

RA

**stichting
mathematisch
centrum**

**Σ
MC**

REKENAFDELING

RA

CR 26/72

SEPTEMBER

JACK ALANEN
CHANGING MC ALGOL 60 PROGRAMS FOR THE
SARA COMPUTER

2e boerhaavestraat 49 amsterdam

BIBLIOTHEEK MATHEMATISCH CENTRUM
 AMSTERDAM

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".

ALGOL-60 (3.0) 02/04/72 13.13 HRS PAGE 1

```

00** "BEGIN" INTEGER PROCEDURE "A(K,X1,X2,X3,X4,X5)";
    "VALUE" K := INTEGER "K,X1,X2,X3,X4,X5";
    "BEGIN" "INTEGER" PROCEDURE "A";
        "DEFINITION" K-1 := A(K,X1,X2,X3,X4,X5);
        "IF" K NOT GREATER 0 THEN "A" := X4+X5 "ELSE" B
            "END";
        OUTREAL(61,A(61,2,3,4,5))
    "END"
FINIS GEN-RY EOR-CARD

```

ALGOL-60	(3.0)	XXALGOL	02/08/72	13.13 HRS	PAGE 1
LINE 0		PROGRAM BEGINS		(MESSAGE)	1
LINE 7		PROGRAM ENDS		(MESSAGE)	1
LINE 7		SOURCE DECK ENDS		(MESSAGE)	1

THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE

```

CORE MAP 13.14.05. NORMAL CONTROL
---TIME---LOAD MODE ---L1---L2---TYPE---USER---CALL---000100 012016 007A16 002200
FWA LOADER--036771--FWA TABLES--035241-----FWA LOAD--LWA LOAD--BLNK COMM--LENGTH--
PROGRAM-----ADDRESS--
XXALGOL 000340
ALGORUN 001340
--Labeled---COMM0N--
000100
000100
000100
000100
REFERENCES
ALGLB04 003656
ALGLB04 007103
-----UNSATISFIED EXTERNALS-----

```

CHANNEL 60=INPUT, PAD, R
CHANNEL 61=OUTPUT, P136, PP60, R

*2.800000000000"+001
END OF ALGOL RUN *V3.0*

02/08/72 *GRONINGEN SCOPE: 3.3 SUM 279 24/01/72
13.13.54. VELDH6L
13.13.54. VELDH.NR9500. VELDHUYZEN VAN ZANTEN
13.13.54. ALGOL.
13.13.54. FILE OPENED---OUTPUT
13.14.02. FILE OPENED---LGO
13.14.02. 0 21A
13.14.02. LGO.
13.14.06. CP 000.332 SEC.
13.14.06. PP 006.086 SEC.

020872- 24 A 2458V.4 JACKALANEN 1

A2458V.4,JACK ALANEN

```
'BEGIN' 'COMMENT' 'MAN OR BOY (ANSWER = 28) FROM VD POEL, PAGE 1,22,31
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** TEST1 "BEGIN" "REAL" Y,EPS; "INTEGER" N;
"PROCEDURE" OUTPUT(X); "BEGIN" NLCR; PRINT(X) "END";
"BOOLEAN" "ARRAY" B(1:800000); "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,M1,C1,D1,E1,F1,G1,H1,I1,J1,K1,L1,M1,N1,O1,P1,Q1,R1,S1,T1,U1,V1,W1,X1,
Y1,Z1,A2,M2,C2,D2,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*100; OUTPUT(REAL Y);
EPS:=1*100; OUTPUT(EPS);
3* N:=2;
Q(1.6*10); OUTPUT(N);
PRINTTEXT("MESSAGE");
E(22):=N:=1;"FORM" E(10/);
PRINT(LN(0)); PRINT(SORT(-3)); PRINT("700"); PRINT(1/0);
A(0,3):=0;
REQD

```

FINIS GEN.BY FOR-CARD

LINE -0- PROGRAM-BEGIN (MESSAGE) 1
LINE 22- PROGRAM-ENDS (MESSAGE) 1
LINE 22 SOURCE DECK ENDS (MESSAGE) 1
LINE 1 DECLARATION MISSING FOR.. PRINT 6
LINE 1
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	3
LINE 14	ARITHMETIC OVERFLOW OF NUMBER/EXPRESSION	1
LINE 16	.. FOLLOWS AN INCORRECT OPERAND	2
LINE 18	NON-FORMAT STRING (MESSAGE)	3
LINE 18	DECLARATION MISSING FOR.. PRINTTEXT	1
LINE 20	ARITHMETIC OVERFLOW OF NUMBER/EXPRESSION	6
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.VELDHIN
14.15.32.VELDMNR9500.
14.15.32.AL60L.
14.15.39.FILE OPENED===OUTPUT
14.15.42.FILE OPENED===L60
14.15.42. 0 279
14.15.43.CP 000.200 SFC.
14.15.43.PP 003.054 SFC.

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.

VFLDH,NDQ500. VFLDHUYZEN VAN 7ANTEN
 COMMENT,EXAMPLE OF ENTRY/EXIT MACROS, ALGOL MANUAL 15-2
 COMDASS,SEALGTEXT.

ALGOL.

IG0.

00000000000000000000 7/8/9 END OF RECORD

**SHRDBACDAM

SST

*PARAMETER VALUE SAVING..

CONF 1

VALUE 1.X6.R.1

SAS SAVE

RETURN

RSSZ 1

*RETURN VALUE..

CONF 2

SAS SAVE

ASSIGN 1.X6

RETURN

END

00000000000000000000 7/8/9 END OF RECORD

"BEGIN"

"BDBACEURP" P1(X);"BDEALUY";"CONF1";

"BDBACEURP" P2(X);"BDEALUY";"CONF2";

"BDEALUY";

A:=3; P1(A); P2(B); OUTDEAL(A1.A)

END

00000000000000000000 6/7/8/9 END OF FILE

*3.000000000000000000

END OF ALGOL RUN #V3.0*

11/09/72 *GRANTINGEN SCODE 3.3 SUM 270 24/01/72

13.31.55.VFLDH61

13.31.55.VFLDH.CM100000.NR9500.

13.31.55.EXAMPLE OF ENTRY/EXIT MACROS.

13.31.55. SEE ALGOL MANUAL CHAPTER 15-2.

13.31.55.COMDASS,SEALGTEXT.

13.31.57.FTIF OPENED---SCP1

13.31.57.FTIF OPENED---COMPSCP

13.31.59.FTIF OPENED---IG0

13.31.59.FTIF OPENED---OUTBIT

13.31.59.ASSEMBLY COMPLETE. 53400H SUM USED.

13.31.59.ALGOL.

13.32.03. 1 160

13.32.03.IG0.

13.32.07.CP 001.245 SEC.

13.32.07.CP 008.453 SEC.

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

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

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

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

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 EOM-CARD

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

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

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

+3.000000000000000000
END OF ALGOL RUN *V3.0*

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

14.46.18.VELDH20
14.46.18.VELDH.NR9500.
14.46.18.ALGOL.
14.46.19.FILE OPENED---OUTPUT
14.46.22.FILE OPENED---1.60
14.46.23. 0
14.46.23.ALGOL.
14.46.24. 0 249
14.46.30.L60.
14.46.37.CP 000.354 SEC.
14.46.37.PP 013.856 SEC.

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

^{*} Punch these symbols on the IBM 029 machines at the MC or at SARA (located at the VU). In addition, the five characters # \$? _ $\neg$ 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).

PAGE 2

11/08/72 12.27 HRS

XXALGOL

(3.0)

ALGOL-60

```

OUTPUT(A1,"(2700)",ARRAY(PRASE/I)).
  "FOR" I=PRASE+1 "STEP" 1 "UNTIL" POINTER-1 "DO"
  OUTPUT(A1,"((",I," 2700)",ARRAY(I/I)).
  OUTPUT(A1,"((",I," 2700)",ADDITIONAL)).
ENDOUT..
"END" AUTO..

```

```

"INTEGRANDARRAY" NUM,REFFIRST,REFSEC(I/2..N/I)..
"INTEGRANDARRAY" STACK,LIST(PRASE..10/I)..
"INTEGR" P=0,SUM,NUMBER,MINSUM,MINSUMNUMBER,DIFFMINSUM
  DIMMY,I,J,REFFIRST,REFSEC,REFPROD

```

```

70** BR00LFAN0 MTX..

```

```

"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

```

```

  THE ARRAYS REFFIRST,REFSEC ARE NOT INITIALIZED

```

```

  "FOR" I=2 "STEP" 1 "UNTIL" NUM "DO"
  NUM(I/I)=I..
  "FOR" J=2 "STEP" 1 "UNTIL" NUM "DO"
  "IF" NUM(I/I) "NOT LESS" I "THEN"
  "BEGIN" "FOR" J=I+1 "STEP" 1 "UNTIL" NUM "DO"
  NUM(J/I)=I..
  J=I..
  "FOR" J=J+1 "UNTIL" "NOT GREATER" N "DO"
  NUM(J/I)=I
  "END"

```

```

00**

```

```

  "FOR" I=2 "STEP" 1 "UNTIL" NUM "DO"

```

```

"COMMENT"
  NOW WE ANALYZE ARRAY NUM

```

```

  "FOR" I=2 "STEP" 1 "UNTIL" NUM "DO"
  MINSUM=6..
  MINSUMNUMBER=5..
  NUMBER=5..
  "COMMENT" THE NUMBERS 1 TO 5 ARE NOT ANALYZED..
  NUMBER=NUMBER+1..
  P=PRASE..
  "IF" NUMBER GREATER N "THEN" "GOTO" FINISH..
  "IF" NUM(I/NUMBER) "GREATER" 0 "THEN"
  "BEGIN" SUM=NUMBER..
  "GOTO" COMPARE "END" ..
  REFIRST=REFIRST(SOPT(NUMBER))..
  REFSEC=REFIRST+1..
  REPROD=REFIRST+REFSEC..
  "FOR" DIMMY=0 "UNTIL" REPROD "NOT EQUAL" NUMBER "DO"
  "IF" REPROD "LESS" NUMBER "THEN"
  "BEGIN"
    REPROD=REFIRST+REFIRST..
    REFS=REFIRST+1
  "END" REFS
  "BEGIN"

```

```

10**

```

```

NEXT..

```

```

11**

```

11/08/72 12.27 HPS

XXALGOL

(1.0)

ALGOL-60

```
RELPROD:=RELPROD*RELSFC..
RELFIRST:=RELFIRST-1
```

```
"END"..
```

120**

```
"COMMENT" WE HAVE NOW TWO POSSIBLE RELATIVE PRIMES
SEARCH THE SIMPLEST COMBINATION..
MIX:=FALSE..
SEARCH(RELFIRST,RELSFC)..
```

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

130**

```
DIFFMINSUM:=NUMBER-MINSUMNUMBER..
"IF" SUM "GREATER" MINSUM-DIFFMINSUM "THEN"
"BEGIN"
OUT(NUMBER,IST,0,DIFFMINSUM,MINSUM,DIFFMINSUM)..
"END" "ELSE"
"IF" SUM = MINSUM-DIFFMINSUM "THEN"
"BEGIN"
OUT(NUMBER,STACK,P,0,SUM)..
OUT(NUMBER,IST,0,DIFFMINSUM)..
"END" "ELSE"
"BEGIN"OUT(NUMBER,STACK,P,0,SUM)..
"END" "ELSE"PRASE "STEP" UNTIL "P-1" "DO"
LIST(I/I)=STACK(I/I)..
0:=0.. MINSUM:=SUM.. MINSUMNUMBER:=NUMBER..
"END"..
```

140**

```
"COMMENT" BOOLEAN MIX IS TRUE WHEN IN THE ANALYSIS
OF RELFIRST AND RELSFC OCCUR THE SAME PRIMES AND
POWERS OF PRIMES..
```

```
"IF" MIX "THEN"
"BEGIN"
IPMX.. "IF" "NOT GREATER" PRASE+2 "THEN" "GOTO" PMIX..
STACK(P/P)=1..J:=PRASE..
"UNTIL" PRASE "STEP" UNTIL "P-1" "DO"
"BEGIN"STACK(I/I)=STACK(I/I)*STACK(I/I+1)..
J:=J+1..
"END"..
P:=J.. "GOTO" IPMX..
```

160**

```
PMIX..
DIFFFIRST(NUMBER/I)=STACK(P/PRASE)..
DIFFSEC(NUMBER/I)=STACK(P/PRASE+1/I)..
"END" "ELSE"
"BEGIN"DIFFFIRST(NUMBER/I)=RELFIRST..
DIFFSEC(NUMBER/I)=RELSFC..
"END"..
```

170**

```
"GOTO" NEXT
```

PAGE 4

11/08/72 12.27 HRS

XXALGOL

ALGOL-60 (3.0)

"END".
 FINISH..
 "END"
 "EOP"

FINIS GEN.BY FOR-CARD

PAGE 1

11/08/72 12.27 HRS

XXALGOL

ALGOL-60 (3.0)

LINE 0 PROGRAM BEGINS (MESSAGE)
 LINE 176 PROGRAM ENDS (MESSAGE)
 LINE 177 SOURCE DECK ENDS (MESSAGE)

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

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

3 * 4 + 2

3 * 5 + 3

4 * 5 + 1

5 * 7 + 1

5 * 7 + 4

5 * 9 + 6

```

964 7 8 * 7 * 17 + 12
965 = 8 * 7 * 17 + 13
966 = 21 * 3 * 7 + 0
967 = 21 * 3 * 7 + 1
968 = 21 * 3 * 7 + 2
969 = 21 * 3 * 7 + 3
970 = 21 * 3 * 7 + 4
971 = 21 * 3 * 7 + 5
972 7 21 * 2 * 3 * 7 + 6
973 = 21 * 2 * 3 * 7 + 7
974 = 21 * 2 * 3 * 7 + 8
975 = 25 * 3 * 13 + 0
976 = 25 * 3 * 13 + 1
977 = 25 * 3 * 13 + 2
978 = 25 * 3 * 13 + 3
979 = 25 * 3 * 13 + 4
980 = 25 * 3 * 13 + 5
981 = 25 * 3 * 13 + 6
982 = 25 * 3 * 13 + 7
983 = 25 * 3 * 13 + 8
984 = 25 * 3 * 13 + 9
985 = 25 * 3 * 13 + 10
986 = 29 * 2 * 17 + 0
987 = 29 * 2 * 17 + 1
988 = 4 * 13 * 19 + 0
989 = 4 * 13 * 19 + 1
990 = 5 * 2 * 9 * 11 + 0
991 = 5 * 2 * 9 * 11 + 1
992 = 5 * 2 * 9 * 11 + 2
993 = 5 * 2 * 9 * 11 + 3
994 = 5 * 2 * 9 * 11 + 4
995 = 5 * 2 * 9 * 11 + 5
996 = 5 * 2 * 9 * 11 + 6
997 = 5 * 2 * 9 * 11 + 7
998 = 5 * 2 * 9 * 11 + 8
999 = 5 * 2 * 9 * 11 + 9
1000 = 5 * 2 * 9 * 11 + 10
      END OF ALGOL RUN *V3.0*
44 .
45 .
46 .
47 .
48 .
49 .
50 .
51 .
52 .
53 .
54 .
55 .
56 .
57 .
58 .
59 .
60 .
61 .
62 .
63 .
64 .
65 .
66 .
67 .
68 .
69 .
70 .
71 .
72 .
73 .
74 .
75 .
76 .
77 .
78 .
79 .
80 .
81 .
82 .
83 .
84 .
85 .
86 .
87 .
88 .
89 .
90 .
91 .
92 .
93 .
94 .
95 .
96 .
97 .
98 .
99 .
100 .

```

```

11/08/72 *GRANTINGEN SCODE 3.3 SIM 279 24/01/72
12.27.32.FRANK4F
12.27.32.FRANK*ARG500.
12.27.32.ALGO1.
12.27.33.FILF OPENFD---OUTPUT
12.27.34.FILF OPENFD---LGO
12.27.42.
12.27.42.LGO.
12.28.00.CP 014.873 SEC.
12.28.00.DP 014.434 SEC.

```

090872-122 B 2458V.11 FRANKGOOSSENS 1

82458V.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
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
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' 1, PBASE+1 'STEP' 1 'UNTIL' POINTER-1 'DO'
44 'BEGIN' PRSYM(66), 'COMMENT' STARSIGN.,
45 ABSFIXT(2,0,ARRAY(/1/)),
46 'END',
47 PRSYM(64), 'COMMENT' PLUSSIGN.,
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' OUTO(N,ARRAY, POINTER, ADDITIONAL),

```

090872-122 8 2458V.11 FRANKGOSSSENS 2
 'VALUE' N, POINTER, ADDITIONAL., 'INTEGER' N, POINTER, ADDITIONAL.,
 'ARRAY' ARRAY.,
 'BEGIN'

56 'IF' N 'LESS' NPRINTMIN 'THEN' 'GOTO' ENDOUT.,
 57 CARRIAGE(0), SPACE(100),
 58 ABSFIXT(2,0,ARRAY(/PBASE/)),
 59 'FOR' I, PBASE+1 'STEP' 1 'UNTIL' POINTER-1 'DO'
 60 'BEGIN' PRSYM(66), ABSFIXT(2,0,ARRAY(/I/)) 'END',
 61 PRSYM(64), ABSFIXT(2,0, ADDITIONAL),
 62 ENDOUT.,
 63 'END' OUTD.,
 64

65 'INTEGER' ARRAY, NUM, REFFIRST, REFSEC(/2,N/),
 66 'INTEGER' ARRAY, STACK, LIST(/PBASE, 10/),
 67 'INTEGER' P, Q, SUM, NUMBER, MINSUM, MINNUMBER, DIFFMINSUM
 68 , DUMMY, I, J, RELFIRST, RELSEC, RELPROD
 69 'BOOLEAN' MIX.,
 70

71 'COMMENT'
 72 INITIALIZATION OF THE ARRAYS NUM, REFFIRST, REFSEC
 73 NUM/ IF INDEX IS PRIME THEN CONTENTS IS INDEX
 74 IF INDEX IS APOWER OF PRIME THEN CONTENTS IS
 75 THAT PRIME
 76 IF INDEX IS NOT ONE OF ABOVE MENTIONED THEN
 77 CONTENTS IS NEGATIVE
 78

THE ARRAYS REFFIRST, REFSEC ARE NOT INITIALIZED

79 'FOR' I, 2 'STEP' 1 'UNTIL' N 'DO'
 80 NUM(/I/) = 1.
 81 'FOR' I, 2 'STEP' 1 'UNTIL' N 'DO'
 82 'IF' NUM(/I/) 'NOT LESS' 1 'THEN'
 83 'BEGIN' 'FOR' J, 1 * 2 'STEP' 1 'UNTIL' N 'DO'
 84 NUM(/J/) = 1.,
 85 J = 1.,
 86 'FOR' J, 2 * I 'WHILE' J 'NOT GREATER' N 'DO'
 87 NUM(/J/) = 1
 88 'END',
 89

HEADER.,

'COMMENT' NOW WE ANALYZE ARRAY NUM

90 MINSUM = 6.,
 91 MINNUMBER = 5.,
 92 NUMBER = 5., 'COMMENT' THE NUMBERS 1 TO 6 ARE NOT ANALYZED.,
 93 NUMBER = NUMBER + 1., P = PBASE.,
 94 'IF' NUMBER 'GREATER' N 'THEN' 'GOTO' FINISH.,
 95 'IF' NUM(/NUMBER/) 'GREATER' 0 'THEN'
 96 'BEGIN' SUM = NUMBER., 'GOTO' COMPARE 'END'.,
 97 RELFIRST = ENTIER(SQRT(NUMBER)),
 98 RELSEC = RELFIRST + 1.,
 99 RELPROD = RELFIRST * RELSEC.,
 100 'FOR' DUMMY = 0 'WHILE' RELPROD 'NOT EQUAL' NUMBER 'DO'
 101 'IF' RELPROD 'LESS' NUMBER 'THEN'
 102 'BEGIN' RELPROD = RELPROD * RELFIRST.,
 103

NEXT.,

104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115

090872-122 B 2450V.11 FRANKGOOSSENS 3

```

116 RELSEC.=RELSEC+1
117 'END' 'ELSE'
118 'BEGIN'
119   RELPROD.=RELPROD*RELSEC.,
120   RELFIRST.=RELFIRST-1
121 'END'..
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175

'COMMENT' WE HAVE NOW TWO POSSIBLE RELATIVE PRIMES
SEARCH THE SIMPLEST COMBINATION.,
MIX.=FALSE.,
SEARCH(RELFIRST,RELSEC),
'COMMENT' THE STACK IS FILLED WITH THE BEST COMBINATION.,
'COMMENT' CALCULATE THE SUM OF THE RELATIVE PRIMES.,
SUM.=0.,
'FOR' I.=PBASE 'STEP' 1 'UNTIL' P-1 'DO'
  SUM.=SUM+STACK(I/),
COMPARE.,
DIFFMINSUM.=NUMBER-MINSUMNUMBER.,
'IF' SUM 'GREATER' MINSUM+DIFFMINSUM 'THEN'
  'BEGIN'
    OUT(NUMBER,LIST,G,DIFFMINSUM,MINSUM+DIFFMINSUM),
  'END' 'ELSE'
    'IF' SUM = MINSUM+DIFFMINSUM 'THEN'
      'BEGIN'
        OUT(NUMBER,STACK,P,0,SUM),
        OUT(NUMBER,LIST,G,DIFFMINSUM),
      'END' 'ELSE'
        'BEGIN'OUT(NUMBER,STACK,P,0,SUM),
        'FOR' I.=PBASE 'STEP' 1 'UNTIL' P-1 'DO'
          LIST(I/).=STACK(I/),
          Q.=P., MINSUM.=SUM., MINSUMNUMBER.=NUMBER.,
        'END'..
      'COMMENT' BOOLEAN MIX IS TRUE WHEN IN THE ANALYSIS
      OF RELFIRST AND RELSEC OCCUR THE SAME PRIMES AND
      POWERSOF PRIMES.,
    'IF' MIX 'THEN'
      'BEGIN'
        LPMIX.,
        'IF' P'NOT GREATER' PBASE+2 'THEN' 'GOTO' RDMIX.,
        STACK(P/).=1.,J.=PBASE.,
        'FOR' I.=PBASE 'STEP' 2 'UNTIL' P-1 'DO'
          'BEGIN'STACK(J/).=STACK(I/)*STACK(I+1/),
          J.=J+1.,
        'END'..
        P.=J., 'GOTO' LPMIX.,
      RDMIX.,
      REFFIRST(/NUMBER/).=STACK(/PBASE/),
      REFSEC(/NUMBER/).=STACK(/PBASE+1/),
    'END' 'ELSE'
      'BEGIN'REFFIRST(/NUMBER/).=RELFIRST.,
      REFSEC(/NUMBER/).=RELSEC.,
    'END'..

```

090872-122 B 2458V.11 FRANKCOOSSENS

4

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

090872-122 B 2458V.11 FRANKGOOSSENS

5

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 * 3 + 4	9
11 = 2 * 3 + 5	10
12 = 2 * 3 + 6	11
13 = 2 * 3 + 7	12
14 = 2 * 3 + 8	13
15 = 2 * 3 + 9	14
16 = 2 * 3 + 10	15
17 = 2 * 3 + 11	16
18 = 2 * 3 + 12	17
19 = 2 * 3 + 13	18
20 = 2 * 3 + 14	19
21 = 2 * 3 + 15	20
22 = 2 * 3 + 16	21
23 = 2 * 3 + 17	22
24 = 2 * 3 + 18	23
25 = 2 * 3 + 19	24
26 = 2 * 3 + 20	25
27 = 2 * 3 + 21	26
28 = 2 * 3 + 22	27
29 = 2 * 3 + 23	28
30 = 2 * 3 + 24	29
31 = 2 * 3 + 25	30
32 = 2 * 3 + 26	31
33 = 2 * 3 + 27	32
34 = 2 * 3 + 28	33
35 = 2 * 3 + 29	34
36 = 2 * 3 + 30	35
37 = 2 * 3 + 31	36
38 = 2 * 3 + 32	37
39 = 2 * 3 + 33	38
40 = 2 * 3 + 34	39
41 = 2 * 3 + 35	40
42 = 2 * 3 + 36	41
43 = 2 * 3 + 37	42
44 = 2 * 3 + 38	43
45 = 2 * 3 + 39	44
46 = 2 * 3 + 40	45
47 = 2 * 3 + 41	46
48 = 2 * 3 + 42	47
49 = 2 * 3 + 43	48
50 = 2 * 3 + 44	49
51 = 2 * 3 + 45	50
52 = 2 * 3 + 46	51
53 = 2 * 3 + 47	52
54 = 2 * 3 + 48	53
55 = 2 * 3 + 49	54
56 = 2 * 3 + 50	55
57 = 2 * 3 + 51	56
58 = 2 * 3 + 52	57
59 = 2 * 3 + 53	58
60 = 2 * 3 + 54	59
61 = 2 * 3 + 55	60
62 = 2 * 3 + 56	61
63 = 2 * 3 + 57	62
64 = 2 * 3 + 58	63
65 = 2 * 3 + 59	64
66 = 2 * 3 + 60	65
67 = 2 * 3 + 61	66
68 = 2 * 3 + 62	67
69 = 2 * 3 + 63	68
70 = 2 * 3 + 64	69
71 = 2 * 3 + 65	70
72 = 2 * 3 + 66	71
73 = 2 * 3 + 67	72
74 = 2 * 3 + 68	73
75 = 2 * 3 + 69	74
76 = 2 * 3 + 70	75
77 = 2 * 3 + 71	76
78 = 2 * 3 + 72	77
79 = 2 * 3 + 73	78
80 = 2 * 3 + 74	79
81 = 2 * 3 + 75	80
82 = 2 * 3 + 76	81
83 = 2 * 3 + 77	82
84 = 2 * 3 + 78	83
85 = 2 * 3 + 79	84
86 = 2 * 3 + 80	85
87 = 2 * 3 + 81	86
88 = 2 * 3 + 82	87
89 = 2 * 3 + 83	88
90 = 2 * 3 + 84	89
91 = 2 * 3 + 85	90
92 = 2 * 3 + 86	91
93 = 2 * 3 + 87	92
94 = 2 * 3 + 88	93
95 = 2 * 3 + 89	94
96 = 2 * 3 + 90	95
97 = 2 * 3 + 91	96
98 = 2 * 3 + 92	97
99 = 2 * 3 + 93	98
100 = 2 * 3 + 94	99
101 = 2 * 3 + 95	100
102 = 2 * 3 + 96	101
103 = 2 * 3 + 97	102
104 = 2 * 3 + 98	103
105 = 2 * 3 + 99	104
106 = 2 * 3 + 100	105
107 = 2 * 3 + 101	106
108 = 2 * 3 + 102	107
109 = 2 * 3 + 103	108
110 = 2 * 3 + 104	109
111 = 2 * 3 + 105	110
112 = 2 * 3 + 106	111
113 = 2 * 3 + 107	112
114 = 2 * 3 + 108	113
115 = 2 * 3 + 109	114
116 = 2 * 3 + 110	115
117 = 2 * 3 + 111	116
118 = 2 * 3 + 112	117
119 = 2 * 3 + 113	118
120 = 2 * 3 + 114	119
121 = 2 * 3 + 115	120
122 = 2 * 3 + 116	121
123 = 2 * 3 + 117	122
124 = 2 * 3 + 118	123
125 = 2 * 3 + 119	124
126 = 2 * 3 + 120	125
127 = 2 * 3 + 121	126
128 = 2 * 3 + 122	127
129 = 2 * 3 + 123	128
130 = 2 * 3 + 124	129
131 = 2 * 3 + 125	130
132 = 2 * 3 + 126	131
133 = 2 * 3 + 127	132
134 = 2 * 3 + 128	133
135 = 2 * 3 + 129	134
136 = 2 * 3 + 130	135
137 = 2 * 3 + 131	136
138 = 2 * 3 + 132	137
139 = 2 * 3 + 133	138
140 = 2 * 3 + 134	139
141 = 2 * 3 + 135	140
142 = 2 * 3 + 136	141
143 = 2 * 3 + 137	142
144 = 2 * 3 + 138	143
145 = 2 * 3 + 139	144
146 = 2 * 3 + 140	145
147 = 2 * 3 + 141	146
148 = 2 * 3 + 142	147
149 = 2 * 3 + 143	148
150 = 2 * 3 + 144	149
151 = 2 * 3 + 145	150
152 = 2 * 3 + 146	151
153 = 2 * 3 + 147	152
154 = 2 * 3 + 148	153
155 = 2 * 3 + 149	154
156 = 2 * 3 + 150	155
157 = 2 * 3 + 151	156
158 = 2 * 3 + 152	157
159 = 2 * 3 + 153	158
160 = 2 * 3 + 154	159
161 = 2 * 3 + 155	160
162 = 2 * 3 + 156	161
163 = 2 * 3 + 157	162
164 = 2 * 3 + 158	163
165 = 2 * 3 + 159	164
166 = 2 * 3 + 160	165
167 = 2 * 3 + 161	166
168 = 2 * 3 + 162	167
169 = 2 * 3 + 163	168
170 = 2 * 3 + 164	169
171 = 2 * 3 + 165	170
172 = 2 * 3 + 166	171
173 = 2 * 3 + 167	172
174 = 2 * 3 + 168	173
175 = 2 * 3 + 169	174
176 = 2 * 3 + 170	175
177 = 2 * 3 + 171	176
178 = 2 * 3 + 172	177
179 = 2 * 3 + 173	178
180 = 2 * 3 + 174	179
181 = 2 * 3 + 175	180
182 = 2 * 3 + 176	181
183 = 2 * 3 + 177	182
184 = 2 * 3 + 178	183
185 = 2 * 3 + 179	184
186 = 2 * 3 + 180	185
187 = 2 * 3 + 181	186
188 = 2 * 3 + 182	187
189 = 2 * 3 + 183	188
190 = 2 * 3 + 184	189
191 = 2 * 3 + 185	190
192 = 2 * 3 + 186	191
193 = 2 * 3 + 187	192
194 = 2 * 3 + 188	193
195 = 2 * 3 + 189	194
196 = 2 * 3 + 190	195
197 = 2 * 3 + 191	196
198 = 2 * 3 + 192	197
199 = 2 * 3 + 193	198
200 = 2 * 3 + 194	199
201 = 2 * 3 + 195	200
202 = 2 * 3 + 196	201
203 = 2 * 3 + 197	202
204 = 2 * 3 + 198	203
205 = 2 * 3 + 199	204
206 = 2 * 3 + 200	205
207 = 2 * 3 + 201	206
208 = 2 * 3 + 202	207
209 = 2 * 3 + 203	208
210 = 2 * 3 + 204	209
211 = 2 * 3 + 205	210
212 = 2 * 3 + 206	211
213 = 2 * 3 + 207	212
214 = 2 * 3 + 208	213
215 = 2 * 3 + 209	214
216 = 2 * 3 + 210	215
217 = 2 * 3 + 211	216
218 = 2 * 3 + 212	217
219 = 2 * 3 + 213	218
220 = 2 * 3 + 214	219
221 = 2 * 3 + 215	220
222 = 2 * 3 + 216	221
223 = 2 * 3 + 217	222
224 = 2 * 3 + 218	223
225 = 2 * 3 + 219	224
226 = 2 * 3 + 220	225
227 = 2 * 3 + 221	226
228 = 2 * 3 + 222	227
229 = 2 * 3 + 223	228
230 = 2 * 3 + 224	229
231 = 2 * 3 + 225	230
232 = 2 * 3 + 226	231
233 = 2 * 3 + 227	232
234 = 2 * 3 + 228	233
235 = 2 * 3 + 229	234
236 = 2 * 3 + 230	235
237 = 2 * 3 + 231	236
238 = 2 * 3 + 232	237
239 = 2 * 3 + 233	238
240 = 2 * 3 + 234	239
241 = 2 * 3 + 235	240
242 = 2 * 3 + 236	241
243 = 2 * 3 + 237	242
244 = 2 * 3 + 238	243
245 = 2 * 3 + 239	244
246 = 2 * 3 + 240	245
247 = 2 * 3 + 241	246
248 = 2 * 3 + 242	247
249 = 2 * 3 + 243	248
250 = 2 * 3 + 244	249
251 = 2 * 3 + 245	250
252 = 2 * 3 + 246	251
253 = 2 * 3 + 247	252
254 = 2 * 3 + 248	253
255 = 2 * 3 + 249	254
256 = 2 * 3 + 250	255
257 = 2 * 3 + 251	256
258 = 2 * 3 + 252	257
259 = 2 * 3 + 253	258
260 = 2 * 3 + 254	259
261 = 2 * 3 + 255	260
262 = 2 * 3 + 256	261
263 = 2 * 3 + 257	262
264 = 2 * 3 + 258	263
265 = 2 * 3 + 259	264
266 = 2 * 3 + 260	265
267 = 2 * 3 + 261	266
268 = 2 * 3 + 262	267
269 = 2 * 3 + 263	268
270 = 2 * 3 + 264	269
271 = 2 * 3 + 265	270
272 = 2 * 3 + 266	271
273 = 2 * 3 + 267	272
274 = 2 * 3 + 268	273
275 = 2 * 3 + 269	274
276 = 2 * 3 + 270	275
277 = 2 * 3 + 271	276
278 = 2 * 3 + 272	277
279 = 2 * 3 + 273	278
280 = 2 * 3 + 274	279
281 = 2 * 3 + 275	280
282 = 2 * 3 + 276	281
283 = 2 * 3 + 277	282
284 = 2 * 3 + 278	283
285 = 2 * 3 + 279	284
286 = 2 * 3 + 280	285
287 = 2 * 3 + 281	286
288 = 2 * 3 + 282	287
289 = 2 * 3 + 283	288
290 = 2 * 3 + 284	289
291 = 2 * 3 + 285	290
292 = 2 * 3 + 286	291
293 = 2 * 3 + 287	292
294 = 2 * 3 + 288	293
295 = 2 * 3 + 289	294
296 = 2 * 3 + 290	295
297 = 2 * 3 + 291	296
298 = 2 * 3 + 292	297
299 = 2 * 3 + 293	298
300 = 2 * 3 + 294	299
301 = 2 * 3 + 295	300
302 = 2 * 3 + 296	301
303 = 2 * 3 + 297	302
304 = 2 * 3 + 298	303
305 = 2 * 3 + 299	304
306 = 2 * 3 + 300	305
307 = 2 * 3 + 301	306
308 = 2 * 3 + 302	307
309 = 2 * 3 + 303	308
310 = 2 * 3 + 304	309
311 = 2 * 3 + 305	310
312 = 2 * 3 + 306	311
313 = 2 * 3 + 307	312
314 = 2 * 3 + 308	313
315 = 2 * 3 + 309	314
316 = 2 * 3 + 310	315
317 = 2 * 3 + 311	316
318 = 2 * 3 + 312	317
319 = 2 * 3 + 313	318
320 = 2 * 3 + 314	319
321 = 2 * 3 + 315	320
322 = 2 * 3 + 316	321
323 = 2 * 3 + 317	322
324 = 2 * 3 + 318	323
325 = 2 * 3 + 319	324
326 = 2 * 3 + 320	325
327 = 2 * 3 + 321	326
328 = 2 * 3 + 322	327
329 = 2 * 3 + 323	328
330 = 2 * 3 + 324	329
331 = 2 * 3 + 325	330
332 = 2 * 3 + 326	331
333 = 2 * 3 + 327	332
334 = 2 * 3 + 328	333
335 = 2 * 3 + 329	334
336 = 2 * 3 + 330	335
337 = 2 * 3 + 331	336
338 = 2 * 3 + 332	337
339 = 2 * 3 + 333	338
340 = 2 * 3 + 334	339
341 = 2 * 3 + 335	340
342 = 2 * 3 + 336	341
343 = 2 * 3 + 337	342
344 = 2 * 3 + 338	343
345 = 2 * 3 + 339	344
346 = 2 * 3 + 340	345
347 = 2 * 3 + 341	346
348 = 2 * 3 + 342	347
349 = 2 * 3 + 343	348
350 = 2 * 3 + 344	349
351 = 2 * 3 + 345	350
352 = 2 * 3 + 346	351
353 = 2 * 3 + 347	352
354 = 2 * 3 + 348	353
355 = 2 * 3 + 349	354
356 = 2 * 3 + 350	355
357 = 2 * 3 + 351	356
358 = 2 * 3 + 352	357
359 = 2 * 3 + 353	358
360 = 2 * 3 + 354	359
361 = 2 * 3 + 355	360
362 = 2 * 3 + 356	361
363 = 2 * 3 + 357	362
364 = 2 * 3 + 358	363
365 = 2 * 3 + 359	364
366 = 2 * 3 + 360	365
367 = 2 * 3 + 361	366
368 = 2 * 3 + 362	367
369 = 2 * 3 + 363	368
370 = 2 * 3 + 364	369
371 = 2 * 3 + 365	370
372 = 2 * 3 + 366	371
373 = 2 * 3 + 367	372
374 = 2 * 3 + 368	373
375 = 2 * 3 + 369	374
376 = 2 * 3 + 370	375
377 = 2 * 3 + 371	376
378 = 2 * 3 + 372	377
379 = 2 * 3 + 373	378
380 = 2 * 3 + 374	379
381 = 2 * 3 + 375	380
382 = 2 * 3 + 376	381
383 = 2 * 3 + 377	382
384 = 2 * 3 + 378	383
385 = 2 * 3 + 379	384
386 = 2 * 3 + 380	385
387 = 2 * 3 + 381	386
388 = 2 * 3 + 382	387
389 = 2 * 3 + 383	388
390 = 2 * 3 + 384	389
391 = 2 * 3 + 385	390
392 = 2 * 3 + 386	391
393 = 2 * 3 + 387	392
394 = 2 * 3 + 388	393
395 = 2 * 3 + 389	394
396 = 2 * 3 + 390	395

21

090872-122 B 2458V.11 FRANKGOOSSENS

```

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 .

```

22

090872-122 B 2458V.11 FRANKGOOSSENS

```

3
2460656 2460804 25
1150 4275

```

Memory for data [2;2-37]

<u>Core memory</u>		
	<u>Bits per word</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

```

ALGOL-60      (3.0)      08/08/72  12.06 HRS      PAGE 1
00** "BEGIN""BOOLEAN""ARRAY" A(1:1000000);
      "INTEGER" I;
      "FOR" I:=1 "STEP" 1 "UNTIL" 1000000 "DO" B(I/1):="TRUE"
"END"
      FINIS GEN.BY FOR-CARD
  
```

```

ALGOL-60      (3.0)      XXALGOL      08/08/72  12.06 HRS      PAGE 1
LINE 0  PROGRAM BEGINS (MESSAGE)
LINE 3  PROGRAM ENDS (MESSAGE)
LINE 3  SOURCE DECK ENDS (MESSAGE)
      THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE
      FILE
  
```

```

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
000010 011232 0000 000000 0000 000001 6144 /11775 0446 011240 +8.000000000000000"+005
+1.000000000000000"+000

THIS BLOCK WAS CALLED FROM LINE 0 IN ALGOL60
  
```

```

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.L60.
12.07.12.CP 600.25H SFC.
12.07.12.PP 013.121 SEC.
  
```

090872-100 A 2458V.7 JACKALANEN

A2458V.7, JACK ALANEN

```

1 'BEGIN' 'BOOLEAN' 'ARRAY' B[1:800000]; 'INTEGER' I;
2 'FOR' I:=1 'STEP' 1000 'UNTIL' 800000 'DO' B[I]:=TRUE;
3 PRINTTEXT("800000 ELEMENT BOOLEAN ARRAY NOT TOO LARGE")
4 'END'

```

090872-100 A 2458V.7 JACKALANEN

2

800000 ELEMENT BOOLEAN ARRAY NOT TOO LARGE

090872-100 A 2458V.7 JACKALANEN

3

```

0 94184 25
1 94184 29748
2 85

```

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

ALGOL-60 (3.0) 08/08/72 14.15 HRS PAGE 1

```

0000 "BEGIN" "REAL" "LARGE"
      "LARGE"="1.234" "OUTREAL" (51 * LARGE)
"END"
      FINIS GEN.BY FOR-CARD

```

ALGOL-60 (3.0) XXALGOL 08/08/72 14.15 HRS PAGE 1

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.VELOMIK
14.15.07.VELDM+HR9500.
14.15.07.ALGOL.
14.15.14.FILE OPENED---OUTPUT
14.15.14.FILE OPENED---L60
14.15.17.
14.15.17.CP 000.037 SFC.
14.15.17.PP 003.067 SFC.

```

090872- 24 A 2458V.4 JACKALANEN 1

A2458V.4, JACK ALANEN

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

LARGE÷1.234,400; PRINT(LARGE)

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.

AL60L-60 (3.0) 08/08/72 12.08 HRS PAGE 1

00** "BEGIN" 39:"G0T0"39"ENDH"
FINIS GEN-BY EOR-CARD

AL60L-60 (3.0) XXAL60L 08/08/72 12.08 HRS PAGE 1

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

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

08/08/72 GROMINGEN SCOPE 3.3 SUM 279 24/01/72
12.07.59.VELDH3N
12.07.59.VELDH.NH9500.
12.07.59.AL60L.
12.08.00.FILE OPENED---OUTPUT
12.08.03.FILE OPENED---L60
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.

ALGOL-60 (3.0) 08/08/72 12.08 HRS PAGE 1

```
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

ALGOL-60 (3.0) 08/08/72 12.08 HRS PAGE 1

```
LINE 00 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.I.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 011346 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 /747 776707 4000 011222

```

THIS BLOCK WAS CALLED FROM LINE 0 IN ALGOLHUN

```

0A/0A/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72
12.0A.04.VELDH30
12.0A.05.VELDH.NP9500.
12.0A.05.ALGOL.
12.0A.05.FILE OPENED---OUTPUT
12.0A.0A.FILL OPENED---LGO 305
12.0A.10. 0
12.0A.10.LGO.
12.0A.20.CP 000.434 SEC.
12.0A.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.

ALGOL-60 (3.0) 08/08/72 12.08 HRS PAGE 1

```

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

```

ALGOL-60 (3.0) 08/08/72 12.08 HRS PAGE 1

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

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. VELDHUIS
12.08.20. VELDHUIS. VELDHUISZEN
12.08.20. ALGOL
12.08.21. FILE OPENED---OUTPUT
12.08.21. FILE OPENED---LGO
12.08.28. 0 --286
12.08.28. LGO.
12.08.34. CP 000.444 SFC.
12.08.34. PP 009.647 SFC.

090872- 26 A 2458V.6 JACKALANEN 1

A2458V.6,JACK ALANEN

```

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

```

090872- 26 A 2458V.6 JACKALANEN 2

ER 502 1 -0

090872- 26 A 2458V.6 JACKALANEN 3

```

0      38911 38911 25
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.

ALGOL-60 (3.0) 08/08/72 12.08 HRS PAGE 1

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

ALGOL-60 (3.0) XXALGOL 08/08/72 12.08 HRS PAGE 1

```
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

*1.000000000000000000
END OF ALGOL RUN *V3.0*

08/08/72 GRONINGEN SCOPE 3.3 SUM 279 24/01/72
12.08.39.VELDH3R
12.08.39.VELOH.NR9500. VELDHUYZEN
12.08.39.AL60L.
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):=1;
3   'FOR' E(1):=0 'WHILE' I<10 'DO' I:=I+1;
4   PRINT(E(2))
5 'END'

```

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").

ALGOL-60 (3.0) 08/08/72 12.08 HRS PAGE 1

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

ALGOL-60 (3.0) TEST 08/08/72 12.08 HRS PAGE 1

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.VELDH35
12.08.35.VELDH.NR9500.
12.08.35.ALGOUL.
12.08.35.FILE OPENED---OUTPUT
12.08.36.FILE OPENED---LG0
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.

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

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

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

14.46.55.VELDH20

14.46.56.VELDH.NR9500.

14.46.56.ALGOL.

14.46.57.FILE OPENED---OUTPUT

14.47.00.FILE OPENED---LG0

14.47.02. 0 135

14.47.02.LG0.

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" A; "OUTREAL"(G,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

*0.00000000000000000000
END OF ALGOL RUN *V3.0*

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

```

14.46.40.VELDHZ
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 SEC.
14.46.55.PP 011.238 SEC.

```

ALGOL-60 (3.0) 08/08/72 14.17 HRS PAGE 1

```

00** "BEGIN" PROCEDURE" PRINT(X) "REAL" X "OUTREAL(61-X) :
PRINT(1/0)
"END"
FINIS GEN BY EOM-CARD

```

ALGOL-60 (3.0) XXALGOL 08/08/72 14.17 HRS PAGE 1

```

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.ALGO.
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:=600; PRINT(A*B);
2 A:=B:=600; PRINT(A*B);
3 PRINT(1/0)
4 'END'
```

090872- 17 ----- A 2458V.1 JACKALANEN 2

+ .1776646197514*+629 +.6188692094765*-616 +.1776646197514*+629

090872- 17 ----- A 2458V.1 JACKALANEN 3

```

0 5 25
59265 59265
33 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.

ALGOL-60 (3.0) 08/08/72 14.17 HRS PAGE 1

```
0000 "BEGIN" "PROCEDURE" PRINT(X); "REAL" X; OUTREAL(S,X);
      "INTEGER" I; "REAL" A; I:=1; A:=2.0; PRINT(I "DIV" A)
      "END"
```

FINIS GEN.BY EOR-CARD

ALGOL-60 (3.0) XXALGOL -- 08/08/72 14.17 HRS PAGE 1

```
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---L60 41
14.17.05. 0
14.17.06.CH 000.042 SEC.
14.17.06.PP 005.234 SEC.
```

090872- 22 A 2458V.2 JACKALANEN 1

A2458V.2, JACK ALANEN

'BEGIN' 'INTEGER' 1; 'REAL' A; 11=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";
```


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.

ALGOL-60 (3.0) 08/08/72 14.16 HRS PAGE 1

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

ALGOL-60 (3.0) XXALGOL 08/08/72 14.16 HRS PAGE 1

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

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 -J.0000000000000000000
UV 000000 -J.0000000000000000000
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

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

```
14.16.33.VELDHU
14.16.33.VELDH.NR9500.
14.16.33.ALGOL.
14.16.40.FILE OPENED---OUTPUT
14.16.42.FILE OPENED---LGO
14.16.43. 0 129
14.16.43.LGO.
14.16.44.CP 000.247 SFC.
14.16.45.PP 006.445 SFC.
```

ALGOL-60 (3.0) 08/08/72 14.17 HRS PAGE 1

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

ALGOL-60 (3.0) XXALGOL 08/08/72 14.17 HRS PAGE 1

```
LINE 0 PROGRAM-BEGIN (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 012541 0000 001335 0001 012454
```

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.VELDH1*
14.17.24.VELDH:NR9500.
14.17.24.ALGO.
14.17.25.FILE OPENED---OUTPUT
14.17.27.FILE OPENED---LGO
14.17.28. 0 126
14.17.28.ALGO.
14.17.33.CP 000.246 SEC.
14.17.33.PP 005.201 SEC.
```

090872- 23 A 2458V.3 JACKALANEN 1

A2458V.3 JACK ALANEN

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

090872- 23 A 2458V.3 JACKALANEN 2

+0 -.1776646197514,+629

090872- 23 A 2458V.3 JACKALANEN 3

0 35205 25
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).

