While Loops

ENGR 1181
MATLAB 10
While Loops in Real Life

Revisiting the previous example of the combustible engine from our first discussion of for loops (fixed number of runs), we can apply this example to while loops (indefinite number of runs). The fixed cycle of four states is the for loop, which runs indefinitely while the engine is on.
Today's Learning Objectives

- After today’s class, students will be able to:
  - Explain the proper application of while loops.
  - Use external inputs (e.g., tic, toc) for real time programming.
Reminders About While Loops

- **While** loops must have a conditional expression, and the conditional expression must have at least one variable.

- Conditional variables must be "initialized" before the loop.

- The conditional variable must advance with each pass. (Otherwise the loop will continue indefinitely!)

- ‘Ctrl + C’ will terminate execution of an indefinite loop.

- Like for loops, must have an `end`
Example: Simple While Loop

```
a=0
while a<10
    a=a+1
    if (a==5)
        continue
    end
    disp(a)
end
```

Note:
- conditional variable is initialized
- variable advances each time
- `one end corresponds to if`
- `one end corresponds to while`
Function: rem()

This function checks for and returns a remainder when dividing $x$ by $y$.

\[ \text{rem}(x, y) \]

Ex: \( \text{rem}(x, 3) \) will check if the value of $x$ is divisible by 3.

Ex: \( \text{rem1} = \text{rem}(23, 5) \)

\[ \text{rem1} = 3 \]
Example: Complex While Loop

Task
Write a program that will use a while loop to find the smallest number divisible by both 3 and 5, with the square of the number greater than 3325.

Steps
1. Initialize variables: sq & n
2. Loop while sq < 3325
3. Advance n by 3 in each pass
4. Check if n is divisible by 5
5. If yes, square the n
6. Repeat loop until sq > 3325.
7. Loop stops & n value will give us the smallest integer to meet conditions
Example: Complex While Loops

n=0; sq=0;
while sq<=3325
    n=n+3
    if rem(n,5)==0
        sq=n^2;
    end
end

disp(n)
Function: ‘tic’ and ‘toc’

- `tic` and `toc` are timing functions that monitor elapsed time in your program.
- `tic` starts a stopwatch that runs in the background.
- `toc` returns the elapsed time since the last `tic` command.
- The value of `toc` can be assigned to a variable.
  - Ex: `time1 = toc;`
Tic/Toc Example

N=input('Please enter a large value for N: ');

 tic
 for i=1:N
 sqroot= sqrt(i);
 end

 SqrtTime=toc;

 fprintf('
 It took MATLAB %f seconds to calculate %i square roots
', SqrtTime, N)
Important Takeaways

- While loops need a conditional statement and conditional variables must be initialized.
- While loop will run indefinitely until condition is met.
- ‘rem()’, ’tic’, and ‘toc’ are very useful tools.
What’s Next?

- Review today’s Quiz #10
- Open the in-class activity from the EEIC website and we will go through it together.
- Then, start working on MAT-10 homework.