ENG1060: COMPUTING FOR ENGINEERS

PASS 5 – Week 7

2020 OCT NOV

TASK 1

The following table gives the approximate values of the coefficient of static friction µ, for various materials

Option	Materials	µ
1	Metal on metal	0.2
2	Wood on wood	0.35
3	Metal on wood	0.4
4	Rubber on concrete	0.7

To start moving a weight *W*, on a horizontal surface, you must push with a force *F*, where *F*=µ*W*.

Write an m-file that achieves the following:

1. Uses fprintf to print out the option and material information to the command window
2. Prompt the user to input a value of *W* and the type of materials/option
    - Check if the input values are valid (e.g. negative weights or non-integer/incorrect option numbers). Your program should continue prompting the user to enter values until they are valid.
3. Use a switch statement to compute the force required. Use fprintf to print a statement including the materials used and the force required.

Hint: You may want to use a while loop to ensure that the user enters valid inputs.

TASK 2

The circumference of a circle is given by C=2πr where *r* is the radius. The circumference of an ellipse can be approximated by $E = \pi\left[3(a+b) - \sqrt{(3a+b)(a+3b)}\right]$where *a* and *b* are the minor and major axis lengths.

A. Write an m-file which prompts the user to specify whether they would like to calculate the circumference of a circle or an ellipse. If the circle option is chosen, ask the user for a radius. If the ellipse option is chosen, ask the user for the minor and major axis lengths.
B. Calculate the circumference of the chosen shape. Use fprintf to write a statement containing the shape chosen, the relevant length(s) and the circumference.
C. Ask the user if they would like to repeat the calculation. If yes is chosen, go through steps A and B again. If no is chosen, end the code.

Hint: Use a while loop to achieve part C.

Your code should look similar to the following:

```
Enter either circle (1) or ellipse (2): 1
Enter the radius length: 1
Circle was chosen with a radius of 1. Circumference = 6.283185e+00
Would you like to repeat the calculation? 1=yes, 0=no: 1
```

TASK 3

The factorial of a positive real integer is defined as follows:

$$n!{=}1{\times}2{\times}{\ldots}{\times}(n{-}1){\times}n$$

A. Write a function that takes an input value n and returns its factorial result $n!$. Ensure that your function checks that the input n is real, positive, and an integer. You may not use the prod() or factorial() functions.

The Euler number e is calculated as the sum of the infinite series shown in Eq. 1 below. Since we cannot actually approach infinity, we can assume a function f_k to take the form shown in Eq. 2, and say that f_k is related to e through Eq. 3.

Eq. 1

$$e = \sum_{n=0}^{n\to\infty} \frac{1}{n!}$$

Eq. 2

$$f_k(n) = \sum_{n=0}^{n=k} \frac{1}{n!}$$

Eq. 3

$$\lim_{k\to\infty} f_k = e$$

We can see that every time we increase k, the value of f_k converges to a specific value.

B. Determine the value of k in which the function value f_k produces an error of less than 6 decimal places (i.e. a precision of 1e-6). The error here is defined as the difference in the function values between subsequent increases in k, i.e. between f_k and f_{k-1}.

TASK 4

Recall that a matrix multiplication is performed by multiplying each value in the rows of **A** with the corresponding values in the columns of **B** and then adding the results together. This is described by the following equation:

$$M_{r,c} = \sum_{n} A_{r,n}\, B_{n,c}$$

The subscripts represent the row *r*, and column *c*, of an element in the matrix. $\mathbf{M}_{\mathbf{r,c}}$ means the value in the rth row and the cth column of **M**, which is addressed as M(r,c) in MATLAB.

For example, if **A**=[1, 2; 3, 4] and **B**=[2, 4; 4, 8], then the matrix multiplication of A*B can be coded as:

M(1,1) = **A**(1,1)***B**(1,1) + **A**(1,2)***B**(2,1)
M(1,2) = **A**(1,1)***B**(1,2) + **A**(1,2)***B**(2,2)
M(2,1) = **A**(2,1)***B**(1,1) + **A**(2,2)***B**(2,1)
M(2,2) = **A**(2,1)***B**(1,2) + **A**(2,2)***B**(2,2)

This will result in **M**=[14, 20; 22, 44]. Note that the code above will only work when multiplying two 2×2 matrices. In MATLAB, A*B will perform the matrix multiplication for matrices of any compatible size.

A. Programming something yourself is arguably the best way to understand how an equation or algorithm works. Write a MATLAB function that multiplies matrices **A** and **B** using nested for loops and works just like MATLAB's * operator for matrices. Your function should work on matrices of any compatible size, not just 2×2 matrices. **It must use 3 for loops.** The function header should look similar to the following:

```
function M = MatrixMul(A,B)
```

B. Write an m-file that uses the function in part A to perform the following matrix multiplication.

I. $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} * \begin{bmatrix} 5 \\ 10 \\ 15 \end{bmatrix}$

II. $\begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix} * \begin{bmatrix} 7 & 6 & 5 & 4 \end{bmatrix}$

Hint: Begin by writing down the steps of a matrix multiplication and identify the patterns in terms of the row and column counters. Additional matrix multiplication information and examples are available at https://en.wikipedia.org/wiki/Matrix_multiplication