executed. Thus in our case F consists of r followed by the program *nil*.) In general, the most important issue during the design was to solve the printing difficulties. For instance, the reader will note the difference between the value stored in (*a*6) and other entries of *a*. In (*a*6) we cannot use a string as in the previous cases since the value we are trying to store has strings already. Thus the use of | | is invoked. The reader should carefully check r to make sure that all details are understood.

3. Discussion

It is trivial to modify r to make it recursive. Then one would have a program which replicates itself on a terminal indefinitely.

Although we have not taken the direction, any optimization version will also be interesting. In this respect, a better problem may be the following:

> Write a self-replicating P such that it does something useful and is of small 'size' (total number of characters in F). (Here the notion of being useful is left to the imagination.)

We invite interested readers to send us copies of their favourite self-replicating programs written in a familiar language. These will be included (with proper credits) in a future anthology we are planning on the subject.

Acknowledgement

I thank Matthijs Kuiper for suggesting this problem to me.

V. AKMAN
Department of Computer Science, University of Utrecht, Budapestlaan 6, P.O. Box 80.012 3508 TA Utrecht, the Netherlands

References

1. J. K. Foderaro and K. L. Sklower, *The FRANZ LISP Manual*. University of California, Berkeley, Calif. (1981).
2. R. Wilensky, *LISPcraft. Norton, New York* (1984).

Correspondence

Algorithms for the Even Distribution of Entities

Dear Sir,

In recent editions of *The Computer Journal* you have published correspondence on algorithms for the even distribution of entities (Compton, Brokate, Elston). In order that readers unfamiliar with the subject should not think this an undeveloped field, the following is a brief review of the major achievements already extant. Many of these works have been published in, or referenced by articles in *The Computer Journal*.

The problem has a particular relevance to computer graphics. This is because drawing straight lines with devices which use the logic of incremental plotters or raster scan tubes requires the even distribution of axial and diagonal moves. All lines that can be drawn must consist of a sequence of such movements, and the straightness of the line depends upon the even distribution of the move types. The sequence of such movements, which represent a straight line, is called its chain code (Freeman, 1970).

In 1965 J. E. Bresenham produced an entirely integer-based algorithm which correctly computed the appropriate move distribution, using nothing more than sign tests and additions. It is still the most regularly implemented line-drawing algorithm, and because of its seminal importance, it is reproduced in the following Pascal formulation.

```
PROGRAM Bresenham (INPUT,OUTPUT)
*constructs the series of 0's and 1's which
produce the*
*best-fit from the origin to (u,v) with u>v)*
VAR u,v,a,b,d,counter: INTEGER;

BEGIN
  READLN(u,v);
  b:=2*v;
  a:=2*u-2*v;
  d:=2*v-u;
  counter:=0;
  WHILE counter<u DO
  BEGIN
    IF d<0 THEN BEGIN
                  WRITE('0');
                  d:=d+b;
                END;
          ELSE BEGIN
                  WRITE('1');
                  d:=d-a;
                END;
    counter:=SUCC(counter);
  END;
END.
```

The algorithm resolves the equal-error anomaly by selecting an upper grid point (thus the line from the origin to (2,1) is 10 and not 01). It generates the 'best-fit' chain code specified by minimising either the vertical or perpendicular error distance (Bresenham, 1963). Bresenham's algorithm has been modified to generate circles and conics (Bresenham, 1977; Pitteway, 1967).

Brons (1974) produced an algorithm for generating chain code by using continued fraction expansion via the division version of Euclid's algorithm (a more sophisticated method than quoted in recent correspondence). This was modified by Archelli and Massarotti (1978) to generate 'best-fit' chain code. Brons (1974) has also produced a grammar for the highly context-sensitive language which encodes the even distribution of entities in one dimension.

Modifications have been made to Bresenham's algorithm to improve its operational efficiency (Pitteway and Green, 1982), and to generate reversible plotter paths (Boothroyd and Hamilton, 1970). More recently, Castle and Pitteway (1985) have produced an algorithm which generates 'best-fit' chain code from the palindromic symmetry inherent in its patterns.

The converse problem, that of recognising chain code as belonging to a particular straight line, has been addressed by Dorst & Duin (1984). They have developed an application of spirographic methods to act as a recogniser and estimator for the accuracy of chain code representations.

The purpose of this note is to describe the solutions to this problem that have already been produced, in order to encourage further novel contributions. It is also interesting to reflect that Compton's original problem was unrelated to the field of computer graphics, but is an application of work already done in that area.

References

1. C. Archelli and A. Massarotti, On the parallel generation of straight digital lines. *Computer Graphics and Image Processing* **7** (1), 67–83 (1978).
2. J. E. Bresenham, An incremental algorithm for digital plotting. *Proc. ACM National Conference* (1963).
3. J. E. Bresenham, Algorithm for computer control of a digital plotter. *IBM Systems Journal*, **4** (1), 25–30 (1965).
4. J. E. Bresenham, A linear algorithm for incremental display of digital arcs. *ACM Communications* **20**, 100–106 (1977).
5. J. Boothroyd and P. A. Hamilton, Exactly reversible plotter paths. *Australian Computer Journal* **2** (1), 20–21 (1970).
6. R. Brons, Linguistic methods for the description of straight lines upon a grid. *Computer Graphics and Image Processing* **3** (1974).
7. C. M. Castle and M. L. V. Pitteway, An application of Euclid's algorithm to drawing straight lines. *Proc. NATO ASI on Fundamental Algorithms for Computer graphic*, pp. 135–140. (1985). Springer-Verlag.
8. C. M. Castle and M. L. V. Pitteway, An efficient structural technique for encoding 'best-fit' straight lines. *The Computer Journal*. (Accepted for publication.)
9. L. Dorst & R. Duin, A framework for calculations on digitised straight lines. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **PAMI-6**, (5) (1984).
10. H. Freeman, Boundary encoding and processing. In *Picture Processing and Psychopictorics*, pp. 241–266. Academic Press, New York (1970).
11. M. L. V. Pitteway, Algorithm for drawing ellipses or hyperbolae with a digital plotter. *The Computer Journal* **10**, 282–289 (1967).
12. M. L. V. Pitteway and A. Green, Bresenham's algorithm with run-line coding shortcut. *The Computer Journal* **25** (1), 114–115 (1982).

Yours faithfully

C. M. CASTLE
Head of Department of Computing and Information Technology, St Mary's College, Twickenham TW1 4SX
M. L. V. PITTEWAY
Professor of Computer Science, Brunel University, Uxbridge UB8 3PH