



*Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.*

*The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.*

## Table of Contents

| <u>Page</u> | <u>Section</u>                         |
|-------------|--|
| 1           | Acknowledgements.                      |
| 2           | Reasons to read this Report.           |
| 3           | Man or boy?                            |
| 6           | An example program.                    |
| 11          | Informal introduction to CDC ALGOL 60. |
| 15          | Recommended character set.             |
| 17          | Source card conventions.               |
| 18          | Compilation and execution speeds.      |
| 31          | Memory for data.                       |
| 34          | Precision of numbers.                  |
| 38          | Integer labels.                        |
| 40          | Formal and actual parameters.          |
| 44          | Subscript checking.                    |
| 47          | Subscripted controlled variable.       |
| 50          | Long identifiers.                      |
| 51          | Labeled outer block.                   |
| 53          | Arithmetic error conditions.           |
| 58          | Determination of types.                |
| 61          | Input/output procedures.               |
| 64          | Standard library procedures.           |
| 68          | Diagnostics (error messages).          |
| 69          | Job control cards.                     |
| 70          | Version 4 of CDC ALGOL 60.             |
| 71          | References.                            |



Acknowledgements

I want to thank Rob F. Veldhuyzen van Zanten, the CDC ALGOL 60 expert at SARA, who tested all the CDC programs listed in this report and who advised me on many differences between MC and CDC ALGOL 60.

Acknowledgement is also made to Dick Grune who corrected and clarified the information herein on MC ALGOL 60.

Reasons to read this Report

Firstly, here are four good reasons to not read any further:

- (i) You do not know/use the MC ALGOL 60 system [3,5,6].
- (ii) You prefer the more precise and complete description of CDC ALGOL 60 to be found in reference [2], which is available from:

Ir. H. te Riele  
 Mathematisch Centrum  
 telephone: 947272 (ext. 40)

- (iii) This Report is incomplete and may contain errors because the SARA computer does not arrive until 1 November (1 December is the acceptance test date) and, furthermore, a new version (4) of CDC ALGOL 60 will be released the first quarter of 1973.
- (iv) You will use and trust the "ALGOL-Convertie Programma X8 CDC" translator to automatically convert your MC ALGOL 60 programs for the SARA computer. In this case, contact either H. te Riele (address above) or:

Ir. R.F. Veldhuyzen van Zanten  
 SARA (Kamer OE23)  
 de Boelelaan 1105  
 Postbus 7161  
 Amsterdam-Buitenveldert  
 telephone: 485428

for details.

Next are two good reasons to read on:

- (1) You know/use MC ALGOL 60 and now want to learn some CDC ALGOL 60.
- (2) You used the "ALGOL-Convertie Programma X8 CDC" translator and:
  - (a) it did not completely translate your MC ALGOL 60 program; or
  - (b) the translated program now runs too slow or requires too much memory; or
  - (c) you want to understand what it did to your MC ALGOL 60 program; or
  - (d) you do not trust it.

Man or boy? [Knuth 1964 and van der Poel 1972]

A good test-program (for recursion and recursive non-locals) to determine whether an ALGOL 60 compiler is a "Man" or a "Boy" follows:

```

begin integer procedure A(k,x1,x2,x3,x4,x5);
  value k; integer k,x1,x2,x3,x4,x5;
  begin integer procedure B;
    begin k:= k-1;
      A:= B:= A(k,B,x1,x2,x3,x4)
    end;
    if k<0 then A:= x4+x5 else B
  end;
  print (A(6,1,2,3,4,5))
end

```

The correct answer is 28; now both MC ALGOL 60 and CDC ALGOL 60 output 28, so they qualify as "Man-compilers".





020872- 24 A 2458V.4 JACKALANEN

1

A2458V.4,JACK ALANEN

```
'BEGIN' 'COMMENT' MAN OR BOY (ANSWER = 28) FROM VD POEL, PAGE 1.22.3;
1 'INTEGER' 'PROCEDURE' A(K,X1,X2,X3,X4,X5);
2 'VALUE' K; 'INTEGER' K;
3 'BEGIN' 'INTEGER' 'PROCEDURE' B;
4 'BEGIN' K:=K-1; A:=B:=A(K,B,X1,X2,X3,X4) 'END';
5 'IF' K<=0 'THEN' A:=X4+X5 'ELSE' B
6 'END';
7 PRINT(A(6,1,2,3,4,5))
8 'END'
```

020872- 24 A 2458V.4 JACKALANEN

2

\*28

020872- 24 A 2458V.4 JACKALANEN

3

|        |        |    |
|--------|--------|----|
| 0      | 1      | 25 |
| 160320 | 160320 |    |
| 99     | 1139   |    |

An example program

To demonstrate that many valid MC ALGOL 60 programs are (even after using the CDC character set) illegal CDC ALGOL 60 programs, an example program follows. Almost every line in the CDC ALGOL 60 version contains at least one syntax or semantics error relative to the MC version. These errors will be explained in later sections.

```
00** TEST: "BEGIN" "REAL" REAL Y, EPS; "INTEGER" N;  
"PROCEDURE" OUTPUT(X); "BEGIN" NLCR; PRINT(X) "END";  
"BOOLEAN" "ARRAY" B(/1:80000/); "INTEGER" "ARRAY" A(/1:3,1:2/),  
E(/1:10/);  
"PROCEDURE" Q(INT, I); "INTEGER" INT, I; I:=INT;  
"PROCEDURE" P(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,  
A1, a1, C1, D1, E1, F1, G1, H1, J1, K1, L1, M1, N1, O1, P1, Q1, R1, S1, T1, U1, V1, W1, X1,  
Y1, Z1, A2, B2, C2, U2, E2, F2, G2, H2, I2, J2, K2, L2, M2, N2, O2, P2, Q2, R2, S2, T2,  
U2, V2, W2, X2, Y2, Z2);  
"VALUE" Z2; "LABEL" Z2;  
10** "INTEGER" A; ;  
  
REAL Y:=0;  
EPS:=0;  
REAL Y:=1.234*400; OUTPUT(REAL Y);  
EPS:=-400; OUTPUT(EPS);  
39: N:=2;  
Q(1.6, N); OUTPUT(N);  
PRINTTEXT("MESSAGE");  
E(ZZ):=N:=1; "FOR" E(ZZ):=0 "WHILE" N<10 "DO" N:=N+1; OUTPUT(E(ZZ));  
20** PRINT(LN(0)); PRINT(SQRT(-3)); PRINT("700"); PRINT("700"); PRINT(1/0);  
A(/0,3/):=0;  
"END"
```

FINIS GEN.BY FOR-CARD

|      |    |                                 |           |   |
|------|----|---------------------------------|-----------|---|
| LINE | 0  | PROGRAM ENDS                    | (MESSAGE) | 1 |
| LINE | 22 | PROGRAM ENDS                    | (MESSAGE) | 1 |
| LINE | 22 | SOURCE DECK ENDS                | (MESSAGE) | 1 |
| LINE | 1  | DECLARATION MISSING FOR.. PRINT |           | 6 |
| LINE | 1  | DECLARATION MISSING FOR.. NLCR  |           | 6 |
| LINE | 1  | SPECIFICATION REQUIRED FOR.. X  |           | 3 |
| LINE | 9  | ILLEGAL "VALUE" SPECIFICATION   |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. B  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. C  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. D  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. E  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. F  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. G  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. H  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. I  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. J  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. K  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. L  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. M  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. N  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. O  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. P  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Q  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. R  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. S  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. T  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. U  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. V  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. W  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. X  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Y  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Z  |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. A1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. B1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. C1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. D1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. E1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. F1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. G1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. H1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. J1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. K1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. L1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. M1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. N1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. O1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. P1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Q1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. R1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. S1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. T1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. U1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. V1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. W1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. X1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Y1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Z1 |           | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. A2 |           | 3 |

|      |    |  |   |
|------|----|--|---|
| LINE | 10 | SPECIFICATION REQUIRED FOR.. B2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. C2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. D2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. E2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. F2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. G2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. H2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. I2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. J2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. K2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. L2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. M2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. N2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. O2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. P2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Q2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. R2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. S2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. T2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. U2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. V2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. W2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. X2          | 3 |
| LINE | 10 | SPECIFICATION REQUIRED FOR.. Y2          | 3 |
| LINE | 12 | CHARACTER                                | 1 |
| LINE | 14 | ARITHMETIC OVERFLOW OF NUMBER/EXPRESSION | 2 |
| LINE | 16 | .. FOLLOWS AN INCORRECT OPERAND          | 3 |
| LINE | 18 | NON-FORMAT STRING (MESSAGE)              | 1 |
| LINE | 18 | DECLARATION MISSING FOR.. PRINTTEXT      | 6 |
| LINE | 20 | ARITHMETIC OVERFLOW OF NUMBER/EXPRESSION | 2 |
| LINE | 20 | ARITHMETIC OVERFLOW OF NUMBER/EXPRESSION | 2 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72  
 14.15.32.VELDMIN  
 14.15.32.VELDM.NR9500.  
 14.15.32.ALGOL.  
 14.15.39.FILE OPENED---OUTPUT  
 14.15.42.FILE OPENED---LGO  
 14.15.42. 0 279  
 14.15.43.CP 000.280 SEC.  
 14.15.43.PP 003.854 SEC.

6

030872- 82 A 2458V.1 JACKALANEN

1

A2458V.1, JACK ALANEN

```

1 TEST: 'BEGIN' 'REAL' REAL Y,EPS; 'INTEGER' N;
2 'PROCEDURE' OUTPUT(X); 'BEGIN' NLCR; PRINT(X) 'END';
3
4 'BOOLEAN' 'ARRAY' B[1:800000]; 'INTEGER' 'ARRAY' A[1:3,1:2],E[1:10];
5
6 'PROCEDURE' Q(INT,I); 'INTEGER' INT,I; I:=INT;
7
8 'PROCEDURE' P(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,
9     A1,B1,C1,D1,E1,F1,G1,H1,I1,J1,K1,L1,M1,N1,O1,P1,Q1,R1,S1,T
10     1,U1,V1,W1,X1,Y1,Z1,
11     A2,B2,C2,D2,E2,F2,G2,H2,I2,J2,K2,L2,M2,N2,O2,P2,Q2,R2,S2,T
12     2,U2,V2,W2,X2,Y2,Z2);
13 'VALUE' Z2; 'LABEL' Z2;
14 'INTEGER' A;
15
16 REAL Y:=0; EPS:=0;
17 REAL Y:=1.234e400; OUTPUT(REAL Y);
18 EPS:=e-400; OUTPUT(EPS);
19 39: N:=2;
20 Q(1.6,N); OUTPUT(N);
21 PRINTTEXT("MESSAGE");
22 E[2]:=N:=1;'FOR' E[N]:=0 'WHILE' N < 10 'DO' N:=N+1; OUTPUT(E[2]);
23 PRINT(LN(0)); PRINT(SQRT(-3)); PRINT(e-700); PRINT(e700); PRINT(1/9);
24 A[0,3]:=0;
25 'END'

```

030872- 82 A 2458V.1 JACKALANEN

2

```

+.1234000000000e+401
+.9999999999999998e-400
      +2      MESSAGE
      +0      -.1776646197514e+629      +0      +.6188692094765e-616 +.1776646197514e+629 +.1776646197514e+629
ER 502 24 -0

```

030872- 82 A 2458V.1 JACKALANEN

3

```

1 1 25
469927 469927
354 30290

```

Informal introduction to CDC ALGOL 60

Compared to MC ALGOL 60, real and integer values in CDC ALGOL 60 contain more significant digits. However, the range of real values is not as great, so that there are both very small and very large real values which can be represented in MC, but not CDC, ALGOL 60.

MC ALGOL 60 programs usually require more words of memory for data (with the important exception of Boolean arrays) than CDC ALGOL 60. Besides, the SARA computer memory is larger than the X8 memory.

The CDC ALGOL 60 input/output procedures are completely different and the CDC printer has only 136 characters per line.

CDC ALGOL 60 will not handle integer labels, unspecified formal parameters, more than 32 nested blocks, and labels declared as value parameters.

"Strange things" happen in CDC ALGOL 60 when the controlled variable of a for statement is subscripted and when you ask for array bound checking.

Source programs in CDC ALGOL 60 are not as nice to read because of a poor character set ("AND" "NOT GREATER" (/ "/" instead of  $\wedge \leq [ \div ]$ ) and because one source line corresponds to columns 1 to 72 of one source card (hence there is no \$ continuation symbol as in MC ALGOL 60).

The CDC ALGOL 60 compiler is fast (one page of source text in less than one second) and has numerous options for faster compilation, more efficient object-code, debugging, relocatable binary object programs, etc.

Finally, the CDC ALGOL 60 system includes useful features not available in the MC ALGOL 60 system such as own arrays (without dynamic array bounds), procedure body in code, comprehensive input/output procedures based upon the ACM proposal [8], debugging directives in the source code, and ALGOL macros. The following three CDC ALGOL 60 programs illustrate some of these features.

VELDH,NR9500. VELDHUYZEN VAN ZANTEN  
COMMENT.EXAMPLE OF ENTRY/EXIT MACROS, ALGOL MANUAL 15-2  
COMPASS.S=ALGTEXT.

ALGOL.  
IGO.  
000000000000000000000000 7/8/9 END OF RECORD

IDENT SAFE  
\*\*SHRPPROGRAM  
SST  
\*PARAMETER VALUE SAVING..

CODE 1  
VALUE 1.X6.R.1  
S44 SAVE  
RETURN  
SAVE R557 1

\*RETURN VALUE..  
CODE 2  
S44 SAVE  
ASSIGN 1.X4  
RETURN  
END

000000000000000000000000 7/8/9 END OF RECORD

"BEGIN"  
"PROCEDURE" P1(X);"REAL"X;"CODE"1;  
"PROCEDURE" P2(X);"REAL"X;"CODE"2;  
"REAL"4.R;  
A:=3; P1(A); P2(R); OUTREAL(61.R)

"END"  
000000000000000000000000 6/7/8/9 END OF FILE

\*3.0000000000000000000000  
END OF ALGOL RUN \*V3.0\*

11/08/72 +GRONINGEN SCORE 3.3 SUM 279 24/01/72  
13.31.55.VELDH6H  
13.31.55.VELDH,CM100000.NR9500.  
13.31.55.EXAMPLE OF ENTRY/EXIT MACROS.  
13.31.55. SEE ALGOL MANUAL CHAPTER 15-2.  
13.31.55.COMPASS.S=ALGTEXT.  
13.31.57.FILE OPENED---SCR1  
13.31.57.FILE OPENED---CMPSCR  
13.31.58.FILE OPENED---IGO  
13.31.58.FILE OPENED---OUTPUT  
13.31.58. ASSEMBLY COMPLETE. 53400R SCM USED.  
13.31.58.ALGOL.  
13.32.03. 1 160  
13.32.03.IGO.  
13.32.07.CP 001.245 SEC.  
13.32.07.PP 008.453 SEC.



ALGOL-60 (3.0) 11/08/72 13.47 HRS PAGE 1

```

00**
"BEGIN" "COMMENT" "TRACE" FAC=L1:
      "COMMENT" "SMAP" I:
"INTEGER" "PROCEDURE" FAC(N): "INTEGER" N: FAC:="IF" N=1 "THEN" 1
      "ELSE" N*FAC(N-1):
"INTEGER" I:
      "FOR" I:=1 "STEP" 1 "UNTIL" 4 "DO"
      L1:OUTPUT(6).("I.".("N=") ".D2R.",(" FACTORIAL(N)=") ".97D")".
      I.FAC(I):
10**
"END"
FINIS GEN.RY FOR-CARD

```

ALGOL-60 (3.0) XXALGOL 11/08/72 13.47 HRS PAGE 1

|      |    |                  |           |   |
|------|----|------------------|-----------|---|
| LINE | 1  | PROGRAM BEGINS   | (MESSAGE) | 1 |
| LINE | 11 | PROGRAM ENDS     | (MESSAGE) | 1 |
| LINE | 11 | SOURCE DECK ENDS | (MESSAGE) | 1 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE D.F.L.X

```

**LINE 7.I.= +1      **LINE 8.(USAGE) I
N=1 . FACTORIAL(N)= **LINE 9.(INVOKF) FAC      1      **LINE 9.(EXIT) I= *1      **LINE 7.I.= +2      **LINE 8.(USAGE) I
N=2 . FACTORIAL(N)= **LINE 9.(INVOKF) FAC      **LINE 4.(INVOKF) FAC      2      **LINE 9.(EXIT) I= +2      **LINE 7.I.= +3
**LINE 8.(USAGE) I
N=3 . FACTORIAL(N)= **LINE 9.(INVOKF) FAC      **LINE 4.(INVOKF) FAC      **LINE 4.(INVOKF) FAC      6      **LINE 9.(EXIT) I= +3
**LINE 7.I.= +4      **LINE 8.(USAGE) I
N=4 . FACTORIAL(N)= **LINE 9.(INVOKF) FAC      **LINE 4.(INVOKF) FAC      **LINE 4.(INVOKF) FAC      **LINE 4.(INVOKF) FAC      24
**LINE 9.(EXIT) I= +4
END OF ALGOL RUN *V3.0*

```

```

11/08/72 +GRONINGEN SCORE 3.3 SUM 279 24/01/72
13.47.16.VELDH60
13.47.16.VELDH.MR9500.
13.47.16.ALGOL.D.
13.47.19.FILE OPENED---OUTPUT
13.47.22.FILE OPENED---LGO
13.47.23. 0 274
13.47.27.LGO.C.D.
13.47.30.CP 000.457 SEC.
13.47.30.DP 008.650 SEC.

```

ALGOL-60 (3.0) 08/08/72 14.46 HRS PAGE 1

00\*\* "CODE"1;"REAL""PROCEDURE"SP(FORMAL-PAR);"REAL"FORMAL-PAR;  
SP:=FORMAL PAR+1;  
FINIS GEN.BY EOR-CARD

ALGOL-60 (3.0) CP00001 08/08/72 14.46 HRS PAGE 1

LINE 0 PROGRAM BEGINS (MESSAGE) 1  
LINE 1 PROGRAM ENDS (MESSAGE) 1  
LINE 1 SOURCE DECK ENDS (MESSAGE) 1

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F,L,X

ALGOL-60 (3.0) 08/08/72 14.46 HRS PAGE 1

00\*\* "BEGIN"  
"REAL""PROCEDURE" P(X);"CODE"1;  
"REAL"ACTUAL PAR.B;  
ACTUAL PAR:=2; H:=P(ACTUAL PAR); OUTREAL(61,B)  
"END"  
FINIS GEN.BY EOR-CARD

ALGOL-60 (3.0) XXALGOL 08/08/72 14.46 HRS PAGE 1

LINE 0 PROGRAM BEGINS (MESSAGE) 1  
LINE 4 PROGRAM ENDS (MESSAGE) 1  
LINE 4 SOURCE DECK ENDS (MESSAGE) 1

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F,L,X

\*3.00000000000000000000\*000  
END OF ALGOL RUN \*V3.0\*

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72  
14.46.18.VELDMZ0  
14.46.18.VELDM,NR9500.  
14.46.18.ALGOL.  
14.46.19.FILE OPENED---OUTPUT  
14.46.22.FILE OPENED---LGO  
14.46.23. 0 114  
14.46.23.ALGOL.  
14.46.24. 0 249  
14.46.30.LGO.  
14.46.37.CP 000.354 SEC.  
14.46.37.PP 013.856 SFC.

Recommended character set

Numerous character set problems exist because of incompatibilities between character codes of different punches and between character conventions of different CDC installations. For example, the Zeeman Lab. uses CDC 026 keypunches, SARA uses IBM 029 keypunches, and the MC uses modified IBM 029 keypunches.

If you use only the keypunches at the MC or SARA, then the following character set is recommended. Otherwise, you need to investigate the character set problems yourself.

Be warned that the SARA CDC ALGOL 60 character set is still tentative; so even if you use the recommended set below, you may have to make changes in 1973.

| ALGOL 60 Reference Language |            | Recommended *<br>Hardware Representation |
|-----------------------------|------------|--|
| Capital letters             | A B C etc. | A B C etc.                               |
| Digits                      | 0 1 2 etc. | 0 1 2 etc.                               |
| Arithmetic operators        | +          | +  |
|                             | -          | -  |
|                             | ×          | *  |
|                             | /          | /  |
|                             | ÷          | "/"                                      |
|                             | ↑          | **                                       |
| Relational operators        | <          | <  |
|                             | ≤          | "NOT GREATER"                            |
|                             | =          | =  |
|                             | ≥          | "NOT LESS"                               |
|                             | >          | >  |
|                             | ≠          | "NOT EQUAL"                              |
| Separators                  | ,          | ,  |
|                             | .          | .  |
|                             | 10         | "  |
|                             | :          | :  |
|                             | ;          | ;  |
|                             | :=         | :=                                       |

| ALGOL 60 Reference Language    |  | Recommended <sup>*</sup><br>Hardware Representation |
|--------------------------------|--|---|
| Brackets                       | (<br>)<br>[<br>]<br>'<br>,   | (<br>)<br>(/<br>/<br>"<br>")                        |
| Logical operators              | $\equiv$<br>$\supset$<br>$\vee$<br>$\wedge$<br>$\neg$                | "EQUIV"<br>"IMPL"<br>"OR"<br>"AND"<br>"NOT"         |
| Reserved <sup>†</sup><br>words | <u>begin</u><br><u>Boolean</u><br><u>end</u><br><u>go to</u><br>etc. | "BEGIN"<br>"BOOLEAN"<br>"END"<br>"GO TO"<br>etc.    |

\* Punch these symbols on the IBM 029 machines at the MC or at SARA (located at the VU). In addition, the five characters # \$ ? \_  $\_$  may be used; the last is "blank" and the next-to-last can not be used to "underline".

† CDC permits blanks in ALGOL symbols. Thus begin may be punched as "B E G I N" and so on.

Source card conventions [2;4-4,5-3] \*

CDC ALGOL 60: Each line in the printed listing of the source program corresponds to one source card, columns 1-72. Hence there is no "\$-convention" for skipping to the next card as in MC ALGOL 60. A source deck is separated from input data by an END-OF-RECORD card (a 7-8-9 multipunch in column 1) and an END-OF-FILE card (a 6-7-8-9 multipunch in column 1) follows the input data (or final end if there is no input data on cards). The additional cards required to identify the user, call the ALGOL compiler, etc. are discussed under "Job control cards".

\* For further information, see reference [2], sections 4-4 and 5-3.

Compilation and execution speeds [2;6-4]

CDC ALGOL 60: Various options increase/decrease compile and/or execution times. For example, the 0 option significantly decreases execution time but should be used only for production programs because it increases compilation time.

A rule-of-thumb is that CDC ALGOL 60 compiles one page of ALGOL text in less than one second.

A rule-of-thumb is that the SARA computer is three to five times faster than the X8.

A three page ALGOL 60 program (written by Frank Goossens) which performs number theory computations is given next. It compiled and executed on the CDC computer in 14.9 seconds (plus 14.4 seconds of peripheral processor time), whereas the X8 version compiled and executed in 16 millihours (57.6 seconds).

```

00** "BEGIN"
      "INTEGER" N,NPRINTMIN,PBASE
      **
      INPUT(A,0,"(N)")N,NPRINTMIN)..
      PBASE:=0..
"BEGIN"

"PROCEDURE" SEARCH(A,R).. "VALUE" A,R.."INTEGER" A,R..
"BEGIN"
  "IF" NUM(/A/) "GREATER" 0 "THEN" INVESTIGATE(A) "ELSE"
10** SEARCH(REFIRST(/A/),PBASE(/A/))..
  "IF" NUM(/R/) "GREATER" 0 "THEN" INVESTIGATE(R) "ELSE"
  SEARCH(REFIRST(/R/),PBASE(/R/))
"END" SEARCH..

"PROCEDURE" INVESTIGATE(A).. "VALUE" A.."INTEGER" A..
"BEGIN" "INTEGER" PHEL P.. PHEL:=PBASE..
  [PINV.."IF" PHEL P "LESS" 0 "THEN"
  "BEGIN" "IF" NUM(/STACK(/PHEL P/)) = NUM(/A/) "THEN"
20**   "BEGIN" MY:=TRUE..
      STACK(/PHEL P/):=STACK(/PHEL P/)*A
      "END" "ELSE"
      "BEGIN" PHEL P:=PHEL P+1..
      "GOTO" PINV
      "END"
  "END" "ELSE"
  "BEGIN" STACK(/P/):=A.. P:=P+1 "END"
"END" INVESTIGATE..

"PROCEDURE" HEADER..
30** "BEGIN"
  OUTPUT(A,1,"(N/((N)*R=PRODUCT+INTEGER))")..
  SYSPARAM(A,2,35)..
  OUTPUT(A,1,"(N/((N)*R))")..
"END" HEADER..

"PROCEDURE" OUT(N,ARRAY,POINTER,ADDITIONAL,SUM)..
"VALUE" N,POINTER,ADDITIONAL,SUM..
"INTEGER" N,POINTER,ADDITIONAL,SUM.."ARRAY" ARRAY..
"BEGIN"
40** "IF" N "LESS" NPRINTMIN "THEN" "GOTO" ENDOUT..
  OUTPUT(A,1,"(N/370,(N)=",270)N,ARRAY(/PBASE/))..
  "FOR" I:=PBASE+1 "STEP" 1 "UNTIL" POINTER-1 "DO"
  OUTPUT(A,1,"(N/((N)=",270)N,ARRAY(/I/))..
  OUTPUT(A,1,"(N/((N)=",270)N,ADDITIONAL)..
  SYSPARAM(A,2,35)..
  OUTPUT(A,1,"(N/270,(N)=",SUM)..
  SYSPARAM(A,2,SUM+40)..
  OUTPUT(A,1,"(N/((N)=",SUM)..
  "ENDOUT..
50** "END" OUT..

"PROCEDURE" AUTO(N,ARRAY,POINTER,ADDITIONAL)..
"VALUE" N,POINTER,ADDITIONAL.."INTEGER" N,POINTER,ADDITIONAL..
"ARRAY" ARRAY..
"BEGIN"
  "IF" N "LESS" NPRINTMIN "THEN" "GOTO" ENDOUT..
  SYSPARAM(A,2,100)..

```

```

        OUTPUT (A1,"(27D)",ARRAY(/PRASE/))..
        *FOR* I.=PRASE+1 "STEP" 1 "UNTIL" P(INTER-1 "D0"
60**  OUTPUT (A1,"(111(1 8))",27D)",ARRAY(/I/))..
        OUTPUT (A1,"(111(1 1))",27D)",ADDITIONAL)..
    ENDOUT..
    *END* OUTD..

        *INTEGER* "ARRAY" NUM,REFFIRST,REFSEC(/2..N/)..
        *INTEGER* "ARRAY" STACK,I (SI(/PRASE..10/))..
        *INTEGER* P,0,SUM,NUMBER,MINSUM,MINSUMNUMBER,DIFFMINSUM
            ,DIMMY,I,J,REFLFRST,REFLSEC,REFLPRD
70**  *ROOFEAN* MIX..

        *COMMENT*      INITIALIZATION OF THE ARRAYS NUM,REFFIRST,REFSEC
            NUM/ IF INDEX IS PRIME THEN CONTENTS IS INDEX
            IF INDEX IS APOWER OF PRIME THEN CONTENTS IS
            THAT PRIME
            IF INDEX IS NOT ONE OF ABOVE MENTIONED THEN
            CONTENTS IS NEGATIVE

80**  *THE ARRAYS REFFIRST,REFSEC ARE NOT INITIALIZED

        *FOR* I.=2"STEP"1"UNTIL" "N" "D0"
            NUM(/I/).=I..
        *FOR* I.=2"STEP"1"UNTIL" "N" "D0"
            *IF* NUM(/I/) "NOT LESS" I "THEN"
            *BEGIN* *FOR* J.=I**2"STEP" I "UNTIL" "N" "D0"
                NUM(/J/).=-1..
                J.=I..
            *FOR* J.=J*I "WHILE" J"NOT GREATER" N "D0"
            NUM(/J/).=I
90**  *END*..

    HEADER..

        *COMMENT*      NOW WE ANALYZE ARRAY NUM

100**  MINSUM.=6..
        MINSUMNUMBER.=5..
        NUMBER.=P..*COMMENT* THE NUMBERS 1 TO 6 ARE NOT ANALYZED..
NEXT.. NUMBER.=NUMBER+1.. P.=PRASE..
        *IF* NUMBER"GREATER" N "THEN" "GOTO" FINISH..
        *IF* NUM(/NUMBER/) "GREATER" 0 "THEN"
            *BEGIN* MINSUM.=NUMBER..*GOTO* COMPARE *END* ..
            REFFIRST.=ENTIER(SQRT(NUMBER))..
            REFLSEC.=REFLFRST+1..
            REFLPRD.=REFLFRST*REFLSEC..
            *FOR* DIMMY.=0"WHILE" REFLPRD "NOT EQUAL" NUMBER "D0"
            *IF* REFLPRD "LESS" NUMBER "THEN"
            *BEGIN*
                REFLPRD.=REFLPRD*REFLFRST..
                REFLSEC.=REFLSEC+1
            *END* *REFLSEC"
            *BEGIN*

```



```

REFLPR00.=REFLPR00-REFLSEC..
REFLFIRST.=REFLFIRST-1
"END"..

120**

"COMMENT" WE HAVE NOW TWO POSSIBLE RELATIVE PRIMES
SEARCH THE SIMPLEST COMBINATION..
MIX.="FALSE"..
SEARCH(REFLFIRST,REFLSEC)..

"COMMENT" THE STACK IS FILLED WITH THE BEST COMBINATION..
130** "COMMENT" CALCULATE THE SUM OF THE RELATIVE PRIMES..
SUM.=0..
"FOR" I.=PRASE "STEP" 1 "UNTIL" P-1 "DO"
SUM.=SUM+STACK(I/I)..
COMPARE..

DIFFMINSUM.=NUMBER-MINSUM*NUMBER..

"IF" SUM "GREATER" MINSUM+DIFFMINSUM "THEN"
"BEGIN"
140** OUT(NUMBER,I,IST,0,DIFFMINSUM,MINSUM+DIFFMINSUM)..
"END" "ELSE"
"IF" SUM = MINSUM+DIFFMINSUM "THEN"
"BEGIN"
OUT(NUMBER,STACK,P,0,SUM)..
OUT(NUMBER,I,IST,0,DIFFMINSUM)..
"END" "ELSE"
"BEGIN"OUT(NUMBER,STACK,P,0,SUM)..
"FOR" I.=PRASE "STEP" 1 "UNTIL" P-1 "DO"
LIST(I/I).=STACK(I/I)..
150** O.P.. MINSUM.=SUM..MINSUM*NUMBER.=NUMBER..
"END"..

"COMMENT" BOOLEAN MIX IS TRUE WHEN IN THE ANALYSIS
OF REFLFIRST AND REFLSEC OCCUR THE SAME PRIMES AND
POWERS OF PRIMES..

"IF" MIX "THEN"
"BEGIN"
160** LPMIX.. "IF" P "NOT GREATER" PRASE+2 "THEN" "GOTO" RDMIX..
STACK(I/P).=1..J.=PRASE..
"FOR" I.=PRASE "STEP" 2 "UNTIL" P-1 "DO"
"BEGIN"STACK(I/J).=STACK(I/I)*STACK(I/I)..
J.=J+1..
"END"..
P.=J.. "GOTO" LPMIX..
RDMIX..
REFLFIRST(/NUMBER/).=STACK(/PRASE/)..
REFLSEC(/NUMBER/).=STACK(/PRASE+1/)..
"END" "ELSE"
170** "BEGIN"REFLFIRST(/NUMBER/).=REFLFIRST..
REFLSEC(/NUMBER/).=REFLSEC..
"END"..
"GOTO" NEXT

```

ALGOL-60 (3.0)

XXALGOL

11/08/72 12.27 HRS

PAGE 4

\*\*\*\*\*  
FINISH\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

FINIS GEN.BY EOB-CARD

ALGOL-60 (3.0)

XXALGOL

11/08/72 12.27 HRS

PAGE 1

LINE 0  
LINE 176  
LINE 177

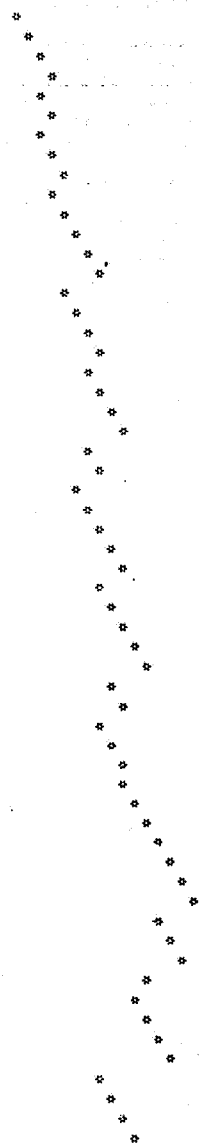
PROGRAM BEGINS (MESSAGE)  
PROGRAM ENDS (MESSAGE)  
SOURCE DECK ENDS (MESSAGE)

1  
1  
1

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

NIMROD=PRODUCT+INTEGR

| NIMROD | PRODUCT | INTEGR | SUM |
|--------|---------|--------|-----|
| 6      | 2       | 3      | 0   |
| 7      | 2       | 3      | 1   |
| 8      | 2       | 3      | 2   |
| 9      | 2       | 3      | 3   |
| 10     | 2       | 5      | 0   |
| 11     | 2       | 5      | 1   |
| 12     | 3       | 4      | 0   |
| 13     | 3       | 4      | 1   |
| 14     | 2       | 7      | 0   |
| 15     | 3       | 5      | 0   |
| 16     | 3       | 5      | 1   |
| 17     | 3       | 5      | 2   |
| 18     | 9       | 2      | 0   |
| 19     | 3       | 5      | 4   |
| 20     | 4       | 5      | 0   |
| 21     | 3       | 7      | 0   |
| 22     | 4       | 5      | 2   |
| 23     | 4       | 5      | 3   |
| 24     | 2       | 3      | 0   |
| 25     | 2       | 3      | 1   |
| 26     | 2       | 3      | 2   |
| 27     | 2       | 3      | 3   |
| 28     | 4       | 7      | 0   |
| 29     | 4       | 7      | 1   |
| 30     | 5       | 2      | 3   |
| 31     | 5       | 2      | 3   |
| 32     | 5       | 2      | 3   |
| 33     | 5       | 2      | 3   |
| 34     | 5       | 2      | 3   |
| 35     | 5       | 7      | 0   |
| 36     | 4       | 9      | 0   |
| 37     | 5       | 7      | 2   |
| 38     | 5       | 7      | 3   |
| 39     | 3       | 13     | 0   |
| 40     | 5       | 2      | 0   |
| 41     | 5       | 2      | 1   |
| 42     | 2       | 3      | 7   |
| 43     | 2       | 3      | 7   |
| 44     | 2       | 3      | 7   |
| 45     | 5       | 9      | 0   |
| 46     | 5       | 9      | 1   |
| 47     | 5       | 9      | 2   |
| 48     | 5       | 9      | 3   |
| 49     | 5       | 9      | 4   |
| 50     | 5       | 9      | 5   |
| 51     | 3       | 17     | 0   |
| 52     | 4       | 13     | 0   |
| 53     | 4       | 13     | 1   |
| 54     | 6       | 13     | 2   |
| 55     | 5       | 11     | 0   |
| 56     | 7       | 2      | 0   |
| 57     | 7       | 2      | 1   |
| 58     | 7       | 2      | 2   |
| 59     | 7       | 2      | 3   |
| 60     | 4       | 3      | 5   |
| 61     | 4       | 3      | 5   |
| 62     | 4       | 3      | 5   |
| 63     | 4       | 3      | 5   |



3 \* 4 + 2

3 \* 5 + 3

4 \* 5 + 1

5 \* 7 + 1

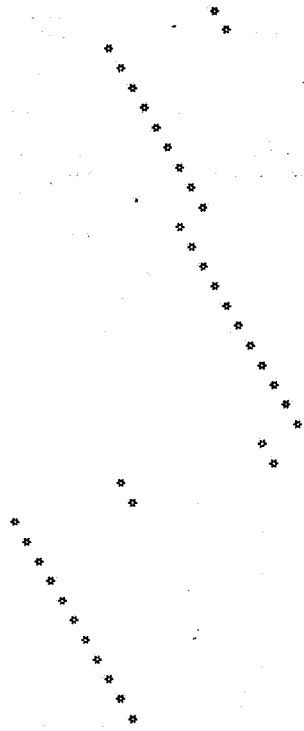
5 \* 7 + 4

5 \* 9 + 6

```

964 7 8 * 7 * 17 + 12      44 .
965 = 8 * 7 * 17 + 13      45 .
966 = 23 * 2 * 3 * 7 + 0   75 .
967 = 23 * 2 * 3 * 7 + 1   76 .
968 = 23 * 2 * 3 * 7 + 2   37 .
969 = 23 * 2 * 3 * 7 + 3   38 .
970 = 23 * 2 * 3 * 7 + 4   39 .
971 = 23 * 2 * 3 * 7 + 5   40 .
972 7 23 * 2 * 3 * 7 + 6   41 .
973 = 23 * 2 * 3 * 7 + 7   42 .
974 = 23 * 2 * 3 * 7 + 8   43 .
975 = 25 * 3 * 13 + 0       41 .
976 = 25 * 3 * 13 + 1       42 .
977 = 25 * 3 * 13 + 2       43 .
978 = 25 * 3 * 13 + 3       44 .
979 = 25 * 3 * 13 + 4       45 .
980 = 25 * 3 * 13 + 5       46 .
981 = 25 * 3 * 13 + 6       47 .
982 = 25 * 3 * 13 + 7       48 .
983 = 25 * 3 * 13 + 8       49 .
984 = 25 * 3 * 13 + 9       50 .
985 = 25 * 3 * 13 + 10      51 .
986 = 29 * 2 * 17 + 0       48 .
987 = 29 * 2 * 17 + 1       49 .
988 = 4 * 13 * 19 + 0       76 .
989 = 4 * 13 * 19 + 1       77 .
990 = 5 * 2 * 9 * 11 + 0    27 .
991 = 5 * 2 * 9 * 11 + 1    28 .
992 = 5 * 2 * 9 * 11 + 2    29 .
993 = 5 * 2 * 9 * 11 + 3    30 .
994 = 5 * 2 * 9 * 11 + 4    31 .
995 = 5 * 2 * 9 * 11 + 5    32 .
996 = 5 * 2 * 9 * 11 + 6    33 .
997 = 5 * 2 * 9 * 11 + 7    34 .
998 = 5 * 2 * 9 * 11 + 8    35 .
999 = 5 * 2 * 9 * 11 + 9    36 .
1000 = 5 * 2 * 9 * 11 + 10   37 .
      END OF ALGOL RUN *V3.0*

```



```

11/08/72 *GRONINGEN SCOPE 3.3 SUM 279 24/01/72
12.27.32.FRANK4F
12.27.32.FRANK.MR9500.
12.27.32.ALGOL.
12.27.33.FILE OPENED---OUTPUT
12.27.34.FILE OPENED---LGO
12.27.42. 1 862
12.27.42.LGO.
12.28.00.CP 014.873 SEC.
12.28.00.PP 014.434 SEC.

```

B2458V.11,FRANK GOOSSENS,R50

```

1  'BEGIN'
2  'INTEGER' N,NPRINTMIN,PBASE
3  N:=READ,, NPRINTMIN:=READ,, PBASE:=0,,
4  'BEGIN'
5
6  'PROCEDURE' SEARCH(A,B), 'VALUE' A,B, 'INTEGER' A,B,,
7  'BEGIN'
8  'IF' NUM(/A/) 'GREATER' 0 'THEN' INVESTIGATE(A) 'ELSE'
9  SEARCH(REFFIRST(/A/),REFSEC(/A/)),
10 'IF' NUM(/B/) 'GREATER' 0 'THEN' INVESTIGATE(B) 'ELSE'
11 SEARCH(REFFIRST(/B/),REFSEC(/B/))
12 'END' SEARCH,,
13
14 'PROCEDURE' INVESTIGATE(A), 'VALUE' A, 'INTEGER' A,,
15 'BEGIN' 'INTEGER' PHELP,, PHELP:=PBASE,,
16 LPINV, 'IF' PHELP 'LESS' P 'THEN'
17 'BEGIN' 'IF' NUM(/STACK(/PHELP/)) = NUM(/A/) 'THEN'
18 'BEGIN' MIX,='TRUE',,
19 STACK(/PHELP/).:=STACK(/PHELP/)*A
20 'END' 'ELSE'
21 'BEGIN' PHELP,=PHELP+1,,
22 'GOTO' LPINV
23 'END'
24 'END' 'ELSE'
25 'BEGIN' STACK(/P/).:=A,, P,=P+1 'END'
26 'END' INVESTIGATE,,
27
28 'PROCEDURE' HEADER,,
29 'BEGIN'
30 NLCR,,
31 PRINTTEXT('('NUMBER=PRODUCT*INTEGER SUM)'),,
32 'END' HEADER,,
33
34 'PROCEDURE' OUT(N,ARRAY,POINTER,ADDITIONAL,SUM),,
35 'VALUE' N,POINTER,ADDITIONAL,SUM,,
36 'INTEGER' N,POINTER,ADDITIONAL,SUM, 'ARRAY' ARRAY,,
37 'BEGIN'
38 'IF' N 'LESS' NPRINTMIN 'THEN' 'GOTO' ENDOUT,,
39 NLCR,,
40 ABSFIXT(4,0,N),,
41 PRSYM(70), 'COMMENT' EQUALSIGN,,
42 ABSFIXT(2,0,ARRAY(/PBASE/)),,
43 'FOR' I,=PBASE+1 'STEP' 1 'UNTIL' POINTER=1 'DO'
44 'BEGIN' PRSYM(66), 'COMMENT' STARSIGN,,
45 ABSFIXT(2,0,ARRAY(/I/)),,
46 'END',,
47 PRSYM(64), 'COMMENT' PLUSIGN,,
48 ABSFIXT(2,0,ADDITIONAL),,
49 CARRIAGE(0), SPACE(35), ABSFIXT(2,0,SUM),,
50 PRSYM(88), 'COMMENT' POINTSIGN,,
51 SPACE(SUM), PRSYM(66), 'COMMENT' STARSIGN,,
52 ENDOUT,,
53 'END' OUT,,
54
55 'PROCEDURE' OUTD(N,ARRAY,POINTER,ADDITIONAL),,

```

```

56 'VALUE' N, POINTER, ADDITIONAL,, 'INTEGER' N, POINTER, ADDITIONAL,,
57 'ARRAY' ARRAY,,
58 'BEGIN'
59 'IF' N 'LESS' NPRINTMIN 'THEN' 'GOTO' ENDOUT,,
60 CARRIAGE(0),,SPACE(100),,
61 ABSFIXT(2,0,ARRAY(/PBASE/)),,
62 'FOR' I,=PBASE+1 'STEP' 1 'UNTIL' POINTER-1 'DO'
63 'BEGIN' PRSYM(66),,ABSFIXT(2,0,ARRAY(/I/)) 'END',,
64 PRSYM(64),,ABSFIXT(2,0,ADDITIONAL),,
65 ENDOUT,,
66 'END' OUTD,,
67
68 'INTEGER' ARRAY' NUM, REFFIRST, REFSEC(/2..N/),,
69 'INTEGER' ARRAY' STACK, LIST(/PBASE..10/),,
70 'INTEGER' P, Q, SUM, NUMBER, MINSUM, MINSUMNUMBER, DIFFMINSUM
71 , DUMMY, I, J, RELFIRST, RELSEC, RELPROD
72 ,,
73 'BOOLEAN' MIX,,
74
75
76 'COMMENT'      INITIALIZATION OF THE ARRAYS NUM, REFFIRST, REFSEC
77                NUM/ IF INDEX IS PRIME THEN CONTENTS IS INDEX
78                IF INDEX IS APOWER OF PRIME THEN CONTENTS IS
79                THAT PRIME
80                IF INDEX IS NOT ONE OF ABOVE MENTIONED THEN
81                CONTENTS IS NEGATIVE
82
83                THE ARRAYS REFFIRST, REFSEC ARE NOT INITIALIZED
84
85 'FOR' I,=2 'STEP' 1 'UNTIL' N 'DO'
86   NUM(/I/),=1,,
87 'FOR' I,=2 'STEP' 1 'UNTIL' N 'DO'
88   'IF' NUM(/I/) 'NOT LESS' 1 'THEN'
89   'BEGIN' 'FOR' J,=I*2 'STEP' 1 'UNTIL' N 'DO'
90     NUM(/J/),=-1,,
91     J,=I,,
92     'FOR' J,=J*1 'WHILE' J 'NOT GREATER' N 'DO'
93       NUM(/J/),=1
94     'END',,
95
96
97 HEADER,,
98
99
100 'COMMENT'      NOW WE ANALYZE ARRAY NUM
101
102   MINSUM,=6,,
103   MINSUMNUMBER,=5,,
104   NUMBER,=5,, 'COMMENT' THE NUMBERS 1 TO 6 ARE NOT ANALYZED,,
105 NEXT,,
106   NUMBER,=NUMBER+1,, P,=PBASE,,
107   'IF' NUMBER 'GREATER' N 'THEN' 'GOTO' FINISH,,
108   'IF' NUM(NUMBER/) 'GREATER' 0 'THEN'
109   'BEGIN' SUM,=NUMBER,, 'GOTO' COMPARE 'END' ,,
110   RELFIRST,=ENTIER(SQRT(NUMBER)),,
111   RELSEC,=RELFIRST+1,,
112   RELPROD,=RELFIRST*RELSEC,,
113   'FOR' DUMMY,=0 'WHILE' RELPROD 'NOT EQUAL' NUMBER 'DO'
114   'IF' RELPROD 'LESS' NUMBER 'THEN'
115   'BEGIN'
     RELPROD,=RELPROD*RELFIRST,,

```

```

116         RELSEC.=RELSEC+1
117         'END' 'ELSE'
118         'BEGIN'
119             RELPROD.=RELPROD-RELSEC.,
120             RELFIRST.=RELFIRST-1
121         'END'.,
122
123
124
125
126         'COMMENT'      WE HAVE NOW TWO POSSIBLE RELATIVE PRIMES
127         SEARCH THE SIMPLEST COMBINATION.,
128     MIX,='FALSE'.,
129     SEARCH(RELFIRST,RELSEC),,
130
131
132     'COMMENT'      THE STACK IS FILLED WITH THE BEST COMBINATION.,
133     'COMMENT'      CALCULATE THE SUM OF THE RELATIVE PRIMES.,
134     SUM.=0.,
135     'FOR' I,=PBASE 'STEP' 1 'UNTIL' P-1 'DO'
136     SUM,=SUM+STACK(/I/),,
137     COMPARE.,
138
139     DIFFMINSUM,=NUMBER-MINSUMNUMBER.,
140
141     'IF' SUM 'GREATER' MINSUM+DIFFMINSUM 'THEN'
142     'BEGIN'
143         OUT(NUMBER,LIST,Q,DIFFMINSUM,MINSUM+DIFFMINSUM),,
144     'END' 'ELSE'
145     'IF' SUM = MINSUM+DIFFMINSUM 'THEN'
146     'BEGIN'
147         OUT(NUMBER,STACK,P,0,SUM),,
148         OUT(NUMBER,LIST,Q,DIFFMINSUM),,
149     'END' 'ELSE'
150     'BEGIN' OUT(NUMBER,STACK,P,0,SUM),,
151             'FOR' I,=PBASE 'STEP' 1 'UNTIL' P-1 'DO'
152             LIST(/I/),=STACK(/I/),,
153             Q,=P., MINSUM,=SUM., MINSUMNUMBER,=NUMBER.,
154         'END'.,
155
156     'COMMENT'      BOOLEAN MIX IS TRUE WHEN IN THE ANALYSIS
157     OF RELFIRST AND RELSEC OCCUR THE SAME PRIMES AND
158     POWERSOF PRIMES.,
159
160     'IF' MIX 'THEN'
161     'BEGIN'
162     LPMIX.. 'IF' P'NOT GREATER' PBASE+2 'THEN' 'GOTO' RDMIX.,
163             STACK(/P/),=1.,J,=PBASE.,
164             'FOR' I,=PBASE 'STEP' 2 'UNTIL' P-1 'DO'
165             'BEGIN' STACK(/J/),=STACK(/I/)*STACK(/I+1/),,
166                     J,=J+1.,
167             'END'.,
168             P,=J., 'GOTO' LPMIX.,
169     RDMIX..
170             REFFIRST(/NUMBER/),=STACK(/PBASE/),,
171             REFSEC(/NUMBER/),=STACK(/PBASE+1/),,
172     'END' 'ELSE'
173     'BEGIN' REFFIRST(/NUMBER/),=RELFIRST.,
174             REFSEC(/NUMBER/),=RELSEC.,
175     'END'.,

```

090872-122 . . . B 2458V.11 FRANKGOOSSENS

4

176. 'GOTO' NEXT  
177 'END',,  
178 FINISH.,  
179 'END'



| NUMBER | PRODUCT | INTEGER | SUM |
|--------|---------|---------|-----|
| 6      | 2 * 3   | 0       | 5   |
| 7      | 2 * 3   | 1       | 6   |
| 8      | 2 * 3   | 2       | 7   |
| 9      | 2 * 3   | 3       | 8   |
| 10     | 2 * 5   | 0       | 7   |
| 11     | 2 * 5   | 1       | 8   |
| 12     | 3 * 4   | 0       | 7   |
| 13     | 3 * 4   | 1       | 8   |
| 14     | 2 * 7   | 0       | 9   |
| 15     | 3 * 5   | 0       | 8   |
| 16     | 3 * 5   | 1       | 9   |
| 17     | 3 * 5   | 2       | 10  |
| 18     | 9 * 2   | 0       | 11  |
| 19     | 3 * 5   | 4       | 12  |
| 20     | 4 * 5   | 0       | 9   |
| 21     | 3 * 7   | 0       | 10  |
| 22     | 4 * 5   | 2       | 11  |
| 23     | 4 * 5   | 3       | 12  |
| 24     | 8 * 3   | 0       | 11  |
| 25     | 8 * 3   | 1       | 12  |
| 26     | 8 * 3   | 2       | 13  |
| 27     | 8 * 3   | 3       | 14  |
| 28     | 4 * 7   | 0       | 11  |
| 29     | 4 * 7   | 1       | 12  |
| 30     | 5 * 2   | 3       | 10  |
| 31     | 5 * 2   | 3       | 11  |
| 32     | 5 * 2   | 3       | 12  |
| 33     | 5 * 2   | 3       | 13  |
| 34     | 5 * 2   | 3       | 14  |
| 35     | 5 * 7   | 0       | 12  |
| 36     | 4 * 9   | 0       | 13  |
| 37     | 5 * 7   | 2       | 14  |
| 38     | 5 * 7   | 3       | 15  |
| 39     | 3 * 13  | 0       | 16  |
| 40     | 5 * 8   | 0       | 13  |
| 41     | 5 * 8   | 1       | 14  |
| 42     | 2 * 3   | 7       | 12  |
| 43     | 2 * 3   | 7       | 13  |
| 44     | 2 * 3   | 7       | 14  |
| 45     | 5 * 9   | 0       | 14  |
| 46     | 5 * 9   | 1       | 15  |
| 47     | 5 * 9   | 2       | 16  |
| 48     | 5 * 9   | 3       | 17  |
| 49     | 5 * 9   | 4       | 18  |
| 50     | 5 * 9   | 5       | 19  |
| 51     | 5 * 17  | 0       | 20  |
| 52     | 4 * 13  | 0       | 17  |
| 53     | 4 * 13  | 1       | 18  |
| 54     | 4 * 13  | 2       | 19  |
| 55     | 5 * 11  | 0       | 16  |
| 56     | 7 * 8   | 0       | 15  |
| 57     | 7 * 8   | 1       | 16  |
| 58     | 7 * 8   | 2       | 17  |
| 59     | 7 * 8   | 3       | 18  |
| 60     | 4 * 3   | 5       | 12  |
| 61     | 4 * 3   | 5       | 13  |
| 62     | 4 * 3   | 5       | 14  |
| 63     | 4 * 3   | 5       | 15  |



3 \* 4 + 2

3 \* 5 + 3

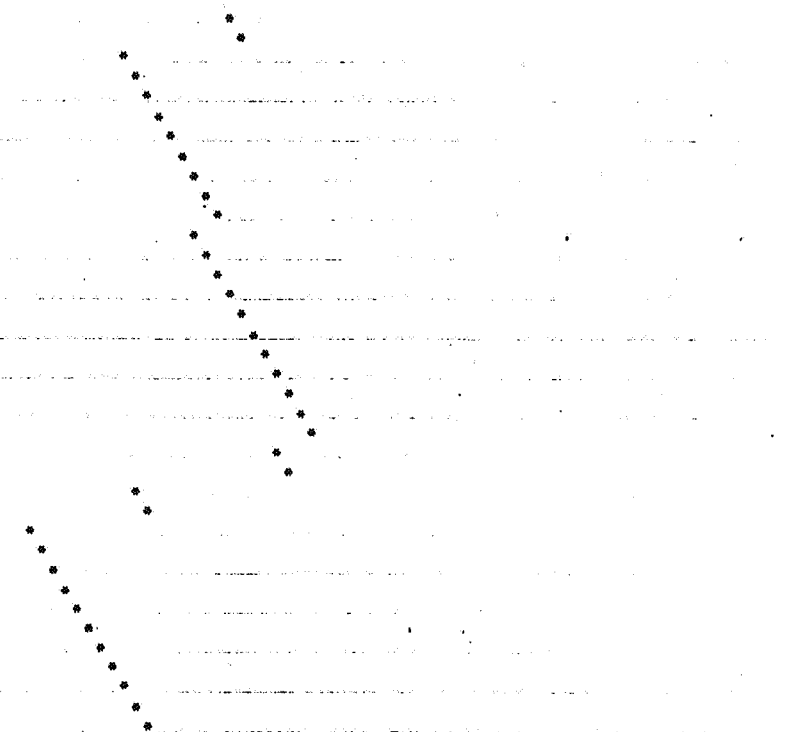
4 \* 5 + 1

5 \* 7 + 1

5 \* 7 + 4

5 \* 9 + 6

964 = 8 \* 7 \* 17 + 12 44  
 965 = 8 \* 7 \* 17 + 13 45  
 966 = 23 \* 2 \* 3 \* 7 + 0 35  
 967 = 23 \* 2 \* 3 \* 7 + 1 36  
 968 = 23 \* 2 \* 3 \* 7 + 2 37  
 969 = 23 \* 2 \* 3 \* 7 + 3 38  
 970 = 23 \* 2 \* 3 \* 7 + 4 39  
 971 = 23 \* 2 \* 3 \* 7 + 5 40  
 972 = 23 \* 2 \* 3 \* 7 + 6 41  
 973 = 23 \* 2 \* 3 \* 7 + 7 42  
 974 = 23 \* 2 \* 3 \* 7 + 8 43  
 975 = 25 \* 3 \* 13 + 0 41  
 976 = 25 \* 3 \* 13 + 1 42  
 977 = 25 \* 3 \* 13 + 2 43  
 978 = 25 \* 3 \* 13 + 3 44  
 979 = 25 \* 3 \* 13 + 4 45  
 980 = 25 \* 3 \* 13 + 5 46  
 981 = 25 \* 3 \* 13 + 6 47  
 982 = 25 \* 3 \* 13 + 7 48  
 983 = 25 \* 3 \* 13 + 8 49  
 984 = 25 \* 3 \* 13 + 9 50  
 985 = 25 \* 3 \* 13 + 10 51  
 986 = 29 \* 2 \* 17 + 0 48  
 987 = 29 \* 2 \* 17 + 1 49  
 988 = 4 \* 13 \* 19 + 0 36  
 989 = 4 \* 13 \* 19 + 1 37  
 990 = 5 \* 2 \* 9 \* 11 + 0 27  
 991 = 5 \* 2 \* 9 \* 11 + 1 28  
 992 = 5 \* 2 \* 9 \* 11 + 2 29  
 993 = 5 \* 2 \* 9 \* 11 + 3 30  
 994 = 5 \* 2 \* 9 \* 11 + 4 31  
 995 = 5 \* 2 \* 9 \* 11 + 5 32  
 996 = 5 \* 2 \* 9 \* 11 + 6 33  
 997 = 5 \* 2 \* 9 \* 11 + 7 34  
 998 = 5 \* 2 \* 9 \* 11 + 8 35  
 999 = 5 \* 2 \* 9 \* 11 + 9 36  
 1000 = 5 \* 2 \* 9 \* 11 + 10 37



30

3 13 25  
 2460656 2460804  
 1150 4275

Memory for data [2;2-37]

|       | <u>Bits per word</u> | <u>Core memory</u> | <u>Number of words</u> |
|-------|----------------------|--------------------|------------------------|
| SARA: | 60                   |                    | 94K                    |
| MC:   | 27                   |                    | 64K                    |

|                      | <u>Number of core words used</u> |                       |
|----------------------|----------------------------------|-----------------------|
| <u>Data type</u>     | <u>MC</u>                        | <u>CDC</u>            |
| <u>integer</u>       | 1                                | 1                     |
| <u>real</u>          | 2                                | 1                     |
| <u>Boolean</u>       | 1                                | 1                     |
| <u>Boolean array</u> | 27 elements per word             | 1 element per word    |
| <u>string</u>        | 3 characters per word            | 8 characters per word |

```

00** "BEGIN""BOOLEAN""ARRAY" B(/1:80000/);
      "INTEGER" I;
      "FOR" I:=1 "STEP" 1 "UNTIL" 80000 "DO" B(/I):="TRUE"
"END"
      FINIS GEN.BY EOR-CARD

```

|      |   |                  |           |   |
|------|---|------------------|-----------|---|
| LINE | 0 | PROGRAM BEGINS   | (MESSAGE) | 1 |
| LINE | 3 | PROGRAM ENDS     | (MESSAGE) | 1 |
| LINE | 3 | SOURCE DECK ENDS | (MESSAGE) | 1 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.A

STACK OVERFLOW

THIS ERROR OCCURRED AFTER LINE 0 IN XXALGOL

THE GLOBAL VARIABLES ARE ..

|          |        |      |        |      |        |
|----------|--------|------|--------|------|--------|
| UA VALUE | 000342 | 2000 | 000000 | 0000 | 000000 |
| UV       | 000000 | 2000 | 000000 | 0000 | 000000 |
| LASTUSED | 011233 | 6144 | 777775 | 0446 | 011240 |

IN THE BLOCK ENTERED AT LINE 0 IN XXALGOL

THE LOCAL VARIABLES ARE ..

|        |        |      |        |      |        |                       |                      |                      |        |
|--------|--------|------|--------|------|--------|-----------------------|----------------------|----------------------|--------|
| 000004 | 011226 | 5757 | 000011 | 0000 | 011222 | +2.47725389H0596"-085 | +8.000000000000"+005 | +1.000000000000"+000 |        |
| 000010 | 011232 | 0000 | 000000 | 0000 | 000001 | 6144                  | 777775               | 0446                 | 011240 |

THIS BLOCK WAS CALLED FROM LINE 0 IN ALGORUN

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72

- 12.06.53.VELDH3L
- 12.06.53.VELDH.NR9500.
- 12.06.53.ALGOL.
- 12.06.56.FILE OPENED---OUTPUT
- 12.07.02.FILE OPENED---LGO
- 12.07.05. 0 158
- 12.07.05.LGO.
- 12.07.12.CP 600.258 SEC.
- 12.07.12.PP 013.721 SEC.

090872-100 A 2458V.7 JACKALANEN

A2458V.7, JACK ALANEN

```
'BEGIN' 'BOOLEAN' 'ARRAY' B[1:80000]; 'INTEGER' I;  
1      'FOR' I:=1 'STEP' 1000 'UNTIL' 80000 'DO' B[I]:='TRUE';  
2      PRINTTEXT("80000 ELEMENT BOOLEAN ARRAY NOT TOO LARGE")  
3      'END'
```

090872-100 A 2458V.7 JACKALANEN

2

80000 ELEMENT BOOLEAN ARRAY NOT TOO LARGE

090872-100 A 2458V.7 JACKALANEN

3

|       |       |    |
|-------|-------|----|
| 0     | 1     | 25 |
| 94184 | 94184 |    |
| 85    | 29748 |    |

Precision of numbers [2;2-37]

For numbers of type real, the precisions are:

|          | <u>Number of bits</u> |                     |
|----------|-----------------------|---------------------|
|          | <u>MC ALGOL 60</u>    | <u>CDC ALGOL 60</u> |
| sign     | 1                     | 1                   |
| exponent | 11 (+1 for sign)      | 10 (+1 for sign)    |
| fraction | 40                    | 48                  |

|  | <u>Number of significant digits</u> |                     |
|--|-------------------------------------|---------------------|
|  | <u>MC ALGOL 60</u>                  | <u>CDC ALGOL 60</u> |
|  | 12 (sometimes 13)                   | 14 (sometimes 15)   |

Range of a nonzero real number y

CDC ALGOL 60:  $3.1 \times 10^{-294} \approx 2^{-1022} (2^{48} - 1) < |y| < 2^{1022} (2^{48} - 1) \approx 1.3 \times 10^{322}$

MC ALGOL 60 :  $10^{-616} < |y| < 10^{628}$

Thus there are some small and large real numbers, such as  $10^{\pm 400}$ , which can be expressed in MC ALGOL 60 but not in CDC ALGOL 60.

For numbers of type integer, the CDC ALGOL 60 uses the same representation as for reals:

Exact range for an integer number i

CDC ALGOL 60:  $|i| \leq 2^{48} - 1 = 281, 474, 976, 710, 655$

MC ALGOL 60 :  $|i| \leq 2^{26} - 1 = 67, 108, 863$

00\*\* "BEGIN" "HEAL" SMALL;  
SMALL:=1.234"-400:OUTREAL(61,SMALL);  
"END"  
FINIS GEN.BY EOR-CARD

|      |   |                  |           |    |
|------|---|------------------|-----------|----|
| LINE | 0 | PROGRAM BEGINS   | (MESSAGE) | 1  |
| LINE | 2 | PROGRAM ENDS     | (MESSAGE) | 1  |
| LINE | 2 | SOURCE DECK ENDS | (MESSAGE) | -1 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

\*0.000000000000000000\*+000  
END OF ALGOL RUN \*V3.0\*

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72  
 14.15.17.VELDHIL  
 14.15.17.VELDH+NR9500.  
 14.15.17.ALGOL.  
 14.15.24.FILE OPENED---OUTPUT  
 14.15.27.FILE OPENED---LGO  
 14.15.28. 0 112  
 14.15.28.LGO.  
 14.15.32.CP 000.213 SEC.  
 14.15.32.PP 005.955 SEC.

0000 "BEGIN" "REAL" "LARGE";  
LARGE:=1.234"400"OUTREAL(6),LARGE);  
"END"  
FINIS GEN.BY EOR-CARD

LINE 0 PROGRAM BEGINS (MESSAGE) 1  
LINE 2 PROGRAM ENDS (MESSAGE) 1  
LINE 2 SOURCE DECK ENDS (MESSAGE) -1  
LINE 1 ARITHMETIC OVERFLOW OF NUMBER/EXPRESSION 2

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

08/08/72 GRONINGEN SCOPE 3.3 SUM -279 24/01/72  
14.15.07.VELDM1K  
14.15.07.VELDM,NR9500.  
14.15.07.ALGOL.  
14.15.14.FILE OPENED---OUTPUT  
14.15.16.FILE OPENED---L60  
14.15.17. 0 36  
14.15.17.CP 000.037 SEC.  
14.15.17.PP 003.667 SEC.



090872- 24 ... A 2458V.4 JACKALANEN 1

A2458V.4, JACK ALANEN

```
1 'BEGIN' 'REAL' SMALL,LARGE; SMALL:=1.234e-400; PRINT(SMALL);  
2 'END' LARGE:=1.234e400; PRINT(LARGE)
```

090872- 24 ... A 2458V.4 JACKALANEN 2

+1.2339999999999999e-399 +.123400000000000e+401

090872- 24 ... A 2458V.4 JACKALANEN 3

|       |       |    |
|-------|-------|----|
| 0     | 1     | 25 |
| 54643 | 54643 |    |
| 32    | 54    |    |

Integer labels [2;2-25]

CDC ALGOL 60: Integer labels are not permitted.

MC ALGOL 60 : The value of integer labels must be  $\leq$  67108863.

00\*\* "BEGIN" 39:"GOTO"39"END"  
FINIS GEN.BY EOR-CARD

|      |   |                                 |           |   |
|------|---|---------------------------------|-----------|---|
| LINE | 0 | PROGRAM BEGINS                  | (MESSAGE) | 1 |
| LINE | 0 | PROGRAM ENDS                    | (MESSAGE) | 1 |
| LINE | 0 | SOURCE DECK ENDS                | (MESSAGE) | 1 |
| LINE | 0 | .. FOLLOWS AN INCORRECT OPERAND |           | 3 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F,L,X

08/08/72 GROWINGEN SCOPE 3.3 SUM 279 24/01/72  
 12.07.59.VELDH3N  
 12.07.59.VELDH.NR9500.  
 12.07.59.ALGOL.  
 12.08.00.FILE OPENED---OUTPUT  
 12.08.03.FILE OPENED---LGO  
 12.08.04. 0 26  
 12.08.04.CP 000.027 SEC.  
 12.08.04.PP 003.921 SEC.

Formal and actual parameters [2;2-35,2-42,2-43]

CDC ALGOL 60: Specifications of all formal parameters must be supplied. A label cannot be specified by value. A maximum of 63 parameters (and no more than 62 constants as actual parameters) are allowed for each procedure.

```

00** "BEGIN" "PROCEDURE" P(A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14,A15
      A16,A17,A18,A19,A20,A21,A22,A23,A24,A25,A26,A27,
      A28,A29,A30,A31,A32,A33,A34,A35,A36);
      "INTEGER" A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14,A15,A16
      A17,A18,A19,A20,A21,A22,A23,A24,A25,A26,A27,A28,A29,
      A30,A31,A32,A33,A34,A35,A36;
      A1:=A2;

```

```

      P(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
      24,25,26,27,28,29,30,31,32,33,34,35,36);

```

```

10** "END"
      FINIS GEN.BY EOR-CARD

```

41

|      |    |                  |           |   |
|------|----|------------------|-----------|---|
| LINE | 0  | PROGRAM BEGINS   | (MESSAGE) | 1 |
| LINE | 10 | PROGRAM ENDS     | (MESSAGE) | 1 |
| LINE | 10 | SOURCE DECK ENDS | (MESSAGE) | 1 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F,L,X

PARAMETER KIND ERROR  
THIS ERROR OCCURRED AFTER LINE 0 IN XXALGOL

THE GLOBAL VARIABLES ARE ..  
UA+VALUE 000340 5716 000000 0000 000043  
UV 000000 5716 000000 0000 000043  
LASTUSED 011346 -2.29715068914800+195

IN THE BLOCK ENTERED AT LINE 0 IN XXALGOL

THE FORMAL VARIABLES ARE ..  
000110 011227 +3.5000000000000000+001  
000107 011230 +3.5000000000000000+001  
000106 011231 +3.4000000000000000+001  
000105 011232 +3.3000000000000000+001  
000104 011233 +3.2000000000000000+001  
000103 011234 +3.1000000000000000+001  
000102 011235 +3.0000000000000000+001  
000101 011236 +2.9000000000000000+001  
000100 011237 +2.8000000000000000+001  
000077 011240 +2.7000000000000000+001  
000076 011241 +2.6000000000000000+001  
000075 011242 +2.5000000000000000+001  
000074 011243 +2.4000000000000000+001  
000073 011244 +2.3000000000000000+001  
000072 011245 +2.2000000000000000+001  
000071 011246 +2.1000000000000000+001  
000070 011247 +2.0000000000000000+001  
000067 011250 +1.9000000000000000+001  
000066 011251 +1.8000000000000000+001  
000065 011252 +1.7000000000000000+001  
000064 011253 +1.6000000000000000+001  
000063 011254 +1.5000000000000000+001  
000062 011255 +1.4000000000000000+001  
000061 011256 +1.3000000000000000+001  
000060 011257 +1.2000000000000000+001  
000057 011260 +1.1000000000000000+001  
000056 011261 +1.0000000000000000+001  
000055 011262 +9.0000000000000000+000  
000054 011263 +8.0000000000000000+000  
000053 011264 +7.0000000000000000+000  
000052 011265 +6.0000000000000000+000  
000051 011266 +5.0000000000000000+000  
000050 011267 +4.0000000000000000+000  
000047 011270 +3.0000000000000000+000  
000046 011271 +2.0000000000000000+000  
000045 011272 +1.0000000000000000+000  
000044 011273 5716 000000 0000 011227  
000043 011274 5716 000000 0000 011230  
000042 011275 5716 000000 0000 011231  
000041 011276 5716 000000 0000 011232  
000040 011277 5716 000000 0000 011233  
000037 011300 5716 000000 0000 011234  
000036 011301 5716 000000 0000 011235  
000035 011302 5716 000000 0000 011236  
000034 011303 5716 000000 0000 011237  
000033 011304 5716 000000 0000 011240  
000032 011305 5716 000000 0000 011241  
000031 011306 5716 000000 0000 011242

|        |        |      |        |      |        |
|--------|--------|------|--------|------|--------|
| 000030 | 011307 | 5716 | 000000 | 0000 | 011243 |
| 000027 | 011310 | 5716 | 000000 | 0000 | 011244 |
| 000026 | 011311 | 5716 | 000000 | 0000 | 011245 |
| 000025 | 011312 | 5716 | 000000 | 0000 | 011246 |
| 000024 | 011313 | 5716 | 000000 | 0000 | 011247 |
| 000023 | 011314 | 5716 | 000000 | 0000 | 011250 |
| 000022 | 011315 | 5716 | 000000 | 0000 | 011251 |
| 000021 | 011316 | 5716 | 000000 | 0000 | 011252 |
| 000020 | 011317 | 5716 | 000000 | 0000 | 011253 |
| 000017 | 011320 | 5716 | 000000 | 0000 | 011254 |
| 000016 | 011321 | 5716 | 000000 | 0000 | 011255 |
| 000015 | 011322 | 5716 | 000000 | 0000 | 011256 |
| 000014 | 011323 | 5716 | 000000 | 0000 | 011257 |
| 000013 | 011324 | 5716 | 000000 | 0000 | 011260 |
| 000012 | 011325 | 5716 | 000000 | 0000 | 011261 |
| 000011 | 011326 | 5716 | 000000 | 0000 | 011262 |
| 000010 | 011327 | 5716 | 000000 | 0000 | 011263 |
| 000007 | 011330 | 5716 | 000000 | 0000 | 011264 |
| 000006 | 011331 | 5716 | 000000 | 0000 | 011265 |
| 000005 | 011332 | 5716 | 000000 | 0000 | 011266 |
| 000004 | 011333 | 5716 | 000000 | 0000 | 011267 |
| 000003 | 011334 | 5716 | 000000 | 0000 | 011270 |
| 000002 | 011335 | 5716 | 000000 | 0000 | 011271 |
| 000001 | 011336 | 5716 | 000000 | 0000 | 011272 |
| 000000 | 011337 | 0044 | 776442 | 4111 | 011222 |

THE LOCAL VARIABLES ARE ..

THIS BLOCK WAS CALLED FROM LINE 8 IN XXALGOL

IN THE BLOCK ENTERED AT LINE 0 IN XXALGOL

THE LOCAL VARIABLES ARE ..

000004 011226 1747 776707 4000 011222

THIS BLOCK WAS CALLED FROM LINE 0 IN ALGORUN

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72  
 12.08.04.VELDH30  
 12.08.05.VELDH.NR9500.  
 12.08.05.ALGOL.  
 12.08.05.FILE OPENED----OUTPUT  
 12.08.08.FILE OPENED----LGO  
 12.08.10. 0 305  
 12.08.10.LGO.  
 12.08.20.CP 000.434 SEC.  
 12.08.20.PP 010.307 SEC.

Subscript checking [2;2-15]

CDC ALGOL 60: Upper and lower subscript bounds are never checked. However, with option F (default active) a run-time check will be made to ensure that the final address calculated for an array element is in fact within the range for all elements of that array; in other words, if integer array  $A[1:3,1:2]$  is stored in locations 103 to 108 inclusive, then a reference to  $A[0,3]$  is always permitted because it corresponds to address 103 ( $A[i,j] \leftrightarrow 103 + (j-1) + (i-1) \times 2$ ) which falls in the range [103,108] even though both subscripts lie outside their declared ranges. Clearly, for arrays with only one subscript this array bound check is equivalent to a subscript bound check.

MC ALGOL 60: Upper and lower subscript bounds are always checked.



```

00** "BEGIN""INTEGER""ARRAY"A(/I;3;112/);
      "INTEGER" I,J;
      "FOR" I:=1"STEP"1"UNTIL"3"DO"
      "FOR" J:=1"STEP"1"UNTIL"2"DO"
      A(/I,J/):=0;
      A(/0,3/):=1;
      "FOR" I:=1"STEP"1"UNTIL"3"DO"
      "FOR" J:=1"STEP"1"UNTIL"2"DO"
      OUTPUT(61,"(/;0;320;320)",A(/I,J/),I,J)
"END"
10** FINIS GEN.BY FOR-CARD

```

```

LINE 0 PROGRAM BEGINS (MESSAGE) 1
LINE 9 PROGRAM ENDS (MESSAGE) 1
LINE 9 SOURCE DECK ENDS (MESSAGE) 1

```

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

```

1 1 1
0 1 2
0 2 1
0 2 2
0 3 1
0 3 2
END OF ALGOL RUN. *V3.0*

```

```

08/08/72 GRONINGEN SCOPE-3.3 SUM 279 24/01/72
12.08.20.VELUH3U
12.08.20.VELUH.NH9500. VELDHUYZEN
12.08.20.ALGOL.
12.08.21.FILE OPENED---OUTPUT
12.08.20.FILE OPENED---LG0
12.08.28. 0 280
12.08.29.LG0.
12.08.34.CP 000.444 SFC.
12.08.34.PP 009.647 SFC.

```

54

090872- 26 A 2458V.6 JACKALANEN 1

A2458V.6,JACK ALANEN

```
1 'BEGIN' 'INTEGER' 'ARRAY' A[1:3,1:2];  
2 'END' A[0,3]:=1
```

090872- 26 A 2458V.6 JACKALANEN 2

ER 502 1 -0

090872- 26 A 2458V.6 JACKALANEN 3

```
0 1 25  
38911 38911  
40 86
```

Subscripted controlled variable [2;2-32]

CDC ALGOL 60: In the case of an array element used as the controlled variable in a for statement,

```
i:= 1; for A[i]:= 0 while i < n do i:= i+1;
```

the same element is always used (i.e., the element referenced by the initial values of the subscripts) regardless of any changes that might occur. In the above example, only A[1] is set to zero.

```

00** "BEGIN""INTEGER""ARRAY"(1:10);
      "INTEGER" I;
      E(2/):=I:=1;
      "FOR"(1/):=0"WHILE"1<10"DO" I:=I+1;
      OUTREAL(61,E(2/))
"END"
      FINIS GEN.BY EOK-CARD

```

|      |   |                  |           |   |
|------|---|------------------|-----------|---|
| LINE | 0 | PROGRAM BEGINS   | (MESSAGE) | 1 |
| LINE | 5 | PROGRAM ENDS     | (MESSAGE) | 1 |
| LINE | 5 | SOURCE DECK ENDS | (MESSAGE) | 1 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F,L,X

\*1.00000000000000000000\*000  
END OF ALGOL RUN \*V3.0\*

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72  
12.08.39.VELDM3R  
12.08.39.VELDH,NR9500. VELDHUYZEN  
12.08.39.ALGOL.  
12.08.40.FILE OPENED---OUTPUT  
12.08.43.FILE OPENED---LG0  
12.08.44. 0 193  
12.08.44.LG0.  
12.08.49.CP 000.295 SEC.  
12.08.49.PP 006.711 SEC.

090872- 25 ... A 2458V.5 JACKALANEN 1

A2458V.5,JACK ALANEN

```
1 'BEGIN' 'INTEGER' 'ARRAY' E(1:10) 'INTEGER' I;  
2 E(2):=I:=1;  
3 'FOR' E(1):=0 'WHILE' I<10 'DO' I:=I+1;  
4 'END' PRINT(E(2))
```

090872- 25 ... A 2458V.5 JACKALANEN 2

+0

090872- 25 ... A 2458V.5 JACKALANEN 3

|       |       |    |
|-------|-------|----|
| 0     | 1     | 25 |
| 93456 | 93456 |    |
| 83    | 136   |    |

Long identifiers [2;2-10]

CDC ALGOL 60: Only the first 256 characters of an identifier are used.

Labeled outer block [2;2-27]

CDC ALGOL 60: The block or compound statement which defines the program cannot be labeled.

MC ALGOL 60: The outermost block or compound statement can be labeled (See section "Integer labels").

00\*\* TEST:"BEGIN" "GOTO" TEST "END"  
FINIS GEN.BY EOK-CARD

|      |   |                          |           |   |
|------|---|--------------------------|-----------|---|
| LINE | 0 | PROGRAM BEGINS           | (MESSAGE) | 1 |
| LINE | 0 | PROGRAM ENDS             | (MESSAGE) | 1 |
| LINE | 0 | SOURCE DECK ENDS         | (MESSAGE) | 1 |
| LINE | 0 | DECLARATION MISSING FOR. | TEST      | 6 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

08/08/72 GRONINGEN SCOPE 3.3 SUM--279-- 24/01/72

12.08.34.VELDH3S  
12.08.35.VELDH.NR9500.  
12.08.35.ALGOL.  
12.08.35.FILE OPENED---OUTPUT  
12.08.38.FILE OPENED---LGO  
12.08.39. 0 26  
12.08.39.CP 000.027 SEC.  
12.08.39.PP 003.783 SEC.



Arithmetic error conditions [2;2-19]

CDC ALGOL 60: There are automatic signals for irregularities in arithmetic (overflow, underflow, division by zero, etc.). Unless the special procedure ARTHOFLW has been called, the object program terminates abnormally with the message "ARITHMETIC OVERFLOW" when a machine arithmetic error condition arises.

MC ALGOL 60: Exceptional arithmetic conditions are defined so that abnormal termination of the object program is avoided. For example, a real arithmetic expression (such as  $3.2/0$ ) which exceeds  $10^{628}$  is defined equal to the largest X8 real number.

```
00** "BEGIN" "PROCEDURE" PRINT(X); "REAL" X; "OUTREAL"(61,X);
      "REAL" A,B;
      A:=B:=200; PRINT(A*B);
      "END"
      FINIS GEN.BY EOR-CARD
```

|      |   |                  |           |   |
|------|---|------------------|-----------|---|
| LINE | 0 | PROGRAM BEGINS   | (MESSAGE) | 1 |
| LINE | 3 | PROGRAM ENDS     | (MESSAGE) | 1 |
| LINE | 3 | SOURCE DECK ENDS | (MESSAGE) | 1 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

\* INFINITE \*
END OF ALGOL RUN \*V3.0\*

08/08/72 GRONINGEN SCOPE 3.3 SUM--279 24/01/72

14.46.55.VELDM20  
14.46.56.VELDM.NR9500.  
14.46.56.ALGOL.  
14.46.57.FILE OPENED---OUTPUT  
14.47.00.FILE OPENED---LGO  
14.47.02. 0 135  
14.47.02.LGO.  
14.47.12.CP 000.235 SEC.  
14.47.12.PP 010.522 SEC.

ALGOL-60 (3.0) 08/08/72 14.46 HRS PAGE 1

```
00** "BEGIN" "PROCEDURE" PRINT(X); "REAL"-X; OUTREAL(61,X);
      "REAL" A,B;
      A:=B:="200; PRINT(A*B);
      "END"
      FINIS GEN.BY EOR-CARD
```

ALGOL-60 (3.0) XXALGOL 08/08/72 14.46 HRS PAGE 1

```
LINE 0 PROGRAM BEGINS (MESSAGE) 1
LINE 3 PROGRAM ENDS (MESSAGE) 1
LINE 3 SOURCE DECK ENDS (MESSAGE) 1
```

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE -- F.L.X

\*0.0000000000000000\*\*000  
END OF ALGOL RUN \*V3.0\*

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72  
14.46.40.VELDH2Z  
14.46.40.VELDH.NR9500.  
14.46.40.ALGOL.  
14.46.41.FILE OPENED---OUTPUT  
14.46.46.FILE OPENED---LGO  
14.46.49. 0 135  
14.46.49.LGO.  
14.46.55.CP 000.237 SFC.  
14.46.55.PP 011.238 SEC.

```

00** "BEGIN" "PROCEDURE" PRINT(X); "REAL" X; OUTREAL(61,X);
      PRINT(1/0)
"END"
      FINIS GEN.BY EOR-CARD

```

|      |   |  |           |   |
|------|---|--|-----------|---|
| LINE | 0 | PROGRAM BEGINS                           | (MESSAGE) | 1 |
| LINE | 2 | PROGRAM ENDS                             | (MESSAGE) | 1 |
| LINE | 2 | SOURCE DECK ENDS                         | (MESSAGE) | 1 |
| LINE | 2 | ARITHMETIC OVERFLOW OF NUMBER/EXPRESSION |           | 2 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

```

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72
14.17.37.VELDHIV
14.17.37.VELDH.NR9500.
14.17.37.ALGOL.
14.17.39.FILE OPENED---OUTPUT
14.17.42.FILE OPENED---LGO
14.17.42. 0 34
14.17.42.CP 000.035 SEC.
14.17.42.PP 003.737 SEC.

```

090872- 17 A 2458V.1 JACKALANEN 1

A2458V.1,JACK ALANEN

```
1 'BEGIN' 'REAL' A,B; A:=B:#m600; PRINT(A*B);  
2 A:=B:#m-600; PRINT(A*B);  
3 PRINT(1/0)  
'END'
```

090872- 17 A 2458V.1 JACKALANEN 2

+.1776646197514<sub>m</sub>+629 +.6188692094765<sub>m</sub>-616 +.1776646197514<sub>m</sub>+629

090872- 17 A 2458V.1 JACKALANEN 3

```
0 5 25  
59265 59265  
53 84
```

Determination of types [2;2-12,2-20,2-21,2-28]

CDC ALGOL 60: Because integer numbers are represented the same as real numbers, there is no integer to real conversion

For an arithmetic expression where the type cannot be determined at compile-time, the type is assumed real. For example, in

```
integer i; real A;  
i := 2 × (if i > 0 then i else A);
```

the parenthesized expression is considered real.

For the special case of an integer base  $i$  and an exponent  $j$  which is a positive integer variable, the result of  $i \uparrow j$  is real.

Both operands of an integer division operation ( $a \div b$ ) must be integer.

```
00** "BEGIN" "PROCEDURE" PRINT(X); "REAL" X; OUTREAL(61,X);
"INTEGER" I; "REAL" A; I:=A; A:=2.0; PRINT(I "DIV" A)
"END"
FINIS GEN.BY EOR-CARD
```

|      |   |                                      |           |   |
|------|---|--------------------------------------|-----------|---|
| LINE | 0 | PROGRAM BEGINS                       | (MESSAGE) | 1 |
| LINE | 2 | PROGRAM ENDS                         | (MESSAGE) | 1 |
| LINE | 2 | SOURCE DECK ENDS                     | (MESSAGE) | 1 |
| LINE | 2 | INCONSISTENT TYPES WITHIN EXPRESSION |           | 6 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F\*L\*X

08/08/72 GRONINGEN SCOPE 3.3 SUM-279 24/01/72

14.16.58.VELDHIS

14.16.58.VELDH.NH9500.

14.16.58.ALGOL.

14.17.00.FILE OPENED---OUTPUT

14.17.04.FILE OPENED---LG0

14.17.05. 0 41

14.17.06.CP 000.042 SEC.

14.17.06.PP 005.234 SEC.

090872- 22 A 2458V.2 JACKALANEN 1

A2458V.2,JACK ALANEN

'BEGIN' 'INTEGER' I; 'REAL' A; I:=6; A:=2.0; PRINT('DIV' A) 'END'

090872- 22 A 2458V.2 JACKALANEN 2

+3

090872- 22 A 2458V.2 JACKALANEN 3

|       |       |    |
|-------|-------|----|
| 0     | 1     | 25 |
| 52564 | 52564 |    |
| 34    | 67    |    |



Input/output procedures [2;3-1 to 3-46]

CDC ALGOL 60: The extensive input/output procedures are based upon the ACM proposal [8]. The following program illustrates some of these and, at the same time, shows how to change the common MC ALGOL 60 input/output procedures (read, printtext, newpage, carriage, nlcr, print).

A printer line on the CDC holds 136 characters; there are 60 lines per printer page. To locate the current printer position and line number, the two respective CDC ALGOL 60 procedures could be:

```
"INTEGER" "PROCEDURE" PRINTPOS;  
"BEGIN" "INTEGER" N;  
      SYSPARAM (61,1,N); PRINTPOS:= N  
"END";  
"INTEGER" "PROCEDURE" LINE NUMBER;  
"BEGIN" "INTEGER" N;  
      SYSPARAM (61,3,N); LINENUMBER:= N  
"END";
```

```

00** "BEGIN"
      "REAL" "PROCEDURE" "READ": "BEGIN" "REAL" "X": INPUT (60, "(X)" "X): "READ": "X" "END":
      "COMMENT" PRINTTEXT REPLACED BY PRINT:
      "PROCEDURE" "PRINT" (S): "STRING" S: "BEGIN" "INTEGER" N: "PROCEDURE" "F":
      "FORMAT" ("(YS)" "N): "PROCEDURE" "L" (I): "PROCEDURE" "D" (S):
      N:=CHLENGTH(S): OUTLIST (6) .F.L) "END" "PRINT":

      "PROCEDURE" "PAGE": SYSPARAM (6) .4.1):

19** "PROCEDURE" "LINE" (N): "VALUE" N: "INTEGER" N: "BEGIN"
      "INTEGER" L: SYSPARAM (6) .3.L): SYSPARAM (6) .4.N+L) "END" "LINE":

      "COMMENT" "FUNCTION" MICR: SYSPARAM (6) .2.0):

      "COMMENT" "TAB" TO POSITION N: SYSPARAM (6) .2.N):

      "INTEGER" I: "REAL" X: "REAL" X: "DATA":
20** "DATA": "TRUNC": "FORM": "1" "WHILE" "DATA" "DO" "BEGIN"
      X:=READ: IF X=999 THEN "DATA": "FALSE" ELSE "BEGIN" "LINE" (1) :OUTREAL (6) .X):
      "END" ELSE "END" "WHILE":

      "PAGE":
      PRINT ("SOME EXAMPLES OF INPUT/OUTPUT"):
      LINE (3) : I:=1000: OUTREAL (6) .1001): OUTREAL (6) .I+2):
      X:=-1234.567: I:=20:
      OUTPUT (6) ."(X.-570.40)" "X): SYSPARAM (6) .2.I): PRINT ("(*)"):
      SYSPARAM (6) .2.0): OUTREAL (6) .X): OUTPUT (6) ."(/" "TEXT ON NEW LINE" " " "
      )" "END":
30** "FINIS GEN. BY FOR-CARD"
  
```

|      |    |                   |           |   |
|------|----|-------------------|-----------|---|
| LINE | 0  | PROGRAM BEGINS    | (MESSAGE) | 1 |
| LINE | 29 | PROGRAM ENDS      | (MESSAGE) | 1 |
| LINE | 29 | SOURCE DECK ENDS  | (MESSAGE) | 1 |
| LINE | 24 | NON-FORMAT STRING | (MESSAGE) | 1 |

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

- +1.00000000000000000000+000
- +2.00000000000000000000+000
- +3.00000000000000000000+000
- +4.00000000000000000000+000
- +5.00000000000000000000+000
- 7.00000000000000000000+077
- +8.97849000000000000000+003
- +1.00000000000000000000+000
- +2.00000000000000000000+000
- +2.12345678901234-123

SOME EXAMPLES OF INPUT/OUTPUT

```

* +1.00100000000000+003 +1.00200000000000+003
  -1234.5670
-1.23456700000000+003
TEXT ON NEW LINE
END OF ALGOL RUN *V3.0*
  
```

```

11/08/72 *GRANTINGEN SCOPF 3.3 SJM 270 24/01/72
13.22.00.VFLD40
13.22.00.VFLD4.NR0500.
13.22.00.ALGOL.
13.22.10.FILE OPENED---OUTPUT
13.22.13.FILE OPENED---I GO
13.22.14.
13.22.14.160.
13.22.20.00 000.717 SEC.
13.22.20.00 007.220 SEC.
  
```

Standard library procedures

CDC ALGOL 60: A program will be aborted with an error message when a standard library procedure (sqrt, ln, etc.) is called with an illegal argument.

```
00** "BEGIN" "PROCEDURE" PRINT(X); "REAL" X; OUTREAL(61,X);
      PRINT(SORT(-3))
      "END"
      FINIS GEN.BY FOR-CARD
```

```
LINE 0 PROGRAM BEGINS (MESSAGE) 1
LINE 2 PROGRAM ENDS (MESSAGE) 1
LINE 2 SOURCE DECK ENDS (MESSAGE) 1
```

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

SQUARE ROOT ERROR

THIS ERROR OCCURRED AFTER LINE 1 IN XXALGOL

THE GLOBAL VARIABLES ARE ..

```
UA.VALUE 000341 -3.000000000000000000+000
UV 000000 -3.000000000000000000+000
LASTUSED 012541 0000 001335 0001 012464
```

IN THE BLOCK ENTERED AT LINE 0 IN XXALGOL

THE LOCAL VARIABLES ARE ..

```
000004 012470 7747 776474 4000 012464
```

THIS BLOCK WAS CALLED FROM LINE 0 IN ALGORUN

```
00** "BEGIN" "PROCEDURE" PRINT(X); "REAL"-X; OUTREAL(61,X);
      PRINT(LN(0))
      "END"
      FINIS GEN.BY EOR-CARD
```

```
LINE 0 PROGRAM BEGINS (MESSAGE) 1
LINE 2 PROGRAM ENDS (MESSAGE) 1
LINE 2 SOURCE DECK ENDS (MESSAGE) 1
```

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE F.L.X

LOGARITHM ERROR

THIS ERROR OCCURRED AFTER LINE 1 IN XXALGOL

THE GLOBAL VARIABLES ARE ..
UA+VALUE 000341 0000 000000 0000 000000
UV 000000 0000 000000 0000 000000
LASTUSED 0125+1 0000 001335 0001 0125+1

IN THE BLOCK ENTERED AT LINE 0 IN XXALGOL

THE LOCAL VARIABLES ARE ..
000004 012470 7747 776474 4000 012464

THIS BLOCK WAS CALLED FROM LINE 0 IN ALGORUN

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72
14.17.24.VELUM1#
14.17.24.VELUM.NR9500.
14.17.24.ALGOL.
14.17.25.FILE OPENED---OUTPUT
14.17.27.FILE OPENED---LGO
14.17.28. 0 126
14.17.28.LGO.
14.17.33.CP 000.246 SFC.
14.17.33.PP 006.201 SEC.

090872- 23 A 2458V.3 JACKALANEN 1

A2458V.3,JACK ALANEN

'BEGIN' PRINT(SQRT(-3)); PRINT(LN(0)) 'END'

090872- 23 A 2458V.3 JACKALANEN 2

+0 -.1776646197514e+629

090872- 23 A 2458V.3 JACKALANEN 3

|       |       |    |
|-------|-------|----|
| 0     | 1     | 25 |
| 3>20> | 35205 |    |
| 20    | 50    |    |

Diagnostics (error messages) [2; 8-1 to 8-15]

CDC ALGOL 60: Compiler diagnostics consist of error descriptions ("DECLARATION MISSING FOR ... <identifier>", "END'S MISSING", "EXCESSIVE NUMBER OF UNIQUE IDENTIFIERS", etc.) printed following the source listing. The line count of the source card in error is included.



Job control cards [2;6-1]

Because of the numerous options available in CDC ALGOL 60 and of the generality of the CDC operating system, there are many possible control cards for running a job. The following are the minimal required:

1. Identification card
  2. JOB card
  3. ALGOL card
  4. Load-and-go card
  5. End-of-record card
  6. Source program cards
  7. End-of-record card
  8. Input data cards
  9. End-of-file card
- } May be both omitted.

The format for the first card is yet to be determined by SARA. Cards 2,3, and 4 in the simplest situation are:

```

Col. 123456
<Your name>.
  ALGOL.
  LGO.

```

An end-of-record card consists of a 7-8-9 multipunch in column 1. An end-of-file card consists of a 6-7-8-9 multipunch in column 1.

Version 4 of CDC ALGOL 60

We have described version 3 of the CDC ALGOL 60 system. However, in the first quarter of 1973, CDC will release a version 4. Most programs written in version 3 of CDC ALGOL 60 will also run under the version 4 compiler. But version 4 will not handle segmentation\* which is available in version 3, and version 4 will use improved source code debugging statements. There is no manual available for version 4 CDC ALGOL 60 yet.

The main important new feature of version 4 is that FORTRAN library subroutines can be called and executed within the ALGOL program. Thus the enormous CDC FORTRAN library of subroutines will become directly accessible to the ALGOL programmer.

\* An overlay facility will, however, be available.

References

1. \_\_\_\_\_, 1966. Control Data 6400/6600 Computer Systems ALGOL General Information Manual, Publication No. 60173200.
2. \_\_\_\_\_, 1971. Control Data 6000 Computer Systems ALGOL 60 Reference Manual, Version 3, Publication No. 60329000.
3. Grune, D., 1971. Handleiding Milli-systeem voor de EL X8, Report LR 1.1 of Stichting Mathematisch Centrum, 2e Boerhaavestraat 49, Amsterdam.
4. Knuth, D.E., 1964. Man or boy? ALGOL Bulletin No. 17 (page 7).
5. Kruseman Aretz, F.E.J., 1969. Het MC-ALGOL 60-systeem voor de X8, Voorlopige programmeurshandleiding. Report MR 81 (4e uitgave) of Stichting Mathematisch Centrum, 2e Boerhaavestraat 49, Amsterdam.
6. Naur, P., 1964. Revised Report on the Algorithmic Language ALGOL 60, Corrected reprint of A/S Regnecentralen, Copenhagen.
7. Poel, W.L. van der, 1972. A comparative study of some higher programming languages. Lecture notes for Advanced Course on Programming Languages and Data Structures, Amsterdam, 12-23 June 1972, page 1.22.-3.
8. \_\_\_\_\_, 1964. A proposal for input-output conventions in ALGOL 60, The Communications of the ACM, vol. 7, no. 5 (May).

