ECE 175: Computer Programming for Engineering Applications

Homework Assignment 5

Due Date: Tuesday October 10, 2017 by 11:59 PM, via D2L and Zylab

Conventions: Name your C programs as *hw***x***p***y***.c*
where *x* corresponds to the homework number and *y* corresponds to the problem number.
For example, the C program for homework 5, problem 1 should be named as *hw***5***p***1***.c.*
Write comments to your programs. *Programs with no comments* will receive PARTIAL credit. For each program that you turn in, at least the following information should be included at the top of the C file:

- Author and Date created:
- Brief description of the program:
    - input(s):
    - output(s):
    - brief description or relationship between inputs and outputs

Submission Instructions: Submit your .c files in **D2L "Assignments" Dropbox** and **Zylab**.

Problem 1 (35 points): Jumbled File Fix

Your boss has given you a file, named Scrambled File.txt, that is completely jumbled. The file is actually a mixture of two separate files, one of which is supposed to be strictly numerical (i.e. contains only digits 0-9), and the other contains only non-numerical characters.

Write a C program that reads the file Scrambled File.txt, one character at a time. If the character is non-numerical then print this character to a file called Unscrambled.txt. If the character is a numeric digit then your boss would like to know the largest product of 4 consecutive digits in the data. For example if the file Scrambled File.txt, contains
15H8e292ll38740o234 482T0093i2423m234
then the file Unscrambled.txt, would contain the message
Hello Tim
and the following message would be printed to the screen
The largest product of a 4-digit sequence is 5 * 8 * 2 * 9 = 720
Note that the character '0' has an ASCII value of 48, and the character '9' has an ASCII value of 57. Write a function isdig as follows

```
int isdig(char c, int *Current_Digit) {
	//This function returns 0 if the character c is non-numerical, or
	//returns 1 if the character c is a digit (ASCII value 48 to 57)
	// If a 1 is returned the *Current_Digit contains the integer number
	// If, for example, c = '7' then *Current_Digit = 7, and a 1 is returned
	// If, for example, c = '+' then a 0 is returned
	// ***** Your code goes here ****
}
```

Use the function Update Samples to help keep track of old values

```
void Update_Samples(int Current_Digit, int *One_Frames_Old, int *Two_Frames_Old, int *Three_Frames_Old) {
	*Three_Frames_Old = *Two_Frames_Old;
	*Two_Frames_Old = *One_Frames_Old;
	*One_Frames_Old = Current_Digit;
	return;
}
```

For this problem, if your program is working, the Unscrambled.txt file will contain the pattern that you can read and the largest product of 4-digit sequence (from Scrambled File.txt) is 5832.

2 Vector Calculations

We wish to write a set of *C* functions to work with 3-D vectors. Write *C* functions as follows:

- *VectorMagnitude*
  - Input: An array that represents a 3-D vector $\bar{V} = \langle V_x, V_y, V_z \rangle$
  - Output: The magnitude of the input vector $\left|\bar{V}\right| = \sqrt{V_x^2 + V_y^2 + V_z^2}$
- *DotProduct*
  - Input: Two arrays of three elements each, that represent vectors $\bar{V}_1$ and $\bar{V}_2$
  - Output1: The dot product of these two vectors $\bar{V}_1 \cdot \bar{V}_2 = V_{x_1} * V_{x_2} + V_{y_1} * V_{y_2} + V_{z_1} * V_{z_2}$
  - Output2: The angle between the two vectors (in degrees) $\theta = \cos^{-1}\left(\frac{\bar{V}_1 \cdot \bar{V}_2}{\left|\bar{V}_1\right| * \left|\bar{V}_2\right|}\right)\left(\frac{180}{\pi}\right)$. Note that this calculation must call the function *VectorMagnitude* twice.
- *CrossProduct*
  - Input: Two arrays of three elements each, that represent vectors $\bar{V}_1$ and $\bar{V}_2$
  - Output1: A 3-element array that represents the cross product of the two input vectors. The cross product is described below.
  - Output2: The angle between the two input vectors (in degrees). The method for calculating the angle between the two input vectors, using the cross product is described below.

The cross product of two vectors is defined as the formal determinant of the following matrix

$$\bar{V}_3 = \bar{V}_1 \times \bar{V}_2 = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ V_{x_1} & V_{y_1} & V_{z_1} \\ V_{x_2} & V_{y_2} & V_{z_2} \end{vmatrix} = \begin{vmatrix} V_{y_1} & V_{z_1} \\ V_{y_2} & V_{z_2} \end{vmatrix} \hat{i} - \begin{vmatrix} V_{x_1} & V_{z_1} \\ V_{x_2} & V_{z_2} \end{vmatrix} \hat{j} + \begin{vmatrix} V_{x_1} & V_{y_1} \\ V_{x_2} & V_{y_2} \end{vmatrix} \hat{k}$$

For $\bar{V}_3 = \langle V_{x_3}, V_{y_3}, V_{z_3} \rangle$ then

$$\begin{aligned} V_{x_3} &= V_{y_1} V_{z_2} - V_{y_2} V_{z_1} \\ V_{y_3} &= -V_{x_1} V_{z_2} + V_{x_2} V_{z_1} \\ V_{z_3} &= V_{x_1} V_{y_2} - V_{x_2} V_{y_1} \end{aligned}$$

The cross product can be used to find the angle between two vectors

$$\theta = \sin^{-1}\left(\frac{\left|\bar{V}_3\right|}{\left|\bar{V}_1\right| * \left|\bar{V}_2\right|}\right)\left(\frac{180}{\pi}\right)$$

where $\bar{V}_3 = \bar{V}_1 \times \bar{V}_2$. Note that this calculation requires three calls to *VectorMagnitude*. Also note that the value inside the parenthesis of the $\sin^{-1}$ function might need limiting between ±1

HINT: Your functions will have to pass arrays back and forth. A pointer argument may also be necessary for the angle between the two vectors.

Write a *C* program that prompts the user to choose between the following options

- **M** for vector magnitude. This option prompts the user to enter in a single 3-D vector.
- **D** for dot product. This option prompts the user to enter in two 3-D vectors.
- **C** for cross product. This option prompts the user to enter in two 3-D vectors.

Print the output of the appropriate calculations to the screen.

2.1 Sample Code Execution

Sample Code Execution: **Red text indicates information entered by the user**

```
Enter M to calculate the magnitude of a single 3-D vector
Enter D to calculate the dot product of two 3-D vectors
Enter C to calculate the cross product of two 3-D vectors
m
Enter in Vx Vy Vz (separated by spaces)
1 2 3
|V| = |<1.00000000, 2.00000000, 3.00000000>| = 3.74165739

Would you like to perform another calcuation? (y/n)j
Enter a valid input
Would you like to perform another calcuation? (y/n)y

Enter M to calculate the magnitude of a single 3-D vector
Enter D to calculate the dot product of two 3-D vectors
Enter C to calculate the cross product of two 3-D vectors
p
Enter a valid choice

Enter M to calculate the magnitude of a single 3-D vector
Enter D to calculate the dot product of two 3-D vectors
Enter C to calculate the cross product of two 3-D vectors
D
Calculating the dot product between V1 and V2
Enter in Vx1 Vy1 Vz1 (separated by spaces)
3 2 1

Enter in Vx2 Vy2 Vz2 (separated by spaces)
3 4 5
V1 dot V2 = 22.00000000, theta = 33.74461333 degrees

Would you like to perform another calcuation? (y/n)y

Enter M to calculate the magnitude of a single 3-D vector
Enter D to calculate the dot product of two 3-D vectors
Enter C to calculate the cross product of two 3-D vectors
c
Calculating the cross product between V1 and V2
Enter in Vx1 Vy1 Vz1 (separated by spaces)
2 3 4

Enter in Vx2 Vy2 Vz2 (separated by spaces)
5 6 7
V1 x V2 = <-3.00000000, 6.00000000, -3.00000000>, theta = 7.47579063 degrees

Would you like to perform another calcuation? (y/n)n
```