

M. M. Fokkinga

Delft University of
Technology, Delft.

An Algol 60 program is presented which prints literally itself, by use of a procedure OUTS (<string>) only.

The concept program is:

```

begin ZZZZZ; for i := 1, SUCCESSOR(i) while i < 3 do OUTS(' ... (1)
begin ZZZZZ; for i := 1, SUCCESSOR(i) while i < 3 do OUTS( ... (2)
'); end ... (3)

```

where integer procedure SUCCESSOR(i) delivers the value i+1 and has side-effect:

```

if i = 1 then it prints: '
if i = 2 then it prints: '); end

```

} NB. by use of proc OUTS

The problem now is that, due to the declaration of SUCCESSOR, there would appear stringquotes in the string (2) (and other characters which are not allowed in a string).

This is not solved by (depending on the implementation) replacing them by special symbol, because of the second time OUTS is evaluated it will not print the string (2) literally!

The solution is: place the call of OUTS between another begin and end and for every occurrence of a not allowed (or not significant) character X in the string (2), replaceX - - - - -
by'); CHARACTER(x̄,i); OUTS(' - - - - -

where x̄ is a unique code for the character X, and

procedure CHARACTER(x̄,i) is such that, in a straight forward way,

```

if i = 1 then it prints: X
if i = 2 then it prints: '); CHARACTER(x̄,i); OUTS('

```

Now, in the PDP9 Algol implementation the output proc. is OUTS(3, "...") (the 3 denoting the output device; left and right stringquotes equal), and special symbols have to be used for quotes (nl. !Q!), exclamation marks (nl. !E!), new lines (nl. !L!), whereas spaces are significant. Thus we have declared a procedure CHAR(x̄,i) with specifications for x̄ according QUOTES, EXCLAM, RETURN respectively.

At last in the declarations we have used CHAR(\bar{x},i) where possible instead of the special symbol for X and we have inserted a "new-line" before each call of CHAR for lay-out only and adapted its declaration accordingly.

The added sheet is the program and at the same time its own output.

Delft, 3rd November 1972.

M.M. Fokkinga
Delft University of Technology
Dept. of Mathematics
'132 Julianalaan
Delft
The Netherlands.

```

'BEGIN' 'INTEGER'QUOTES,EXCLAM,RETURN,I)
'PROCEDURE'CHAR(X,I);'INTEGER'I,X)
'IF' I=1 'THEN'
'BEGIN' 'IF' X=QUOTES 'THEN' OUTS(3,"|Q|") 'ELSE'
      'IF' X=EXCLAM 'THEN' OUTS(3,"|E|") 'ELSE'
      'IF' X=RETURN 'THEN' OUTS(3,"|L|")
'END' 'ELSE'
'BEGIN' CHAR(RETURN,1);CHAR(QUOTES,1);
      'IF' X=QUOTES 'THEN' OUTS(3,"");CHAR(QUOTES,I);OUTS(3,"") 'ELSE'
      'IF' X=EXCLAM 'THEN' OUTS(3,"");CHAR(EXCLAM,I);OUTS(3,"") 'ELSE'
      'IF' X=RETURN 'THEN' OUTS(3,"");CHAR(RETURN,I);OUTS(3,"");
      CHAR(QUOTES,1)
'END';
'INTEGER' 'PROCEDURE' SUCCESSOR(I); 'INTEGER' I)
'BEGIN' CHAR(QUOTES,1);
      'IF' I=2 'THEN' OUTS(3,"");'END''END'");
      SUCCESSOR:=I+1
'END';
QUOTES:=1;EXCLAM:=2;RETURN:=3;
'FOR' I:=1,SUCCESSOR(I) 'WHILE' I<3 'DO'
'BEGIN'
  OUTS(3,"'BEGIN' 'INTEGER'QUOTES,EXCLAM,RETURN,I)
  ");CHAR(RETURN,I);OUTS(3,"'PROCEDURE'CHAR(X,I);'INTEGER'I,X)
  ");CHAR(RETURN,I);OUTS(3,"'IF' I=1 'THEN'
  ");CHAR(RETURN,I);OUTS(3,"'BEGIN' 'IF' X=QUOTES 'THEN' OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,"
  ");CHAR(EXCLAM,I);OUTS(3,"Q
  ");CHAR(EXCLAM,I);OUTS(3,"
  ");CHAR(QUOTES,I);OUTS(3," 'ELSE'
  ");CHAR(RETURN,I);OUTS(3," 'IF' X=EXCLAM 'THEN' OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,"
  ");CHAR(EXCLAM,I);OUTS(3,"E
  ");CHAR(EXCLAM,I);OUTS(3,"
  ");CHAR(QUOTES,I);OUTS(3,") 'ELSE'
  ");CHAR(RETURN,I);OUTS(3," 'IF' X=RETURN 'THEN' OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,"
  ");CHAR(EXCLAM,I);OUTS(3,"L
  ");CHAR(EXCLAM,I);OUTS(3,"
  ");CHAR(QUOTES,I);OUTS(3,")
  ");CHAR(RETURN,I);OUTS(3,"'END''ELSE'
  ");CHAR(RETURN,I);OUTS(3,"'BEGIN' CHAR(RETURN,1);CHAR(QUOTES,1);
  ");CHAR(RETURN,I);OUTS(3," 'IF' X=QUOTES 'THEN' OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,"");CHAR(QUOTES,I);OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,") 'ELSE'
  ");CHAR(RETURN,I);OUTS(3," 'IF' X=EXCLAM 'THEN' OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,"");CHAR(EXCLAM,I);OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,") 'ELSE'
  ");CHAR(RETURN,I);OUTS(3," 'IF' X=RETURN 'THEN' OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,"");CHAR(RETURN,I);OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,"");
  ");CHAR(RETURN,I);OUTS(3," CHAR(QUOTES,1)
  ");CHAR(RETURN,I);OUTS(3,"'END'");
  ");CHAR(RETURN,I);OUTS(3,"'INTEGER' 'PROCEDURE' SUCCESSOR(I); 'INTEGER' I)
  ");CHAR(RETURN,I);OUTS(3,"'BEGIN' CHAR(QUOTES,1);
  ");CHAR(RETURN,I);OUTS(3," 'IF' I=2 'THEN' OUTS(3,
  ");CHAR(QUOTES,I);OUTS(3,");'END''END'");
  ");CHAR(QUOTES,I);OUTS(3,");
  ");CHAR(RETURN,I);OUTS(3," SUCCESSOR:=I+1
  ");CHAR(RETURN,I);OUTS(3,"'END'");
  ");CHAR(RETURN,I);OUTS(3,"QUOTES:=1;EXCLAM:=2;RETURN:=3)
  ");CHAR(RETURN,I);OUTS(3,"'FOR' I:=1,SUCCESSOR(I) 'WHILE' I<3 'DO'
  ");CHAR(RETURN,I);OUTS(3,"'BEGIN'
  OUTS(3,");'END''END'");

```