Data Structure Assignment 1
1024160095 - Mitali Sadhav

1.
```
#include <iostream>
using namespace std;

int main() {

    int array[5];
    bool isCreated = false;
    int choice = 0;
    int size=5;

    while(choice!=6){
        cout<<"---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<<"Choose an option: ";
        cin>>choice;
        if(choice==1){
            for(int i=0;i<sizeof(array)/sizeof(array[0]);i++){
                array[i] = i+1;
            }
            cout<<"Array of size 5 created. \n";
            isCreated = true;
        }else if(choice==2){
            if(isCreated){
                cout<<"array is: \n";
                cout<<array[0]<<"\n";
                cout<<array[1]<<"\n";
                cout<<array[2]<<"\n";
                cout<<array[3]<<"\n";
                cout<<array[4]<<"\n";
            }else{
                cout<<"Array not created \n";
            }
        }else if(choice==3){
            int position,value;
            cout<<"Enter position where you wanna add: ";
            cin>>position;
            cout<<"Enter what you wanna add: ";
            cin>>value;
            for(int i=size;i>=position-1;i--){
                array[i]=array[i-1];
            }
            array[position-1]=value;
            for(int i=0;i<=sizeof(array)/sizeof(array[0]);i++){
                cout<<array[i]<<"\n";
            }
        }else if(choice ==4){
            if(isCreated==true){
                int position;
                cout<<"Enter position where you wanna delete: ";
            cin>>position;
```

```
            for(int i=position-1;i<=size;i++){
                array[i]=array[i+1];
            }
            size--;
            for(int i=0;i<=size-1;i++){
                cout<<array[i]<<"\n";
            }
            }
        }else if(choice==5){
            if(isCreated==true){
                int value;
                cout<<"Enter element you want to find: ";
                cin>>value;
                for(int i=0;i<size-1;i++){
                    if(array[i]==value){
                        cout<<"Your element was found at: "<<i;
                        break;
                    }
                }
            cout<<"\n";
            }else{
                cout<<"Array not created";
            }
        }else if(choice==6){
            cout<<"Exiting program \n";
        }else{
            cout<<"Invalid argument";
        }

    }

    return 0;
}
```

2.

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

int main() {
    int size;
    cout << "Enter size of array: ";
    cin >> size;

    int arr[100];
    cout << "Enter elements: ";
    for(int i = 0; i < size; i++) {
        cin >> arr[i];
    }

    for(int i = 0; i < size; i++) {
        for(int j = i + 1; j < size; j++) {
            if(arr[i] == arr[j]) {
                for(int k = j; k < size - 1; k++) {
                    arr[k] = arr[k + 1];
                }
                size--;
                j--;
            }
        }
    }

    cout << "Array after removing duplicates: ";
```

```
    for(int i = 0; i < size; i++) {
        cout << arr[i] << "\n";
    }
    return 0;
}
```
3. 10000
4. a.
```
#include <iostream>
using namespace std;

int main() {
    int size;
    cout << "Enter size of array: ";
    cin >> size;

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

    for (int i = 0; i < size / 2; i++) {
        int temp = arr[i];
        arr[i] = arr[size - 1 - i];
        arr[size - 1 - i] = temp;
    }

    cout << "Reversed array: ";
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
}
```
b.
```
#include <iostream>
using namespace std;

int main() {
    int r1, c1, r2, c2;
    cout << "Enter rows and cols for first matrix: ";
    cin >> r1 >> c1;
    cout << "Enter rows and cols for second matrix: ";
    cin >> r2 >> c2;

    if (c1 != r2) {
        cout << "Matrix multiplication not possible!";
        return 0;
    }

    int A[r1][c1], B[r2][c2], C[r1][c2] = {0};

    cout << "Enter elements of first matrix:\n";
    for (int i = 0; i < r1; i++)
        for (int j = 0; j < c1; j++)
            cin >> A[i][j];

    cout << "Enter elements of second matrix:\n";
    for (int i = 0; i < r2; i++)
        for (int j = 0; j < c2; j++)
            cin >> B[i][j];

    for (int i = 0; i < r1; i++)
        for (int j = 0; j < c2; j++)
```

```
            for (int k = 0; k < c1; k++)
                C[i][j] += A[i][k] * B[k][j];

    cout << "Result matrix:\n";
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++)
            cout << C[i][j] << " ";
        cout << "\n";
    }
}
```

c.

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

int main() {
    int rows, cols;
    cout << "Enter rows and cols: ";
    cin >> rows >> cols;

    int A[rows][cols], T[cols][rows];

    cout << "Enter elements:\n";
    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            cin >> A[i][j];

    for (int i = 0; i < rows; i++)
        for (int j = 0; j < cols; j++)
            T[j][i] = A[i][j];

    cout << "Transpose:\n";
    for (int i = 0; i < cols; i++) {
        for (int j = 0; j < rows; j++)
            cout << T[i][j] << " ";
        cout << "\n";
    }
}
```

5.

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

int main() {
    int arr[3][3] = {{5, 7, 2}, {1, 8, 6}, {2, 4, 7}};
    int rowSum[3] = {0};
    int colSum[3] = {0};

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            rowSum[i] += arr[i][j];
        }
    }

    for (int j = 0; j < 3; j++) {
        for (int i = 0; i < 3; i++) {
            colSum[j] += arr[i][j];
        }
    }

    cout << "Row Sums: ";
    for (int i = 0; i < 3; i++) {
        cout << rowSum[i] << " ";
```

```
    }
    cout << endl;

    cout << "Column Sums: ";
    for (int j = 0; j < 3; j++) {
        cout << colSum[j] << " ";
    }
    cout << endl;

    return 0;
}
```