**National University of Computer and Emerging Sciences**



Lab Manual # 02

Object oriented programming

(Section BSE-2A)

|  |  |
| --- | --- |
| Course Instructor | Ms. Arooj Khalil |
| Lab Instructor(s) | Ms. Fareeha Maqbool  Ms. Amara Nasir |
| Semester | Spring 2023 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

**Objectives:**

After performing this lab, students shall be able to:

Dynamically allocate and deallocate memory.

Create and manipulate dynamic 1D and 2D arrays.

**Question no 1:**

Write a program following the instructions given below:

Take two arrays of integers A and B of sizes M and N respectively (M and N taken from User). The program should allocate arrays dynamically. Then you need to mix these arrays into a third array named C such that the following sequence is followed.

All even numbers of A from left to right are copied into C from left to right.   
All odd numbers of A from left to right are copied into C from right to left.   
All even numbers of B from left to right are copied into C from left to right.   
All old numbers of B from left to right are copied into C from right to left.

A, B and C are the arrays to Mix. e.g., A is {3, 2, 1, 7, 6, 3} and B is {9, 3, 5, 6, 2, 8, 10} the resultant array C is {2, 6, 6, 2, 8, 10, 5, 3, 9, 3, 7, 1, 3}

Display all of the arrays with proper label to show the result.

**Question no 2:**

Write a program which allows user to enter as many numbers as he likes until -1 is entered by the user in the input of Array\_1 and concurrently there will be Array\_2 which will have input right after of every input in Array\_1 element. Array\_1 will have input of numbers between 4 & 32 and Array\_2 will have input of Binary Numbers. Program should store these numbers in arrays. The program should allocate array dynamically and expand these array if the size of the input goes beyond the size of the array. Your program should also de-allocate acquired dynamic memory at the end of the life cycle.

Your program should expand array by handling the following conditions:

* Make a new array of SIZE+1 of original array.
* Copy all the elements of original arr in the new array
* The elements from oldSize to newSize should be initialized to zero
* Deallocate the original array and point the arr pointer to new array
* Assign NULL to the pointer used to make new array

e.g.

1st Element in Array\_1: 6

1st Element in Array\_2: 000110

2nd Element in Array\_1: 33

Error please input in the range 4 & 32

2nd Element in Array\_1: 32

2nd Element in Array\_2: 100000

2nd Element in Array\_1: -1

Thank you for input.

Both arrays are same in Binary value Congrats.

**Note: If the actual binary value of any variable in array\_1 is not equal to binary value entered in array\_2 then the arrays are not same.**

**Question no 3:**

Write a program that inputs the number of rows and columns from the user. It then inputs the elements to store in the matrix. The program calculates the sum of each row and each column and displays on the screen. If it is square matrix. It also calculates the sum of diagonal elements and displays it on screen.

**Example:**

Enter number of rows and columns in matrix:

3 3

Enter elements of matrix:

9 8 7 6 5 4 3 2 1

Row sum and column sum:

9 8 7 **24**

6 5 4 **15**

3 2 1 **6**

**18 15 12**

**Question no 4:**

i) 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.

ii) Write a function void InputMatrix(char\*\* matrix, const int rows, const int cols) which takes input the values in matrix from user(console).

iii) Write a function void DisplayMatrix(char\*\* matrix, const int& rows, const int& cols) that displays the matrix in proper format.

iv) 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, if the Sample Matrix is  
A 1v @  
+ 9s 6  
P # ^4  
Your function will return the following arrays: alphabets = A v s P  
numbers = 1 9 6 4  
specialchar = @ + # ^