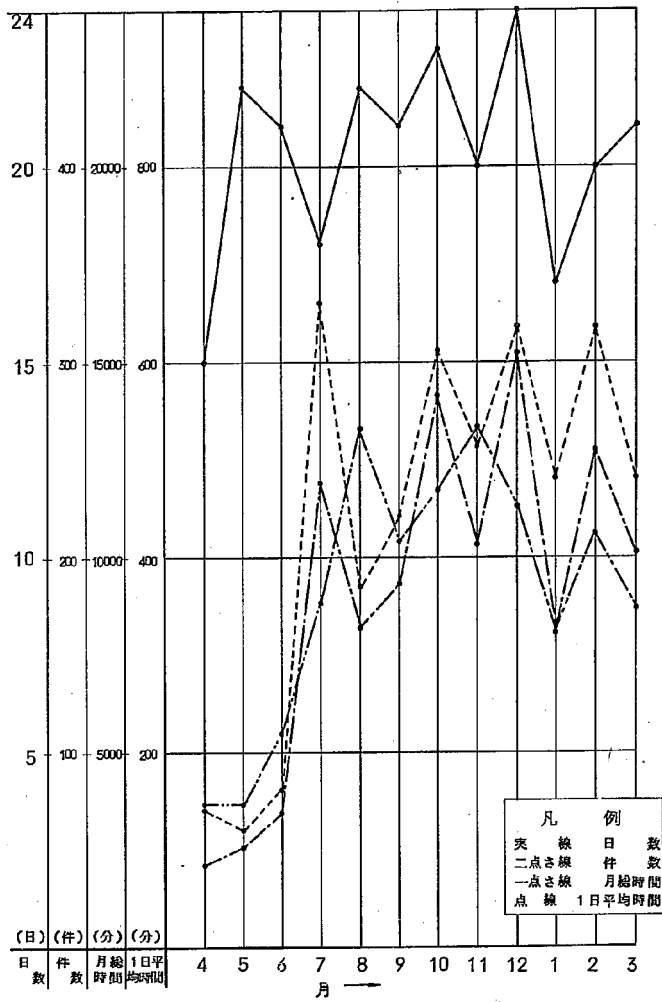


図-2 使用時間図

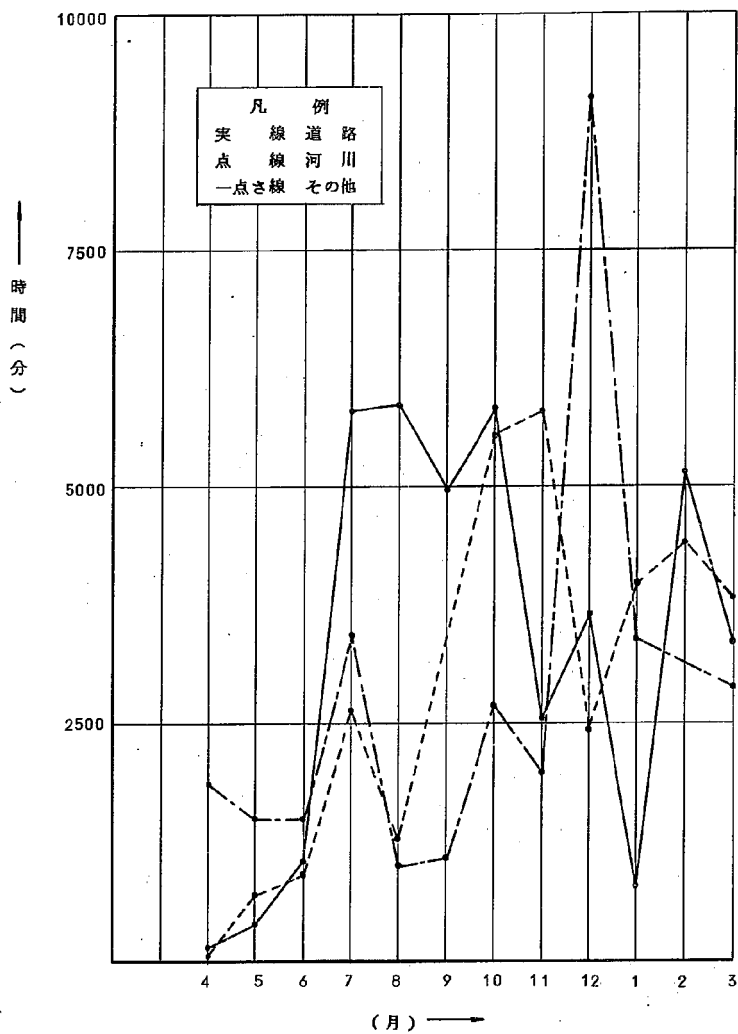


図-3 全使用時間に対する百分率図

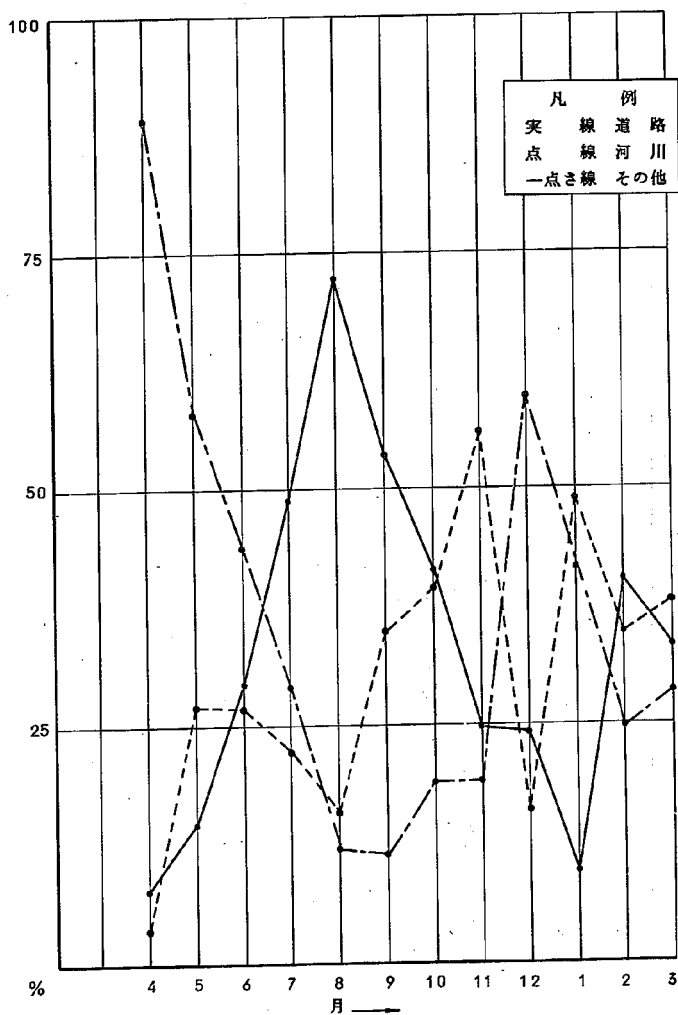


図-4 200時間に対する百分率図

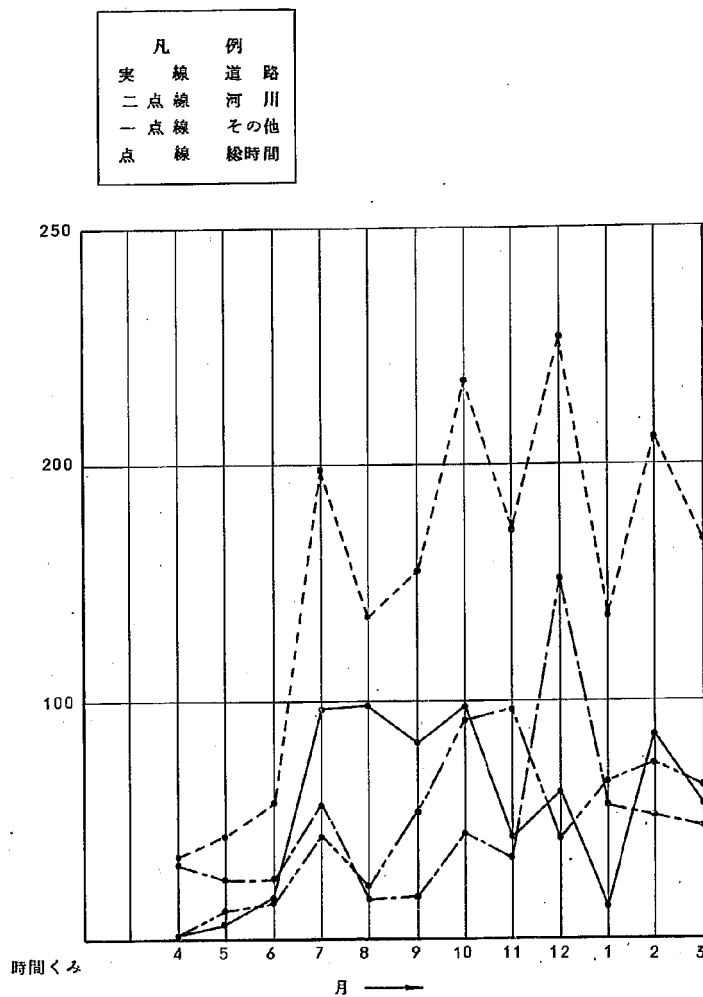


図-5 使用件数図

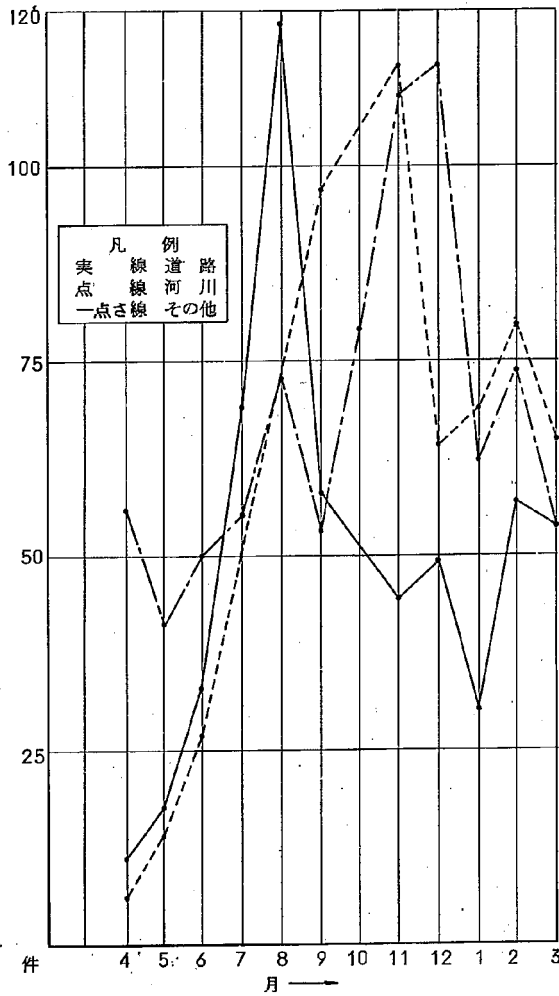


図-6 総件数に対する百万率図

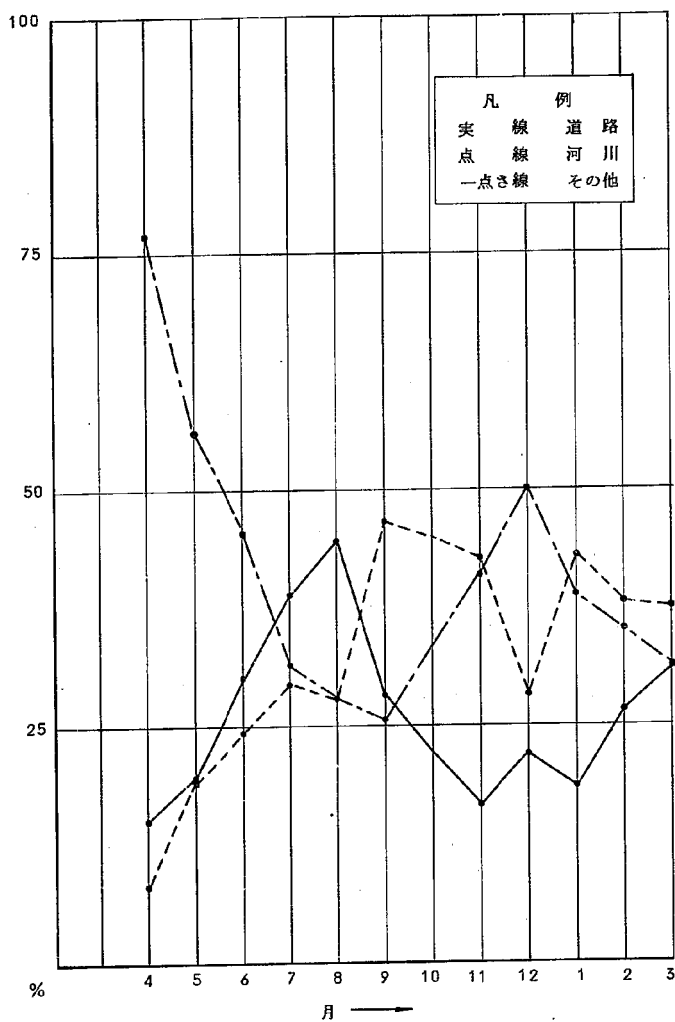


図-7

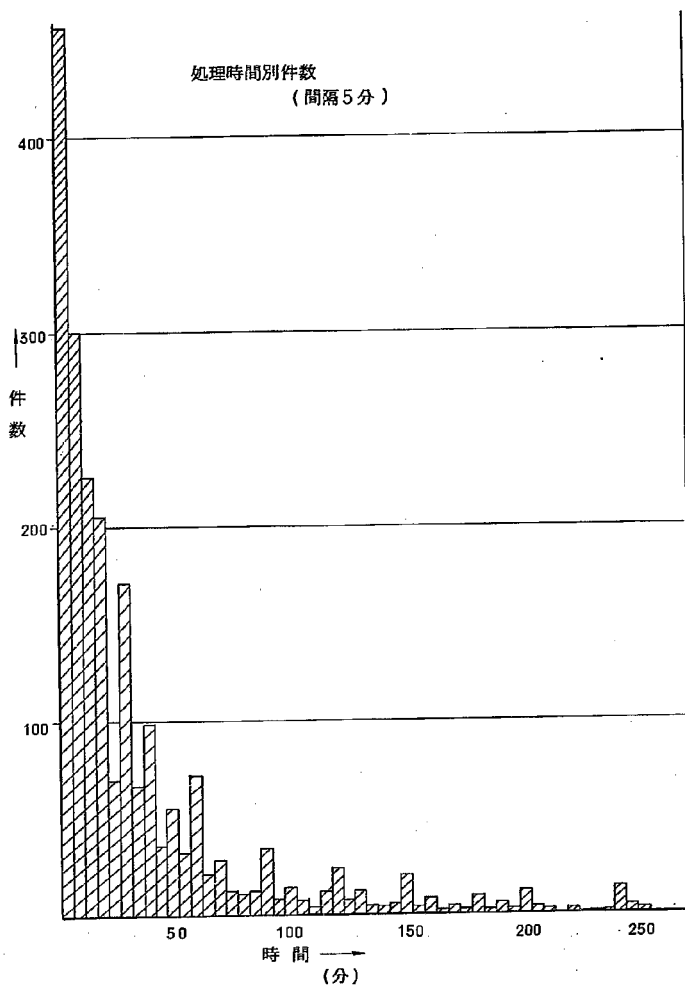


図-8

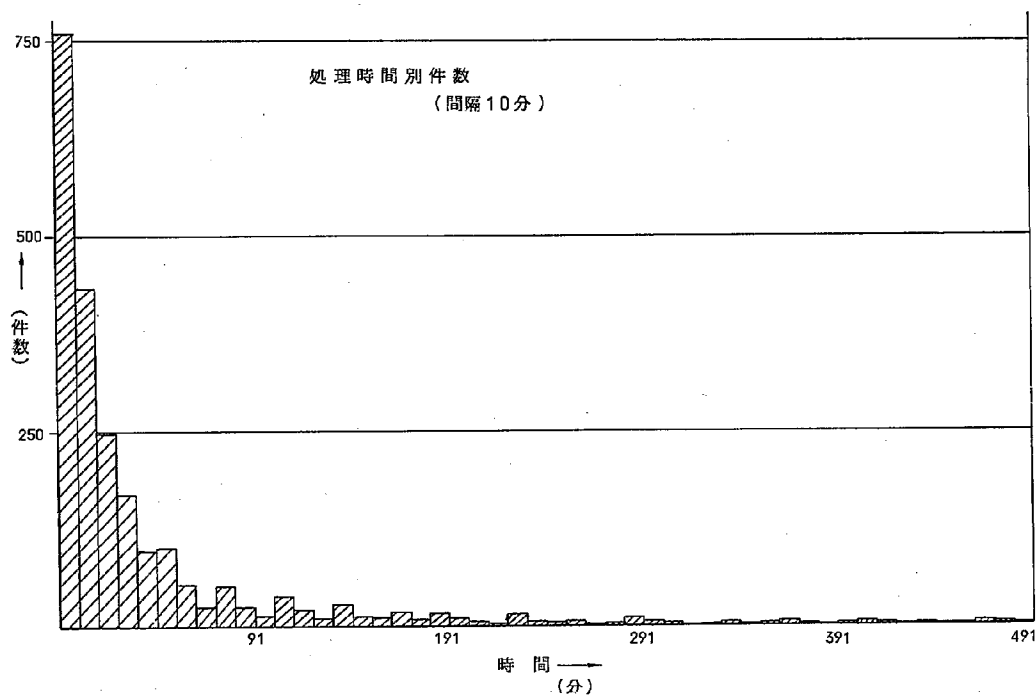


図-9

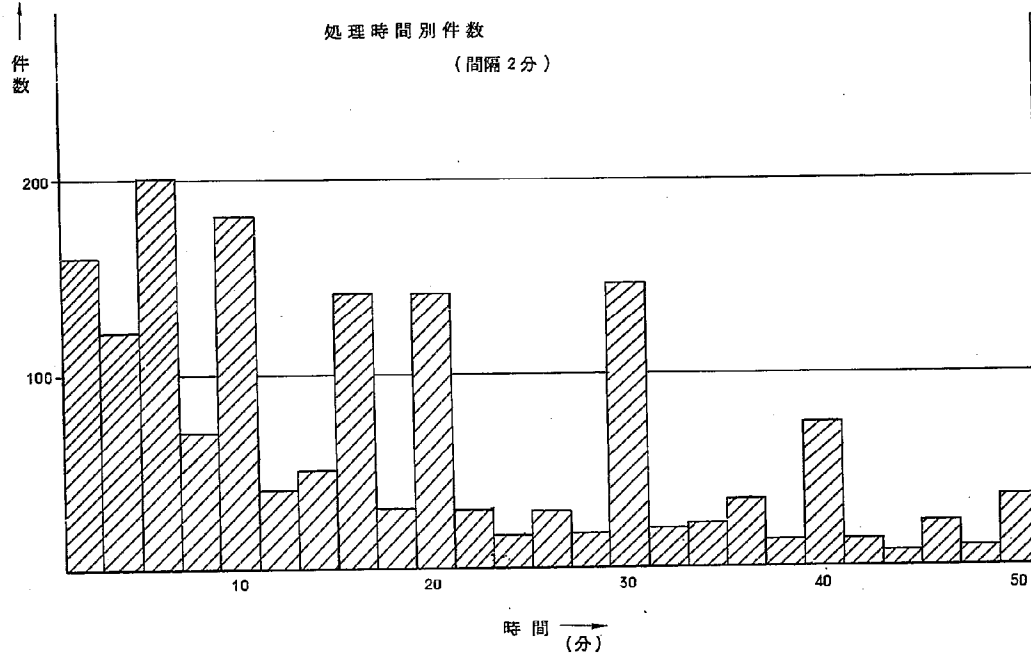
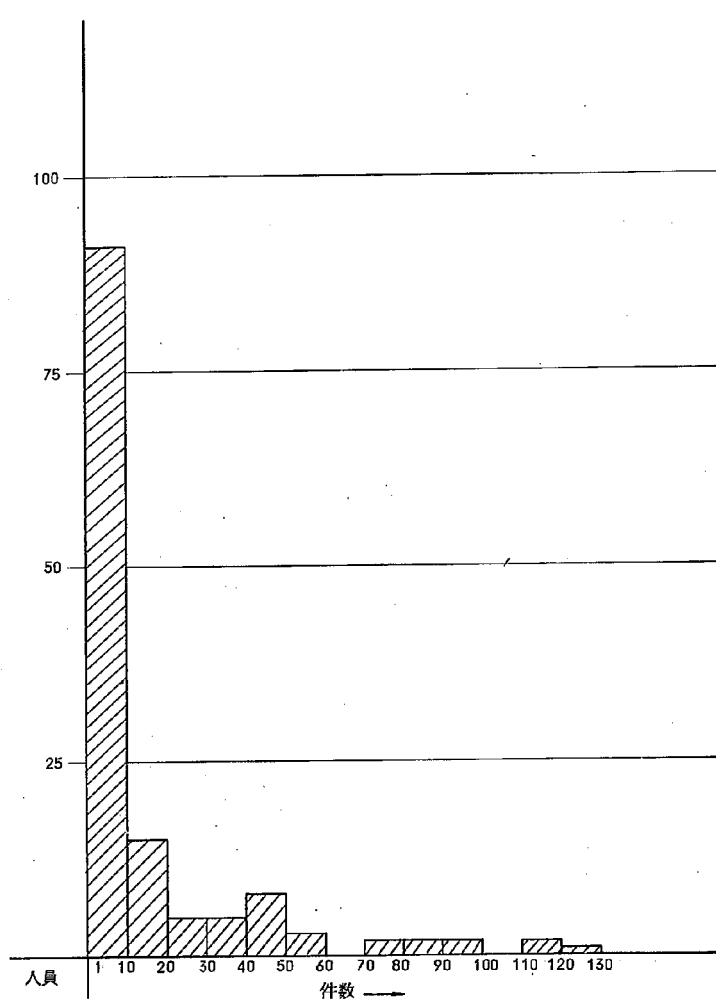


図-10 個人別件数



別 表

機 械 の 種 類			単価(1/100) 時間
CDC 3600電子計算組織	中央処理機(98,304語)	1	3,000円
	磁気テープ装置	14	オペレーシ ョン料を含む
	ディスクバック装置	1	
	高速度ラインプリンタ装置	3	
	仮名文字プリンタ装置	1	
	紙テープ読取, せん孔装置	1	
	カード 読取機	2	
	カードせん孔機	2	
	X-Y グラフプロッタ	1	
	3200衛星計算機(入出力専用)	1	
プログラム 作成料金	1人1時間当り		1,500円
カード読取・さん孔料金	カード10枚当り		4円
カ ー ド 代 金	カード1枚当り		1円
(標準) 印 刷 料 金	16インチ巾2パートの場合		20円
	” 3パート ”		30円
	” 4パート ”		35円
カードさん孔及びさん孔 検査	カード1枚当り(カード代を含む)		15円
紙テープ読取・さん孔料 金	1,000字当り		8円
紙テープ代金(1巻)	220メートル当り		320円

☒ — 1 0

所 長	総務部長	会計課長 企画室長	会 計 課長補佐	用 度 係 機械管理係	所属室課名	部 長	室 課 長	担 当 者
月 日				月 日		月 日	月 日	月 日
計 算 請 求 書								
土木研究所費		事業費(地 建 名)		委 託 費		分 類		
研 究 名								
題 目		規 格	数 量	称 呼	細分類	概 算 額	摘 要	
(例) ダムの挙動解析								
使 用 目 的								
取得希望時期		月 日	納入場所		土木研究所			
備 考								
供 用 請 求 書								
上記計算請求します					物品供用官		④	
物品管理官					殿			

図-11

計 算 仕 様 書

1. 名 称

2. 納 入 場 所

3. 納 期

4. 概 要

5. 仕 様

5-1 予 想 時 間

5-2 打 切 時 間

5-3 予 想 印 刷 枚 数

5-4 テー プ 読 込 順 位

5-5 パンチの有無

5-6 そ の 他

5-7 この仕様書に明示されていない事項については、単価契約時の仕様書による。

〔2〕プログラマー追加名簿（その3）

プログラマー登録番号総括表

所 属	記号	先号掲載人員	追加登録者	登録者人員
土 木 研 究 所 本 所	A	3 1	6	3 7
千 葉 支 所	B	7 9	1 2	9 1
赤 羽 分 室	C	5 6	9	6 5
篠 崎 分 室	D	1 2		1 2
鹿 島 水 理 試 験 所	E	2		2
新 潟 試 験 所	F	2	1	3
小 計		1 8 2	2 8	2 1 0
本 省 関 係	G	7 9	2 1	1 0 0
地 建 関 係		1 8 2	4 4	2 2 6
東 北 地 建	K	1 2	4	1 6
北 陸 "	L	3 2	4	3 6
関 東 "	M	4 0	1 1	5 1
中 部 "	N	2 9	6	3 5
近 畿 "	P	2 1	5	2 6
中 国 "	Q	3 2	5	3 7
四 国 "	R	7	4	1 1
九 州 "	S	9	5	1 4
都 府 県 関 係	T	2 8	5 6	8 4
公 団 ・ 公 社	U	1 5	5	2 0
そ の 他	V	1 1	6	1 7
小 計		5 4	6 7	1 2 1
合 計		4 9 7	1 6 0	6 5 7

プログラマー名簿（追加）その3

（S41.1～4月研修）

登録番号	氏 名	所	属
A017	樽 沼 昌 明	コンクリート研究室	土 木 研 究 所
A032	高 倉 敏 雄	技術管理室	"
A033	松 本 重 憲	"	"
A034	佐々木 隆 士	コンクリート研究室	"
A035	灘 波 憲 一	化学研究室	"
A036	松 原 美 和 子	企 画 室	"
B046	川 島 博 子	橋梁研究室	"
B081	渋谷 光 男	道路 "	"
B082	森 山 英 一	橋梁 "	"
B083	赤 堀 健 一	機械 "	"
B084	神 保 信 一	施工 "	"
B085	吉 岡 晴 之	" "	"
B086	宮 越 博 文	" "	"
B087	二 馬 健二郎	" "	"
B088	齊 藤 栄一郎	機械 "	"
B089	亀 甲 勝 信	基礎 "	"
B090	佐々木 康	土質 "	"
B091	柴 田 正 雄	道路 "	"
G057	青 木 典 夫	河川 "	"
G058	小 林 貞 夫	海岸 "	"
G059	米 道 久	ダム構造 "	"
G060	本 木 博	ダム水理	"
G061	仲 道 憲 一	河川 "	"
G062	菅 原 正 孝	下水道 "	"

登録番号	氏 名	所	属
C063	馬 場 洋 二	河川研究室	土 木 研 究 所
C064	田 畑 茂 清	砂防 "	"
C065			
F003	倉 繁 敏	新潟試験所	"
G080	森 悠		計 画 局
G081	林 雄 作		"
G082	中 野 和 義		都 市 局
G083	藤 条 邦 裕		"
G084	亀 田 泰 武		"
G085	小 浪 博 英		"
G086	溜 水 義 久		"
G087	奥 村 俊 雄		"
G088	北 山 武 征		"
G089	深 沢 日 出 男		河 川 局
G090	田 辺 善 彦		"
G091	服 部 嘉 徳		道 路 局
G092	藤 木 顕 興		住 宅 局
G093	松 山 雅 昭		"
G094	上 野 公 成		"
G095	大 沼 嘉 明		"
G096	島 崎 勉		"
G097	野 村 信 之		"
G098	石 川 二 郎		営 繕 局
G099	松 島 清 夫		"
G100	末 永 保 夫		"
K013	野 村 和 正	郡山国道工事々務所	東 北 地 建

登録番号	氏 名	所	属
K014	齊 藤 正 勝	釜房ダム工事々務所	東 北 地 建
K015	大 湾 嘉 昭	郡山国道工事々務所	"
K016	山 田 俊 郎	仙台技術事務所	"
L033	須 田 公 男	富山工事々務所	北 陸 地 建
L034	伊 藤 宏 美	旧信濃川工事々務所	"
L035	山 田 功	金沢工事々務所	"
L036	岡 田 勝 彦	神通川工事々務所	"
M041	三 浦 一 宇	京浜工事々務所	関 東 地 建
M042	高 橋 五十二	利根川管理事務所	"
M043	山 根 豊	横浜国道管理事務所	"
M044	佐 藤 武 司	首都国道工事々務所	"
M045	小 池 達 男	利根川下流工事々務所	"
M046	西 健 吾	首都国道工事々務所	"
M047	山 田 直 重	横浜国道工事々務所	"
M048	吉 田 三 郎	利根川水系砂防事務所	"
M049	三 木 繁 生	営繕部監督室	"
M050	吉 田 邦 雄	建築第一課	"
M051	山 田 勝 己	東京国道工事々務所	"
N030	杉 浦 英 雄	木曾川上流工事々務所	中 部 地 建
N031	鈴 木 敏 夫	名阪国道工事々務所	"
N032	今 村 宏 昭	営繕部建築課	"
N033	伊 糖 仁 史	名古屋技術事務所	"
N034	和里田 義 雄	企画室	"
N035	高 橋 尚 城	名古屋技術事務所	"
P022	恒 松 道 信	大滝ダム工事々務所	近 畿 地 理

登録番号	氏 名	所	属
P023	福 田 収	第二阪神国道工事々務所	近 畿 地 建
P024	中 岡 智 信		"
P025	宮 井 宏		"
P026	瀬 尾 克 美		"
Q033	岩 田 泰 之	出雲工事々務所	中 国 地 建
Q034	田 岡 荘 二	岡山国道工事々務所	"
Q035	下 村 周		"
Q036	馬 場 紘 一		"
Q037	川 端 徹 哉		"
R008	裏 戸 勉	松山工事々務所	四 国 地 建
R009	中 島 英 輔	土佐国道工事々務所	"
R010	松 浦 伝	企 画 室	"
R011	水 野 正 憲	"	"
S010	山 口 一 弘	緑川ダム調査事務所	九 州 地 建
S011	勇 直 允	宮崎工事々務所	"
S012	針 具 武 紀	企 画 室	"
S013	吉 川 勝 敏	道路計画課	"
S014	横 田 茂 行	河川計画課	"
T029	工 藤 智	道 路 課	青 森
T030	山 館 清 士	砂 防 課	"
T031	三 橋 至	道土木事務所	宮 城
T032	石 田 豊 明	迫川改良事務所	"
T033	吉 田 浩	原町土木事務所	福 島
T034	三 沢 邁 策	浦和土木事務所	埼 玉
T035	前 野 巖	日光土木事務所	栃 木

登録番号	氏 名	所	属
T036	三宅 勇喜	道路課	千葉
T037	滝沢 和男	道路工事課	東京
T038	早川 完治	埋立工事部	"
T039	山本 從徳	道路監理課	横浜市
T040	佐々木 謙次	河港課	神奈川
T041	木村 稔	道路課	"
T042	千葉 寿郎	河川部	群馬
T043	北沢 文教	館山建設事務所	長野
T044	斉藤 健	河川部	山梨
T045	松倉 良信	道路課	富山
T046	佐賀 定義	芦原町役場	福井
T047	佐竹 計之	大垣市土木課	岐阜
T048	稲葉 治雄	道路課	静岡
T049	森下 則雄	河川課	"
T050	坂井 尚弘	道路建設課	愛知
T051	山下 洋	道路課	三重
T052	駒田 明克	雲出川開発事務所	"
T053	山本 厳	道路課	大阪
T054	小林 忠美	幹線課	"
T055	藤井 孝雄	池田土木出張所	"
T056	深津 懿敬	西大阪防潮	"
T057	野村 尚弘	高田土木出張所	奈良
T058	船戸 富	道路課	和歌山
T059	中村 肇	河川課	"
T060	若宮 誠	大井土木事務所	愛媛
T061	岩井 栄	河川課	茨木
T062	保田 清	河川開発	石川
T063	北柄 敬輔	久慈土木事務所	岩手
T064	野本 宏	長野建設事務所	長野

登録番号	氏 名	所	属
T065	岩 本 正 彦	計画課	神 戸 市
T066	高 杉 卓 伸	津山土木事務所	岡 山
T067	坂 本 義 一	河川課	広 島
T068	岡 山 弘 親	道路課	"
T069	高 橋 仁	"	滋 賀
T070	坂 田 滋	"	兵 庫
T071	井 上 俊 明	鳴門土木事務所	徳 島
T072	真 家 武 二	道路課	"
T073	山 下 靖 雄	坂出土木出張所	香 川
T074	山 田 専 一	観音寺土木出張所	"
T075	田 中 宏 明	河津課	鳥 取
T076	田 中 頼 安	河川課	兵 庫
T077	松 田 二 郎	河港課	岩 手
T078	高 橋 正 人	河川砂防	長 崎
T079	楠 木 正 孚	道路課	"
T080	運 尾 純 蔵	道路建設課	福 岡
T081	宮 崎 義 男	河川開発	"
T082	吉 良 文 至	道路課	大 分
T083	越 山 達 夫	"	宮 崎
T084	柿 元 郁 雄	大島土木課	鹿 児 島
U016	堀 敏	矢木沢ダム	水 資 源
U017	高 木 康 男	川越道路調査事務所	道路 公 団
U018	富 岡 好 雄	第一設計課	首都 公 団
U019	太 田 利 龍	"	"
U020	龍 野 浩	工事第二課	阪 神
V012	北 条 紘 次	旭川開発	開 発 局
V013	小 林 豊 明	札幌 "	"

登録番号	氏 名	所	属
V014	若 月 修 一	函 館 開 発	開 発 局
V015	金 森 恒 則	小 樽 開 発	"
V016	辻 靖 三		
V017	山 口 嘉 之		

〔8〕 操作指示書・時間外申込書

操作指示書の様式が変更になったのでお知らせします。
尚、時間外に計算を申込む場合は時間外申込書を添付して下さい。

操 作 指 示 書

担当者

受 付 年 月 日		受 付 番 号		受 付 者 氏 名	
室 課 名	登録番号	氏 名	計 算 時 間		
			打 切 時 間		
プログラム 種 類	H2・H1 (チェック要・不要)		予 想 時 間	分	
返 送 方 法	本人・連絡者・依頼者 ()				
題 目 計算 { 内 容					
注意 この欄より上の太線内は確実に記入して下さい					
P U E T	有 ・ 無	専 用 M T			
MT WRITE READ	有 (1 ・ 2) 無	穿 孔 依 頼		要 ・ 不要	
備 考 { 完了希望日 其 の 他					
計 算 日 時		計 算 番 号		事 故 の 状 況	
使 用 時 間		分	終 了 月 日		
L P 用 紙		枚	穿 孔 者		
			操 作 者		
依頼方法		返送日		返送方法	

- 注意 1. 計算申込者は、太線枠内のみ書入れて下さい。
2. 計算完了プログラムは、procedure にして技術管理室に提出して下さい。

時間外申込書

室長印

室 課 名	登録番号	氏 名		計 算 時 間	
					分
連 絡 先		予想印刷枚数	枚	予想時間	分
プログラム チェック	要 不要				
計 算 題 目			使 用 M T 石		
申 込 日					
希 望 日 時					
決 定 日 時					
計 算 担 当 者					
使 用 時 間	分	終 了 月 日			
L P 用 紙	枚	穿 孔 者			
		操 作 者			
備 考					
注 希望日時が、外と重なる場合は技術管理室に於いて日時を繰り返えて 通知致します。					