General Computer Science for Engineers CISC 106 Lecture 13

Roger Craig
Computer and Information Sciences
3/13/2009

Lecture Overview

- Midterm Review (March 17, 4:30pm to 7pm) in Gore Hall 318
- Midterm review sheet will be posted online
- Go over labs and class notes
- While loop with complex conditions

Matrix looping cont.

Note: this code is found on the course website under Examples

```
function [output] = change8to9( inputMatrix)
%changes the first 8 in each row to a 9 in a matrix and returns that altered matrix
%input: matrix
%output: matrix
    lengths = size(inputMatrix); %get size of matrix and store in 1x2 array
    fprintf('size of the matrix m by n is ')
    disp(lengths)
    for m = 1:lengths(1)
        for n = 1:lengths(2)
            if  (inputMatrix(m,n) == 8) %if that position equals an 8
                inputMatrix(m,n) = 9;  %change that position in the matrix to a 9 using an assignment statement
                break; %break out of this inner FOR loop
            end  %ends the IF
        end  %ends the inner FOR loop
    end   %ends the outer FOR loop
    output = inputMatrix; %assign the matrix to the output value (or in other words, "return the (possibly) altered
      matrix")
end %end the function
```

For loop and break statement

- break; lets us stop a for loop before it would normally stop
- However, break; statement makes for unclear code and should be avoided
- What to use instead?
- While loop

While Loop

- while (logical expression)

  %block of code to execute

  end

- First, the logical expression is tested.
- If it is true, we do one iteration of the loop.
- If it is false, we go to the next line after the 'end' statement.
- After every iteration of the loop, we test the logical expression again to see if it is still true.

While loop

- How many times will these run?

```
while (true) %here true is the matlab constant 'true'
   %do something
end
%it runs forever


while (false) %here false is the matlab constant 'false'
    %do something
End
%never goes inside the while loop
```

- If we use a counter, we can do the same thing as a FOR loop

While loop with counter

```
for counter = 1:2:11
   disp(counter);
end
```

- The following while loop produces the same exact output

```
counter = 1;
while(counter <= 11)
    disp(counter);
    counter = counter + 2;
end
```

ChangeOnlyOne8to9.m

Warning , this code will produce an infinite loop on purpos

```
function [output] = changeOnlyOne8to9( inputMatrix)
%changes only the first 8 to a 9 in a matrix and returns that altered matrix
%(The matrix is traversed row by row.)
%input: matrix
%output: matrix
    lengths = size(inputMatrix); %get size of matrix and store in 1x2 array
eightNotSeenYet = true; %initialize flag to true
    m = 1; %initialize m to 1
n = 1; %initialize n to 1


    while(eightNotSeenYet & (m < lengths(1)))
        n = 1; %reset n to 1 as we begin a new row
        while(eightNotSeenYet & (n < lengths(2)))
            if  (inputMatrix(m,n) == 8) %if that position equals an 8
                inputMatrix(m,n) = 9;  %change that position in the matrix to a 9 \
using an assignment statement
                eightNotSeenYet = false; %set flag to false
            end  %ends the IF
        end  %ends the inner While loop
    end   %ends the outer While loop
    output = inputMatrix; %assign the matrix to the output value (or in other word\
s, "return the (possibly) altered matrix")
end %end the function
```

Matrix loops

- How to break out of an infinite loop?
- Ctrl-C
- (Note: Ctrl – C will also end processes on the unix command line if they are running too long and you want them to stop.)

- Other examples
- How to loop through a matrix and set the entire diagonal (starting at 1,1) to zero?
- Is that easier to do with FOR loops or while loops?
- How to write a loop to sum the elements in a matrix x.