## Objectives

After performing this lab, students shall be able to:

* Dynamic memory allocation (Revision)
* Double Pointers. (2D structures)
* Double pointers and dynamic allocation of 2D arrays.
* Memory Leak and Dangling pointer issues.

**TASK 1: (1+1+1)+2+(3+3+3+1)**

A C++ program “**Encrypt**” that creates a dynamic char array of **size defined by the user**. Then takes input of that array from the user and resize the original array to size equal to the number of characters entered by the user.

Your function now asks the user to enter encryption key that can be any alphabet **(a-z, A-Z)**. Encrypt function adds the ASCII of encryption key to each char of your input array to generate encrypted sentence. If the shift takes you past the end of the alphabet, just rotate back to the front of the alphabet. In the case of a shift factor of 3 the message “**wxyz”** would map to “**zabc”** i.e. w becomes z, x becomes a, y becomes b and z becomes c.

**Note:** Array subscript notation cannot be used.

**For Example:**

|  |
| --- |
| **INPUT: (3)**  **Enter Size of array:** 100  **Enter the array elements:** This is object oriented class  **Enter the encryption key**: d  **OUTPUT: (2+10)**  **New Size of array:** 29  **Encrypted array:** Pdeojeojkxfaypjkneajpazjyhwoo |

**TASK 2: 2+2+2+3+3+3=15**

1. Write a function **char\*\* AllocateMemory(int& rows, int& cols)** that takes size of matrix (rows and columns) from user, allocates memory for the matrix and return its pointer.
2. Write a function **char\*\* InputMatrix(char\*\* matrix, const int rows, const int cols)** which takes input the values in matrix from user(console) and return the matrix to main function.
3. Write a function **void DisplayMatrix(char\*\* matrix, const int& rows, const int& cols)** that displays the matrix in proper format.
4. Write a function that does the following:

* Creates three dynamic char arrays namely **alphabets, numbers, and specialchar.** (Define the sizes yourself).
* Iterate the 2D array and separate alphabet elements and save them in the alphabets array, separate number elements and save them in numbers array and separate special character elements and save them in the specialchar array.
* The function returns the three arrays **alphabets, numbers, and specialchar.**
* **Note:** The three arrays should not consume any extra space. **Resize the arrays accordingly.**

**For Example:**

|  |
| --- |
| **INPUT:**  **Sample Matrix** is:  A 1 v @  + 9 s 6  P # ^ 4  **OUTPUT:**  **Function will return the following arrays:**  alphabets = A v s P  numbers = 1 9 6 4  specialchar =@ + # ^ |