Loops 2: Accumulating an Answer

By the end of this Lab, you should be able to:

- use a for loop to solve problems requiring accumulating an answer in steps
- use a while loop to solve problems requiring accumulating an answer in steps

Download ***fact.m***
to your working directory
Try it out, e.g. >> fact(5)

Laboratory Outcomes:

1. solve engineering problems using computer tools
2. apply arrays and array manipulations
3. use and explain files and data structures
4. write a function with multiple inputs and outputs at the command line
5. write a function that results in a non-numerical output
6. write programs using logical expressions and conditional statements
7. **write programs using loop structures**
8. fit data that follows linear, quadratic, or power law forms
9. properly communicate a solution based on computer calculation or program

Handouts:
- **For Loop Indexing**
- **Loop Lab**

Web/Network Drive Materials:
elastic.m, fact.m

Introduction

- Questions from MATLAB work

Outline ➔

1. Intro
2. Review:
3. **Slide Intro:** Loops & the For structure
   Accumulation Logic and Recursion Equations
   a) Background – loop index & overview
   b) Example loop: Factorial Function
   ~~c) Student Exercise: Geometric Series Function~~

Lab Handout: Accumulation Loops
1. Example Loop: Spring energy
2. Exercise: charge on a capacitor vs. time
3. Accumulation with while loops
   a) While Loops (slides available)
   b) Example Loop: length to Spring energy
   c) Exercise: time to charge on a capacitor
5. Assignment: 2 capacitor charging programs

For Loop Practice Quiz

- Complete and turn in.

Example: An accumulation problem

Set Up/ Introduction Type of Program: ☐ Script ☑Function

1. Problem Statement:

Develop a function that can calculate the factorial of any integer base.

2. Inputs: (full name, variable to be used, units)

Variable Name	Description	Units or Values	Input Source*
n	factorial base	integer	command line

3. Output: (full name, variable to be used, units)

Variable Name	Description	Units or Values	Output type*
y	resulting factorial value	integer	command line

4. Plan & Test Calculations

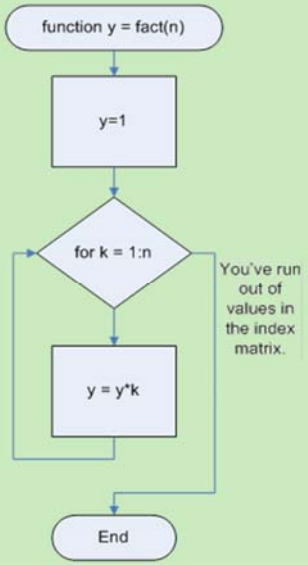
Test Calculations			
n	**n!**	**= n * (n-1)!**	**y**
1	1! = 1		= 1
2	2! = 2 * 1	= 2 * 1!	= 2
3	3! = 3 * 2 * 1	= 3 * 2!	= 6
4	4! = 4 * 3 * 2 * 1	= 4 * 3!	= 24
5	5! = 5 * 4 * 3 * 2 * 1	= 5 * 4!	= 120

Recursion Formula: $k! = k\,(k\,-1)!$

or $y_k = k\,(y_{k-1})$

Calculation Process:

- apply recursion successively
- starting at k = 1 & y_0 = 1
- repeat n times
  increasing k by one each time
- until k = n



Note on variables n & k being used in this example (in calculations, flowchart and code)
k is used to represent the current base value in the recursive calculation
n is used to represent the base input

when the calculation (i.e. the loop) is done k = n.

Programs based on recursion equations usually end up with two program variables for one algebraic variables

1. the current value in the recursion calculation
2. the final value

It is important to not get them confused.

Try it: A Factorial function (Accumulation)

```
function y = fact(n)
% function y = fact(n)
% This function calculates the factorial of n
% input:    n
% output:  y = the factorial of n (n!)

% internal variable k = loop variable
% initialize accumulation variable
y = 1;

%  use for loop to accumulate the
% factorial product in the variable y
for k = 1:n
   y = y * k;
end
```

The accumulation variable (y) must have an initial value before the loop starts.

You can initialize the **accumulation variable** by: carrying out the first calculation outside the loop or

- 1 to start a product
- 0 to start a summation
- [ ] to start a concatenation

Inside the loop the **accumulation variable** appears on both the LHS and RHS to allow information to pass from one loop to the next.

Recursion Formula

$k! = k\,(k\,-1)!$

or $y_k = k\,(y_{k-1})$

Loops: Accumulation to an Answer (Lab Handout)

I. Elastic Energy Example (For loop)

II. Capacitor Charging Exercise (For loop)

III. Impact of Step Size

IV. Accumulating with a While loop
   Example: Spring compression for a given energy
   Important Note: ^c to stop "crash out" of a program
   Problem: Charging time for a given charge

V. Lab Assign – Capacitor Charging (Also try new Pig Latin program)

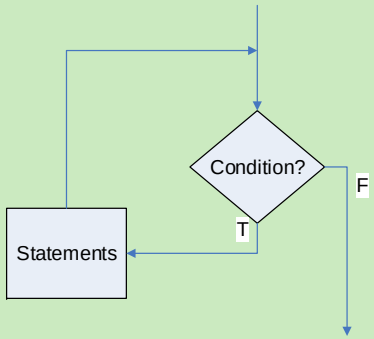
1) **Voltage at a given time** (from part II)
2) **Time to reach a given voltage** (analogous to example in part IV)
3) Bonus: Write a function to plot the charge vs time given R, C and V_s
   Take it farther: Make it animate the charging in real time!

- use following slide toward end of period – material in handout appendix (goes with changing a capacitor to a voltage target).

- Encourage students to carry suggested exercise to get use to the while loop.

While Loops: Appendix A

```
%  program whileeg
%  this script demonstrates a simple while loop
x=6
k=0;                    % this initializes the loop vbl.
while k <= x            % logical condition controls loop
   k                    % echo print loop variable to see
what is happening
   k=k+1;               % change the loop variable
end
```



- vary index increment step sizes (in k = k + 1)
- use a different operation (e.g., try k = k*2)
- vary the starting point (the k = 1 line)