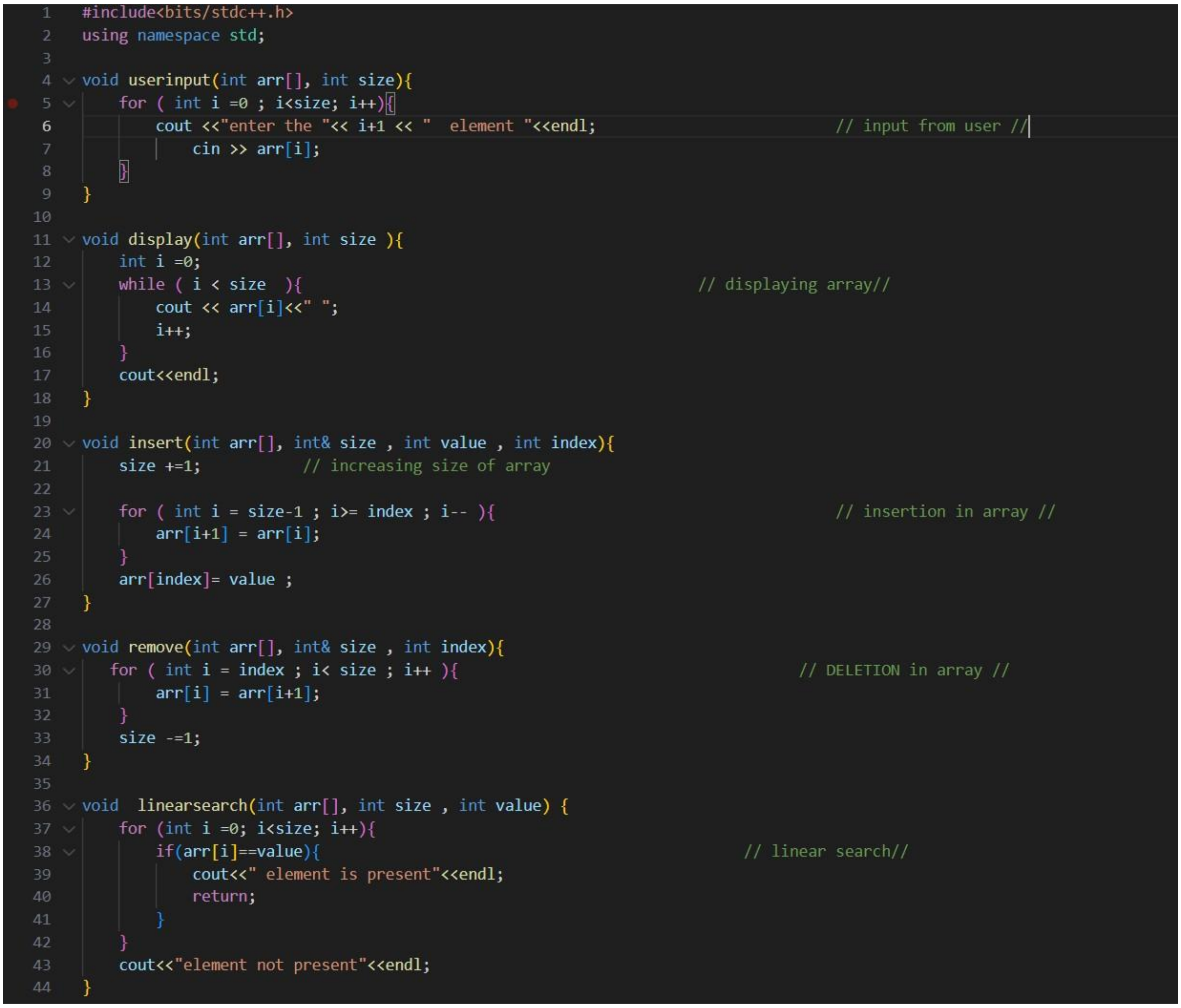
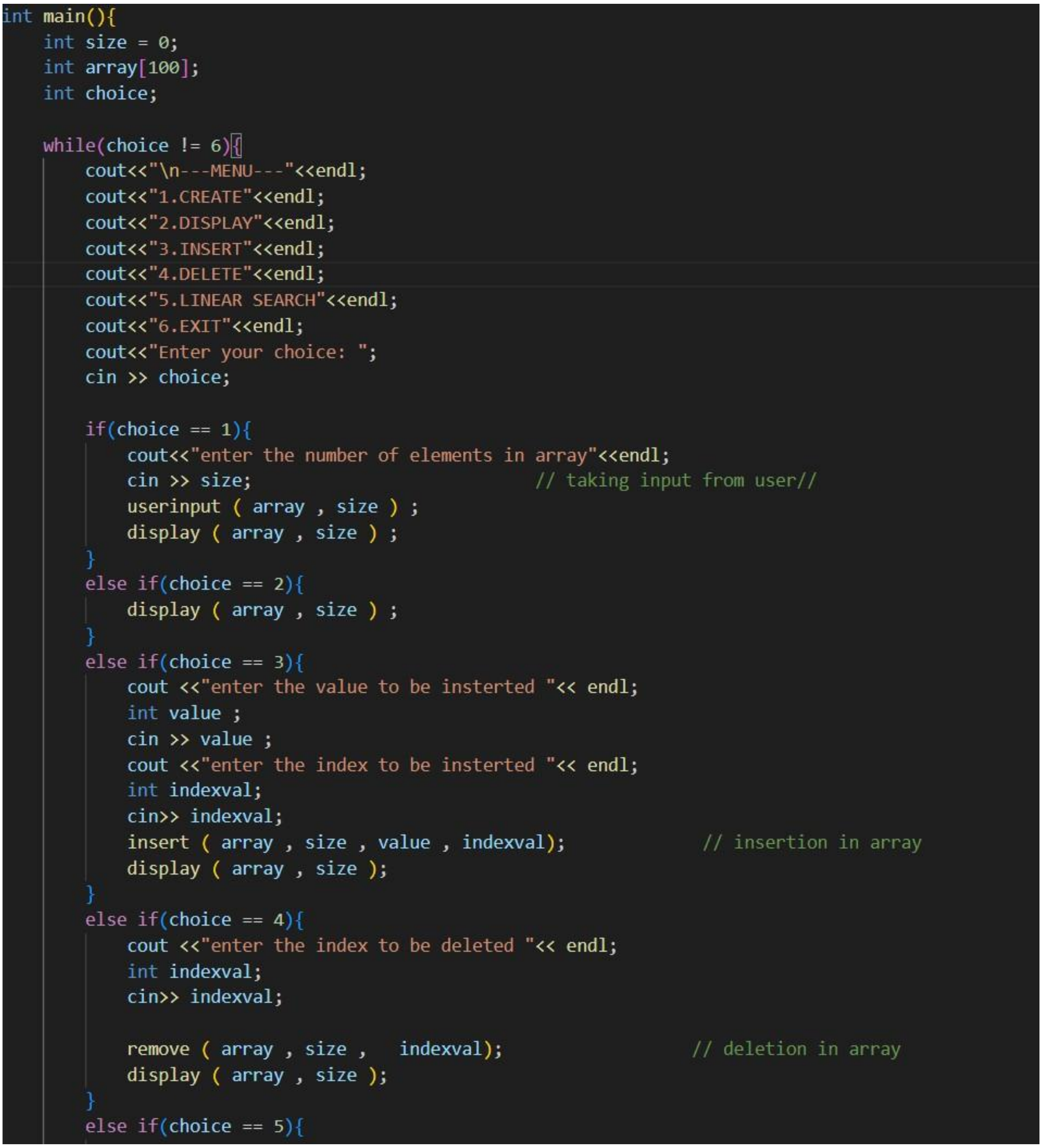
ASSIGNMENT 1:

NAME: Ayush garg
ROLL NUMBER: 1024030878

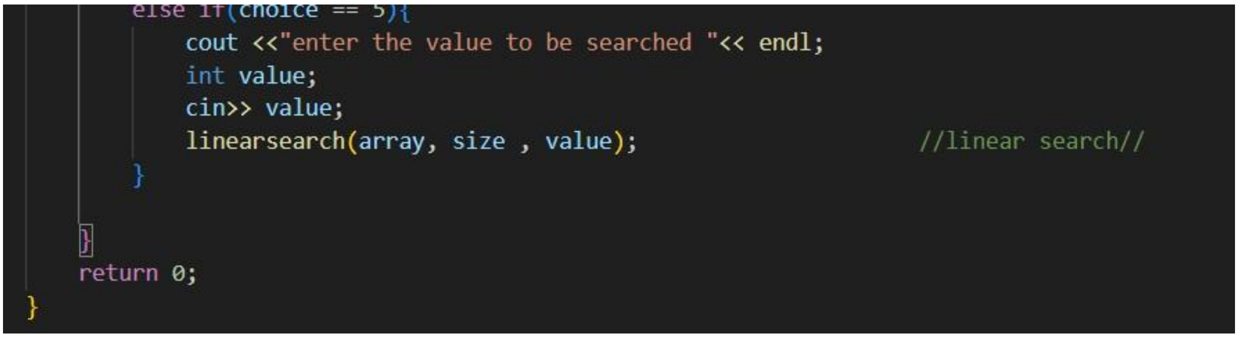
QUE1: Develop a Menu driven program to demonstrate the following operations of Arrays
——MENU——- 1.CREATE
2.DISPLAY
3.INSERT
4.DELETE
5.LINEAR SEARCH
6.EXIT
CODE:



```cpp
#include<bits/stdc++.h>
using namespace std;

void userinput(int arr[], int size){
    for ( int i =0 ; i<size; i++){
        cout <<"enter the "<< i+1 << "  element "<<endl;                           // input from user //
            cin >> arr[i];
    }
}

void display(int arr[], int size ){
    int i =0;
    while ( i < size  ){                                           // displaying array//
        cout << arr[i]<<" ";
        i++;
    }
    cout<<endl;
}

void insert(int arr[], int& size , int value , int index){
    size +=1;           // increasing size of array

    for ( int i = size-1 ; i>= index ; i-- ){                                     // insertion in array //
        arr[i+1] = arr[i];
    }
    arr[index]= value ;
}

void remove(int arr[], int& size , int index){
   for ( int i = index ; i< size ; i++ ){                                     // DELETION in array //
        arr[i] = arr[i+1];
    }
    size -=1;
}

void  linearsearch(int arr[], int size , int value) {
    for (int i =0; i<size; i++){
        if(arr[i]==value){                                              // linear search//
            cout<<" element is present"<<endl;
            return;
        }
    }
    cout<<"element not present"<<endl;
}
```



```cpp
int main(){
    int size = 0;
    int array[100];
    int choice;

    while(choice != 6){
        cout<<"\n---MENU---"<<endl;
        cout<<"1.CREATE"<<endl;
        cout<<"2.DISPLAY"<<endl;
        cout<<"3.INSERT"<<endl;
        cout<<"4.DELETE"<<endl;
        cout<<"5.LINEAR SEARCH"<<endl;
        cout<<"6.EXIT"<<endl;
        cout<<"Enter your choice: ";
        cin >> choice;

        if(choice == 1){
            cout<<"enter the number of elements in array"<<endl;
            cin >> size;                                 // taking input from user//
            userinput ( array , size ) ;
            display ( array , size ) ;
        }
        else if(choice == 2){
            display ( array , size ) ;
        }
        else if(choice == 3){
            cout <<"enter the value to be insterted "<< endl;
            int value ;
            cin >> value ;
            cout <<"enter the index to be insterted "<< endl;
            int indexval;
            cin>> indexval;
            insert ( array , size , value , indexval);              // insertion in array
            display ( array , size );
        }
        else if(choice == 4){
            cout <<"enter the index to be deleted "<< endl;
            int indexval;
            cin>> indexval;

            remove ( array , size ,   indexval);                  // deletion in array
            display ( array , size );
        }
        else if(choice == 5){
```

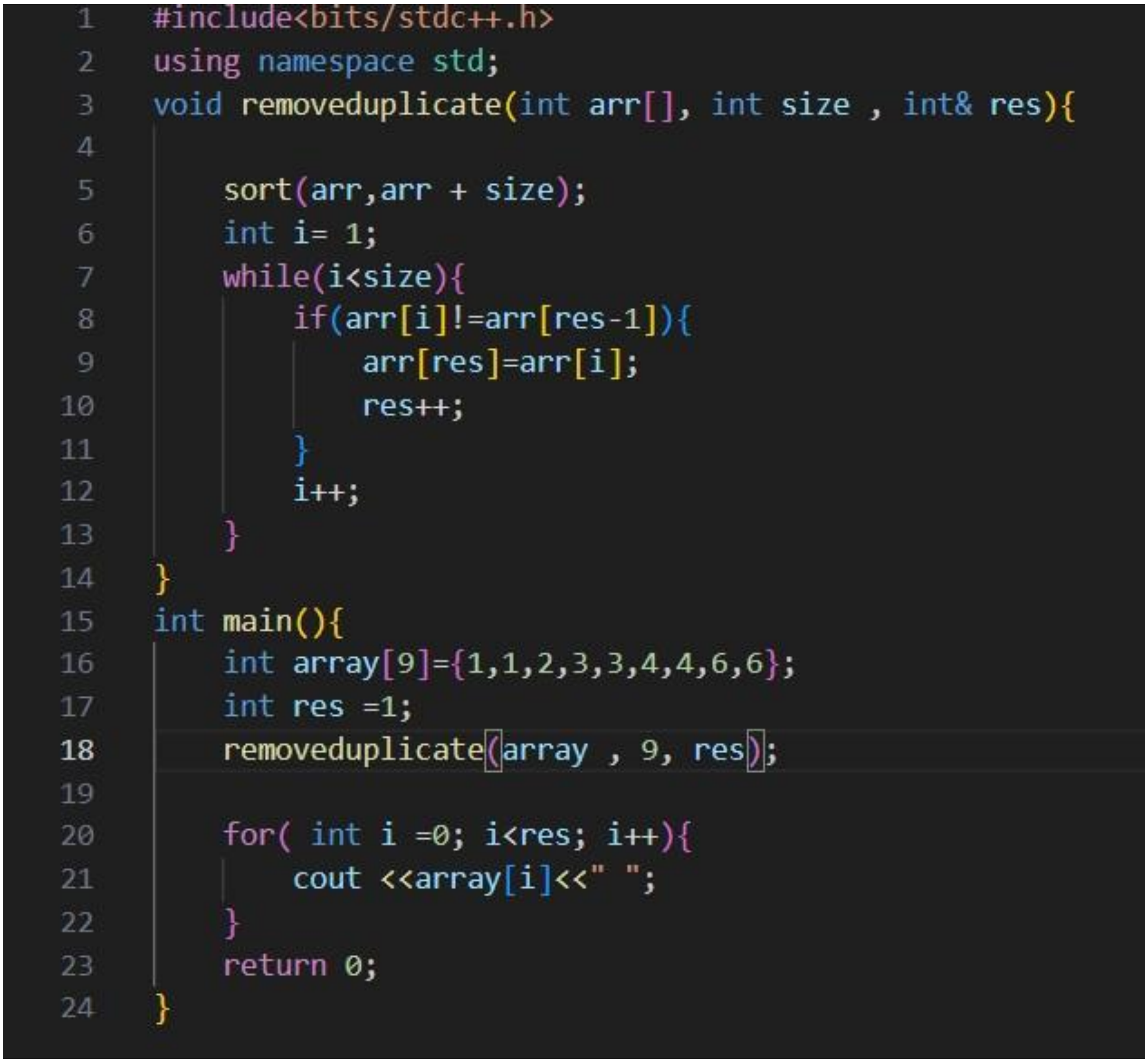


```cpp
        else if(choice == 5){
            cout <<"enter the value to be searched "<< endl;
            int value;
            cin>> value;
            linearsearch(array, size , value);                       //linear search//
        }

    }
    return 0;
}
```

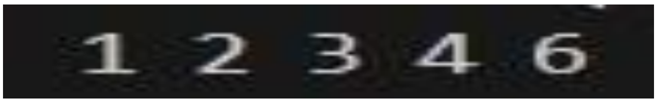
------------------X----------------------X----------------------X------------------------

QUE2: Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements.

CODE:



```
#include<bits/stdc++.h>
using namespace std;
void removeduplicate(int arr[], int size , int& res){

    sort(arr,arr + size);
    int i= 1;
    while(i<size){
        if(arr[i]!=arr[res-1]){
            arr[res]=arr[i];
            res++;
        }
        i++;
    }
}
int main(){
    int array[9]={1,1,2,3,3,4,4,6,6};
    int res =1;
    removeduplicate(array , 9, res);

    for( int i =0; i<res; i++){
        cout <<array[i]<<" ";
    }
    return 0;
}
```



```
1 2 3 4 6
```

------------------X----------------------X----------------------X------------------------

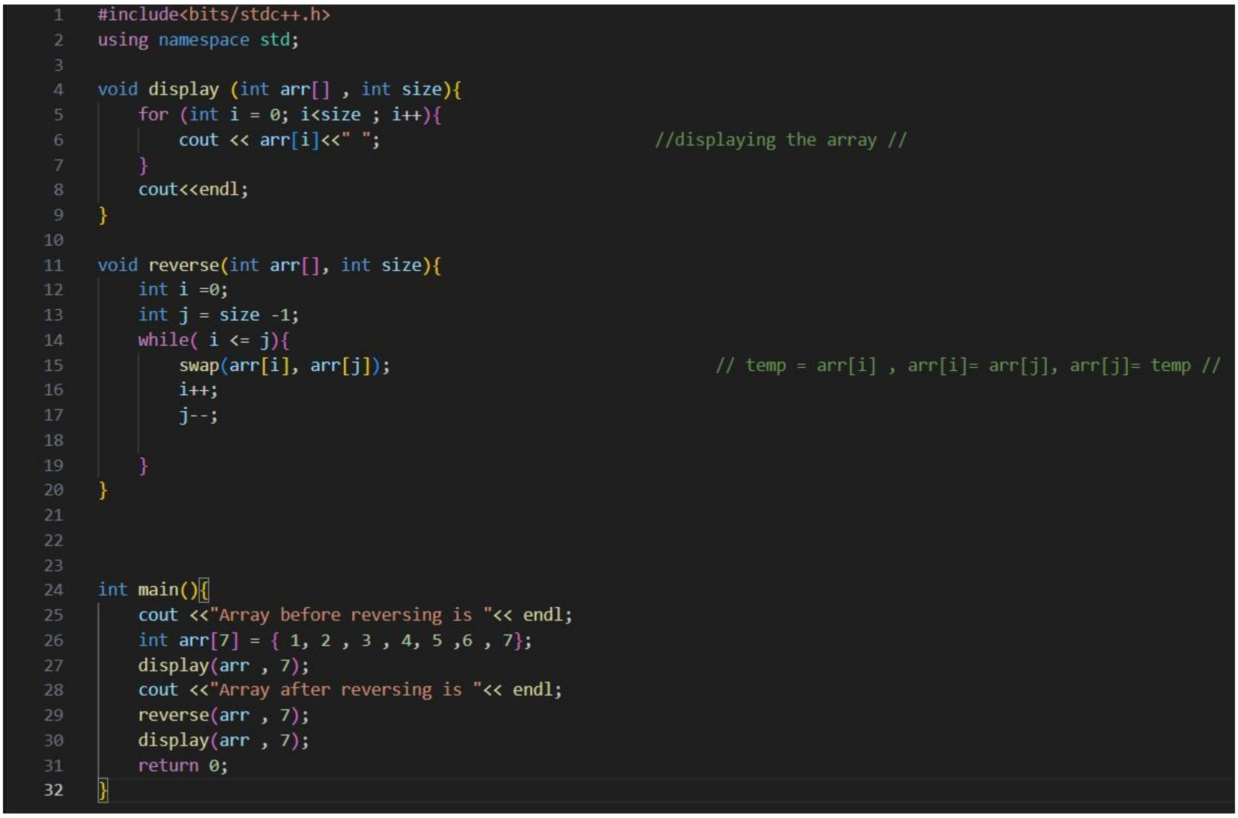
QUE3: Predict the Output of the following program int main() { int i; int arr[5] = {1}; for (i = 0; i < 5; i++) printf("%d",arr[i]); return 0; }
SOL: {1,0,0,0,0}

------------------X----------------------X----------------------X------------------------

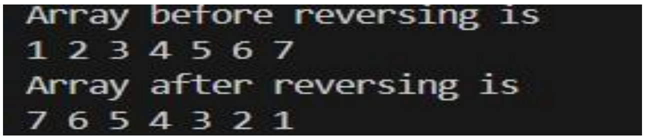
QUE4: Implement the logic to

a. Reverse the elements of an array
b. Find the matrix multiplication
c. Find the Transpose of a Matrix

CODE A:

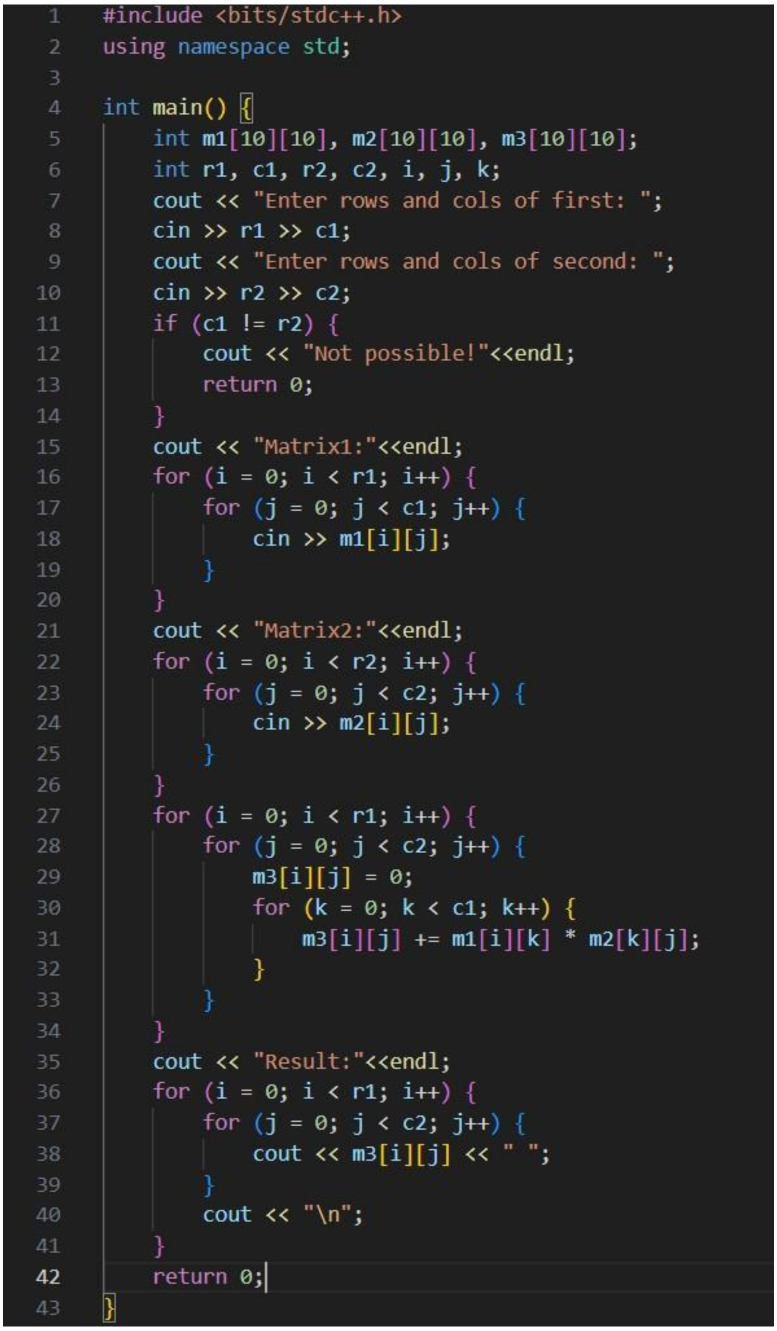


```
#include<bits/stdc++.h>
using namespace std;

void display (int arr[] , int size){
    for (int i = 0; i<size ; i++){
        cout << arr[i]<<" ";                                    //displaying the array //
    }
    cout<<endl;
}

void reverse(int arr[], int size){
    int i =0;
    int j = size -1;
    while( i <= j){
        swap(arr[i], arr[j]);                                   // temp = arr[i] , arr[i]= arr[j], arr[j]= temp //
        i++;
        j--;

    }
}



int main(){
    cout <<"Array before reversing is "<< endl;
    int arr[7] = { 1, 2 , 3 , 4, 5 ,6 , 7};
    display(arr , 7);
    cout <<"Array after reversing is "<< endl;
    reverse(arr , 7);
    display(arr , 7);
    return 0;
}
```

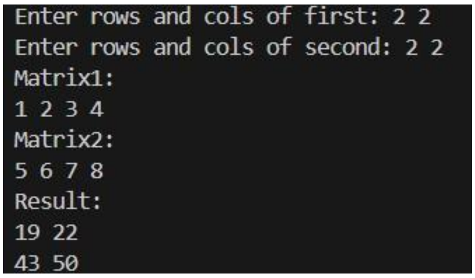


```
Array before reversing is
1 2 3 4 5 6 7
Array after reversing is
7 6 5 4 3 2 1
```

CODE B:

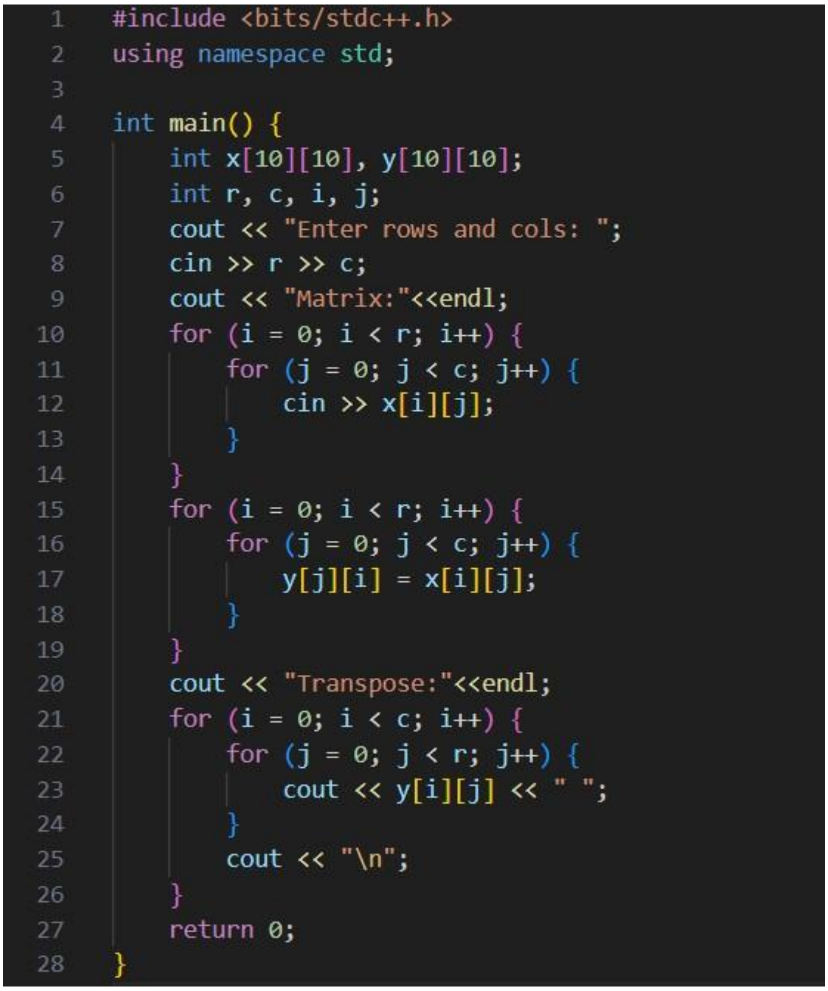


```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    int m1[10][10], m2[10][10], m3[10][10];
    int r1, c1, r2, c2, i, j, k;
    cout << "Enter rows and cols of first: ";
    cin >> r1 >> c1;
    cout << "Enter rows and cols of second: ";
    cin >> r2 >> c2;
    if (c1 != r2) {
        cout << "Not possible!"<<endl;
        return 0;
    }
    cout << "Matrix1:"<<endl;
    for (i = 0; i < r1; i++) {
        for (j = 0; j < c1; j++) {
            cin >> m1[i][j];
        }
    }
    cout << "Matrix2:"<<endl;
    for (i = 0; i < r2; i++) {
        for (j = 0; j < c2; j++) {
            cin >> m2[i][j];
        }
    }
    for (i = 0; i < r1; i++) {
        for (j = 0; j < c2; j++) {
            m3[i][j] = 0;
            for (k = 0; k < c1; k++) {
                m3[i][j] += m1[i][k] * m2[k][j];
            }
        }
    }
    cout << "Result:"<<endl;
    for (i = 0; i < r1; i++) {
        for (j = 0; j < c2; j++) {
            cout << m3[i][j] << " ";
        }
        cout << "\n";
    }
    return 0;
}
```

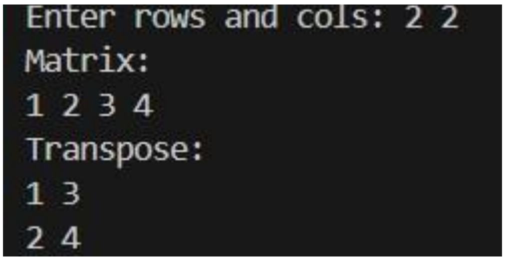


```
Enter rows and cols of first: 2 2
Enter rows and cols of second: 2 2
Matrix1:
1 2 3 4
Matrix2:
5 6 7 8
Result:
19 22
43 50
```

CODE C:



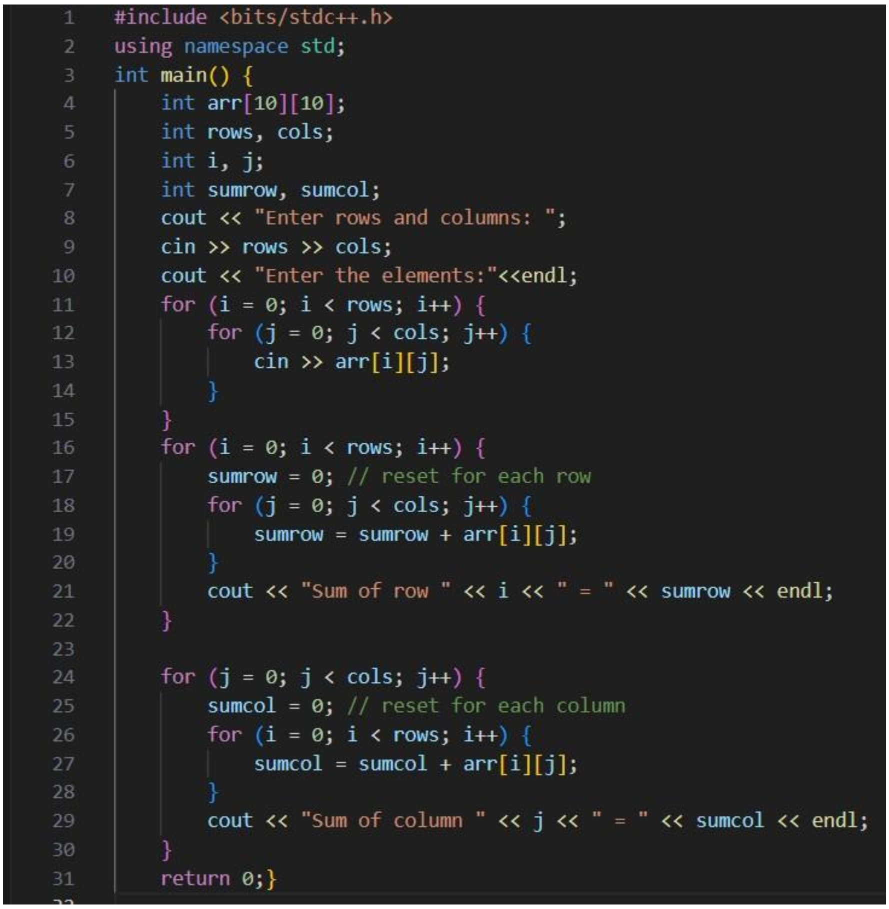
```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    int x[10][10], y[10][10];
    int r, c, i, j;
    cout << "Enter rows and cols: ";
    cin >> r >> c;
    cout << "Matrix:"<<endl;
    for (i = 0; i < r; i++) {
        for (j = 0; j < c; j++) {
            cin >> x[i][j];
        }
    }
    for (i = 0; i < r; i++) {
        for (j = 0; j < c; j++) {
            y[j][i] = x[i][j];
        }
    }
    cout << "Transpose:"<<endl;
    for (i = 0; i < c; i++) {
        for (j = 0; j < r; j++) {
            cout << y[i][j] << " ";
        }
        cout << "\n";
    }
    return 0;
}
```



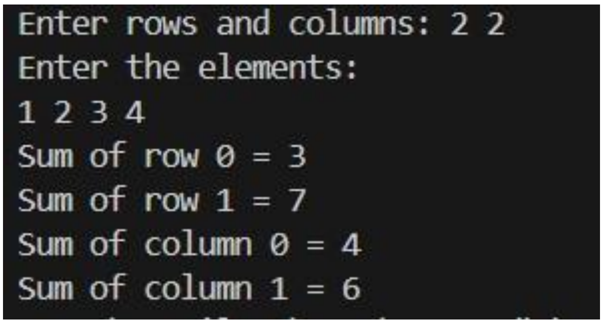
```
Enter rows and cols: 2 2
Matrix:
1 2 3 4
Transpose:
1 3
2 4
```

------------------X----------------------X----------------------X------------------------

QUE5: Write a program to find sum of every row and every column in a two dimensional array.
CODE:



```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int arr[10][10];
    int rows, cols;
    int i, j;
    int sumrow, sumcol;
    cout << "Enter rows and columns: ";
    cin >> rows >> cols;
    cout << "Enter the elements:"<<endl;
    for (i = 0; i < rows; i++) {
        for (j = 0; j < cols; j++) {
            cin >> arr[i][j];
        }
    }
    for (i = 0; i < rows; i++) {
        sumrow = 0; // reset for each row
        for (j = 0; j < cols; j++) {
            sumrow = sumrow + arr[i][j];
        }
        cout << "Sum of row " << i << " = " << sumrow << endl;
    }

    for (j = 0; j < cols; j++) {
        sumcol = 0; // reset for each column
        for (i = 0; i < rows; i++) {
            sumcol = sumcol + arr[i][j];
        }
        cout << "Sum of column " << j << " = " << sumcol << endl;
    }
    return 0;}
```



```
Enter rows and columns: 2 2
Enter the elements:
1 2 3 4
Sum of row 0 = 3
Sum of row 1 = 7
Sum of column 0 = 4
Sum of column 1 = 6
```

-------------------X-----------------------X-----------------------X--------------------------