Workshop 5 - While Loops and Conditionals

- Due Friday by 23:59
- Points 1

This workshop is designed to provide more practice with working with conditional execution in MATLAB. To prepare for this workshop you should create a folder in your account or on your computer called **workshop5**.

Each question has the pattern of problem definition, followed by tests, followed by coding. The Demonstrators will work through questions 1 and 2 with you and for the remaining questions you should be able to use code and ideas from previous concepts in this course.

During the demonstration, if possible it is good to type along with the demonstrator or take notes *(bring pen and paper!)*. If you are sharing a terminal make sure that take turns at working during this practical so every group member has a chance to save some working code.

When you have finished your workshop you should make sure that you show the work that you have done to the laboratory supervisor so your participation can be marked. Be sure to check your participation is recorded before you leave your workshop session or you won't receive credit.

Question 1 - Using a while loop to read in data (example from lecture)

Problem Definition

In this question we have to prompt the user repeatedly for the ages and names of people and add them to the end of vectors to hold the ages and hold the names. The reading stops when the user enters -1 for an age. The program should then use disp to display the names and ages vectors.

Testing

The input will consist of ages and names typed by the user. The input, excluding prompts, from the program might be

```
30
Beatrice
40
Alan
50
Celia
-1
```

The desired array entries corresponding to these lines, printed with names array first, are:

```
"Beatrice" "Alan" "Celia"
30 40 50
```

when printed out in MATLAB.

(we would save text like this in a file called q1.txt)

Coding

To start this workshop navigate to your workshop 5 folder in MATLAB and type

clear all

into the MATLAB console. This is a way to clear previous definitions that might get in the way of code working as required. In general, it is good to type this command at the beginning of a coding session.

One possible solution for the program is the script.

```
% read in ages and names
% set up result vectors
ages=[];
names=[];

% prompt for first age input
age=input('enter age ');

% read in both repeatedly and add to vectors
while(age~=-1)   % while the user has not entered -1 for age
    name=input('enter name ','s'); % read in string
    ages=[ages age];  % append to ages end
    names=[names string(name)]; % convert name to string and append
    age=input('enter age '); % read in new age.
end
% display resulting vectors
disp(names);
disp(ages);
```

(we would save this script in a file called q1.m). Note the use of the function string() to convert a character array as read in by input() to a string. Also note the use of the ‘s’ parameter in input() so that we let MATLAB know we are reading a string rather than a numerical value.

Question 2: Using a while loop to print the elements of a vector while the sequence ascends (example from lecture)

Definition

For this question we use a while loop to print the elements of a non-empty vector while the next number is greater or equal to the previous number in the vector. The first element of the vector should always be printed.

Testing

These are some example test cases.

```
sequence          output
[ 1 ]             1
[ 1 2 3]          1 2 3
[ 2 3 1]          2 3
[ 5 8 9 9 2 10]   5 8 9 9
[ 6 5 ]           6
```

(we would save these in a file called q2.txt)

Coding

One solution for this program is the code:

```
% define a vector
A=[ 1 5 5 7 3 2 9 10 ];

% set the first index
i=1;

% print the first element of the vector and increment index
fprintf('%0.0f ',A(i));
i=i+1;

% while we haven't reached end and we are ascending
while( i<=length(A) && A(i)>=A(i-1))  % why won't this cause an error?
    fprintf('%0.0f ',A(i));  % print current elements
    i=i+1;                   % increment index
end
fprintf('\n');               % print newline
```

Note that the key line in this code is the test for the while statement:

```
while( i<=length(A) && A(i)>=A(i-1))  % why won't this cause an error?
```

which means:

> *check that i is less than or equal to the length of A and, after that, check the current element of A is greater than equal that the previous element and, if both are true, then continue with the loop.*

Note that the && operator will only check the second clause: A(i)>=A(i-1) clause if the first clause: i<=length(A) is true. This is fortunate because the second clause will generate an error when we are beyond the end of the array. Also note that an, equally nice, way to write the above loop is:

```
% while we haven't reached end and we are ascending
for i=[2:length(A)]
    if (A(i)<A(i-1))  % we have started to descend
        break;        % exit the loop
    end
    fprintf('%0.0f ',A(i));  % print current elements
end
```

Where the break; statement is a MATLAB keyword that means “jump out of the current loop”.

Note also the if-statement has the test: A(i)<A(i-1) because now it is checking whether it should *leave* the loop.

Question 3: Using a while-loop to print a vector’s values while the values are greater than zero **(do this, and the following questions, in groups)**

Definition

In this question you will write code to print the values from a vector called vals *while* those values are greater than zero.

Testing

On paper, or in a text file, write three new tests for the above program As an example, three tests for this program might be.

```
vals          output
[ -2 ]        empty output
[ 3 -2 7 9]   3
[ 4 9 8 0 7]  4 9 8
```

If you are using a text file save the file with the name q3.txt

Coding

If you are sharing a computer, swap a new group member to the terminal. Write a MATLAB script called q3.m that implements the problem definition above. Your program should require some changes from the code presented in question 2.

Question 4: Detecting an Equilateral Triangle

Definition

In this question we define code that reads in three positive numbers representing the lengths of the sides of a triangle and prints out a message saying “it’s equilateral” if all three sides are of equal length.

Testing

On paper, or in a text file, write three new tests for the above program. As an example, two tests for this program might be.

```
inputs output
5 5 5  “it’s equilateral”
2 1 2  <no output>
```

Coding

If you are sharing a computer, swap a new group member to the terminal. Write a MATLAB script called q4.m that implements the definition above using an if-statement.

Question 5: Detecting an Isosceles Triangle

Definition

For this question we have to write MATLAB code to detect an isosceles triangle by reading in three numerical values representing the lengths of the three sides and printing “it’s an isosceles” if any two of the sides are equal and the triangle is not equilateral.

Testing

On paper, or in a text file, write three new tests for the above program. As an example, three tests for this program might be.

```
inputs output
5 5 5  <no output>
2 1 2  “it’s isosceles”
3 2 1  <no output>
```

Coding

If you are sharing a computer, swap a new group member to the terminal. Write a MATLAB script called q5.m that implements the definition above.

Question 6: Detecting a Scalene Triangle

Definition

For this question, we have to write MATLAB code to detect a scalene triangle by reading in three numerical values representing the lengths of the three sides and printing “it’s a scalene” if the triangle is not an isosceles or equilateral triangle and the sum of any two side’s length is *greater than* the length of the other side.

Testing

On paper, or in a text file, write three new tests for the above program. As an example, three tests for this program might be.

```
inputs output
5 5 5  <no output>
2 1 2  <no output>
3 2 1  <no output>
3 4 5  “it’s a scalene”
```

Coding

If you are sharing a computer, swap a new group member to the terminal. Write a MATLAB script called q6.m that implements the definition above using at least if-statements.

Question 7: Converting a Number to Text using a Switch Statement

Definition

For this question we have to write MATLAB code to generate a random integer value which is either 0, 1 or 2 and output the message: "Rock" if the generated number is zero, "Paper" if the generated number is 1, and "Scissors" if the generated number is 2.

Testing

On paper, or in a text file, write two new tests for the above program. As an example, one tests for this program might be.

```
random number output
0             "Rock"
```

Coding

If you are sharing a computer, swap a new group member to the terminal. Write a MATLAB script called q7.m that implements the definition above using a switch statement. If you still have time, you might want to see if you can implement the same functionality by indexing into an array of strings!

Finishing up.

Make sure you let a practical supervisor see your work for this session.

End of Workshop.