Object Oriented Programming Lab

SPRING 2023

LAB 03

FAST National University of

Computerand Emerging Sciences

Learning Outcomes

In this lab you are expected to learn the following:

- Double Pointers
- Dynamic Memory Allocation

POINTERS

A pointer is a variable that stores an address of a memory location.

Pointer Declaration and initialization

Pointers is declared as follow.

Type* <variable Name>;

```
#include<iostream>
using namespace std;

int main(){

    int *x=NULL;
    // or
    int* x=NULL;
    //or
    int * x=nullptr;

    return 0;
}
```

Address-of operator (&)

The ampersand, **&**, called the **address of operator**, is a unary operator that returns the addressof its operand.

```cpp
#include<iostream>
using namespace std;

int main(){

    //declare and initialize string* variable
    string *str=nullptr;
    //declare and initialize string variable
    string temp="hello";
    //store the address of temp to str, both will
    //start pointing to same memory location
    str=&temp;


    //print the address of temp
    cout<<"temp Address "<<&temp<<endl;

    //print value in str
    cout<<"str Value "<<str<<endl;

    return 0;
}
```

```
temp Address 0x7ffdc2d7c440
str Value 0x7ffdc2d7c440
```

Dereference operator (*)

The unary operator * is the dereferencing operator that returns the value of its operand.

```cpp
#include<iostream>
using namespace std;

int main(){

    //declare and initialize string* variable
    string *str=nullptr;
    //declare and initialize string variable
    string temp="hello";
    //store the address of temp to str, both will
    //start pointing to same memory location
    str=&temp;


    //print the address of temp
    cout<<"temp Value "<< temp<<endl;

    //Dereferencing to print value in str
    cout<<"str Value "<< *str<<endl;

    return 0;
}
```

```
temp Value hello
str Value hello
```

Initializing double pointers

```
#include <iostream>
using namespace std;

int main()
{
    int** dpointer=new int*[5];
    for(int i=0;i<5;i++)
    {
        *(dpointer+i)=new int[6];
            //or
        dpointer[i]=new int[6];

    }


    return 0;
}
```

Traversing double pointers

```
for(int i=0; i<5; i++)
{
    for(int j=0; j<6;j++)
    {
        cout<<*(*(dpointer+i)+j); //pointer notation
        //or
        cout<<dpointer[i][j]; //Array notation
    }
}
```

Problem 1:

In this task, your job is to create a function Swapper() that takes three arguments num1, num2, and product. The first argument should be passed by reference, second argument should be passed by pointer, and third argument should be passed by value. Your Swapper function should first swap the values of num1 and num2 with each other, and then compute their product and store it in product variable. In the end, you need to display the values of all three variables inside your Swapper method as well as in your main function.
Note: You are not allowed to create any other variable or pointer inside your Swapper method

Problem 2

In this task you have to make following two functions

Write a function named as create2DArray which creates a 2-Dimensional array. Your function receives three arguments:

(i) an alias to a 2D pointer
(ii) number of rows
(iii) number of columns

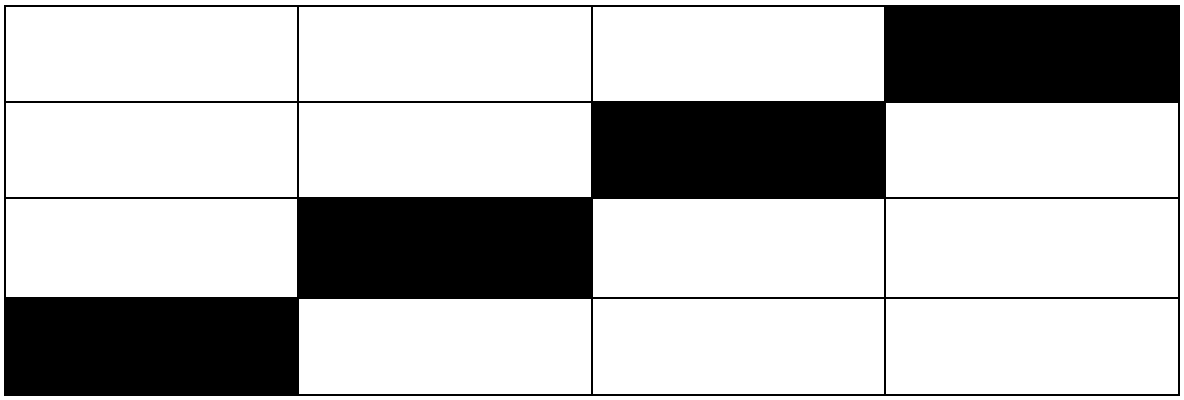
Your goal is to first allocate the memory for rows and then for columns. Initialize it with random values

Write a function named as calDiagonal that receives three arguments:

(i) a 2D pointer p;
(ii) number of rows sizeA;
(iii) number of columns sizeB;

The function will return sum of diagonal columns which are shown as black area in the following figure of the 2D array. Remember you have to calculate sum of values of the array shown in black area in the given fiure only and return the sum

Prototype: int calDiagonal(intp,int sizeA,int sizeB)**



Problem 3:

Write a function named as rotateArray that takes an array of type integer, its size, input n whichshifts elements of an array to the right n times, and a number m which divide array into m partsas function parameters. You first need to need to divide array into m number of parts and then shifts elements of an array to the right n times. Array under consideration is circular.

Function Prototype: rotateArray (int *arr, int sizeofArray, int n, int m)

Original Array

1	2	3	4	5	6

Updated Array (n=2 and m=2)

2	3	1	5	6	4

Problem 4:

In this task ,create a function that will take a

(i) 2D pointer
(ii) Number of rows
(iii) Number of cols as argument

For main: Initialize 2D array with random values

Create a 1D dynamic array Result of size 4 to store the sum. Sum will be calculated on the basis of following conditions

(i) At zero index of Result array, store sum of all values with even row no. and even column no.
(ii) At 1st index of Result array, store sum of all values with even row no. and odd column no.
(iii) At 2nd index of Result array, store sum of all values with odd row no. and even column no
(iv) At 3rd index of Result array, store sum of all values with odd row no. and odd column no

Return the result array.

Function Prototype: int* sumArray (int arr,int rows,int cols)**

Note: Use pointer notation

Submission Instructions:

1. Create a single cpp file containing all the functions of the problems and main function.Save the cpp file as Q1.cpp.
2. Now create a new folder with name ROLLNO_SEC_LAB01 e.g. i22XXXX_A_LAB02.
3. You need to display your roll no and name before the output of each question.
4. comment out the main Function and then move to this newly created directory andcompress it into a .zip file.
5. Now you have to submit this zipped file on Google Classroom.
6. If you don't follow the above-mentioned submission instruction, you will be marked zero.
7. Plagiarism in the Lab Task will result in zero marks in the whole category.