Lab Exercise 1

CSDC102: Intermediate Programming

Before your codes...

```
//******************************************************************
// Filename :
// Date :
// Subject :
// Second Semester, SY 2019 - 2020
// Activity : Lab 1A
// Problem Title :
// Input :
// Output :
//
// Honor Code : *insert honor code here*
//
// Complete Name :
// ID Number :
// Year-Course : 1-BSCS
// DCS, College of Computer Studies
// Ateneo de Naga University
//******************************************************************
```

Honor Code : This is my own program. I have not received any unauthorized help in completing this work. I have not copied from my classmate, friend, nor any unauthorized resource. I am well aware of the policies stipulated in the handbook regarding academic dishonesty.
If proven guilty, I won't be credited any points for this exercise.

Lab 1A: Arithmetic Operators

Program Description:

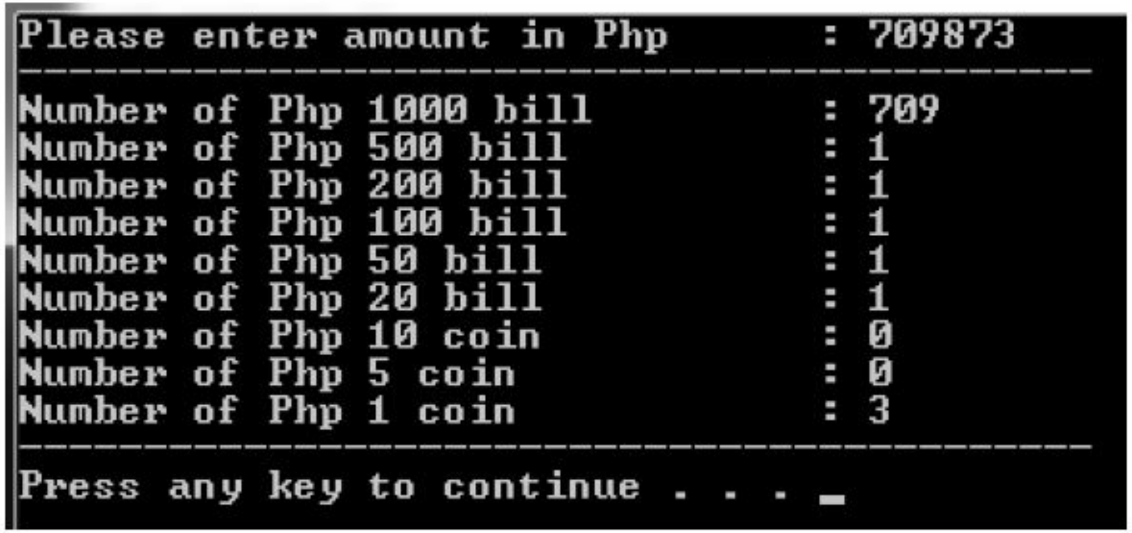
- Your task is to write a code that asks the user to enter amount in peso.
- You may assume the user will input a positive whole number
- The code then will output the number of Php1000 bill, Php500 bill, Php200 bill, Php100 bill, Php50 bill, Php20 bill, Php10 coin, Php5 coin and Php1 coin.
- Filename: **Lab1A_SURNAME.cpp**

Lab 1A: Arithmetic Operators

- Specifications:
  - Implement using Functions

Lab 1A: Arithmetic Operators

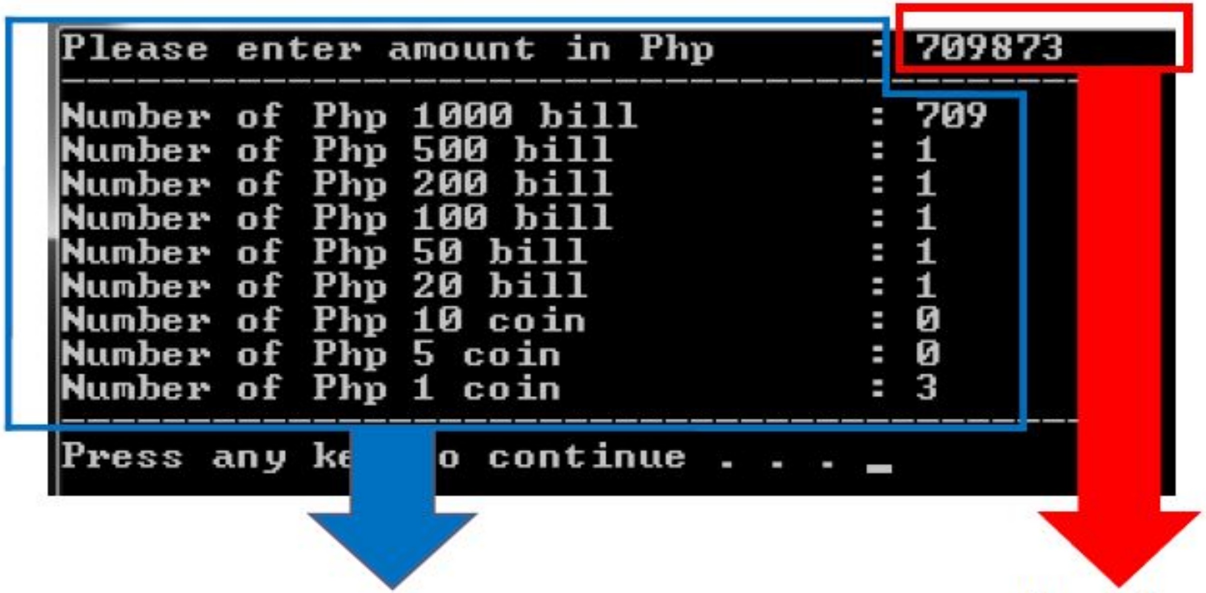
The program output should look like this:



```
Please enter amount in Php          : 709873
------------------------------------------------
Number of Php 1000 bill             : 709
Number of Php 500 bill              : 1
Number of Php 200 bill              : 1
Number of Php 100 bill              : 1
Number of Php 50 bill               : 1
Number of Php 20 bill               : 1
Number of Php 10 coin               : 0
Number of Php 5 coin                : 0
Number of Php 1 coin                : 3
------------------------------------------------
Press any key to continue . . . _
```

Lab 1A: Arithmetic Operators

The program output should look like this:



```
Please enter amount in Php           : 709873
---------------------------------------------
Number of Php 1000 bill              : 709
Number of Php 500 bill               : 1
Number of Php 200 bill               : 1
Number of Php 100 bill               : 1
Number of Php 50 bill                : 1
Number of Php 20 bill                : 1
Number of Php 10 coin                : 0
Number of Php 5 coin                 : 0
Number of Php 1 coin                 : 3
---------------------------------------------
Press any ke   o continue . . . _
```

Lab 1B: Strings

Program Description:

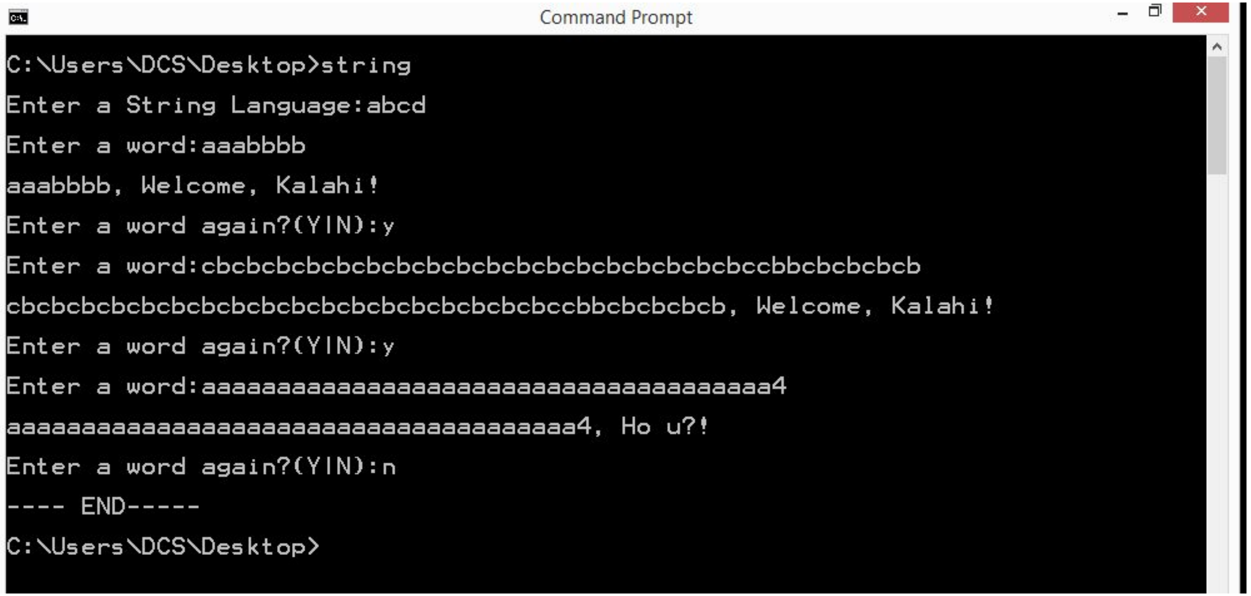
- Create a program that will accept a string **LANGUAGE** and string **WORD** from the user. Then, identify whether the WORD is a valid string from a **LANGUAGE**.
- If the **WORD** is valid, display **WORD , Welcome Kalahi!** . Otherwise, **WORD , Ho u?!**.
- Filename: **Lab1B_SURNAME.cpp**

Lab 2b: Strings

- Specifications:
  - Implement using Functions

Lab 2b: Strings

The program output should look like this:



Command Prompt

```
C:\Users\DCS\Desktop>string
Enter a String Language:abcd
Enter a word:aaabbbb
aaabbbb, Welcome, Kalahi!
Enter a word again?(Y|N):y
Enter a word:cbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbccbbcbcbcbcb
cbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbcbccbbcbcbcbcb, Welcome, Kalahi!
Enter a word again?(Y|N):y
Enter a word:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa4, Ho u?!
Enter a word again?(Y|N):n
---- END-----
C:\Users\DCS\Desktop>
```

Lab 1C: 2D Arrays

Program Description:

- Asks the user to enter the number of rows and the number of columns
- Display a table where each cell corresponds to the **product of the row value and column value**
- The first line should display a welcome message with the following format:
    - “Welcome to <Your First Name> ’s Multiplication Table Creator!”

Lab 1C: 2D Arrays

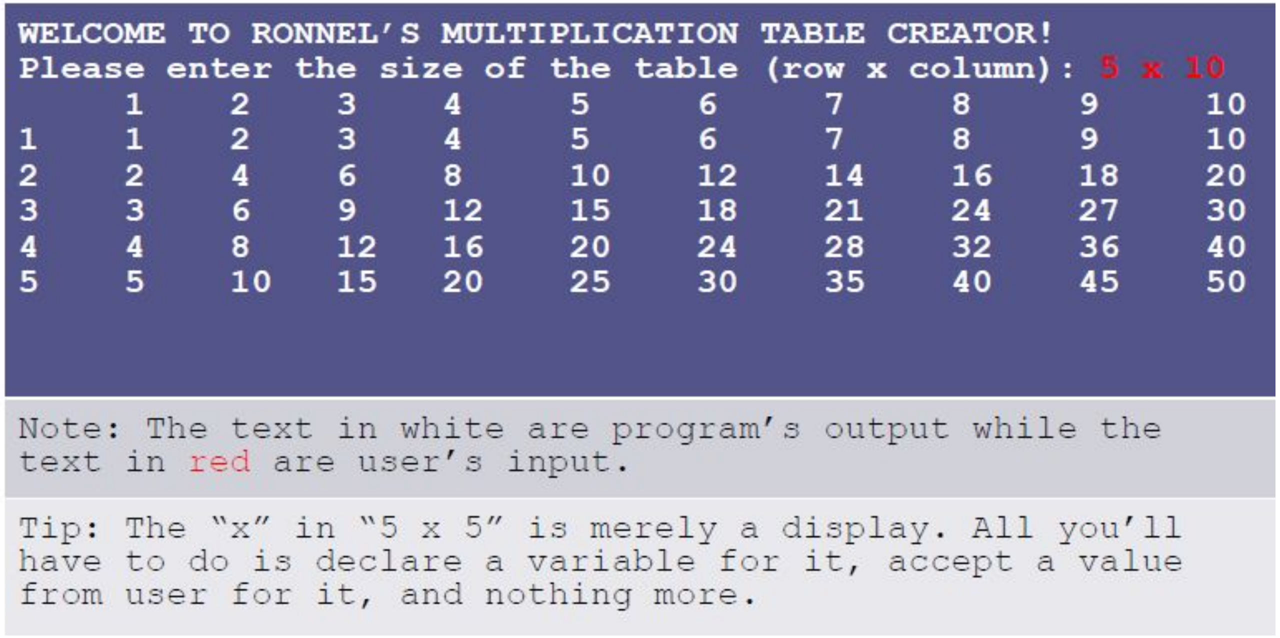
- Specifications:
  - The multiplication table must be stored in array of integers named **`mulTable`** which can hold a maximum of 100 integers in row and column.
  - A void function named **`CreateMulTable`** which takes 3 parameters - **`name_Table`** of type **`int[100][100]`** , **row_size_mulTable** of type **`int`** ***(from user input)***, and **col_size_mulTable** of type **`int`** ***(from user input)*****.** This function will create the multiplication table (**row_size_mulTable** x **col_size_mulTable** size) and store the results in **`name_Table`** array.

Lab 1C: 2D Arrays

- Specifications:
  - A void function named **`print`** which takes takes 3 parameters - **`name_Table`** of type **`int[row][col]`**, **row_size_mulTable** of type **`int`** ***(from user input)***, and **col_size_mulTable** of type **`int`** ***(from user input)*****.** This function display a **row_size_mulTable** x **col_size_mulTable** size multiplication table in the screen.

Lab 1C: 2D Arrays

The program output should look like this:



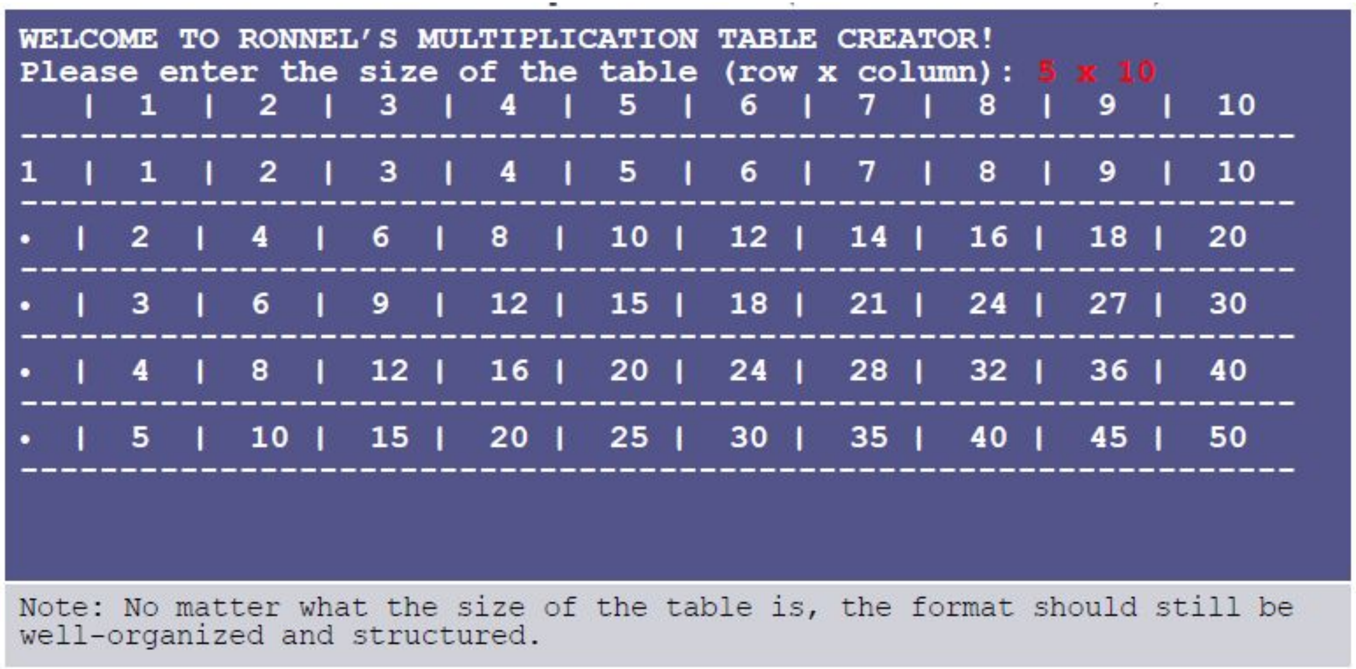
```
WELCOME TO RONNEL'S MULTIPLICATION TABLE CREATOR!
Please enter the size of the table (row x column): 5 x 10
     1    2    3    4     5     6     7     8     9     10
1    1    2    3    4     5     6     7     8     9     10
2    2    4    6    8     10    12    14    16    18    20
3    3    6    9    12    15    18    21    24    27    30
4    4    8    12   16    20    24    28    32    36    40
5    5    10   15   20    25    30    35    40    45    50
```

Note: The text in white are program's output while the text in red are user's input.

Tip: The "x" in "5 x 5" is merely a display. All you'll have to do is declare a variable for it, accept a value from user for it, and nothing more.

Lab 1C: 2D Arrays

Sample output with +10 bonus points



```
WELCOME TO RONNEL'S MULTIPLICATION TABLE CREATOR!
Please enter the size of the table (row x column): 5 x 10
    |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10
----------------------------------------------------------------
1   |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10
----------------------------------------------------------------
•   |  2  |  4  |  6  |  8  |  10 |  12 |  14 |  16 |  18 |  20
----------------------------------------------------------------
•   |  3  |  6  |  9  |  12 |  15 |  18 |  21 |  24 |  27 |  30
----------------------------------------------------------------
•   |  4  |  8  |  12 |  16 |  20 |  24 |  28 |  32 |  36 |  40
----------------------------------------------------------------
•   |  5  |  10 |  15 |  20 |  25 |  30 |  35 |  40 |  45 |  50
----------------------------------------------------------------
```

Note: No matter what the size of the table is, the format should still be well-organized and structured.