Programming Fundamentals *COURSE & LAB* – Fall 2022
(BS-IT-F22 Morning)
Assignment # 1

*Assigned on: **Monday, Nov 20, 2023***

*Submission Deadline: **Friday, Nov 24, 2023 (till 6:30 AM)***

Instructions:

- **This is an individual assignment. Absolutely NO collaboration is allowed. Any traces of plagiarism/cheating would result in an "F" grade in this course.**
- Do **NOT** copy even a single sentence/statement or line of code from any other person or book or Internet or any other source.
- This assignment will ALSO be counted towards your marks in **PF-Lab**.
- Late submissions will NOT be accepted, whatever your excuse may be.
- This assignment needs to be submitted in **Soft** form. The **Submission Procedure** is given below.

Note:

- Implement all the task as **functions**
- You are not allowed to take input in the main function and not in the function which performs a specific task for input make a separate function like if you want to take height as an input write function which has a prototype like this **int getHeight();** Perform all the input validation in this function.

Submission Procedure:

i. You must write **C code** (using the notations/conventions we have discussed) in a **single zip Document**. Also make sure that you **write your Roll Number and Name at the top of your VSCode file.**.

ii. The name of your submission file must be exactly according to the format:
Mor-A1-BITF22M032
where the text shown in **BLUE** should be your **complete Roll Number**.

iii. Finally, upload and submit your file through **Google Classroom**. Make sure that you press the SUBMIT button after uploading your file.

Note: If you do not follow the above submission procedure, your Assignment will NOT be graded and you will be awarded a ZERO.

You are required to write a C program for the following tasks. Keep the following instructions in mind when writing your algorithms:

- Use meaningful variable names. Use ***camelCase*** notation to name variables.
- ***Indent*** the code properly.
- Use meaningful prompt lines for all input, and use meaningful labels for all output that is performed by your program.

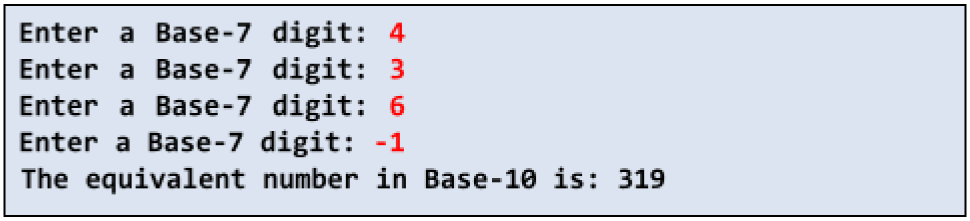
1. Take two positive integers ***n*** and ***m*** from the user. Find out the closest integer(s) to ***n*** which is/are a multiple of ***m***. Four sample runs of your algorithm are given below (text shown in red is entered by the user):

   Note: You are NOT allowed to use any loop in this algorithm.

   Function prototype: void nearestMultiple(int num);

Sample run # 1: **Enter n: 480** **Enter m: 50** **The multiple of 50 closest to 480 is 500.**	***Sample run # 2:*** **Enter n: 150** **Enter m: 50** **The multiples of 50 closest to 150 are 100 and 200.**
Sample run # 3: **Enter n: 94** **Enter m: 3** **The multiple of 3 closest to 94 is 93.**	***Sample run # 4:*** **Enter n: 40** **Enter m: 4** **The multiples of 4 closest to 40 are 36 and 44.**

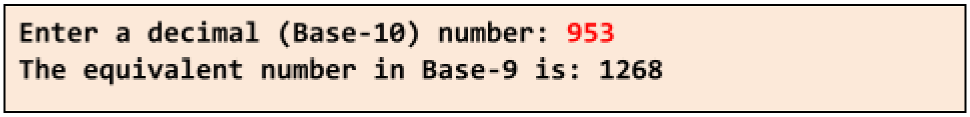
2. Input a base-7 number, **digit-by-digit**, then convert it into the equivalent decimal number (base-10). Digits of the input will be entered one-by-one in order from **Least significant to Most significant**. Remember that the valid digits in base-7 are from **0** to **6**. User will enter the value **-1** to terminate the input taking process. The following sample run corresponds to the conversion: $(\mathbf{634})_7 = (\mathbf{319})_{10}$



```
Enter a Base-7 digit: 4
Enter a Base-7 digit: 3
Enter a Base-7 digit: 6
Enter a Base-7 digit: -1
The equivalent number in Base-10 is: 319
```

   Function prototype: int conversioFromBaseSevenToTen(int baseSeven);

3. Input a decimal number (base-10) and store its equivalent in base-9 as a **single numeric value**, and display it on screen. The following sample run corresponds to the conversion:

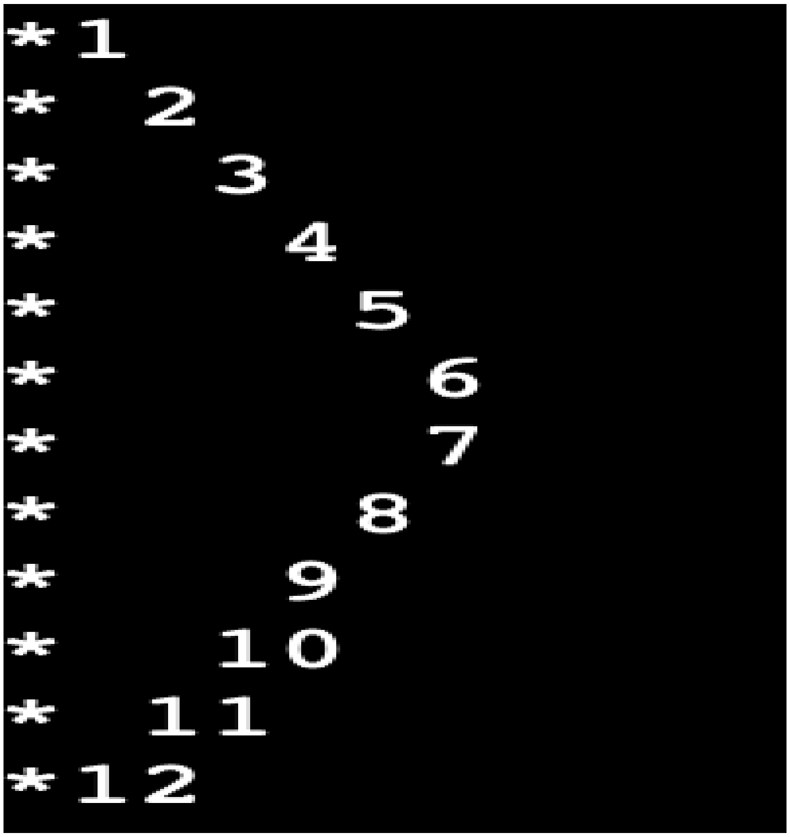


```
Enter a decimal (Base-10) number: 953
The equivalent number in Base-9 is: 1268
```

   $(\mathbf{953})_{10} = (\mathbf{1268})_9$

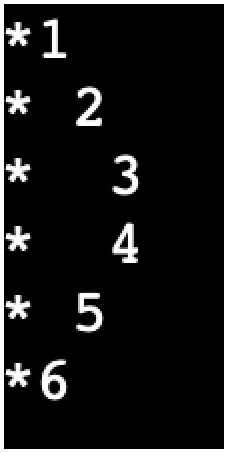
   Function prototype: int conversioFromBaseTenToNine(int baseTen);

4. Write a C program which prints the following pattern by taking the size (dimensions) from the user. For example, if the user enters the size to be 6, the following 6 by 6 pattern should be displayed on screen:



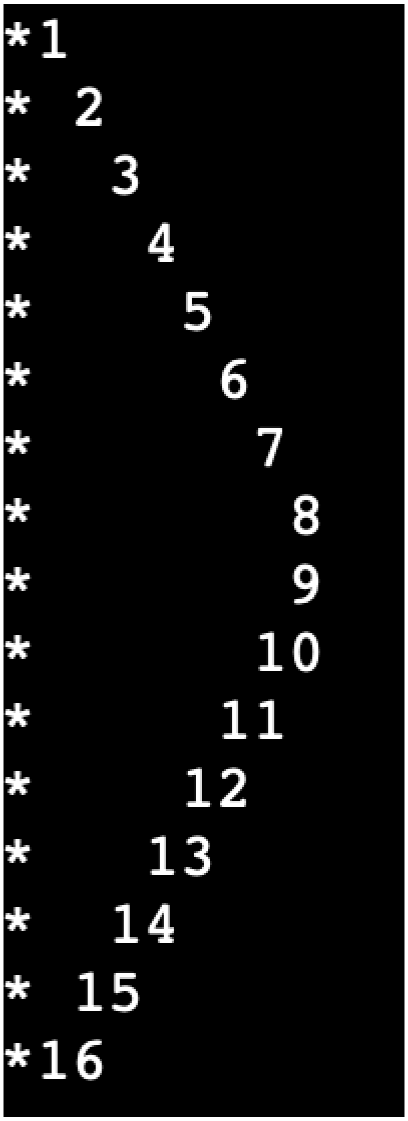
```
*1
* 2
*  3
*   4
*    5
*     6
*     7
*    8
*   9
*  10
* 11
*12
```

if size = 3



```
*1
* 2
*  3
*  4
* 5
*6
```

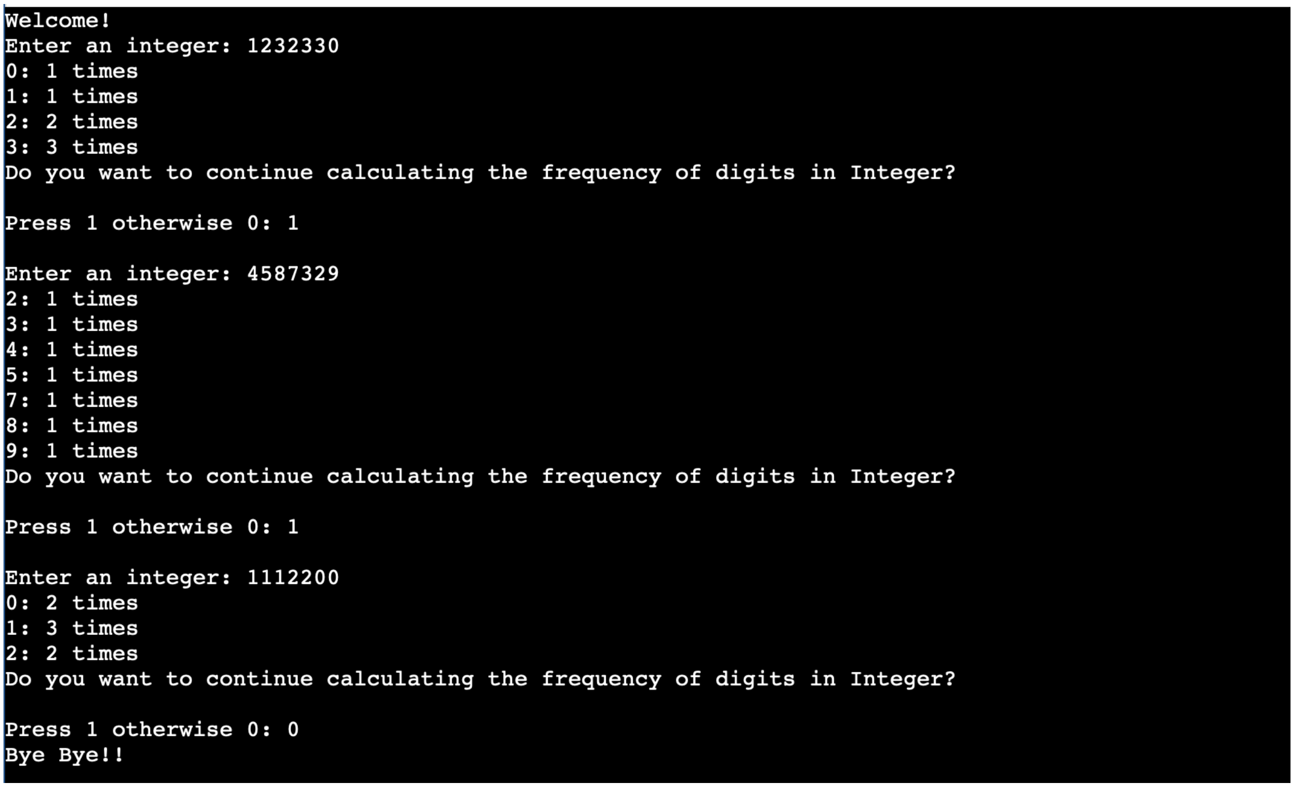
if size = 8



```
*1
* 2
*  3
*   4
*    5
*     6
*      7
*       8
*       9
*      10
*     11
*    12
*   13
*  14
* 15
*16
```

Note: size should be greater than 1.
Function prototype: void diagonalNumberTriangle();
use a separate function to take size from user(read out instruction).

5. Given an integer, find out the frequency of each digit in it.
***Sample Run*:**



```
Welcome!
Enter an integer: 1232330
0: 1 times
1: 1 times
2: 2 times
3: 3 times
Do you want to continue calculating the frequency of digits in Integer?

Press 1 otherwise 0: 1

Enter an integer: 4587329
2: 1 times
3: 1 times
4: 1 times
5: 1 times
7: 1 times
8: 1 times
9: 1 times
Do you want to continue calculating the frequency of digits in Integer?

Press 1 otherwise 0: 1

Enter an integer: 1112200
0: 2 times
1: 3 times
2: 2 times
Do you want to continue calculating the frequency of digits in Integer?

Press 1 otherwise 0: 0
Bye Bye!!
```

Function prototype: void countDigitFrequency(int num);

Note: This function and your main function will not perform input operation you should use a separate function.

6. Write a C program which uses the following FIVE functions to calculate and display the area of a rectangle whose length and width are taken from the user. Each of these functions is explained
below:

▪ **bool isValid (int num, int start, int end);**

This function will return true if the value of the variable num is in the range from start to end (both inclusive), and it should return false otherwise. There should be NO input or output performed within this function.

▪ **int getLength ();**

This function will prompt the user to enter the length of a rectangle in the range from 5 to 500 (both inclusive). In case of invalid input, this function should keep prompting the user again and again till the user provides valid input. This function MUST use the function isValid(...) (which you have implemented above) to validate the input. The valid input provided by the user should be returned from this function.

▪ **int getWidth ();**

This function will prompt the user to enter the width of a rectangle in the range from 8 to 200 (both inclusive). In case of invalid input, this function should keep prompting the user again and again till the user provides valid input. This function MUST use the function isValid(...) (which you have implemented above) to validate the input. The valid input provided by the user should be returned from this function.

- **void calculateArea (int length, int width);**

This function will receive the length and width of a rectangle (see the first two parameters). It should calculate and call the function displayArea which is below. There should be NO input or output performed within this function.

- **void displayArea (int area);**

This function will display the area of the rectangle on screen. Use meaningful label(s) in output.

☺ **GOOD LUCK!** ☺

☺ **Code your dreams into reality!** ☺