What Does This Program Do? (Looping & Arrays)

GBW ACSL - Contest #3

Clayton O'Neill — clayton@oneill.net

2015-2016

How Do Loops Work?

Loops in BASIC have the following form:

```
1 FOR X = 0 TO 30 STEP 10
2    PRINT X
3 NEXT X
```

This program will print 0, 10, 20, 30. The loop starts on line 1. The FOR X part specifies that X is the variable that will change each time the loop is run.

The 0 TO 30 part of line 1 sets the initial value of X when the loop starts, and specifies when the loop will end.

The last part part of line 1, STEP 10, specifies how much will be added to X every time the loop ends. The STEP is optional and defaults to 1.

Finally, on line 3 we see NEXT X. This marks the end of the loop where X is the loop variable. This is important for nested loops.

Nested Loops

This is an example an outer and inner loop, or *nested* loops.

```
1 FOR X = 1 TO 3
2   FOR Y = 1 TO 3
3     PRINT X,Y
4   NEXT Y
5 NEXT X
```

The output will be:

```
1 1 1
2 1 2
3 1 3
4 2 1
5 ...
6 3 3
```


Backwards Loops

The STEP option can also be negative to loop backwards:

```
1 FOR X = 30 TO 0 STEP -10
2   PRINT X
3 NEXT X
```

This will print:

```
1 30
2 20
3 10
4 0
```

Decimal Numbers as STEP

The STEP can take non-whole, or decimal numbers as shown below:

```
1 FOR X = 30 TO 0 STEP -7.5
2    PRINT X
3 NEXT X
```

This will print:

```
1 30
2 20.5
3 11
4 1.5
```

How does the loop decide to end in this case?

Technical Implementation

When a For...Next loop starts, Visual Basic evaluates start, end, and step. Visual Basic evaluates these values only at this time and then assigns start to counter. Before the statement block runs, Visual Basic compares counter to end. If counter is already larger than the end value (or smaller if step is negative), the For loop ends and control passes to the statement that follows the Next statement. Otherwise, the statement block runs.

Each time Visual Basic encounters the Next statement, it increments counter by step and returns to the For statement. Again it compares counter to end, and again it either runs the block or exits the loop, depending on the result. This process continues until counter passes end or an Exit For statement is encountered.

The loop doesn't stop until counter has passed end...[1]

[1] https://msdn.microsoft.com/en-us/library/5z06z1kb.aspx

No Nothing Loops

Loops can result in **nothing** being done!

```
1 FOR X = 30 TO 0
2    PRINT X
3 NEXT X
```

This will print out nothing, because X will be assigned 30, which is greater than 0, so the loop doesn't even run once.

Common Pattern: Checking If Evenly Divisible

One pattern that is common in ACSL question is shown below:

```
1 FOR X = 1 TO 20
2    IF INT(X/2) = X/2 THEN PRINT X
3 NEXT X
```

This program prints all numbers between 1 and 20 that are divisible by two. It does this by dividing the number by 2 and rounding it it towards zero, then comparing that to just the number divided by 2. If those are the same, that means the number divided by two has no remainder.

For example, if X were 3, then the first part of the comparison on line 2 would be, 1, but the second part would be 1.5. Those two are are not equal, so the 3 isn't evenly divisible by 2.

What Does This Program Do: Arrays

- Arrays are variables that hold more than one value.
- Another word commonly used for arrays is *matrix*.
- One way to think about them is as a table like we saw with the adjacency matrix.
- In ACSL you will normally see arrays that either have one or two *dimensions*.
- The number of dimensions is determines how many variables you use to access the array.
- One dimension arrays are accessed, or indexed, with one variable.
- Two dimension arrays are accessed, or indexed, with two variables.
- You'll commonly see arrays paired with loops.

One Dimensional Array Example

The example below initializes the array A using the READ command then counts all array elements that are greater than 2 and prints it out.

```
1  DATA 3, 3, 2, 1, 2, 4, 7
2  FOR X = 1 to 7
3     READ Y
4     A(X) = Y
5  NEXT X
6  C = 0
7  FOR X = 1 TO 7
8     IF A(X) > 2 C = C + 1
9  NEXT X
10 PRINT C
```

Two Dimensional Array Example

Assume that the array A has the values given in the table below. What would this program print?

```
1 FOR X = 1 TO 4
2   C = 0
3   FOR Y = 1 TO 3
4     IF INT(A(X,Y)/2) = A(X,Y)/2 THEN C = C + 1
5   NEXT Y
6   PRINT C
7 NEXT X
```

37	14	10	2
2	3	11	6
21	0	1	8

Bottom left corner is 1,1 (21).

Miscellanous Hints and Tips

- In ACSL problems arrays are sometimes declared using the DIM declaration. You can ignore these. In ACSL BASIC, declaring arrays ahead of time isn't necessary.
- ACSL array problems frequently take the form: Initialize array/Change Array/Summarize Array.
- When you are given a two dimensional array in a problem, be very careful about where the two axes start.