| ***Computer Engineering Department*** |
| --- |
| ***CE100L: Computing Fundamentals & Programming*** |

| ***Course Instructor: Usama Bin Shakeel*** | ***Dated: 28/01/2022*** |
| --- | --- |
| ***Teaching Assistant: Aqsa Khalid*** | ***Semester: Fall 2021*** |
| ***Lab Engineer: Nadir Abbas*** | ***Batch: BSCE2021*** |

# **Lab 14B. Data Manipulation in 2 Dimensions using Dynamic Memory Allocations**

| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| --- | --- | --- | --- | --- |
| NIMRA MAQBOOL | BSCE21012 |  |  |  |

Checked on: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Objective**

The objective of this lab is to understand open ended problem solving.

## **Equipment and Component**

| **Component Description** | **Value** | **Quantity** |
| --- | --- | --- |
| Computer | Available in lab | 1 |

## **Conduct of Lab**

1. Students are required to perform this experiment individually.
2. In case the lab experiment is not understood, the students are advised to seek help from the course instructor, lab engineers, assigned teaching assistants (TA) and lab attendants.

## **Theory and Background**

A dynamic array is quite similar to a regular array, but its size is modifiable during program runtime. DynamArray elements occupy a contiguous block of memory. Once an array has been created, its size cannot be changed. However, a dynamic array is different. A dynamic array can expand its size even after it has been filled. During the creation of an array, it is allocated a predetermined amount of memory. This is not the case with a dynamic array as it grows its memory size by a certain factor when there is a need.

**Lab Task**

1. Write a program in C++ for adding and multiplication of 2D arrays.

| void additionOF2dArray() {  int col;  int row;  int arr[50][50]; //initializing  int arr1[50][50];  int arr2[50][50];  cout << "please enter rows of arr" << endl;  cin >> row; //taking rows and columns as input  cout << "please enter columns of arr" << endl;  cin >> col;  cout << "please enter the values of arr1= ";  for (int i = 0; i < row; i++) {  for (int j = 0; j < col; j++) { //taking the values of array  cin >> arr1[i][j];  }  }  cout << " The arr1 is = ";  for (int i = 0; i < row; i++) { //displaying the array  for (int j = 0; j < col; j++) {  cout << arr1[i][j] << " ";  }  cout << endl << "\t\t";  }  cout << endl;  cout << "please enter the values of arr2= ";  for (int i = 0; i < row; i++) { //taking the values of array2  for (int j = 0; j < col; j++) {  cin >> arr2[i][j];  }  }  cout << " The arr2 is = ";  for (int i = 0; i < row; i++) {  for (int j = 0; j < col; j++) { //displaying array2  cout << arr2[i][j] << " ";  }  cout << endl << "\t\t";  }  cout << endl;  cout << " The sum is =";  for (int i = 0; i < row; i++) {  for (int j = 0; j < col; j++) {  arr[i][j] = arr1[i][j] + arr2[i][j]; //taking sum of arr1 and arr2  cout << arr[i][j] << " ";  }  cout << endl << "\t\t";  }  }  void multiplicationOf2dArray() {  int row1;  int row2;  int col1;  int col2; //initializing rows and col  cout << "please enter rows of arrA" << endl;  cin >> row1;  cout << "please enter columns of arrA" << endl; //taking rows and col for arrays  cin >> col1;  cout << "please enter rows of arrB" << endl;  cin>>row2;  cout << "please enter columns of arrB" << endl;  cin>>col2;  int arrA[50][50];  int arrB[50][50];  int arrC[50][50];  if(col1==row2){  cout << "please enter the values of arr1= ";  for (int i = 0; i < row1; i++) {  for (int j = 0; j < col1; j++) { //taking values of array1  cin >> arrA[i][j];  }  }  cout << " The arr1 is = ";  for (int i = 0; i < row1; i++) {  for (int j = 0; j < col1; j++) { //displaying array2  cout << arrA[i][j] << " ";  }  cout << endl << "\t\t";  }  cout << endl;  cout << "please enter the values of arr2= ";  for (int i = 0; i < row2; i++) {  for (int j = 0; j < col2; j++) { //taking the values of array 2  cin >> arrB[i][j];  }  }  cout << " The arr2 is = ";  for (int i = 0; i < row2; i++) {  for (int j = 0; j < col2; j++) { //displaying array2  cout << arrB[i][j] << " ";  }  cout << endl << "\t\t";  }  cout << endl;  int i;  for ( i = 0; i < row2; i++) {  for (int j = 0; j < col2; j++) { //taking the values of arrays and multiplying them  for(int k=0;k<col1;k++){  arrC[i][j] = ((arrC[i][j])+ (arrA[i][k]\*arrB[k][j]));  }  }  }  cout<<"The multiplication = ";  for ( i = 0; i < row1; i++) {  for (int j = 0; j < col2; j++){  cout<<arrC[i][j]<<" "; //displaying the array  }  cout<<endl;  }  }  else{  cout<<"arrays can't be multiplied"<<endl;  }  } |
| --- |

2. Write a function in C++ which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column.

[Assuming the 2D Array to be a square matrix with odd dimensions i.e. 3x3, 5x5, 7x7 etc...]

Example, if the array contents is

3 5 4

7 6 9

2 1 8

Output through the function should be :

Middle Row : 7 6 9

Middle column : 5 6 1

| void middleRowAndColumn(int size){  int arrX[50][50];  cout<<"please enter size"<<endl;  cin>>size; //taking size  if(size%2==1 ){  cout << "please enter the values of arrX= ";  for (int i = 0; i < size; i++) { //taking values of arr  for (int j = 0; j <size; j++) {  cin >> arrX[i][j];  }  }  cout << " The arrX is = ";  for (int i = 0; i < size; i++) { //displaying the array  for (int j = 0; j < size; j++) {  cout << arrX[i][j] << " ";  }  cout << endl << "\t\t";  }  cout<<"the middle row is= ";  for (int i = 0; i <size; i++) {  for (int j = 0; j <size; j++) { //displaying middle row  if(i==size/2){ //dividing by 2 for taking middle number  cout<<arrX[i][j]<<" ";  }  }  cout << endl << "\t\t";  }  cout<<"the middle column is= "<<endl;  for (int i = 0; i <size; i++) {  for (int j = 0; j <size; j++) {  if(j==size/2){ //dividing by 2 for taking middle number  cout<<"\t\t"<<arrX[i][j]<<" "; //displaying middle column  }  }  cout << endl;  }  }  else {  cout<<"The size is not of the odd number...0\_0";  }  } |
| --- |

3. Write a user defined function named Upper-half() which takes a two dimensional array A, with size N rows and N columns as argument and prints the upper half of the array.

e.g.,

2 3 1 5 0 2 3 1 5 0

7 1 5 3 1 1 5 3 1

2 5 7 8 1 Output will be: 1 7 8

0 1 5 0 1 0 1

3 4 9 1 5 5

| void upperHalf(int nRows,int nCol){  cout<<"please enter row size"<<endl;  cin>>nRows;  cout<<"please enter column size"<<endl; //taking rows and columns  cin>>nCol;  int arr[40][40];  cout << "please enter the values of arrX= ";  for (int i = 0; i < nRows; i++) {  for (int j = 0; j <nCol; j++) {  cin >> arr[i][j]; //entering values of array  }  }  cout << " The arr is = ";  for (int i = 0; i < nRows; i++) {  for (int j = 0; j < nCol; j++) {  cout << arr[i][j] << " "; //displaying the array  }  cout << endl << "\t\t";  }  cout<<endl;  cout<<"the upper half of array is ="<<endl;  for (int i = 0; i < nRows; i++) {  for (int j = 0; j < nCol; j++) {  if(i<=j){ //array i must be less than j  cout<<"\t" <<arr[i][j]; //displaying the upper half elements of array  }  else{  cout<<"\t";  }  }  cout<<endl ;  }  } |
| --- |

#### **Assessment Rubric for Lab**

**Method for assessment:**

Lab reports and instructor observation during lab sessions. Outcome assessed:

a. Ability to conduct experiments, as well as to analyze and interpret data (P) b. Ability to function on multi-disciplinary teams (A)

c. Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (P)

| Performance metric | Mapping (task no. and description) | | Max marks | Exceeds expectation | Meets expectation | Does not meet expectation | Obtained marks |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Realization of experiment (a) | 1 | Functionality | 40 | Executes without errors excellent user prompts, good use of symbols, spacing in output. Through testing has been completed (35-40) | Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed (20-34) | Does not execute due to syntax errors, runtime errors, user prompts are misleading or non-existent. No testing has been completed (0-19) |  |
| 2. Teamwork (b) | 1 | Group Performance | 5 | Actively engages and cooperates with other group member(s) in effective manner (4-5) | Cooperates with other group member(s) in a reasonable manner but conduct can be improved (2-3) | Distracts or discourages other group members from conducting the experiment (0-1) |  |
| 3. Conducting experiment (a, c) | 1 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 2 | Viva | 10 | Answered all questions (8-10) | Few incorrect answers (5-7) | Unable to answer all questions (0-4) |  |
| 4. Laboratory safety and disciplinary rules (a) | 1 | Code commenting | 5 | Observes lab safety rules; handles the equipment and parts with care and adheres to the lab disciplinary guidelines aptly (4-5) | Generally observes safety rules and disciplinary guidelines with minor lapses (2-3) | Disregards lab safety and disciplinary rules (0-1) |  |
| 5. Data collection (c) | 1 | Code Structure | 5 | Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap (4-5) | Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables (2-3) | Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy (0-1) |  |
| 6. Data analysis (a, c) | 1 | Algorithm | 20 | Solution is efficient, easy to understand, and maintain (15-20) | A logical solution that is easy to follow but it is not the most efficient (6-14) | A difficult and inefficient solution (0-5) |  |
| 7. Computer use (c) | 1 | Documentation | 5 | Timely documented (4-5) | Late documented (2-3) | Not documented (0-1) |  |
|  | Max Marks (total): | | 100 | Obtained Marks (total): | | |  |

Lab Engineer Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_