ENME 303 LAB

Week 4: For-Loops

Lab_4

Week 4: Control Flow III

I. Refresher: Loops
II. The `for` Loop
III. Nested Loop
IV. Break statement

I. Review of Loops

- Two classifications of loops:
  - **Condition-controlled statements** aka a `while` loop
    - Repeats one or more statements while some condition is true
    - In general, once loop commences, **you don't know** how long it will iterate
  - **Count-controlled statements** aka a `for` loop
    - Repeats one or more statements for a fixed number of times
    - In general, once loop commences, **you know** how long it will iterate

II. The `for` Loop

The `for` loop has the following syntax:

```
for <loop control variable> = <vector expression>

 <one or more statements>

end
```

The `for` loop behavior:

- <vector expressions>: evaluated to form a row vector of values. If row vector empty-->code skips to **end**
    - Has form `first:last` or `first:increment:last` (ex 1:10 or 100:10:200)
- Otherwise, the loop control variable is set to the first element of the row vector and then the code between the vector expression and the **end** (the loop body) is executed
    - After execution, loop control is augmented to next row vector variable
- This sequence repeats until the loop control variable is set to the last value of the row vector and the loop body executes for a final time

II. The `for` Loop

Example:

```
for Count = 1:10
         fprintf('Count = %d\n', Count');
end
```

The one-line `for` statement performs the *initialization*, *testing*, and *modification* (ITM) of the loop control variable

What does this `for` loop do?

What is the control variable?
What is the vector expression?

II. The `for` Loop

- Most times, the # of times the loop repeats = # of elements in the row vector created from the <vector expression>
    - i.e **1:10** means 10 iterations, **1:2:10** means 5 iterations counting by 2
- Other times row vector is uncertain and number of iterations not known until execution of the program

```
Max_Number = input('Enter a number to count up to: ');

for Count = 1:Max_Number

     fprintf('Count = %d\n', Count')

end
```

Note that this loop repeats 0 times if the user enters a number less than 1

II. The `for` Loop

- Commonly used for **processing** each element of an array.
  - Take this example, which adds 1 to each element in its row vector each loop:

```
Vector = [5 3 2 6 1 1 4 6 3 2];
VecLength = length(Vector)
for Index = 1: VecLength
        Vector(Index) = Vector(Index) + 1;
end
disp(Vector);
```

Whats the length of Vector? How many iterations occur?

This takes the index of each value in Vector

What does Vector look like after loop fully executed?

Examples: Simple For Loop

```
b = 3;

for k = 1:5

 b^k

end
```

```
sum1 = 0;

for k = 1:2:9

 sum1 = sum1+k;

end

sum1
```

How many iterations occur for each? What does the output look like for each example?

III. Nested Loop

Matlab allows us to put compound statements like `if`, `while`, and `for` statements inside other compound statements.

```
for m = 1:5           %Outer loop executes 5 times, once for each value of m
     for n = 1:5 %For each value of m, inner loop updates 5 entries (for each n) into A matrix
          A(m,n) = m*n;        %Creating A matrix
     end
end
disp(A)                                             %Displaying A matrix
```

```
1    2    3    4    5
2    4    6    8    10
3    6    9    12   15
4    8    12   16   20
5    10   15   20   25
```

Examples: Nested `for` loops

```
sum1 = 0;
for n = 1:2
 for m = 1:3
 sum1 = sum1 + n*m;
 end
end
sum1
```

Sum1 =18

vs

```
for n = 1:2
 for m = 1:3
 fprintf('n = %d m = %d \n', n, m)
 end
end
```

```
n = 1 m = 1
n = 1 m = 2
n = 1 m = 3
n = 2 m = 1
n = 2 m = 2
n = 2 m = 3
```

IV. Break Statement

- Used to terminate loop from any location in the body of the loop
- When `break` executed→ execution jumps to the code after the **end** of loop

```
N = 1                                                                   %initialize N

while N < 100                                             %while statement that checks if N is less
than 100

     if N <= 0                                            %Nested if statement that eliminates chance
of never ending loop

          break                                           %if N is negative or zero, break forces the
     while loop to end

     end

N = N*(N+1);                                              %If N is not negative or zero, execute
operation

disp(N)                                                            %Displays N each iteration of
while
```

Acknowledgement

The lab slides you see are not made by one person. All the TA/TFs served for this course have contributed their effort and time to the slides. Below are the leading TFs for each semester:

- 2021 FA - Karla Negrete (GTA)
- 2022 SP - Justin Grahovac
- 2022 FA - Kelli Boyer and Yisrael Wealcatch
- 2023 SP - Matt Moeller and Mahamoudou Bah
- 2024 SP - Mohammad Riyaz Ur Rehman and Michael Mullaney