903 ALGOL Maintenance

This document gives details of changes made to the 903 ALGOL system since Issue 3 (about January 1967). It includes all Translator changes made since C.A.P. (Computer Analysts and Programmers) finished work on the Translator.

The document starts with a summary list of the changes. The numbering starts arbitarily at Change 30.

Summary of Changes

Change number		Brief Description
30	Translator	Large integer constant treated as real
31	Translator	Declaration following read or print gives error now
32	Translator Interpreter	Correct code now given for Formal switch with subscript used as Actual parameter to a call by name parameter /α.ℓ.ℓ./ parameter
33	Translator	Characters " and ← printed with correct parity now
34	Interpreter	Use of pointer FP abolished except in code
35	Translator	all own code procedures now picked up correctly from library tape.
36	Translator	(not implemented)
37	Translator	if A>B than print A+B else how translated correctly again
38	Interpreter	Input of 0.00 and $10_{10}-24$ now gives exactly zero
39	Interpreter	Program run on level 4
40	Interpreter	Change to Character input-output allows plotter etc.
41	Translator	Type procedure on its own gives error
42	Translator	Easy to add names to "built-in" namelist
43	Translator	Detect illegal character in parameter comment.
44	Translator	: = detected as error in read or print
45	Translator	Two commas in print statement not detected as error

Change Number		Brief Description
46	Interpreter	After error No. 18 to 14
47	Interpreter	Correction in print string
48	Trdes. Interp Array access changes	

All these changes except number 36 and 46 incorporated in

Issue 4.

49	*Dump* ~~Dump~~ facility and "freeze namelist".	Interpreter Translator
50	Formal procedure parameterless and with parameters confused, ~~Prench~~ correct by changing ~~Proein~~ *PRAMCH*	Translator
51	~~Aretan~~ errors corrected *ARCTAN*	Library Interpreter
52	Library scan, avoid error of library procedure too big, but not wanted anyway	Translator
53	Change COMPIL to allow 16K (LG) system	Translator
54	Tidy up Issue 4 mistakes	Translator Interpreter
55	Wrong code produced for parameter will array plus	*Translator*

Algol change 30

Fault   X:= 1234567;

gives error 8, as the constant is treated
as an integer.

No reason why constant too big for integer
should not be treated as real (unless change
too difficult)

Correction

Change tests in Translator rountine NUMBER
(Page 23 of flow charts)

For number without point or 10, if > 131071
branch to section to process real constants.

Flowchart Change

Translator flow page 23   9.3.67.

Coding Change

in number, OKQ

change 8 FAll - 8 to 8 FIN
       9 FAll - 8 to 9 FIN

Consequent Changes

None

Fault

    begin integer n1;

        real a,x;

    read n1,x;

        procedure P(b): real (b);

    Not detected as error.

Correction

    In block OUT of translator
    in READ, PRINT
    set DECSTA to statement level

Flowchart change

    Page 123 Translator flow charts
    after COMPIL [INOUT 20]
    change SV:= /0   0   to
    DECSTA:=   SV:= /0   0

Coding Change   Vol 3 Translator

    Block OUT, label PRINT   and add DECSTA to globals
    after   5 SV
    insert   5 DECSTA

Consequent changes

    Extra store. 1 location only.

Algol Change 32

Fault

Corrupt code produced from program with
Formal switch with subscipt used as actual parameter to
a procedure with label parameter

e.g. begin   switch S: L1, L2

procedure P(A); value A; label A; goto A;

procedure Q(B); switch B; P(B (2));

Q(S);

Correction

CAP corrected Translator in December, 1966. New Translator
now gnerates new *cord* function 5 (INDFS) (at the call of P
in procedure Q above)

Interpreter must take action on INDFS

Documentation

Pord Brand Manual updated with description of INDFS action

Flow chart

Page 26)
Page 4) of Interpreter Flow charts

Coding

1) Label FBAJ+5
   Change 8 SPARE to 8 INDFS

2) Label INDSFL
   after 8 ENFAIL
   insert (L)

   INDFS   11 FINDFP
            8 FINDFP +1
            8 ; +2 (ADDRESS := CONTENT of 3N + FP)

Consequent Changes

Storage 3 extra words in Interpreter

In the example above
    procedure P(A) will be translated
$\begin{cases} PE \\ GTF\ Bn,1 \\ RETURN \end{cases}$

procedure Q(B) will be translated
$\begin{cases} PE \\ TIC\ (+2) \\ INDFS\ Bn,1 \\ CF \end{cases}$

Procedure call Q(S) will be
    translated
$\begin{cases} TICA \\ CF \end{cases}$

Algol Change 33

Fault

Characters " and ⟵ are printed with wrong parity on
output from Translator and Interpreter

Correction   Alter TABLE
                    in Translator and Interpreter

Documentation

No change

Coding

In Translator and Interpreter Block TABLE
at section beginning /4 160
        alter /4 2082 to 4 2082

at section beginning /11 56
        alter /11 7363 to .11 7363

Consequent Changes

None

Fault

The store location FP was not updated correctly on certain
cases of ~~unit~~ exit from <u>for</u> ~~lessons~~ loops and procedures

Correction

Abolish FP as far as ~~pads~~ pords are concerned i.e. FP will only
be updated on entry to a machine code procedure.

Elsewhere the value of FP is always found from the ~~stall~~ stack
entry on ~~FP~~ via EP

Documentation

Pord Manual already updated

Flow

Interpreter sheets 2, 12, 17, 13

Coding

    alter   FINDFP
             EXEÇUT
             RETURN
             PE
             GTL4

Consequent changes

Total coding reduced by 11 locations

Fault

Machine code procedure "added to" library tape was not
copied onto end of pord tape

Correction   (Translator only)

The "topmost" entry in the namelist was not being checked
(i.e. the last *thing* added to the namelist in the first
block of the program)
The test to complete ~~running~~ *scanning* the namelist in ENDPRO must
be altered.

Documentation

None

Flow

No Change

Coding

In ENDPRO alter NOT + 4 (9 Loop)
     insert 7 Loop

Consequent changes

Total coding (Translator) increased by 1

Not Implemented

Improvement

Make translator library soon ignore loader stop code, and detect special character (erase) as end of library tape.

Convenience when producing library tape it will not be necessary to remove loader stop codes, by copying on a flexowriter.

This will also prevent the program going wild if a loader code $>$ 7 is punched on tape i.e. a mispunching caused "parity bit" to be one.

Coding

In ENDPRO

after   JOIN ≠ 8 (≠5 0)
   insert 2 ≠ 7
            9 OUTEND
after L1-2 (8 test)
Change 8 OUTEND to 8 NEXT

This is not adequate

Consequent changes

Total translator code increased by 2

Do not implement in current changes as Algol Manual describes old system.

Fault

　　　With translator of December 1966 the construction

　　　　　if A>B then print A+B else B; = B+1,

　　　is translated wrongly

Cause

　　　In attempting to allow:
　　　　　print if A>B then C else D;
　　　this error has been introduced

Correction　　(Translator only)

　　　1) Put test in IF
　　　　　if MREAD or MPRINT ≠ 0
　　　　　　then FAIL

　　　2) Alter ELSE back to original ie call INOUT (2)
　　　　　before testing TS

　　　3) see also flow chart pages 117 and 119
　　　　　MPRINT is stacked at (and unstacked at ')'
　　　　　to allow print (if X=0 then A else B)

Documentation

　　　Change Manual and facts card. (Error 67 if misused)
　　　Change to 67 if misused or used in read or print

Flow Chart

　　　Translator　⎰Page 93
　　　　　　　　　　⎱Page 89

Coding

　　　Insert in ELSE call of INOUT (2) immmediately after ELSE

　　　Insert in IF　　check that sum of M + MREAD + MPRINT = 0
　　　　　　　　　　　　(can only be true if each one is zero
　　　　　　　　　　　　as negative values not used.)

Consequent changes　　+2 words of Translator

　　　Programs with print if A>B then C else D will not translate
　　　+ 2 words of Translator

　　　　　print if A = B then C else D is now illegal
　　　but　print (if A = B then C else D) is OK now

Algol Change 38

(Interpreter only)

Fault

1) 0.00 on data tape is input as a small non-zero number (corrected by AIC2)

2) $1_{10}-20$ input as zero correctly but $1_{10}-24$ input as small positive non zero number !

Cause

The DIVTEN routine is sufficiently accurate for all standardised numbers except the special case of zero. Zero divided by 10 gives a very small non zero-result. When standardised this result becomes significant.

AIC2 tested for zero before the division loop in input was started, thus curing 0.00 input. Case (2) occurs because the result goes zero during the loop.

Correction

DIVTEN

Test for zero before using DITEN each time, on RDNM

Flow chart

Sheet 52 (Interpreter)

Coding

At RDNM16 insert   4   W3
                       7   RDNM20

Consequent changes

Total Interpreter engle increased by + 2

Improvement   (interpreter only)

It is highly desirable that Algol programs should be obeyed
from level 4. This makes it easier to control the program
with interrupts.

In particular, a dump facility and a debugging facility
are required, controlled by interrupts.

Details

Entry to start running a program, (at 10) and to continue
after a wait, (entry at 9) cause drop to level 4. Continue
after wait is in the standard 903 systems manner by
continuing at the address held in location 20.

On level 1 interrupt, a routine is entered which places a break
in NXPORD. At completion of the current pord the program
enters a wait stop on level 4. *This program may be stopped to do a dump.*

Allowance has been made for a possible pseudo-"time sharing"
routine, by placing suitable jumps in locations NXPORD and
NXPORD + 1, a program can be entered after each pord is
obeyed. A jump would be placed in NXPORD, to enter the
"time sharing" program after every pord. A jump is placed in
NXPORD + 1 (replacing a dynamic stop) which jumps to the
time sharing routine to complete it if the Algol comes to a
WAIT, or STOP, or final "END" (or halt code on data tape or
run *non* time error stop)

After each entry from NXPORD, the time sharing program must re-
enter at NXPORD + 2.

On completion of all current processing, the time sharing
program must reset the content of NXPORD + 2 and NXPORD + 3
into NXPORD and NXPORD + 1 respectively. NXPORD will be pointed
to by an address in NXPDAD which will occupy "absolute"
location.—

*This facility has not been used up to
now (Aug 68)*

Coding Changes

At NXPORD
at end of EXECUT
at SWAIT and CONTIN
and insert NXPDAD near beginning of Interpreter

Documentation

Must put seperate note in Algol Master file.   ⚹

Consequent changes

Interpreter increase 21 extra words.

Improvement (Essential)

Change in Method of output/input of individual characters.

A table of addresses of routines for input and output of individual characters will be provided. EXECUT will set all these jumps to the entry for paper tape input or output. The input and output to paper tape routines will do a special check to see if the required device is teleprinter. EXECUT will also clear a BUFFER area of 9 locations, one for each possible input device.

Special device routines must be entered by a call of a machine code procedure before their use for input or output. These procedure will place an appropriate address in the ϕtable position for the device.

Consequent changes

About 45 locations added to Interpreter. Description of machine code procedures for using special I/O devices put in Master file.

<u>Fault</u>

Type procedure with parameters standing on its own
as if non-type, is not detected as error. Also
I believe, A + B; as a statement, is not detected as
error.

~~Cause~~ <u>Cure</u>

Test of last delimeter was (;) or <u>begin</u> at Left
Hand Round Bracket ₗ₁ (and at AOP Arithmetic operator)

<u>Flow</u>

Sheet 117 (Translator) LRBRAK
add to test for FAIL 61
FAIL (61) if LASTDL = <u>begin</u> / (;)

(AOP) Sheet 107 Add test before call of EXP (3)
    FAIL (57) if LASTDL = begin / (;)

<u>Coding</u>

in LRBRACK	4 LASTDL	(FAIL IF LAST DELIMETER WAS)
	1 & 756000	
	7 FAIL - 61	(Right round bracket)
	1 & 734000	(;)
	7 FAIL	
	1 & 674000	( ])
	7 FNL	
	1& &754000	
	7 FAIL	("BEGIN")

(in AOP)

4 LASTDL
2 & 216000
7 FAIL - 57   (FAIL IF LASTDL BEGIN)
2 & 130000
7 FAIL - 57   (FAIL IF LASTDL ;)

Algol Change 42

Change

Make it easier to add names to the permanant "built-in"
namelist of the Translator.

Method

Allocate a "fixed" location near beginning of store to
hold address of start of permanent namelist

Documentation

The sheet in the Algol Master File, "How to add names to
the built-in namelist" December, 1966, has been updated
(to May, 1967) with the necessary changes.

Flow Chart

No change

Coding

Insert
  PERM ← 7795
  in option after 8 OUT2$
  in START after 2 + 7795
          to 2 PERM
  replace 4 — 200   by 4 PERM
          1 — NLP+1
          1 — NLP+1

          4 −200 } by 4 PERM
          1 NLP+1 }

Improvement   (Translator only)

Detect illegal characters within parameter comment. This
is important because of way parameter comments are detected,
easy to get one as an error, which does not show up till
many lines later.

Only seperators and letters are allowed in parameter
comment. Delimeter will show up as error following
closely on genuine mistake.

Documentation

No change

Flow Chart

Sheet 16 (TAKCHA)
 updated

Coding

Inserted in LETTER of TAKCHA

Consequent changes

Total Translator coding increased by 4.

Algol Change 44

(Translator only)

Fault

: = not delected as error inside read or print list

Cure

Test in BECOMS for MREAD, MPRINT = O

Flow Charts

Translator Sheet 105

FAIL 28 if MREAD or MPRINT = 1

Coding

Insert after 7 FAIL – 28
```
4 MREAD
1 MPRINT
7 ; + 2
8 FAIL –28
```

Consequent changes

+ 4 words to total Translator

Fault

Comma followed by comma in print statement,
not detected as an error, could produce corrupt
code.

Cure

In SETPRO (which is called from INOUT)
put in test for that there is an identifier
to find.

Flow

Translator Sheet 80

Put in test if $M = 0$
then FAIL 35

Coding

At SETPRO (in INOUT) at SETPRO + 2 insert
7 FAIL --35   .

Consequent Changes

Total Translator coding increased by + 1 words.

Interpreter

Fault in Print Statement

Fault

B (number) in inner string quotes just before 'message'
in print statement, causes the output of extra lines
or spaces, after the 'message' has been on that. *output.*

print *"L B10` ABCDEF* ~~B6~~ *'L``; causes 10 extra*
newlines after
ABCDEF printed

Cause

Sheet 55 of flow diagram.

When N is <u>not</u> negative, and the previous character <u>was</u>
B, N is not set negative. When the 'message' has been
output, the same loop i.e. N not negative will be taken
again.

Cure

Coding

1 = /0 0   } *storing*
5 WS7   i.e. ~~stairs~~ a neg. no. in WS6

4 -- 1
5 WS 6
OST9 ~~OST9~~ 4 WS 7   *PT20*

Changes for different method of array access.

The main purpose of this change is to allow more than 8192 words of object code program. Arrays were referred to by a T.A. operation which pointed to the array pair information contained in the object code. T.A. is limited to a 13 bit address. The total of object code will be reduced slightly (by 2 words per array) but the total amount of store taken for a program + data store will be increased by one word per MAMPS pord. ~~New system.~~

New system
MAMPS will have only one word after it, this will have a real/integer indicator and a relative address in the National Data Area (QAVNDA). Two locations in NDA will be reserved for each array.

For all array references *pord ↑A* T̶.̶A̶.̶ will be replaced by *pord TIA* T̶.̶I̶.̶A̶.̶ which will form the address of the array address pair in QAVNDA.

At run time, MAMPS (which has unchanged format) will store two addresses in the pair of locations in ~~QANDA~~ *QAVNDA* associated with each array.

The first address will point to the actual array with ~~1st~~ *bit* 18 set of the array is real.

The second address will point to the array map, which will be in the run time stack, and will be shared between all arrays declared by the ~~smae~~ *same* MAMPS pord (as before).

The first word of the map will contain the number of dimensions of the arrays. The following words will be the previous layout of the map., that is the whole map will be of the form.

No. of dimensions
Total size
Offset
Lowbound 1
C1
Lowbound 2
C2
Lowbound 3
etc.

The interpreter is changed to allow all this (at MAMPS, INDR, INDA, PE (COPYAR).

Translator Changes

Page 113 of flow chart

In RSBRAK after COMPIL [MAMPS, DIM, ARRCOU]
test if type [I] = real array then LIV:= /0 0 else LIV:= 0
  LIV: = LIV + NDAP (current free location of NDA)
  [COMPIL (LIV)]

followed by LOOP taking the form

LOOP : ADDRES (I):=NDAP (address:=relative NDA address)
      DIM (I): = DIM
      NDAP:= NDAP + 2 (reserve 2 words in NDA)
                    for each array
      I:= I - 4
      ARRCOU:= ARRCOU - 1
      if ARRCOU ≠ 0 goto LOOP
      BCR(D) etc

Page 58 of flow chart

after ARR and if f[J] ≠ 1 COMPIL (TIA,addr. (I))
instead of TA with loader code 1 instead of 2

Page 47 of flow chart

after NAMOK if array then COMPIL (TIA, addr. (I)) with
loader code 1 instead of TA with loader code 2.

Addition

1)   Provide entry point at 16 in Translator so that current contents of Namelist may be "frozen" i.e. incorporated into the permenant namelist.

2)   Provide entry at 14 in Interpreter to enter dump facility at 8000, first testing to see if SP > 8000.

Documentation

Algol Manual change.

Coding

Marked on Issue 4 listing	Interpreter
at 16 put in jump to SETPERM	at 14 put in
put in SETPERM after START	DUMP

Consequent changes

Translator increased by 3 words.

NB note at bottom

Fault

Something of the form

procedure PROC(F); real procedure F;

begin
     X : = SIN(F(X));
end ;

this failed due to an error in PARMCH. The test in
FRED for parameterless procedures apparently branches
the wrong way. This appears to give spurious error.

Connection

Alter the exit from FRED inside PARMCH

Flow

The flow diagram shows the correct decision already,
only the coding is wrong.

Consequent changes

None.   I hope!

N.B.   This change was wrong, original is correct. Translator
        has been patched back to original. Error is still there.

Note that the following program translates correctly:—

procedure PROC(F); real procedure F;

begin
     X := F(x);
     X: = SIN(F(X));
end;

Algol Change 51

Errors in ARCTAN(X) Issue 4 library

Faults

1) if $X < -1$ wrong value given

2) if $X > 6 \times 10^9$ wrong value

Correction

1) Replace the 9 instruction ommitted from Algol
   library version of Arctan.

2) Arctan uses PRIM 31 to add two numbers.
   Alter jump at PRIM 31 + 6 to ensure that result
   always set, PRIM 31 did assume that if 2nd value
   was zero the result was already in correct position
   on stack.

These errors were due to incorporation of new ARCTAN
and other Maths routines from I.C.I. These rountines
were faster by factor of 5 to 10 than the orginals.

Algol Change 52

Library ~~non~~ scan changes

Reason

With Issue 4 the plotter procedures in QAPLT2
were large enough to give trouble to people
compiling in Library mode. Any program with more
than about 50 identifiers declared in the outermost
block failed on the scan, even though the plotter
procedures were not required.

Changes

When the total possible space is filled, instead of
failing immediately, the check is done to find whether
the procedure is wanted. If not wanted, the library
scan continues. If wanted, the FAIL message is given.

Algol Change 53.

Change COMPIL to allow 16K (LG) system

Reason

At present COMPIL, if loader code 3 is compiled,
outputs 6 blanks afterwards. 3 of these represent
a relocatable binary word, increment zero (Code = zero
zero word = zero).

Change so that the zero word is actually output by
PUNGRP, with word = code = 0

Flow

Translator flow sheet 26

Coding

Insert call of PUNGRP

Consequent changes

Total translator coding increased by 2. Paper tape
output is as before.

16K(LG) system gets zero word handed over to loader ,
before the blanks were ignored and therefore the next
word of pord code output by translator was treated as
an increment to the code 3.

Algol Change 54

Tidy up Issue 4 mistakes

Translator

Correct the values in PERM and SP + 1.
Correct ommission and mistake in GETCHA.
Delete output of block number in FAIL.

Interpreter

Correction to make COPYAR work
    (remove sign bit from array address)

Correction to storage of lowbound values.

Fault

SETORIGIN (200, +200)

gave corrupt code.

Reason

Unary plus, although ignored, changes value of E in
Translator.

Cure

In AOP
Set ESAVE (new local workspace)
    ESAVE:=E before using EXP
if unary plus detected set E:=ESAVE

N.B.

Something still wrong here SETORIGIN (200, + 200) now
gives error 5. This is better than corrupt code, but
I don't understand why!

Algol Change 56

Fault

boolean B; real R;

boolean procedure BP(X); real X; BP:=X > 0

B:= BP(R);

always gives *false* result because a PRIM RTOI is
generated before ST to store the boolean result.
(B:= checkb (BP(R)): ~~Checkb gives true correctly~~
~~or false)~~ Checkb give *true* or *false* correctly.

Change

In RRBRAK (Translator)
When Type (I) is boolean   for E = 0
sety TYPBOX: = 0 specifically

*ie say that Boolean
is effectively integer.*

Flow diagram

Page 120

Coding

After EZ in RRBRAK change

2 & 035100   to   2 & 035100
7 32   7; +5

Consequent changes

None.   I hope!

Fault

Translator

When error 40 displayed (end misused)
the error is output repeatedly and indefinitely.

Cause

In processing of END in Translator when Error 40
detected the FAIL rountine is entered.
The FAIL rountine detects that current delimeter is
end and branches to ENT 2 once the error message is
displayed.

ENT2
ENT 2 is on the END routine, before the test for error
40, therefore a closed loop is completed.

Change

In FAIL test for Error 40 (*ENO* = 40)
before branching to ENT 2

Flow diagrams

Page 8 and Page 90

Coding

Insert global label END40F in END rountine.
Insert test for END = 40 in FAIL and branch
to global label END 40F

Put in to Issue 5 but fault still comes up!

CAP — COMPUTER ANALYSTS & PROGRAMMERS LTD

62-63 Queen Street   London E.C.4.

Our ref: DGN/BCP   Your ref:

City 8427

Norman Spink Esq.,
Scientific Computing Division,
Elstree Way,
BOREHAMWOOD,
Herts.

21st March, 1967.

Dear Norman,

903 ALGOL

Thank you for sending me the flowcharts for the 903 Algol interpreter. I wonder if you would mind sending me a printout of the translator coding because the copy is all overwritten with the changes introduced last December? Ideally, I should like to have listings of the issue two translator and interpreter.

Yours sincerely,

Don

Don Hunter.

Directors: A. d'Agapeyeff   C. Strachey   B. J. Gibbens

visit "end"s /   Cut out output of blocks now from Translator and
unkept mode   listing   Interpreter /

NB   PAT beginning of Translation output ≠ OO ✓
       blocks on the pencil   (Translator)

Alter loader to test for CS before storing	Interpreter	
Input-output changes to allow special devices L.	Interpreter	✓
Changes to array access	Interpreter Translator	✓
Detection of type/procedure used as non-type	Translator	Trans) ✓
Large integer constant treated as real	Trans	Loaded ✗rejected
Detection of illegal char in inner string	Trans	Loaded ✓
Sloveman ~~Sloveman~~ Add Buffer Advance Decode QAPLOT, DENWUNE etc to library namelist	Translator	✓)
Add QAPLOT, Buffers to library type	Library	✓ no
Add Buffer Advance Decode to lib top Sloveman ~~Sloveman~~	Library	✓
Check that (⋮≡) is not in print list	Trans	✓

Correction changes to Translator of December 66 suggested by D.H. a	Trans	Re Corr coded ✓
INOFS put into Interpreter	Interp	Documented ✓
Alter array permanent name list is allowed for in START	Trans	✓
Change to level 4 and alter PARTD	Interp	✓ ✓
Dump program NB Interpreter must stop interpreter at NXPORD allow content q Input of numbers < 10⁻²¹	Interp	✓
	Interp	✓ Documented
AICR incorporated in Interpreter	Interp	✓
in READ, PRINT put DECSTA:=/00 to cure declaration following read not deleted	Trans	Documented ✓
Possible shorter Boolean primitives	Interp	Coded ✓
920 Algol Mods	Interp Trans	)

(Error No 14 instead of error no 18)

Detect open string
quote in RDNM
instead of close string

New		Intpps	✓
Put in new double length Divide			✓
Make newline follow table at run time	Trans		✓
Cut out references to VBHR from OT CHA 8 cores saved?	Intp	(? 8)	✓
Translator Error messages and all except body output in Teleprinter	Trans		✓
Replace S 32 and S 33 in Translator by S OUT & S OUT2 & insert globals	Trans	(8)	
Change library scan to recognise erase as end of til tape and "ignore" ladder stop code	Trans		
Change TABLE for output of " and ←	Trans Intp	Documented	✓ ✓
Possibly in Intp rule 1 or @ cause copying of table on date line	Intp		✓ ✓

ISSUE 5 Tests   19 June 68

BGIRAH   Translated and Run   (for 11 min)   ✓
  PARACH   "   "   ✓
  INTRNR   ".   ".   ✓

Test of library scan
        OK for large number of declarations
    (60 declarations in outer block)   ✓
              if SIN picked up

    Fails if WAY (Plotter) picked up   ✓

    ' JAW1   ✓

    Addition of numeric list as in Manual   ✓
              (entry 16)
        Report Mode   ✓

    Fault found, corrected by changing
      FRED in PARACH back to original
      SQRT (ABS (I)) failed, OK now   ✓

    let , procedure K(F); real procedure F ;
          X := SIN(F(X)); s
    now fails again

    Error 40 not output repeatedly now   ✓

    Unary +   (?)
      SQRTGIN(+10,20) still gives error

Changes for 16 K Algol

Array changes already done
Pages 113, 47, 58, 45 of flow charts

Other references to ~~TA~~ compilation of TA

Pages 45   (array or string)
      47   (array or procedure)
      58   (array only)
      121 (string only)
These are the only references to (compile (TA))
in the flowcharts