21-Loops Part 2

text: Chapter 6.4-6.6

ECEGR 101

Engineering Problem Solving with Matlab

Professor Henry Louie

Overview

- While Loop
- Infinite Loops
- Break and Continue

WHILE Loop

- Used to repeat a set of commands an unknown number of times (until a condition is met).
- Syntax

```
while condition
        set of commands
end
```

Example

```
fprintf('Testing the WHILE loop structure.\n\n')
i = 1;
while i <= 4
    fprintf('Testing the WHILE loop. i = %d.\n', i);
    i = i + 1;
end
fprintf('\nEnd of WHILE loop. i = %d.\n', i);
```

remember to initialize the loop counter

```
>>
Testing the WHILE loop structure.

Testing the WHILE loop. i = 1.
Testing the WHILE loop. i = 2.
Testing the WHILE loop. i = 3.
Testing the WHILE loop. i = 4.

End of WHILE loop. i = 5.
>>
```

Example

```
fprintf('Testing the WHILE loop structure.\n\n')
i = 0;
while i <= 4
    i = i + 1;
    fprintf('Testing the WHILE loop. i = %d.\n', i);
end
fprintf('\nEnd of WHILE loop. i = %d.\n', i);
```

It does matter where the fprintf command is placed with regard to the counter incrementing command.

```
>>
Testing the WHILE loop structure.

Testing the WHILE loop. i = 1.
Testing the WHILE loop. i = 2.
Testing the WHILE loop. i = 3.
Testing the WHILE loop. i = 4.
Testing the WHILE loop. i = 5.

End of WHILE loop. i = 5.
>>
```

Infinite WHILE Loops

- If the condition never becomes false, an infinite loop results
- To stop an infinite loop from executing, press Ctrl-c

Exercise

Explain why the following loops are infinite:

```
while 1
end
```

```
i = -5;
while i <= 0
    k = k + 1;
    i = i - 1;
end
```

```
i = 0;
while i >= 0
    j = j^2;
    h = sum(j);
end
```

Exercise

Explain why the following loops are infinite:

```
while 1
end
```

Loop condition never changes.

```
i = -5;
while i <= 0
    k = k + 1;
    i = i - 1;
end
```

Loop condition always true.

```
i = 0;
while i >= 0
    j = j^2;
    h = sum(j);
end
```

The variable i and the loop condition never change.

Initializing the WHILE Loop Variable

The WHILE loop variable has to be properly initialized for the loop to work.

```
while x < 10
    x = x + 1;
    y = 2*x;
end
```

What is x equal to when the loop starts??

Exercise

Write a script that uses a WHILE loop to display the elements of vector x on the command window

Hint: you need to initialize the index of the vector

Exercise

```
i = 0;
while i < length(x)
    i = i + 1;
    disp(x(i))
end
```

```
i = 0;
while i ~= length(x)
    i = i + 1;
    disp(x(i))
end
```

```
i = 1;
while i ~= (length(x) +1)
    disp(x(i))
    i = i + 1;
end
```

Exercise

Modify the guessing game program (from previous lectures) so the user guesses till the correct number is entered.

When the user finally guesses correctly then the program prints out the total number of guesses.

Exercise

```
number = round(10*rand);
counter = 1;
guess = input('Guess an integer between 0 and
   10:  ');
while (guess ~= number)
    guess = input('Guess again:   ');
    counter = counter + 1;
end
fprintf('Congratulations! You guessed the
   correct number.\n');
fprintf('It took you %d attempts to guess this
   number.\n', counter);
```

Exercise

```
>> guessing_game
Guess an integer between 0 and 10:  4
Guess again:   1
Guess again:   5
Guess again:   7
Guess again:   8
Guess again:   3
Guess again:   9
Congratulations! You guessed the correct
   number.
It took you 7 attempts to guess this number.
```

Example

Write a script that finds the smallest integer N such that

$$\sum_{i=1}^{N} i$$

is greater than a specified limit.

Example: Find N if limit = 5.
The answer is N = 3 since 1+2+3=6>5
but 1+2=3<5.

Exercise

Write a program to find the first odd integer whose cube is greater than 3000.

Exercise

```
x = 1;
while x^3 <= 3000
    x = x + 2;
end
```

```
>> x
x =
    15
>> 15^3
ans =
        3375
>> 13^3
ans =
        2197
```

The Break Command

The break command terminates the for or while loop (by jumping to the *end* command) and continues executing the rest of the program.

```
for k = 1:10
    x = 50 - k^2;
    if x < 0
        break
    end
    y = sqrt(x);
end
```

In most cases, a program that uses the break command can be rewritten using the WHILE loop.

Exercise

Are the following two programs equivalent?

```
for k = 1:10
    x = 50 - k^2;
    if x < 0
        break
    end
    y = sqrt(x);
end
```

```
k = 0; x = 1;
while (k <= 10) & (x >= 0)
    k = k + 1;
    x = 50 - k^2;
    y = sqrt(x);
end
```

Exercise

```
for k = 1:10
    x = 50 - k^2;
    if x < 0
        break
    end
    y = sqrt(x);
end
```

Answer: No.

```
k = 0; x = 1;
while (k <= 10) & (x >= 0)
    k = k + 1;
    x = 50 - k^2;
    y = sqrt(x);
end
```

x changes after the condition was checked.

```
>> [k x y]
ans =
     8   -14     1
```

```
>> [k x y]
ans =
        8                -14                0 +    3.7417i
```

Exercise

How can we modify the 2nd program to make it equivalent to the 1st program?

```
for k = 1:10
    x = 50 - k^2;
    if x < 0
        break
    end
    y = sqrt(x);
end
```

```
k = 0; x = 1;
while (k <= 10) & (x >= 0)
    k = k + 1;
    x = 50 - k^2;
    y = sqrt(x);
end
```

Exercise

How can we modify the 2nd program to make it equivalent to the 1st program?

```
for k = 1:10
    x = 50 - k^2;
    if x < 0
        break
    end
    y = sqrt(x);
end
```

```
k = 0; x = 1;
while (k <= 10) & (x >= 0)
    k = k + 1;
    x = 50 - k^2;
    if (x >= 0)
        y = sqrt(x);
    end
end
```

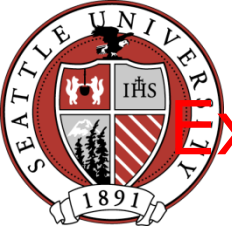
```
>> [k x y]
ans =
     8   -14     1
```

Exercise

Write a script that finds the smallest integer N such that

$$\sum_{i=1}^{N} i > limit$$

Use a FOR loop with the BREAK command



Exercise

$$\sum_{i=1}^{N} i > \text{limit}$$

```
%Exercise 10  Using the break command

limit = input('enter the value of the limit:  ');
if limit < 1
    fprintf('you must enter a number greater than 1\n');
else
    sum = 0;
    for i = 1 : 1e10
        sum = sum + i;
        if sum > limit
            break
        end
    end
    disp('the value of N is:'); disp(i);
end
```

The Continue Command

The continue command passes control to the next iteration of the FOR or WHILE loop in which it appears, skipping any remaining statements in the body of the loop.

```
x = [10, -1000, 10000, -100000];
y = -ones(size(x));
for k = 1:length(x)
    if x(k) < 0
        continue;
    end
    y(k) = log10(x(k));
end
```

```
>> k
k =
     4
```

```
>> y
y =
    1    -1     4    -1
```

Exercise

Rewrite the following program without the use of the continue command.

```
x = [10, -1000, 10000, -100000];
y = -ones(size(x));
for k = 1:length(x)
    if x(k) < 0
        continue;
    end
    y(k) = log10(x(k));
end
```

Exercise

Rewrite the following program without the use of the continue command.

```
x = [10, -1000, 10000, -100000];
y = -ones(size(x));
for k = 1:length(x)
    if x(k) < 0
        continue;
    end
    y(k) = log10(x(k));
end
```

```
x = [10, -1000, 10000, -100000];
y = -ones(size(x));
for k = 1:length(x)
    if x(k) >= 0
        y(k) = log10(x(k));
    end
end
```

Exercises

Solve the following exercises using either FOR or WHILE loops.

Exercise

The Fibonacci numbers are defined by the sequence such that each value is the sum of the previous two values. The first two numbers in the Fibonacci sequence are given below:

$F_1 = 1, F_2 = 1.$

The formula for calculating the successive numbers in the sequence is the following:

$F_i = F_{i-1} + F_{i-2}$, where i is an integer greater than 2.

Write a function that takes one argument – a positive integer N and outputs a vector of N Fibonacci numbers.

Exercise

Use a WHILE loop to determine how many terms in the series 2^k, k = 1, 2, 3, …, are required for the sum of the terms to exceed 2000. What is the sum for this number of terms?

Exercise

```
s = 0;
k = 1;
while s < 2000
    s = s + 2^k;
    k = k + 1;
end
k = k - 1;
```

```
s = 0;
k = 0;
while s < 2000
    k = k + 1;
    s = s + 2^k;
end
```

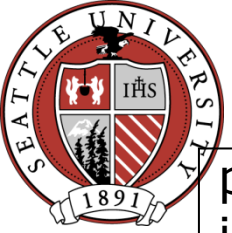
k = 10
s = 2046

Exercise

Write a program that asks the user to enter positive real numbers until the product of the numbers is greater than 500.

The program should check for invalid input: the user should not be able to enter strings, negative numbers, complex numbers, or arrays.

When a total of 500 is exceeded, the program should display the product of the numbers, how many numbers were entered, and list the numbers.

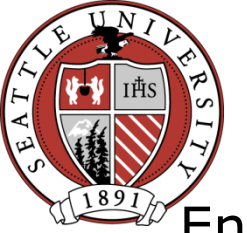


Exercise

```
p = 1; % product
i = 0; % number counter
while p < 500
    x = input('Enter a number = ');
    if ~isreal(x) % test if the number is complex
        fprintf('You have entered a complex number. Please, try
    again.\n');
    elseif ~isnumeric(x) % test if a string was entered
        fprintf('You have entered a string. Please, try again.\n');
    elseif x < 0
        fprintf('You have entered a negative number. Please, try
    again.\n');
    elseif length(x) > 1
        fprintf('You have entered an array. Please, try again.\n');
    else
        i = i + 1;
        n(i) = x;
        p = p*x;
    end
end
```

Exercise

```
fprintf('\n');
fprintf('%d numbers entered = ', i);
fprintf('%.2f ', n);
fprintf('\n');
fprintf('Product = %.2f\n', p);
```



Exercise

Enter a number = -8
You have entered a negative number. Please, try again.
Enter a number = 'y'
You have entered a string. Please, try again.
Enter a number = 4+5i
You have entered a complex number. Please, try again.
Enter a number = [1 7 8 9]
You have entered an array. Please, try again.
Enter a number = 3
Enter a number = 5
Enter a number = 6.7
Enter a number = 3
Enter a number = 9

5 numbers entered = 3.00 5.00 6.70 3.00 9.00
Product = 2713.50

Exercise

Use the WHILE loop to repeatedly generate three random numbers, each between 0 and 9. Stop the WHILE loop when at least two or more of the numbers are equal to 7. Display the numbers and the number of iterations it took to achieve this result.

Exercise

```
x = 0; y = 0; z = 0; test = 0; i = 0;
while ~test
    i = i + 1;
    x = round(rand*9);
    y = round(rand*9);
    z = round(rand*9);
    test = ((x == 7) & (y == 7)) | ((x == 7) & (z ==
   7)) | ((y == 7) & (z == 7)) | ((x == 7) & (y == 7)
   & (z == 7));
end
fprintf('i = %d, x = %d, y = %d, z = %d\n', i, x, y, z);
```

Exercise

i = 8, x = 7, y = 7, z = 4

i = 20, x = 3, y = 7, z = 7

i = 74, x = 7, y = 7, z = 1

i = 5, x = 7, y = 7, z = 1

i = 67, x = 7, y = 6, z = 7

i = 14, x = 4, y = 7, z = 7

Exercise

Write a simple program that simulates gambling. A player starts with $100. He plays a simple machine such that he can either win $10 or lose $10. It also costs a $1 to play the machine each time.

Use the rand function to generate a win or a loss. For example if x = rand, then if x > 0.5, the player loses $10, and if x < 0.5, the player wins $10.

For each iteration, display the iteration number and the players money.

Exercise

```
s = 100; i = 0;
fprintf('i = %d, s = %d\n', i, s);
while s > 0
    x = rand;
    s = s - 1;
    i = i + 1;
    if x > 0.5
        s = s - 10;
    else
        s = s + 10;
    end
    if s >= 0
        fprintf('i = %d, s = %d\n', i, s);
    end
end
```

Exercise

i = 0, s = 100
i = 1, s = 89
i = 2, s = 98
i = 3, s = 107
i = 4, s = 96
i = 5, s = 85
i = 6, s = 94
i = 7, s = 83
i = 8, s = 72
i = 9, s = 61
i = 10, s = 70
i = 11, s = 59
i = 12, s = 48

i = 13, s = 37
i = 14, s = 26
i = 15, s = 35
i = 16, s = 24
i = 17, s = 33
i = 18, s = 42
i = 19, s = 31
i = 20, s = 20
i = 21, s = 9
i = 22, s = 18
i = 23, s = 7
i = 24, s = 16
i = 25, s = 5

Exercise

Write a program that asks the user to enter a string and counts the number of characters in a string up to the first occurrence of an uppercase or lowercase letter ‘X’. If no ‘X’ or ‘x’ occurs, all characters are counted.

Exercise

```
s = input('Type a string = ');
count = 0; i = 1;
while (i <= length(s)) & (s(i) ~= 'x') & (s(i) ~= 'X')
    count = count + 1;
    i = i + 1;
end
fprintf('There are %d characters in %s before the
   first X or x.\n', count, s);
```

Exercise

Type a string = 'kjhfdjbvdbvxehdhfhf'
There are 11 characters in kjhfdjbvdbvxehdhfhf before the first X or x.
Type a string = 'xfkldhglifkhdg'
There are 0 characters in xfkldhglifkhdg before the first X or x.
Type a string = 'Xjh'
There are 0 characters in Xjh before the first X or x.
Type a string = 'hfhgXlkhg'
There are 4 characters in hfhgXlkhg before the first X or x.
Type a string = 'lkjfhdX'
There are 6 characters in lkjfhdX before the first X or x.
Type a string = 'ax'
There are 1 characters in ax before the first X or x.

Exercise

Using a Taylor series expansion, the function e^x can be approximated by the series

$$\sum_{n=0}^{\infty} \frac{x^n}{n!}$$

for values of x close to zero. Write a program that determines how many terms in the sum are required so that the difference between e^x and the sum is less than 10^{-6}.

Exercise

```
close all;
clear all;clc
N=0;
s=0;
e=1;
x=input('Enter x: ');
while e>10^-6
    s=s+(x^N)/factorial(N);
    e=abs(exp(x)-s);
    N=N+1;
end
fprintf('Number of terms %5d \n',N)
fprintf('The error is %5d \n',e)
```

Exercise

Command Window

```
Enter x: .1
Number of terms      5
The error is 8.474231e-08
```

Command Window

```
Enter x: 10
Number of terms     37
The error is 9.829419e-07
```

```
Enter x: 1
Number of terms     10
The error is 3.028859e-07
```