COMSATS University Islamabad, Lahore Campus

Lab assignment #03 (Project based on strings)

QNO:01

Take a string from user as input and do following in individual switch cases by just calling each function as instructed below:

Note: Don't use any predefined function from library otherwise it will deduct your mark.

1) **Create a function named ReverseString() to reverse the string i.e. user entered Hello and it will return olleH.**
2) **Create function named FindFrequency() to find frequency of each element.**
3) **Create a function named StrLen() which will return length of the string.**
4) **Create a function named ConditonOfString() which will tell the string consist of how many vowels, consonants or number if exist.**
5) **Create a function named toUpper() and toLower() to covert cases of string.**
6) **Create a function named strcat() which will take another string from user and concatenate it with other string. i.e. str1="COMSATS", str2=" University" New str= "COMSATS University"**
7) **Create a function named PrintAscii() which will print ascii value of each character of string.**
8) **Create another function named ReverseEachWord() to reverse each word not whole string. i.e. str1=" This is a cat"; new str="cat a is This"; Hint: [Use two arrays one will take original string while other will store the reversed string.]**
9) **Create another function named strcmp() which will take two strings from user and compare them.**
10) **Create another function named strcpy() which will copy strings the string to another string and display it.**
11) **Create your own version of strncpy().**

Reading Task 2: The Magic Square

The **magic square** is a square matrix, whose **order is odd** and where the sum of the elements for each row or each column or each diagonal is same. The sum of each row or each column or each diagonal can be found using this formula. n(n2+ 1)/2 where **'n'** is the order of the matrix.

8	1	6
3	5	7
4	9	2

Figure 1: *Magic Square of order 3*

A method for constructing magic squares of odd order was published by the French diplomat de la Loubère in his book, *"A new historical relation of the kingdom of Siam"* (Du Royaume de Siam, 1693), in the chapter entitled *The problem of the magical square according to the Indians*. The method operates as follows:

1. The method prescribes starting in the central column of the first row with the number 1.
2. After that, the fundamental movement for filling the squares is diagonally up and right, one step at a time.
3. If a filled square is encountered, one moves vertically down one square instead, then continues as before.
4. When an "up and to the right" move would leave the matrix, it is wrapped around to the last row or first column, respectively.

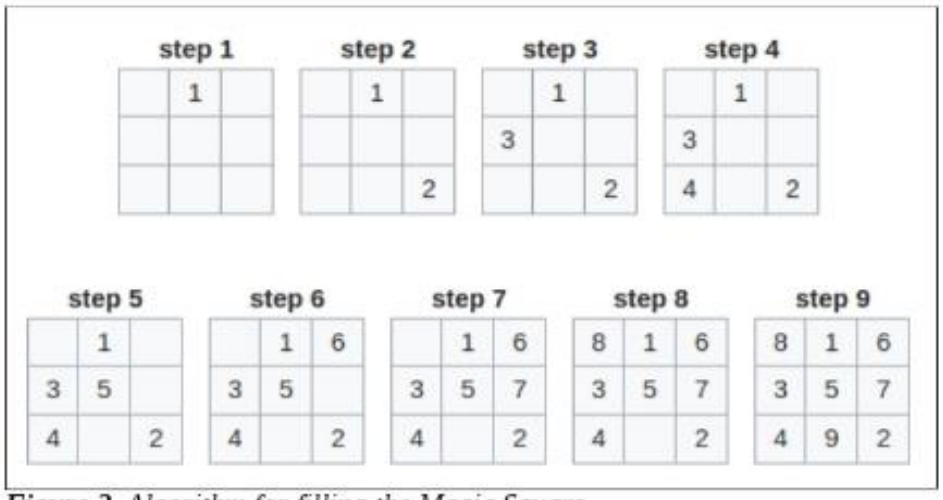

Figure 2. *Algorithm for filling the Magic Square*

In-Lab Task 1: 2D Arrays and user defined functions

- Your task is to declare a 2D array whose dimensions should be entered by the user of the program.
- Then you should initialize the array with ones (using nested loops).
- Next you have to write a function **array_multiply()** which takes in the array or its pointer as argument and multiplies all its entries with a user input number. (**Hint:** you will also need to pass in the dimensions of the matrix to this function).
- Similarly write a function **array_add()** that adds a constant number to all the entries in a 2D array.
- Print the results of calling these functions.

In-Lab Task 2: Implementing the algorithm for the Magic Square

In this task you have to make a magic square and display it on the screen. You are given a Starter Code (Annex I), that does the following:

- Asks the user to enter the order **'n'** of the magic square (odd numbers only).
- Declares a 2D array of size **n** x **n** and initializes it with zeros.
- Prints the magic square on the screen.

Your job is to complete this code by implement the algorithm discussed in the **Reading Task 2**.

References

1. **https://en.wikipedia.org/wiki/Magic_square#A_method_for_constructing_a_magic_square_of_odd_order**