

Kidsgrove ALGOL Run Time Error Messages

0. Explanation

The following is reproduced from the original EE documentation of the Kidsgrove ALGOL run time system, with minor redactions for consistency of format, and to correct obvious errors.

1. Failures at run time

When running ALGOL programs which have been compiled using the Kidsgrove Compiler, there is a variety of checks made on the object program at run time. These checks have been built into the object program by the compiler and function automatically. The following table shows what these checks are and the action taken when failure occurs. These messages are output to the line printer. Each consists of of a short character string and an associated numerical value.

Failure	Reason and message
Real to integer conversions	The limit of 39 bits in any ALGOL conversion is checked every time a real number is converted to an integer. In the event of failure, the character string 'HX' and the value of the floating point number are output.
Non-integral arguments to ÷	The Kidsgrove ALGOL ÷ is defined for integral arguments only. In the event of failure, the character string NOT I and the value of the real argument are output.
Overflow	This is checked on the leftmost assignment of the left part list (excluding those generated as the result of <u>for</u> clauses). The character string is VR and the value of the expression which caused the overflow to be set is output.
Storage capacity	This is checked whenever a procedure is entered, an array is declared, or an array is value copied. The character string is STORAGE CAP and the value is the amount of core store required to get past the failure point.
Invalid use of expression called by name	When a parameter is called by name (and its corresponding formal parameter appears in a left part list) it must either: (1) be a simple or Subscripted variable, or (2) correspond identically in type with the formal parameter. Violation of this semantic rule causes failure. The character string is SECT. 4. 7. 5. 2. and the value is 0.
Array by value conversions	When an array is called by value it is copied into the procedure data space, with appropriate type conversion. The character string is A-B TRANSFER and the value is 0.
Switch subscript check	A switch subscript i must satisfy $1 \leq i \leq n$, where n is the number of switch elements. If it is outside this range then failure will occur. The character string is SW. →UNDEFINED and the value is the subscript value. The failure number is the number of elements in the switch.
The ↑ operator	This operator is undefined for various combinations and values of arguments. For failure the character string is SEC. . 3. 3. 4. 3. and the value indicates the particulars leading to failure. 1 Exponent negative, zero mantissa (in $x \uparrow i$) 2 Exponent zero, zero mantissa (in $x \uparrow i$) 3 Exponent negative or zero, zero mantissa (in $x \uparrow y$) 4 Negative mantissa (in $x \uparrow y$) 6 Overflow in $x \uparrow i$ 7 y too big in $x \uparrow y$ 8 y too big in $i \uparrow y$
Result of exp	If the result of $\exp(x)$ would have been out of range, then failure will occur. The character string is EXP FAIL 1.
All peripheral devices have been closed	If at the end of the program all devices used have not been closed a failure will occur. The character string is GRAND CLOSE FAIL 1 and the value is the number of devices that have been left open. All these devices are then closed. No information is lost.

2. Check Modification

If a program or segment is translated WITHOUT TEST then an overflow check is made only. If WITHOUT TEST is not present, then an entry is made into a checking routine on every assignment.

In non-segmented Kidsgrove ALGOL, this routine prints the value being assigned in octal (so as to give the precise value), and is controlled by the value of VOP244 in the run time control routine. If this store is zero, no printing takes place; but if it is non-zero then its value, followed by the assignment value, are output in the following format:

```
1111      dddddddddddddddd
```

where 1111 is a four digit octal line number held in V0 and dddddddddddddddd is the 16-digit octal assignment value. The value of V0 is incremented by one every time the routine is entered.

In segmented Kidsgrove ALGOL, the output format is as follows, either:

dddd ±D.DDDDDDDDDDD±DD LLLL LE AAAAAAAAAAAAA

or

dddd ±DDDDDDDDDDDDDDDD LLLL LE AAAAAAAAAAAAA

where:

(a) dddd is the line number in decimal (contents of V0P244).

(b) ±D.DDDDDDDDDDD±DD is the decimal value, in floating point, if the number in the assignment is a real number.

(c) ±DDDDDDDDDDDDDDDD is the decimal value of the integer being assigned, and if negative, the minus sign appears.

Certain machine bit patterns can be interpreted as either real or integer quantities. They fall into two groups; the first group represents reals in the range $+1.469368_{10-39}$ to $+2.93876_{10-39}$ and integers in the range $+274877906944$ to $+54975581388$, and the second represents reals in the range -8.507059_{10+37} to -1.701412_{10+38} and integers in the range -274877906945 to -549755813888 . The failure routine, if given a value in either group, will interpret it as an integer. It follows that a real quantity which happens to lie in one of the ranges above may, by the failure routine, be erroneously printed as an integer. Since both ranges are at the extreme end of the area of representation of KDF9 floating numbers the problem should rarely, if ever, occur in practice. The number is zero suppressed.

(d) LLLL is the value in decimal of the last label passed.

(e) LE is the decimal value of the level number of the last level passed, that is, the last procedure entered.

(f) AAAAAAAAAAAAA is the twelve character identifier of the segment from which entry to this routine was made. If entry to this routine was not made from one of the segments, then MAIN PROGRAM will be output. This message only appears on the first assignment in any segment or the main program.

Control of the testing routine is provided by the following code procedure, in both segmented and non-segmented Kidsgrove ALGOL.

procedure TRACE(i); value i; integer i;

KDF9 1/0/0/0;

 [i]; =V0P244; EXIT;

ALGOL;

NOTES

- TRACE (0) switches the printing off.
- TRACE (5) switches the printing on, starting the output line count as 5.
- The line count may be changed at any point in the program.
- The TRACE mechanism is normally switched off, and must be switched on if required.
- The output stream is stream 30₈, and is output via OUT 8.
- If a program has the procedure TRACE present, and it is required for this procedure to be ineffective, then this is achieved by recompiling the program 'WITHOUT TEST'.

3. I/O failure messages

All I/O failures are typed on the console Flexowriter and are in the form:

KALGOL AT R last label passed FAIL failure number

IN descriptive string

VALUE informative number OCTAL

where, when the descriptive string is SP . /READ/device no./:

failure number	informative number	Reason for failure
1	exponent	exponent out of range
2	exponent	10 or . in exponent
3	number read so far	more than one decimal point
4	number read so far	More than one sign
5	number read so far	sign not at the beginning of the number
6	-1	+ - 10 . without a digit
7	number read so far	no digit after the decimal point
8	number that set overflow	number out of range
9	terminating basic symbol	invalid terminator

The **write** procedure may also give warning messages which do not result in a failure. These take the form:

WF *n D m*

where *m* is the device number and *n* is the failure number, as follows:

<i>n</i>	Reason for failure
2	Negative number with empty sign in format
3	No digits in the format
4	More characters than bits in the field
5	Digit before start of initial spaces
6	Digits left over at decimal point position
7	More digits to left of decimal point than given in format before roundoff
8	More digits to left of decimal point than given in format after roundoff
9	Only guard digit in layout field
10	No digit after decimal point
11	more than eight characters in exponent
12	More than two, but less than nine, characters in exponent

From the **format** function, only warning messages come out and these are in the form:

I **FOR** *invalid string*

Invalid characters on input are indicated by the message:

I **C** **H** *cdd*

where *dd* is the KDF9 internal character code in octal and *c* is the case marker: *c* = 0 for Case Normal (figures shift) and *c* = 1 for Case Shift (letters shift).

Invalid basic symbols on output are indicated by the message:

O **C** **H** *ddd*

where *ddd* is the KDF9 basic symbol code in octal.