**Experiment No.3**

**OBJECTIVES:**

Things that will be covered in today’s lab:

* + Double Pointers
  + Dynamic Two Dimensional Arrays

#### THEORY:

**Double Pointer:** A double pointer is a pointer to pointer. Normally, a pointer contains the address of a variable. When we define a pointer to a pointer, the first pointer contains the address of the second pointer, which points to the location that contains the actual value as shown in Figure 3.1:

Pointer Pointer Variable

Addres

Address

Value

##### Figure 3.1

In C++, you declare a double pointer variable by using two asterisks (\*\*) between the data type and the variable name. The general syntax to declare a pointer variable is as follows:

datatype \*\* identifier;

Int var=3000;

int \*ptr;

int \*\*pptr;

ptr = &var;

pptr = &ptr;

// assigns the address of var to ptr

// assigns the address of ptr to pptr

cout<<var<<endl; cout<< \*ptr<<endl; cout<< \*\*pptr<<endl;

// pointer points to var

// Pointer points of ptr

**Dynamic 2-D Array:** First, we will allocate memory for an array which contains a set of pointers. Next, we will allocate memory for each array which is pointed by pointers. The de- allocation of memory is done in the reverse order of memory allocation.

int \*\*Array = 0;

Array = new int \*[ROWS];//memory allocated for elements of row

for(int i=0; i<ROWS; i++)//memory allocated for each col.

Array[i] = new int[COLUMNS];

for(int i=0; i<ROWS; i++ )//free the allocated memory

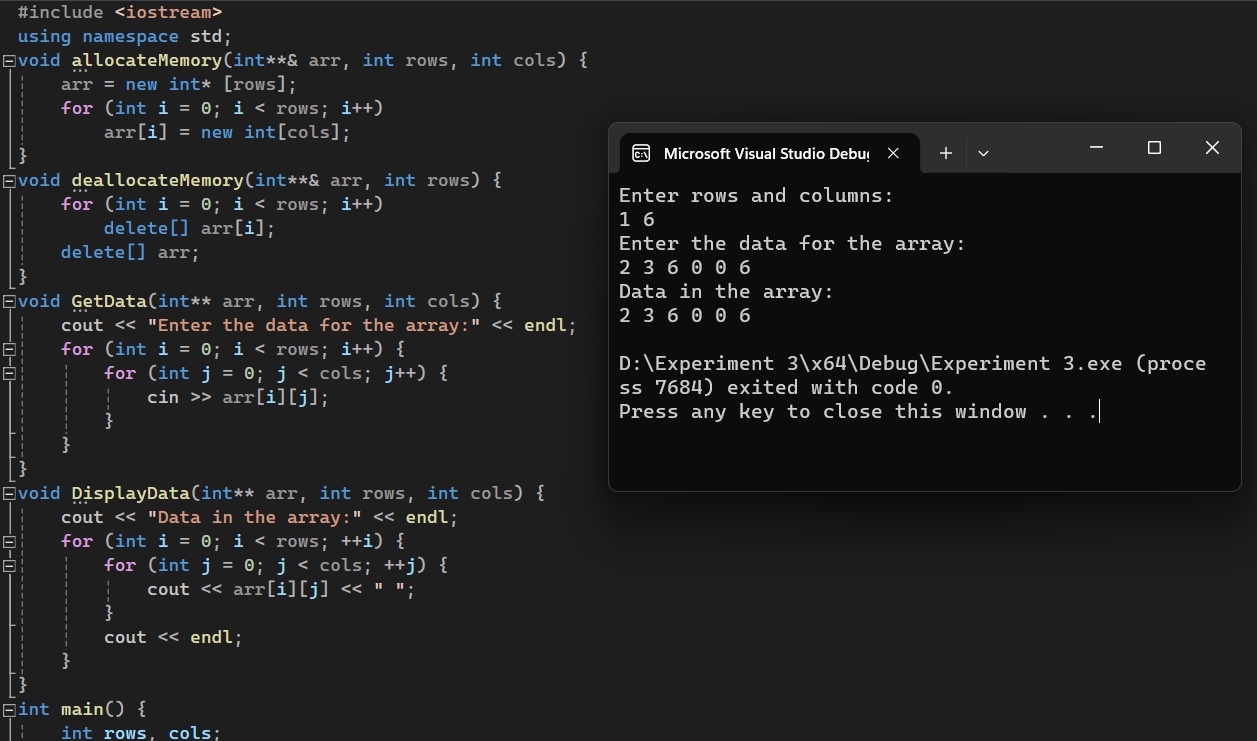
delete [] Array[i] ;

delete [] Array ;

**For all of the exercises below save your code on the learning management system (LMS) and give the screen shot of the output you get on the console in the space provided after every exercise.**

**Exercise: 1**

Using the above example as guideline please allocate memory to a 2-dimensional array via function call, then populate and display data and finally deallocate memory via functions. Your code should run for the following main function, please include your rollnumber as the first row of the data e.g, 2 3 1 1 2 7



int main()

{

int rows, cols;

cout<<"Enter rows and columns"<<endl;

cin>>rows;

cin>>cols;

int \*\*\_2DArr;

allocateMemory(\_2DArr, rows, cols);

GetData(\_2DArr, rows, cols);

DisplayData(\_2DArr, rows, cols);

deallocateMemory((\_2DArr, rows);

return 0;

}

**Exercise: 2**

A double pointer is used for declaring two-dimensional arrays dynamically. For example,

int \*\*p;

p=new int \*[rows];

for(inti=0;i<rows; i++)

p[i]=new int [cols];

We want to implement a **triangular 2D** array in which each row has one column greater than the previous one. i.e., the first row has one column, the second one has two columns, and the third one has three columns and so on. You have to take the total number of rows from the user. Figure 3.2 shows an example of a triangular 2D array with four rows.

##### Figure 3.2

int \*\*T2D

|  |
| --- |
| \* |
| \* |
| \* |
| \* |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Write following functions with given prototype in the space provided:

1. Void AddColumns (int \* &, int size)

This function takes a single pointer by reference and dynamically allocates memory to it. You will call this function in main to allocate number of columns to each row turn by turn.

1. void RowWiseInput(int \*, int size)

This function simply takes a pointer as an argument and takes input in it from the user. The second argument is the size of 1D array pointed by pointer.

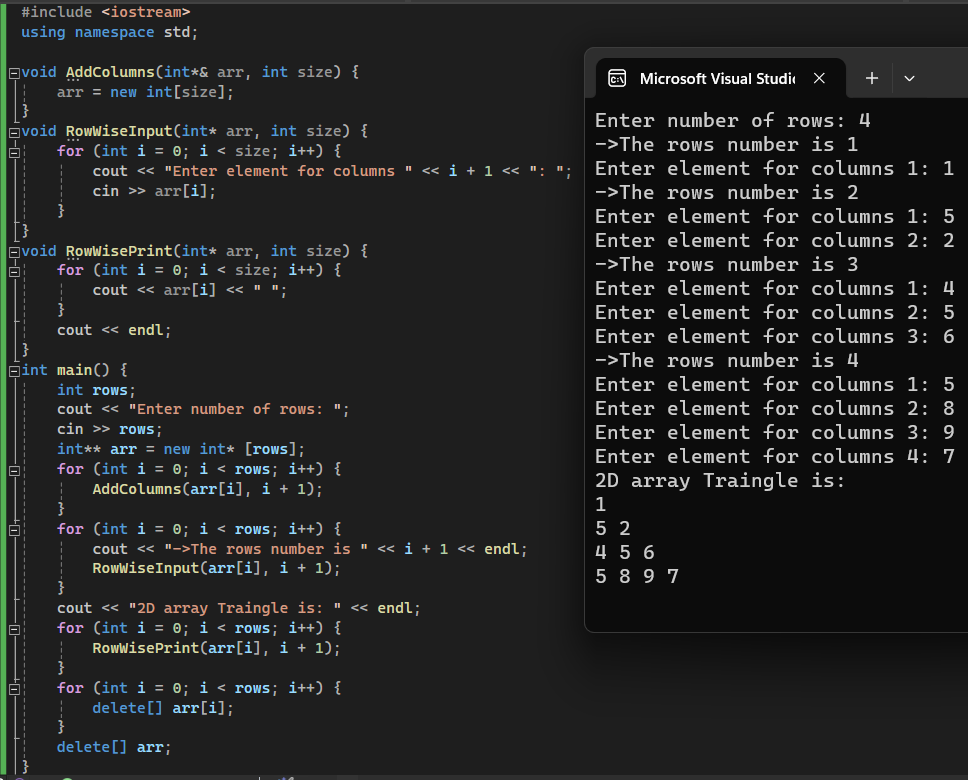
1. void RowWisePrint(int \*, int size)

This function takes a pointer as argument and prints its contents. The second argument is the size of 1D array pointed by pointer.

d. int main()

In main function you have to do the following tasks:

1. Ask the user to enter the number of rows.
2. Declare a 2D array.
3. Allocate memory for its columns in this array using AddColumns Function defined above.
4. Take data input in the 2D array using RowWiseInput function.
5. Print the 2D array using RowWisePrint function.

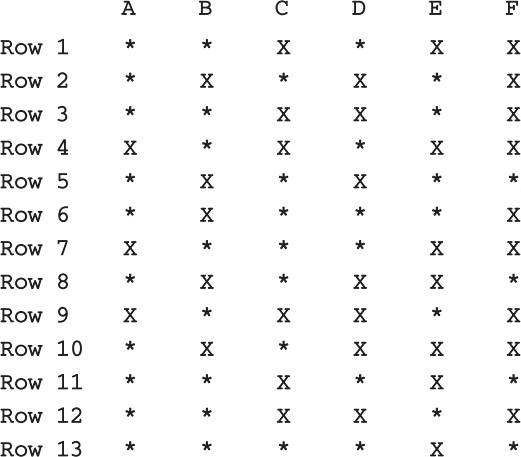


#### Exercise 3: (10 points)

(Airplane Seating Assignment) Write a program that can be used to assign seats for a commercial airplane. The airplane has 13 rows, with six seats in each row. Rows 1 and 2 are first class, rows 3 through 7 are business class, and rows 8 through 13 are economy class. Your program must prompt the user to enter the following information:

* Ticket type (first class, business class, or economy class)
* Desired seat

Output the seating plan in the form given in Figure 3.3:



##### Figure 3.3

Here, \* indicates that the seat is available; X indicates that the seat is occupied. Make this a menu-driven program; show the user’s choices and allow the user to make the appropriate choices. In the space below please give the code for your implementation and a screenshot of the output.

