Two Computing Exercises with Mathematical Overtones

John G. Heuver

It is sometimes necessary to have exercises that demonstrate the need for a sound knowledge of mathematics while teaching the intricacies of computer programming. The programming language I have chosen is Pascal, which was developed by Niklaus Wirth in the 1970s for the specific purpose of teaching programming. Pascal is being maintained and upgraded as a suitable language that works well on present-day computers. It is not overburdened with unnecessary detail. Version 7 is apparently the latest by Borland. One has to keep in mind that in general a high correlation exists between the maintenance of software and its effectiveness. The programming language C is more flexible in the sense that it allows close control of the hardware and the interaction with assembler is easier. The use of pointers in C makes programming more agile. However, to get a good insight in basic programming, Pascal is simpler because it has fewer trappings and is less cumbersome in more instances. I have chosen the following two examples simply because I am personally intrigued by the questions they raise.

The $3x + 1$ Conjecture

The $3x + 1$ problem has been attributed to Lothar Collatz. Pick a positive integer; when it is odd, multiply it by three and add one, and if it is even, divide by two, continuing the process until you end up with one. The sequence 3, 10, 5, 16, 8, 4, 2, 1 is an example. The program that follows performs this task for a number entered from the keyboard. The solution to the question of why the sequence ends with 1 or, simply put, why it ends in finite time is still beyond our present-day knowledge. Note that the entire sequence is not uniquely reversible while this is the case in the second example.

```
program COLLATZ;
var
    a:integer;
    begin
    writeln('Enter a number.');
    readln(a);
    repeat
    if odd(a) then
    a:= 3*a + 1
else
    a:=a div 2;
    write(a:5);
    until a = 1;
    readln;
end.
```

The Problem of Cups and Stones

Barry Cipra (1992, 1993) proposed the following problem. Suppose we have n cups arranged in a circle and k unmarked stones are placed in each cup. Mark these cups 1, 2, 3 . . . , n, clockwise. Pick all the stones out of the first cup, and put one in every subsequent cup moving clockwise, leaving your hand by the cup in which you dropped the last stone. Pick up all the stones in that cup, and start over again. This process is simulated in the program that follows with four cups and one stone per cup. Depending on one's type of computer, the number of cups and stones cannot be increased much beyond 10 and 4, respectively. Otherwise, it takes a long time to watch the stream of output. The questions that this problem raises are (1) why does the process end in accumulating all the stones piled up in the first cup? and (2) how many moves accomplish this process? The first question was answered and the second only solved for the case of two cups. This is not the place to consider the proofs.

```
program CIPRA;
type
container = array[1..20] of integer;
var
    cups:container;
stones, num, a, b, c, startcup, stonesincup:integer;
begin
    b:= 0;
    num:= 4;
    stones:= 1;
for a := 1 to num do
```

```
begin
    cups[a] :=stones;
end;
    startcup:= 1;
repeat
    stonesincup:= cups[startcup];
    cups[startcup] := 0;
for c:= 1 to stonesincup do
begin
    startcup := startcup + 1;
if startcup num then
    startcup := startcup - num;
    cups[startcup]:= cups[startcup] + 1;
end;
    b:=b + 1;
for c:= 1 to num do
    write(cups[c]:5);
    writeln;
until cups[1] = num * stones;
    writeln(b:5,'iterations');
    readln;
end.
```

```
1111;0211;0022;1120;1030;2101;
2011;2002;3100;3010;3001;4000;
11 iterations
```

Conclusion

The problems are meant as exercises in programming techniques while focusing on some of the more subtle questions about the mathematics in the background. The programs can also be used as exercises for designing a file for printable output. In the $3x + 1$ case, a loop can be built to obtain the number of iterations, for example, from 2 to 500, while suppressing the output of the actual sequences. Programs for graphs of mathematical functions are less suitable because they are easier to do with Maple V, the mathematics software package developed at the University of Waterloo, or equivalent software.

Bibliography

Char, B.W., et al. *First Leaves: A Tutorial Introduction to Maple V.* New York: Springer-Verlag, 1992.

Cipra, B. *The Mathematics Magazine.* Problem 1388. The Mathematical Association of America, 1992, p. 56; 1993, p 58.

Engel, A. *Elementary Mathematics from an Algorithmic Standpoint.* Staffordshire, U.K.: Keele Mathematical Education Publications, University of Keel, 1984.

Lagarias, J.C. "The $3x + 1$ Problem and its Generalizations." *The American Mathematical Monthly,* 1985.