Question5.cpp

```cpp
//<----Lab 01 - Arrays and Dynamic Memory Allocation---->

// Q5. Write a program that creates a 2D array of 5x5 values of type boolean. Suppose indices
//     represent cities and that the value at row i, column j of a 2D array is true just in case i and
//     j are direct neighbors and false otherwise. Use initializer list to instantiate and initialize
//     your array to represent the following configuration: (* means “neighbors”)

//     Write a method to check whether two cities have a common neighbor. For example, in the
//     example above, 0 and 2, 3 and 4 are neighbors with 1 (so they have a common neighbor),
//     whereas 0 and 1 have no common neighbors.

#include<iostream>
using namespace std;

class array2d{
    const bool city[5][5]={{false,true,false,false,false},{true,false,true,false,false},{false,true,false,false,true},{false,true,true,false,false},{false,true,true,true,false}};
    int row;
    int col;
    bool** arr;
    public:
        array2d(int r=5,int c=5):row(r),col(c){
            arr = new bool*[5];
            for(int i=0;i<5;i++){
                arr[i]=new bool[5];
            }
            for(int i=0;i<5;i++){
                for(int j=0;j<5;j++){
                    arr[i][j]=city[i][j];
                }
            }

        }
        void display(){
            cout<<endl;
            for(int i=0;i<row;i++){
                cout<<"\t";
                for(int j=0;j<col;j++){
                    cout<<arr[i][j]<<"\t";
                }
                cout<<endl;
            }
            cout<<endl;
        }
        void searchneighbours(int city1,int city2){
            for(int i=0;i<5;i++){
                if(arr[city1][i]==arr[city2][i] && arr[city1][i]== true && arr[city2][i]==true){
                cout<<"City "<<city1<<" and city "<<city2<<" share neighbour at index "<<i<<". "<<endl;
                }
            }
```

```
        }
};

int main(){
    int c1,c2;
    array2d cities;
    cities.display();
    cout<<"Enter cities to check"<<endl;
    cin>>c1>>c2;
    cities.searchneighbours(c1,c2);
}

//First Attempt

//#include<iostream>
//using namespace std;
//class Array{
//  int row;
//  int column;
//  bool arr[5][5];
//  public:
//  Array(int r=5,int c=5):row(r),column(c){
//      for(int i=0;i<r;i++){
//          for(int j=0;j<c;j++){
//              arr[i][j]=false;
//          }
//      }
//      arr[0][1]=true;
//      arr[1][0]=true;
//      arr[1][2]=true;
//      arr[2][1]=true;
//      arr[2][4]=true;
//      arr[3][1]=true;
//      arr[3][2]=true;
//      arr[4][1]=true;
//      arr[4][2]=true;
//      arr[4][3]=true;
//  }
//  void display(){
//      for(int i=0;i<5;i++){
//          for(int j=0;j<5;j++){
//              cout<<arr[i][j]<<" ";
//          }
//          cout<<endl;
//      }
//  }
//  void search(int city1,int city2){
//      for(int i=0;i<5;i++){
//          if(arr[city1][i]==arr[city2][i] && arr[city2][i]==true && arr[city1][i]==true ){
//              cout<<"Common Neighbour of City1 and City2 are index "<<i<<endl;
//          }
//      }
//  }
//};
//
//int main(){
```

```
//  Array cities;
//  cities.search(0,2);
//  cities.search(1,4);
//  cities.display();
//}
```