

TECHNICAL REPORT #1
MAY 1961

SOME ALGOL INFORMATION

MKC
9C7392t

by

Jesusa V. Kinderman

I. DECK STRUCTURE

```

JOB, T10, CM60000.00000000
ALGOL(L,X)                (ALGOL. compiles and lists diagnostics only)
NOREDUCE.                 (necessary for ALGOL programs)
LGO.
7
8
9
COMMENTS
'BEGIN'
      ALGOL program
'END'
      'EOP'                (use columns 10-14)
7
8
9
CHANNEL cards, if any
CHANNEL, END
DATA cards, if any
5
6
7
8
9

```

The channel cards define the characteristics of the channel through which input or output is performed. For example, the line printer which is the most common output channel will have parameters such as maximum width of a line, the maximum number of lines on a page, etc., specified on its channel cards. The form of these parameters on a channel card is given in the CDC 6000 Series Computer Systems ALGOL 60 Reference Manual (pub. No. 60306100)

The line printer and the most common input channel, the card reader, do not require any channel cards. They are automatically supplied by the ALGOL system.

II. Character Representation of ALGOL Symbols

ALGOL Symbol	48-Character Representation	ALGOL Symbol	48-Character Representation
A - Z	A - Z	<u>true</u>	'TRUE'
a - z	-	<u>false</u>	'FALSE'
0 - 9	0 - 9	<u>go to</u>	'GO TO'
+	+	<u>if</u>	'IF'
-	-	<u>then</u>	'THEN'
x	*	<u>else</u>	'ELSE'
/	/	<u>for</u>	'FOR'
††	POWER'	<u>do</u>	'DO'
÷	'/' or 'DIV'	<u>step</u>	'STEP'
>	'GREATER'	<u>until</u>	'UNTIL'
≥	'NOT LESS'	<u>while</u>	'WHILE'
=	= or 'EQUAL'	<u>comment</u>	'COMMENT'
≠	'NOT EQUAL'	<u>begin</u>	'BEGIN'
≤	'NOT GREATER'	<u>end</u>	'END'
<	'LESS'	<u>own</u>	'OWN'
∧	'AND'	<u>Boolean</u>	'BOOLEAN'
∨	'OR'	<u>integer</u>	'INTEGER'
≡	'EQUIV'	<u>real</u>	'REAL'
⌋	'NOT'	<u>array</u>	'ARRAY'
⊃	'IMPL'	<u>switch</u>	'SWITCH'
•	•	<u>procedure</u>	'PROCEDURE'
,	,	<u>string</u>	'STRING'
:	..	<u>label</u>	'LABEL'
;	.,	<u>value</u>	'VALUE'
10	'	<u>code††</u>	'CODE'
⌊	⌋	<u>eop††</u>	'EOP'
((
:=	..= or ..=		
))		
[(/		
]	/)		
'	'('		
,	'),'		

†In a format string, † is represented by an asterisk.

††Not defined in the ALGOL-60 Revised Report; code is defined in Sec.5.4.1, Chapter 2; eop in Chapter 4.

III. The Input-Output Statements

The most frequently used input-output statements in ALGOL are:

```
INPUT (Channel No., ('Format'),' ,List)
OUTPUT(Channel No., ('Format'),' ,List)
```

Channel No. is an integer assigned to identify an input-output channel. 60 is the card reader and 61 is the printer.

Format is a string of symbols describing the format with which we want the LIST of items to be read in or printed out. A maximum of 61 items is allowed in the LIST.

Examples are:

```
INPUT (60, ('3ZD,BBB,N'),' ,X,Y)
OUTPUT(61, ('* ,/// ,+DDD.5D'),' ,ANS)
```

Here, X is an integer variable while ANS and Y are real variables. The following description of symbols clarifies the meaning of these input-output statements.

FORMAT CODES

<u>Symbol</u>	<u>Description</u>
*	transfer to the next page.
/	transfer to the next card (or line). Note: Unless a / is used, ALGOL continues reading the same card (or writing on the same line) even with the next input (or output) statement.
B	a blank space; can use BBBB or 4B.
D	a digit; can use DD or 2D.
Z	a zero is suppressed, i.e., a zero digit is replaced by a blank when all digits to its left are zero. For example, a format of 4D will output the number ten as 0010 while 3Zd will output ten as 10.
10	exponent part indicator as in the format, D.5D102D. Note: The maximum number of D's + Z's appearing before the exponent part is 24; in the exponent part the maximum number of D's + Z's is 4.
.	decimal point
+	the sign will appear as + or - on the external medium.

- the sign will appear if minus and will be suppressed if plus. Note: If neither + or - appears in the format and the number is negative, an error occurs; if the number is positive, there is no error.

P To input or output Boolean quantities only. True is represented by 1 and false by 0.

F To input or output Boolean quantities only. T is for true and F is for false. Note: For input and output, the words 'true' or 'false' are not used. Also, the internal representations of these quantities are:

T: 4000 000000 0000000000B

B: 00000 00000 00000 00000B

S This string format is used to output titles or labels as in, OUTPUT(61,>('13S'),'('SIMPSONS RULE'))'. Note: The symbols '(' and ')' are used to enclose the string to be printed.

N standard format for both integer and real variables. On input, the N format accepts any number written according to the ALGOL syntax for a number. Numbers may be separated by

- a) a letter or character except ., +, -, digit, or 10
- b) k or more blank spaces. k=2 unless otherwise specified on the channel card.

For output of integer and real variables, the format N is +D. 13D 10+3 D and numbers are separated by k blank spaces.

Note:

- 1) OUTPUT(61,('5(N)'),'X,Y,Z,U,V) is used instead of just ('5N')'.
- 2) If OUTPUT(61,('D.5D'),'X,Y,Z,U,V) is used, no format is specified for Y,Z,U and V. In this case, these variables are printed using standard format. Similarly, in input, if no format is specified, N is assumed.

A alpha format as in

INPUT(60,('A'),'BAT).

The A format will input only the first character of BAT even if a maximum of 8 characters is allowed. Although BAT is composed of alphabetic characters, it has to be declared integer. This also applies to the output statement.

H a non-format code that inputs or outputs 8 consecutive display (BCD) characters from a single integer variable. INPUT(60,('H'),BAT) will input all 8 characters of BAT. Note the difference between the A and H format.

I
R
L
M

are non-format codes which input or output the 20 octal digits which constitute the internal representation of the variable. I is used for variables of type integer, R for real variables, L for Boolean variables while M accepts all types. Note: Internally, integer variables are represented in the same way as real variables, i.e., in the form of a coefficient and biased exponent.

J Skip the character pointer to the next tabulation position; similar to the tab operation on a typewriter. It is used with the procedure TABULATION which will be described in the next section.

IV. Inlist-Outlist Statements

The use of inlist-outlist statements is best illustrated by an example:

```
'BEGIN'
  'INTEGER' I.,
  'INTEGER' 'ARRAY' A(/ 1: 4/)..
  'PROCEDURE' LAYOUT.,
  'BEGIN' FORMAT('('4(-4ZD,3B)'))' 'END'..
  'PROCEDURE' LIST(ITEM)..
  'PROCEDURE' ITEM.,
  'BEGIN' 'FOR' I..= 1 'STEP' 1 'UNTIL' 4
          'DO' ITEM(A(/I/)) 'END'..
  IN LIST(60,LAYOUT,LIST)..
  :
  :
```

'END'

The first parameter of INLIST (or OUTLIST) is the channel number; the other two are procedures. LAYOUT sets down the format controlling the input or output of data while LIST specifies the list of quantities. ITEM(A(/I/)) means that A(/I/) is the next item of the list. LAYOUT, LIST, and ITEM are arbitrary names given to the procedures. Any other name will do as well.

As it is, using the INLIST statement has no advantage over the following equivalent coding:

```
'FOR' I..= 1 'STEP' 1 'UNTIL' 4 'DO'  
INPUT(60, '(' -4ZD, 3B')', A(/I/) )..
```

However, Inlist-Outlist has greater capabilities. It allows programmers to change certain parameters or "hidden variables" normally inaccessible to them. This is done through 7 descriptive procedures which compose the procedure LAYOUT. The procedure FORMAT used in our example is the first one. The others are H LIM, V LIM, H END, V END, TABULATION, and NO DATA. We shall now discuss each procedure.

- 1) FORMAT('('format string)'). The format string will be made up of the format codes previously described. If FORMAT is not used, the standard format, N, will be assumed.
- 2) H LIM (L,R) sets the horizontal limits of a line. L specifies the left margin while R specifies the right margin. The maximum size of a line, P, is set on the channel card. It is normally 80 for a card and 136 for a printed line. R can be smaller or larger than P.

If H LIM is not used, L and R are set to 1 and ∞ respectively.

- 3) V LIM (L',R') sets vertical limits. L' sets the top margin while R' sets the bottom margin. P', the maximum number of lines on a page is also fixed by the channel card.

If V LIM is not used, L' and R' are set to 1 and ∞ respectively

- 4) H END (PN, PR, PP) indicates that control is transferred to the procedures PN, PR or PP; PN when a / is found in the format, PR in the case of R overflow, i.e., the right margin is reached and $R < P$, and PP in the case of P overflow, i.e., the end of the line is reached and $P < R$.
- 5) V END (PN', PR', PP') indicates that control is transferred to the procedures PN', PR' or PP'; PN' when an * is found in the format, PR' in the case of R' overflow, and PP' in the case of P' overflow.

When H END or V END are not used, PN, PR, PP, PN', PR', and PP' are set equal to a procedure DUMMY which is essentially a do nothing procedure.

- 6) TABULATION (N). When this procedure is used, numbers (or words) to be read or written start in columns L, L+N, L+2N, etc., until the right margin or the end of the line is reached. TABULATION is used in conjunction with the J format. Everytime a J is encountered, the character pointer jumps to the next tabulation position. When TABULATION is not used, N is set to 1.
- 7) NO DATA (L). L is a label to which the program will jump if the list of items is not yet exhausted but no more data is available. When this happens on input and NO DATA is not used, the End of File mark (6-7-8-9 card) is detected and the program is aborted with the error 'unchecked EOF'.

As examples, the following two programs are given.

EXAMPLE 1

ALGOL-60

(1.1)

07/22/71

```
00** #BEGIN# #INTEGER# I.,
      #INTEGER# #ARRAY# A,B,C(/1..12/),

#COMMENT# SETTING THE FORMAT AND THE LABEL TO GO TO WHEN DATA IS
          EXHAUSTED.,
#PROCEDURE# LAYIN., #BEGIN# FORMAT ( (#12(30.58)*)*).,
          NO DATA(L2)., #END#.,

#COMMENT# THIS PROCEDURE ENABLES ELEMENTS OF THE ARRAYS A AND B TO BE
          READ ALTERNATELY.,
10** #PROCEDURE# LIST(AIN).,
      #PROCEDURE# AIN.,
      #BEGIN# #FOR# I..=1 #STEP#1 #UNTIL#12 #DO# #BEGIN# AIN(A(/I/)).,
          AIN(B(/I/)) #END#., #END#.,

      OUTPUT(61, #1'//ISS*)#, #(*THIS IS AN EXPT*)#).,

#COMMENT# THIS INITIALIZES THE ELEMENTS OF THE ARRAYS TO ZERO.,
      #FOR# I..=1 #STEP#1 #UNTIL#12 #DO#
      #BEGIN# A(/I/)..=0., B(/I/)..= 0., #END#.,
20** #COMMENT# THE ARRAYS ARE READ IN, THEIR SUMS COMPUTED AND PRINTED OUT.,
      IN LIST(60-LAYIN,LIST).,
L2.. #FOR# I..=1 #STEP#1 #UNTIL#12 #DO#
      #BEGIN# C(/I/)..=A(/I/)+B(/I/),
      OUTPUT(61, #1',Z000*)#, C(/I/)) #END#.,

#END#
      #EOP#
```


EXAMPLE 2

07/22/71

ALGOL-60 (1.1)

```

00** *BEGIN* *INTEGER*I.,
      *INTEGER* *ARRAY* C(/1..1000/)..

*COMMENT* LAYOUT SETS THE HORIZONTAL AND VERTICAL MARGINS, TABULATION,
          FORMAT AND PROVISION TO GO TO PROCEDURE VPR WHEN BOTTOM
          MARGIN IS REACHED..

*PROCEDURE* LAYOUT..
  *BEGIN*
  H LIM(10,110)..
  V LIM(15,50)..
10**  TABULATION(10)..
      FORMAT(*(*1000(3ZU,J)*)*)..
      V END(VPR,VPR,VPR)..
      *END*..

*COMMENT* VPR CHANGES VERTICAL MARGIN, TABULATION AND FORMAT WHEN
          CONTROL IS TRANSFERRED TO IT..

*PROCEDURE* VPR..
  *BEGIN*
  V LIM(10,55)..
20**  TABULATION(5)..
      FORMAT(*(*1000( 4U,J)*)*)..
      *END*..

*COMMENT* ANS MAKES UP LIST TO BE OUTPUT..
*PROCEDURE* ANS(ELEM)..
*PROCEDURE* ELEM..
  *BEGIN* *FOR*I..=1*STEP*1 *UNTIL*1000*DO*ELEM(C (/I/)) *END*..

      C(/I/)..= 1.0..
30**  *FOR* I..= 2 *STEP* 1 *UNTIL* 1000*DO*
      C(/I/)..= C(/I-1/) + 1..

*COMMENT* ARRAY C CONTAINS INTEGERS FROM 1 TO 1000 AND OUTLIST
          OUTPUTS THESE NUMBERS ACCORDING TO LAYOUT INSTRUCTIONS..
      OUT LIST(61,LAYOUT,ANS)..

  *END*
  *EOP*

```

V. Other Input-Output Procedures

Some other useful I/O procedures will be described in this section.

1) SYSPARAM(CHANNEL NO,FUNCTION,QUANTITY)

Normally, P (the maximum size of a line), P' (the maximum size of a page) and k (the number of blank spaces separating numbers in the standard format) are set on the channel card. SYSPARAM allows the programmer to access or modify these parameters as well as two others, ρ and ρ' . ρ' is the line pointer while ρ specifies the position along the line.

FUNCTION takes on values from 1 to 10 depending on our intentions (see table below) while QUANTITY is an integer variable called by name which will either represent the new value of the parameter or be assigned the present value of the parameter.

<u>FUNCTION</u>	<u>ACTION TAKEN BY SYSPARAM</u>
1	Assigns the present value of ρ to QUANTITY
2	Changes ρ to new value given by QUANTITY
3	Assigns the present value of ρ' to QUANTITY
4	Changes ρ' to new value given by QUANTITY
5	Assigns the present value of P to QUANTITY
6	Changes P to new value given by QUANTITY
7	Assigns the present value of P' to QUANTITY
8	Changes P' to new value given by QUANTITY
9	Assigns the present value of k to QUANTITY
10	Changes k to new value given by QUANTITY

2) IN REAL(CHANNEL NO,DESTINATION) OUT REAL(CHANNEL NO,SOURCE)

SOURCE is a variable of type real which can also be an arithmetic expression. IN REAL will assign the next value on the input device to DESTINATION while OUT REAL will transfer the value of SOURCE to the output device. IN REAL and OUT REAL handle numbers in standard format. In the case where a value has been transferred to a temporary file by OUT REAL, it can be transferred back to a variable by IN REAL.

3) IN ARRAY (CHANNEL NO,DESTINATION) OUT ARRAY(CHANNEL NO,SOURCE)

SOURCE and DESTINATION are arrays of type real. These procedures are analogous to IN REAL and OUT REAL except now we are dealing with arrays. The elements of the arrays

are transferred in or out according to lexicographic order of the values of the subscripts. For a 3x3x3 array, the order of subscripts will be:

111,112,113,	121,122,123,	131,132,133
211,212,213,	221,222,223,	231,232,233
311,312,313,	321,322,323,	311,332,333

Note: These procedures provide a short and easy way to input or output arrays as long as the order of transfer described above is kept in mind.

4) IN CHARACTER (CHANNEL NO,STRING,DESTINATION)
OUT CHARACTER(CHANNEL NO,STRING,SOURCE)

Here SOURCE and DESTINATION are integer variables. IN CHARACTER examines the next basic character on the channel. If it matches the Jth character of STRING, DESTINATION is set to the value J. If no match is found, DESTINATION is set to 0. OUT CHARACTER examines the value of SOURCE. If it is in the range of 1 to J where J is the length of the string, the corresponding character of the string is output; otherwise an object program error results.

5) CHLENGTH(string)

The value of CHLENGTH(string) is equal to the number of characters enclosed between the outermost string quotes. Each embedded string quote, '(' or ')', counts as 3 characters.

A sample program which uses these procedures is shown below.

ALGOL-60

(1.1)

08/09/

```

00**  *BEGIN*
      *INTEGER* QUANT,PRO,,
      *INTEGER* DEST,SOUT,NY,,
      *REAL* X,Y,,
      *REAL**ARRAY* C(/1..10/),,

```

COMMENT IN ARRAY WILL INPUT ARRAY VALUES FROM CARDS AND OUT ARRAYS PRINTS THEM OUT USING STANDARD FORMAT.,

```

      IN ARRAY(60,C),,
      OUT ARRAY(61,C),,

```

10**

COMMENT IN SYSPARAM, FUNCTION = 4 AND SO THE LINE POSITION IS SET EQUAL TO PRO (PRO = 7) AND THE NEXT PRINTING STARTS AT LINE

```

      PRO..= 7 ,,
      SYSPARAM(61,4,PRO),,

```

COMMENT NOTE THAT IN OUT REAL, THE SECOND ARGUMENT CAN BE AN ARITHMETIC EXPRESSION.,

```

      IN REAL(60,X),,
      OUT REAL(61,5*X + 1),,

```

20**

COMMENT HERE SYSPARAM ASSIGNS THE VALUE OF K TO QUANT AND THE VALUE IS PRINTED OUT WITH THE FOLLOWING STATEMENT.,

```

      SYSPARAM(61,9,QUANT),,
      OUTPUT(61,*(#///,2ZD#)*, QUANT),,

```

COMMENT THE USE OF IN CHARACTER, OUT CHARACTER AND CHLENGTH IS ILLUSTRATED IN THE NEXT STATEMENTS.,

```

      IN CHARACTER(60,*(#HELLO#)*,DEST),,
      NY..= CHLENGTH(*(#MAGANDA#)*),,

```

30**

```

      OUTPUT(61,*(#/,34D#)*,DEST),,
      OUTPUT(61,*(#/,3ZD#)*,NY),,
      *FOR* SOUT..= 1 *STEP* 1 *UNTIL* 5 *DO*
      OUT CHARACTER(61,*(#HELLO#)*,SOUT),,
      *END*

```

ENP