Turki Hany Alqunaibot

391110069

CS214

Assignment number 2

This program will ask the user to enter elements to be stored in the array and then do one of the operations

1-delete an Element from the array

2-bubble sort the array 3-search the array using linear search

4-search the array using binary search (array most be sorted first)

5-exit the program

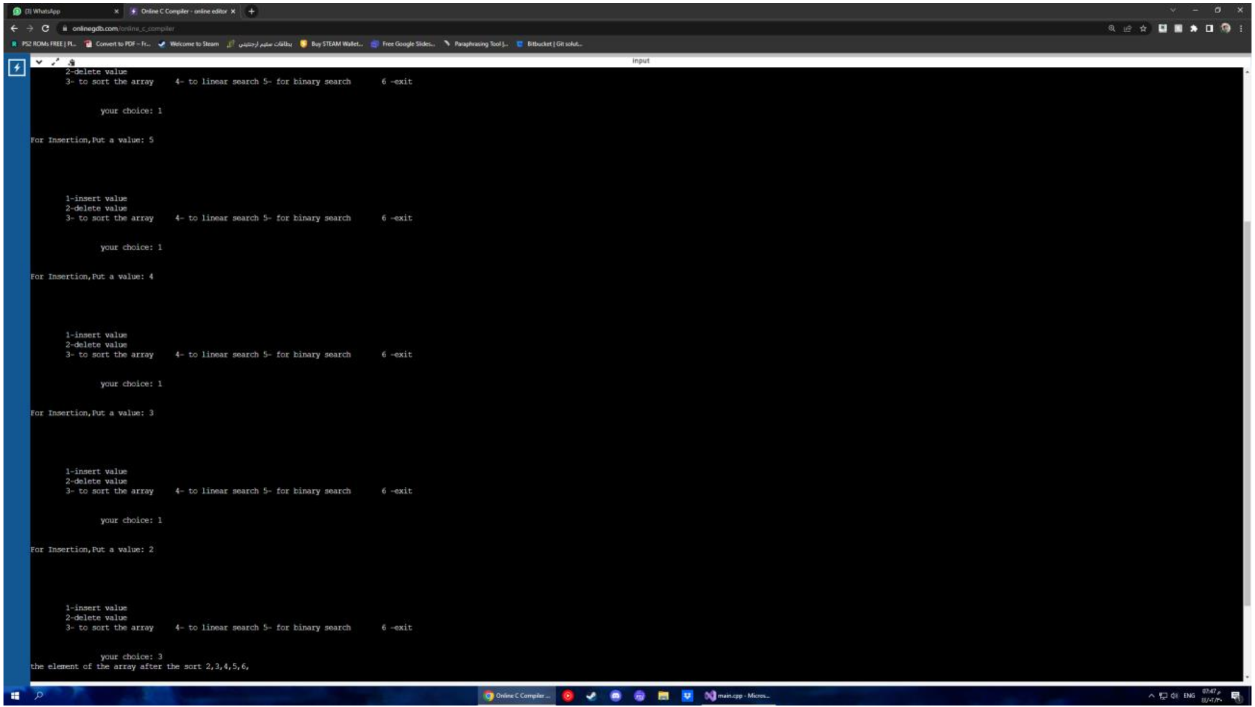
Code:

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

int const SIZE = 10; //size of liner array
int A[SIZE], N = 0; //A[SIZE] declared array of size 10 and N=0 number of element

//this method return true when N is 0 else return false
bool IsFull()
{
    if (N == SIZE) //SIZE(10) == N(number of element)
        return true; //return true
    else //return false
        return false;
}

bool IsEmpty() //when number of element N is 0 return true else return false
{
    if (N == 0)
        return true;
    else
        return false;
}
//we pass a ITEM(array element) to insert it into Linear array in ascending order
// so it return the index i according to sorted order
int Add_location(int ITEM)
{
    //run a loop from 0 to N(excluded)
    for (int i = 0; i < N; i++)
    {
        if (A[i] > ITEM) //when Array element greater than argument ITEM return
index to insert
```

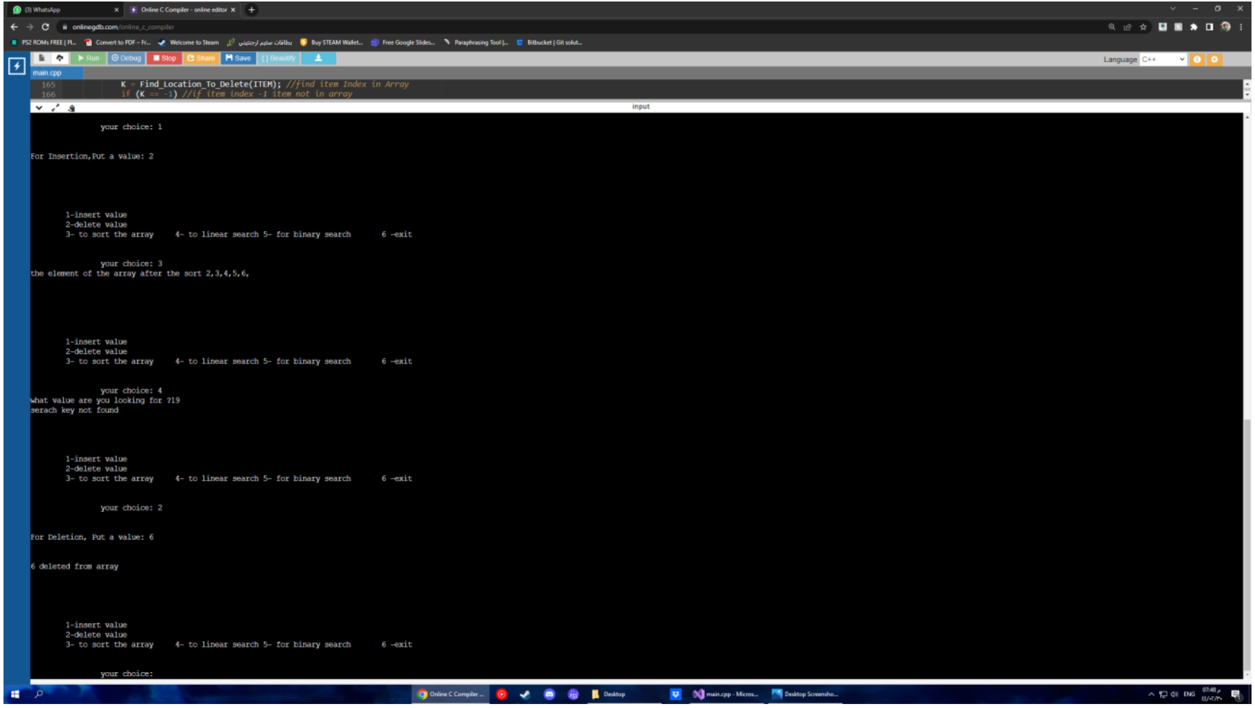
```
            return i;

    }
    return N; //if ITEM maximum in ascending order return N index
}
//now Insert method
//argument  K index to insert argument ITEM to insert at K location of Linear
Array A
void Add(int K, int ITEM)
{
    //run a loop from N to K(excluded)
    for (int i = N; i > K; i--)
    {
        A[i] = A[i - 1]; //shift each element right by 1 and make a space for ITEM
at K index
    }
    A[K] = ITEM; //assign ITEM at K index
    N++; //increment N by 1
}
//print each element of Linear Array separated by space

void sorting(int A[],int N){

      bool swap;
      int pass;
      int K;
      int hold;
      for (pass = 1; pass <= N - 1; pass++) {
             swap = false;

             for (K = 0; K < N - pass; K++) {
                   if (A[K] > A[K + 1]) {
                         hold = A[K];
                         A[K] = A[K + 1];
                         A[K + 1] = hold;
                         swap = true;

                   }


             }

             if (swap == false) {
                   break;

             }
      }
}

int LinearSearch (int A[], int skey)
{ int K;
    for(K=0; K<= N-1; K++)
    if(A[ K ] == skey)
    return K;

else
return -1;
}

int BinarySearch (int A[], int skey) // function definition
```

```
{ int START=0, END= N-1, MID=int((START+END)/2), LOC; while(START <= END && A[MID]
!= skey)
{ if(skey < A[MID])
END = MID - 1;
else
START = MID + 1;
MID=int((START+END)/2);
}
if(A[MID] == skey) LOC=MID; else LOC= -1; return LOC;
}



//return the Index of ITEM to delete from Linear Array
int Find_Location_To_Delete(int ITEM)
{
    for (int i = 0; i < N; i++)
    {
        if (A[i] == ITEM) //when ITEM find in Linear Array return its Index
            return i;
    }
    return -1; //if not found than return -1
}
//delete item  from Array
// run a loop from K to N-1
//shift each element it left index by 1
int Delete(int K, int ITEM)
{
    for (int i = K; i < N - 1; i++)
    {
        A[i] = A[i + 1]; //assign to index of previous element
    }
    N--; //decrement N by 1
    return ITEM; //return ITEM
}

int main()
{
    int ch, ITEM, K,SKEY,LOC; //ch to store ,ITEM= array element to insert or
delete K=index of element
    while (1)
    {
        //print menu
        cout << "\n\n\n\n\n";
        cout << "\t1-insert value\n";
        cout << "\t2-delete value\n";
        cout<<"\t3- to sort the array";
        cout<<"\t 4- to linear search ";
        cout<<"5- for binary search";
        cout << "\t6 -exit\n\n";
        cout << "\n\t\tyour choice: ";
        cin >> ch; //get choice of user

        switch (ch)
        {
        case 1: //if user select to insert
            if (IsFull()) //if array already full can not insert
            {

                cout << "\n\nArray is full\n\n";
                break;
            }
            //else get ITEM and insert at index K
            cout << "\n\nFor Insertion,Put a value: ";
            cin >> ITEM;
            K = Add_location(ITEM);
            Add(K, ITEM);
            break;

        case 2:
            if (IsEmpty()) //if user select delete and array already empty can not
delete
            {
                cout << "\n\nArray is Empty\n\n";
                break;
            }
            //else get item
            cout << "\n\nFor Deletion, Put a value: ";
            cin >> ITEM;
            K = Find_Location_To_Delete(ITEM); //find item Index in Array
            if (K == -1) //if item index -1 item not in array
                cout << "\n\n" << ITEM << " Not Found in Array \n";
            else//delete
                cout << "\n\n" << Delete(K, ITEM) << " deleted from array\n";
            break;
     case 3:


      sorting(A,N);
      cout<<"the element of the array after the sort ";
      for (int I = 0; I < N; I++) {
                          cout << A[I] << ",";
                   }
                   cout << "\n\n";



     break;

     case 4:
    cout<<"what value are you looking for ?";
    cin>>SKEY;
    LOC=LinearSearch ( A, SKEY);

    if(LOC==-1)
    cout<<"serach key not found";

   break ;

   case5:
   cout<<"enter the search key";
   cin>>SKEY;
   LOC=BinarySearch(A, SKEY);
   if(LOC == -1) cout<<"\n The search key is not found in the array\n";
else
cout<<"\n The search key "<<SKEY<< " exist at location "<<LOC<<endl;

   break;
        case 6:
            exit(0);
        default:
            cout << "\n\nInvalid choice\n";

        }
    }
return 0;}
```

output:



```
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 1


For Insertion,Put a value: 5





        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 1


For Insertion,Put a value: 4





        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 1


For Insertion,Put a value: 3





        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 1


For Insertion,Put a value: 2





        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 3
the element of the array after the sort 2,3,4,5,6,
```



```
                K = Find_Location_To_Delete(ITEM); //find item Index in Array
                if (K == -1) //if item index -1 item not in array
```

```
                your choice: 1


For Insertion,Put a value: 2





        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 3
the element of the array after the sort 2,3,4,5,6,






        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 4
what value are you looking for ?19
serach key not found





        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice: 2


For Deletion, Put a value: 6


6 deleted from array





        1-insert value
        2-delete value
        3- to sort the array     4- to linear search 5- for binary search       6 -exit


                your choice:
```