LAB 05

NAME: M. Huzaifa Mustafa

SECTION: AM

SUBJECT: Data Structure and algorithms Lab

- **LINEAR SEARCH:**
  Source Code:

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


class linearSearch{
public:
        int arr[];
        int Snum;
        int index;
        int size;
        int i;
        int counter = 1;

                void LS(){
                        cout<<"Enter the size of an array : "<<endl;
                        cin>>size;
                        int arr[size];

                        cout<<"Enter the element : "<<endl;
                   for(i = 0; i < size; i++){
                   cin>>arr[i];
                   counter++;
                }
                   cout<<"\nEnter a Number to Search: ";
                   cin>>Snum;

                   for(i = 0; i < size; i++)
                   {
                     if(arr[i] == Snum)
                     {
                        index = i;
                        break;
```

```cpp
                }
            }
            cout<<"\nFound at Index No."<<index;
            cout<<endl;

            cout<<"The performance of the program is : "<<counter<<endl;
        }

};

int main(){

        linearSearch ls;
        ls.LS();
}
```

Picture:

- **BINARY SEARCH:**
  Source Code:
  ```cpp
  #include <iostream>
  using namespace std;


  class BinarySearch{
          public:
                  int size;
                  int arr[100];
                  int counter = 1;
                  int i;
  ```

```cpp
int Snum;
int index;


void SetAllData(){
        cout<<"Enter the Size of an array : "<<endl;
        cin>>size;
        arr[size];
}

void SortanArray(){
        cout<<"Enter the element of array : "<<endl;
        for(i = 0;i < size; i++){
                cin>>arr[i];
                counter++;
        }

        for(i = 0;i < size; i++){
                for(int j = i+1; j < size; j++){
                        if(arr[i] > arr[j]){
                                int temp = arr[i];
                                arr[i] = arr[j];
                                arr[j] = temp;
                        }
                }
        }
        cout<<"Array After Sorting : "<<endl;
        for( i = 0; i < size; i++){
                cout<<arr[i]<<" ";
                cout<<endl;
        }
}

void binarysearch(){

        cout<<"Enter the Number for Search : "<<endl;
        cin>>Snum;
        for(i = 0; i < size; i++){
                if(arr[i] == Snum){
                        index = i;
                        break;
                }
        }
        int start = 0;
```

```cpp
            int end = size - 1;
            int mid = (start+end)/2;

            while(start <= end){

                    if(Snum == arr[mid]){
                            cout << "Your element is found at: " << mid << endl;
                    break;
                    }

                    else if(Snum > arr[mid] ){
                            start = mid+1;
                            mid = (start+end)/2;
                    }

                    else{
                            end = mid-1;
                            mid = (start+end)/2;
                    }

            }
        }

};



int main(){
        BinarySearch b1;
        b1.SetAllData();
        b1.SortanArray();
        b1.binarysearch();

}
```
Picture:
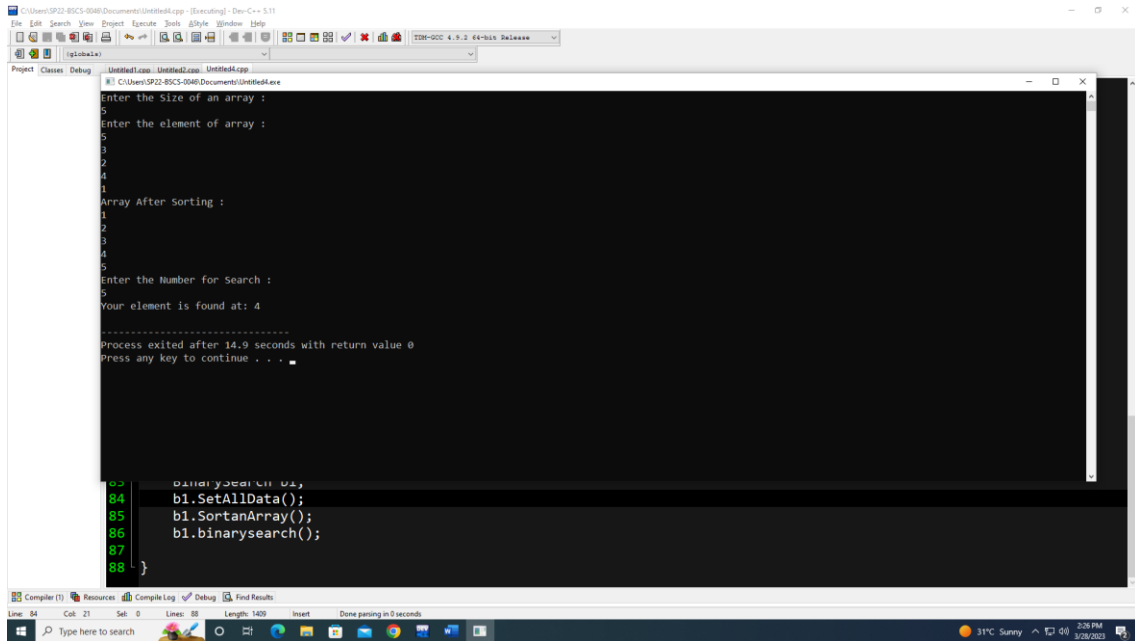
- **EXPONENTIAL SEARCH:**
  Source code:
  #include<iostream>
  using namespace std;

  class ExponentialSearch{
          public:

   int arr[100];
          int size;
          int i;
          int index;
          int Snum;

          void SortAnArray(){
          cout<<"Enter the size of an array : "<<endl;
          cin>>size;

          int arr[size];

          cout<<"Enter the element of an array : "<<endl;
          for(i = 0; i < size; i++){
          cin>>arr[i];
          }
          for(i = 0; i < size; i++){

```cpp
            for(int j = i+1; j < size; j++){
                    if(arr[i] > arr[j]){
                        int temp = arr[i];
                        arr[i] = arr[j];
                        arr[j] = temp;
                    }
            }

}
        cout<<"Sorted Array : "<<endl;
        for(i = 0;i < size; i++){
        cout<<arr[i]<<" ";
        cout<<endl;
        }

}



        void exponentialsearch(){
                int start = 0;
                int end = size-1;
                int mid = (start+end)/2;

                cout<<"Enter the number to search : "<<endl;
                cin>>Snum;

                if((end - start) < 0){
                        cout<<"Element Found at Index -1 : "<<endl;
                }

        int k = 1;

        while(k < (end - start)){
        if(arr[k] < Snum){
                k *= 2;
                index = k;
                cout<<"Element Found at index : "<<index<<endl;
                    }
        else{
                break;
                    }
    }
```

```cpp
				while(start <= end){
						if(Snum == arr[mid]){
								cout<<"Your element is found at mid : "<<mid<<endl;
						}
						else if(Snum > arr[mid]){
								start = mid + 1;
								mid = (start+end)/2;
						}
						else{
								end = mid - 1;
								mid = (start+end)/2;
						}
				}

		}

};



int main(){

		ExponentialSearch e1;

		e1.SortAnArray();
		e1.exponentialsearch();

}
```
Picture:
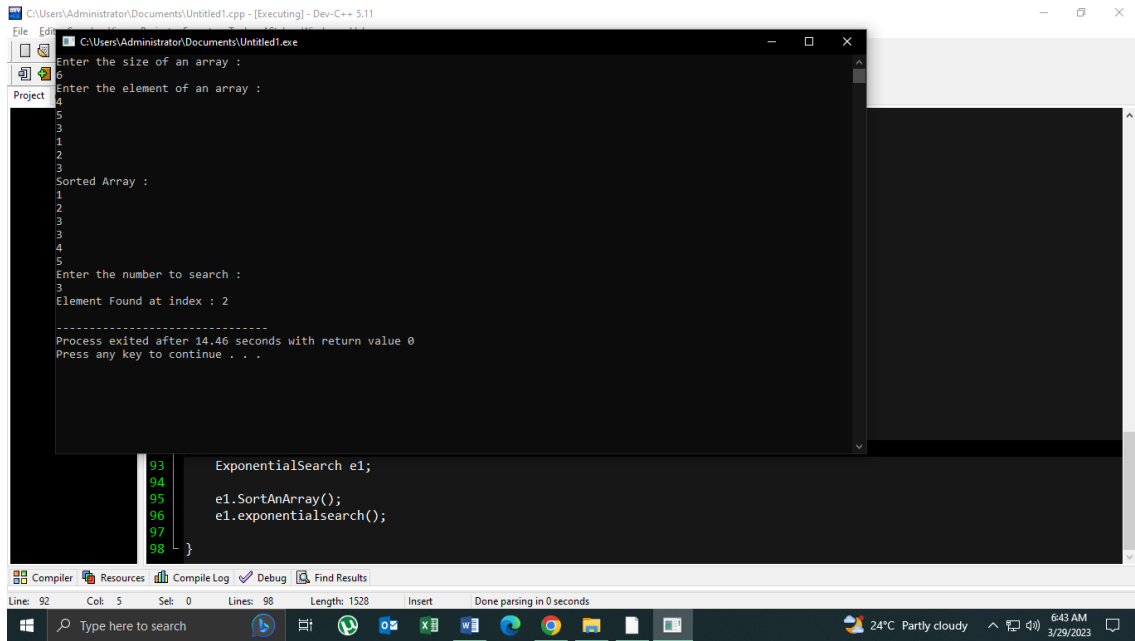
```
C:\Users\Administrator\Documents\Untitled1.exe

Enter the size of an array :
6
Enter the element of an array :
4
5
3
1
2
3
Sorted Array :
1
2
3
3
4
5
Enter the number to search :
3
Element Found at index : 2

--------------------------------
Process exited after 14.46 seconds with return value 0
Press any key to continue . . .
```

```
93        ExponentialSearch e1;
94
95        e1.SortAnArray();
96        e1.exponentialsearch();
97
98    }
```

- **JUMP SEARCH:**

Source code:

```cpp
#include<iostream>
using namespace std;
class JumpSearch{
        public:
    int x,size;
    int arr[100];

    void elementarray(){
                cout << "Enter the number of elements in the array: \n";
        cin >> size;
                cout << "Enter the elements of the array: \n";
       for (int i = 0; i < size; i++)
          cin >> arr[i];
    }

    void bubbleSort(){
      int temp;
      for (int i = 0; i < size; i++) {
         for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
               temp = arr[j];
               arr[j] = arr[j + 1];
               arr[j + 1] = temp;
            }
         }
      }
    }
```

```cpp
}
void search()
{
cout << "Enter the element to be searched: ";
cin >> x;
int step = 0;
while (step * step < size)
       step++;
int prev = 0;
while (arr[min(step, size) - 1] < x) {
   prev = step;
   step += step;
   if (prev >= size) {
      cout << "Element not found";
      return ;
   }
}
while (arr[prev] < x) {
   prev++;
   if (prev == min(step, size)) {
      cout << "Element not found";
   }
}
if (arr[prev] == x) {
   cout << "Element found at index " << prev << endl;

}
}
};
int main()
{
        JumpSearch Js;
        Js.elementarray();
        Js.bubbleSort();
           cout << "Array after sorting: ";
   for (int i = 0; i < Js.size; i++)
      cout << Js.arr[i] << " ";
   cout << endl;
   Js.search();
    return 0;
}
```
Picture:

- **INTER-POLATION SEARCH:**
  Source code:

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

class Interpolate{
public:
    int size, x;
    int arr[100];

    void elementarray(){
        cout << "Enter the number of elements in the array: ";
        cin >> size;

        cout << "Enter the elements of the array: ";
        for (int i = 0; i < size; i++)
            cin >> arr[i];
    }
    void bubbleSort(){
        int temp;
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
```
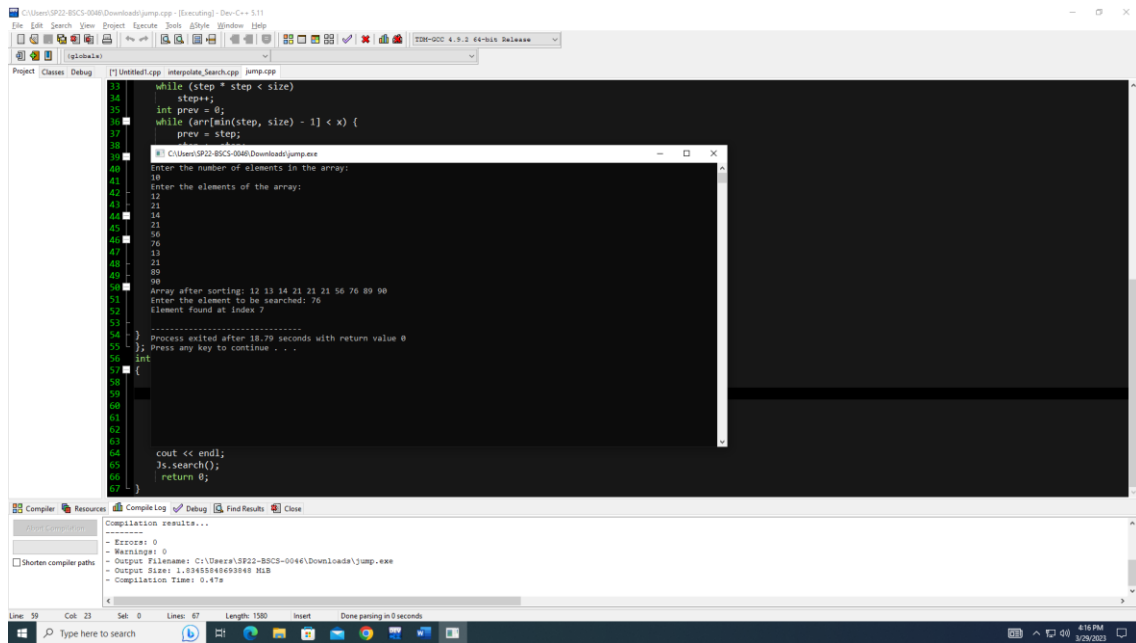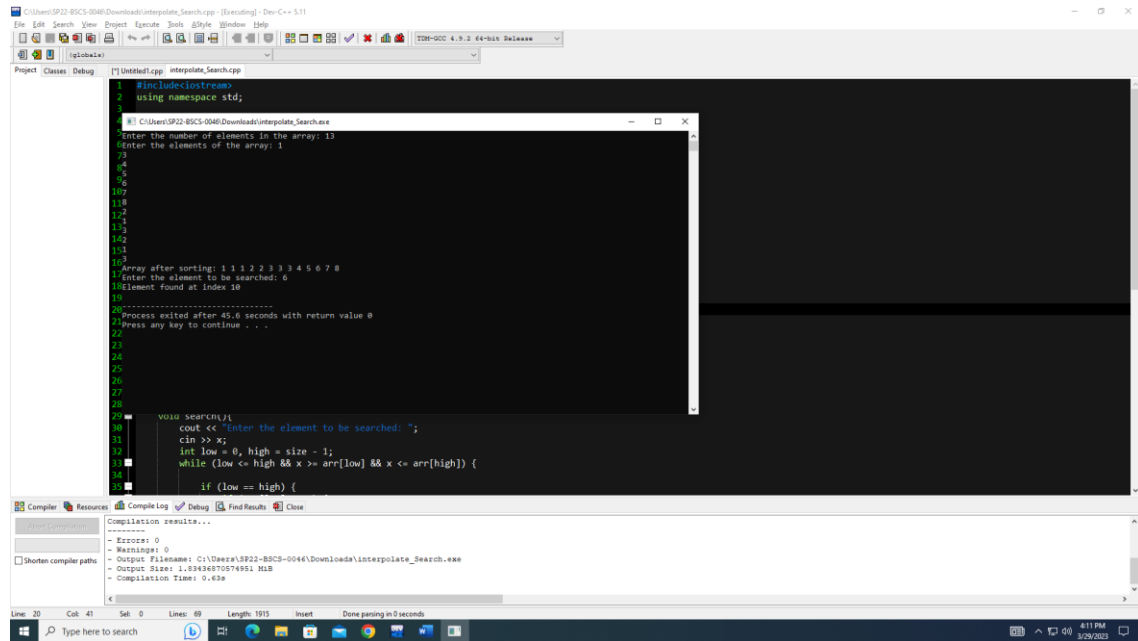
```cpp
        }
      }
    }
    void search(){
      cout << "Enter the element to be searched: ";
      cin >> x;
      int low = 0, high = size - 1;
      while (low <= high && x >= arr[low] && x <= arr[high]) {

        if (low == high) {
          if (arr[low] == x) {
            cout << "Element found at index " << low << endl;
            return;
          }
          else {
            cout << "Element not found" << endl;
            return;
          }
        }
        int pos = low + ((x - arr[low]) * (high - low)) / (arr[high] - arr[low]);
        if (arr[pos] == x) {
          cout << "Element found at index " << pos << endl;
          return;
        }
        if (arr[pos] < x)
          low = pos + 1;
        else
          high = pos - 1;
      }
      cout << "Element not found" << endl;
    }
};

int main(){
  Interpolate ip;
  ip.elementarray();
  ip.bubbleSort();
  cout << "Array after sorting: ";
  for (int i = 0; i < ip.size; i++)
    cout << ip.arr[i] << " ";
  cout << endl;
  ip.search();
  return 0;
}
```

Picture: