arrayImplement.h

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

//functions for 1D array
void print1DArray(int *array, int size){

    cout << "Printing the 1D Array          : ";
    for(int i=0; i<size; i++){
        cout << array[i] << " ";
    }
    cout << "\n";
}

void reversePrint1DArray(int *array, int size){

    cout << "Printing the reverse 1D Array  : ";
    for(int i=size-1; i>=0; i--){
        cout << array[i] << " ";
    }
    cout << "\n";
}

void input1DArray(int *array, int size){

    cout << "Enter elements: ";
    for(int i=0; i<size; i++){
        cin >> array[i];
    }
}

void searchIn1DArray(int *array, int size, int searchKey){
    int temp;
    int flag=0;
    for(int j=0; j<size; j++){
        if(array[j]==searchKey){
            flag=1;
            temp=j;
            break;
        }
    }
    if(flag==1){
        cout << "Element was found at index no  : " << temp;
    }
    else{
        cout << "Element was found at index no  : Not found!";
    }
    cout << "\n";
}

void findMax1DArray(int *array, int size){
    int max;
    //for maximum value
    max = array[0];
    for(int i=0; i<size; i++){
        if(array[i]>max){
            max = array[i];
        }
    }
    cout << "Maximum Value in 1D Array is   : " << max;
    cout << "\n";
}

void findMin1DArray(int *array, int size){
    int min;
    //for minimum value
    min = array[0];
    for(int i=0; i<size; i++){
        if(array[i]<min){
            min = array[i];
        }
    }
    cout << "Minimum Value in 1D Array is   : " << min;
    cout << "\n";
}

void copy1DArray(int *sourceArray, int *destinationArray, int size){

    for(int i=0; i<size; i++){
        destinationArray[i]=sourceArray[i];
    }

    cout << "The source 1D Array            : ";
    for(int i=0; i<size; i++){
        cout << sourceArray[i] << " ";
    }
    cout << "\n";
    cout << "The destination 1D Array       : ";
    for(int i=0; i<size; i++){
        cout << destinationArray[i] << " ";
    }
    cout << "\n";
}


//functions for 2D array
void print2DArray(int **array, int size){

    int row=size;
    int col=size;
    cout << "Printing the 2D Array          : \n";
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            cout << array[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "\n";
}

void reversePrint2DArray(int **array, int size){

    int row=size;
    int col=size;
    cout << "Printing the reverse 2D Array  : \n";
    for(int i=row-1; i>=0; i--){
        for(int j=col-1; j>=0; j--){
            cout << array[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "\n";
}

void input2DArray(int **array, int size){

    int row=size;
    int col=size;
    cout << "Enter elements: \n";
    for(int i=0; i<row; i++){
        array[i] = new int[col];
    }
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            cin >> array[i][j];
        }
    }
}

void searchIn2DArray(int **array, int size, int searchKey){

    int temp1, temp2;
    int flag=0;

    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){
            if(array[i][j]==searchKey){
                flag=1;
                temp1=i;
                temp2=j;
                break;
            }
        }
    }
    if(flag==1){
        cout << "Element was found at index no  : Array[" << temp1 << "][" << temp2 << "]";
    }
    else{
        cout << "Element was found at index no  : Not found!";
    }
    cout << "\n";
}

void findMax2DArray(int **array, int size){

    int row=size;
    int col=size;

    int max;
    //for maximum value
    max = array[0][0];
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            if(array[i][j]>max){
                max = array[i][j];
            }
        }
    }
    cout << "Maximum Value in 2D Array is   : " << max;
    cout << "\n";
}

void findMin2DArray(int **array, int size){

    int row=size;
    int col=size;

    int min;
    //for minimum value
    min = array[0][0];
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            if(array[i][j]<min){
                min = array[i][j];
            }
        }
    }
    cout << "Minimum Value in 2D Array is   : " << min;
    cout << "\n";
}



void copy2DArray(int **sourceArray, int **destinationArray, int size){

    int row=size;
    int col=size;
    /*
        //using the C++ built in library function

        copy(sourceArray, sourceArray+size, destinationArray);
    */

    for(int i=0; i<size; i++){
        for(int j=0; j<size; j++){                    //manually copying source array to destination array
            destinationArray[i][j]=sourceArray[i][j];        //works when dinamically is created
        }
    }


    cout << "The source 2D Array          :\n";
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            cout << sourceArray[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "The distination 2D Array     :\n";
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            cout << destinationArray[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "\n";
}
```

test.cpp

```
#include<iostream>
#include"arrayImplement.h"

using namespace std;

int main()
{

    //for 1D Array
    cout << "\n";
    int size1D;

    cout << "Enter the 1D Array size: ";
    cin >> size1D;

    int sourceArray1D[size1D];

    int copysize1D=50;
    int destinationArray1D[copysize1D];

    input1DArray(sourceArray1D, size1D);

    int searchKey1D;
    cout << "Enter a key to search in 1D Array: ";
    cin >> searchKey1D;

    cout << "-----------------------------------------------\n";
    cout << "-------------------Output-----------------------\n";
    print1DArray(sourceArray1D, size1D);
    reversePrint1DArray(sourceArray1D, size1D);
    searchIn1DArray(sourceArray1D, size1D, searchKey1D);
    findMax1DArray(sourceArray1D, size1D);
    findMin1DArray(sourceArray1D, size1D);
    copy1DArray(sourceArray1D, destinationArray1D, size1D);


    cout << "\n";
    cout << "---------------------------------------------------------------------------------------\n";

    //for 2D Array
    int size2D;

    cout << "Enter the 2D Array size n (where n x n): ";
    cin >> size2D;

    int **sourceArray2D = new int *[size2D];

    int **destinationArray2D = new int *[size2D];

    /*
        int copysize2D=50;
        int **destinationArray2D = new int *[copysize2D];

        The above declaration was for built in copy function
    */


    for(int i=0; i<size2D; i++){                        //for manually copying
        destinationArray2D[i]=new int[size2D];          //this is for dinamically memory allocation for col
    }



    input2DArray(sourceArray2D, size2D);

    int searchKey2D;
    cout << "Enter a key to search in 2D Array: ";
    cin >> searchKey2D;

    cout << "------------------------------------------------\n";
    cout << "------------------Output-----------------------\n";
    print2DArray(sourceArray2D, size2D);
    reversePrint2DArray(sourceArray2D, size2D);
    findMax2DArray(sourceArray2D, size2D);
    findMin2DArray(sourceArray2D, size2D);
    searchIn2DArray(sourceArray2D, size2D, searchKey2D);
    copy2DArray(sourceArray2D, destinationArray2D, size2D);

    return 0;
}
```