**NAME – ABHIRAJ YOGESH SRIVASTAVA**

**ROLL NO. – 1906137**

**SUBJECT NAME – DESIGN AND ANALYSIS OF ALGORITHMS LAB**

**SUBJECT CODE – CSL4403**

**DATE – 11TH JAN, 2021**

**BRANCH – CSE 2**

**ASSIGNMENT-3**

**Q3. WAP to search for an element in the array using Binary Search algoritm.**

**Source Code:**

#include <stdio.h>

int main()

{

int n; // Number of elements

int a[100]; // Array to store elements

int s; // Element to search

int p; // To sotre the position of element where the element s found in the array

int k; // To store the input is in the required form for binary search or not

n = Read\_Number\_of\_Elements(); // Function reads the number of elements

Read\_Elements\_to\_Array(a, n); // Function to read n elements into an array

k = Verify\_Input(a, n); // Function to check the input

if(k==1)

{

s = Read\_Element\_to\_Search(); // Function to read the searching element

p = Binary\_Search(a, 0, n, s); // Function implements the linear search

if(p==-1)

printf("\n Element not found");

else

printf("\n Element found at position: %d",p);

return 0;

}

else printf("wrong input entered");

return 0;

}

int Read\_Number\_of\_Elements() // Read the number of elements in the Array and return it

{

int n;

scanf("%d",&n);

return n;

}

void Read\_Elements\_to\_Array(int \*p, int n) // Read the n elements into array from the keyboard

{

for(int i=0;i<n;i++)

scanf("%d",&p[i]);

}

// Function to check the input is in the required form for binary search or not

// Return 0 if input is not in the proper format

// Return 1 if the input is proper format

int Verify\_Input(int \*p,int n)

{

int ci=0;

for(int i=0;i<(n-1);i++)

{

if(p[i]<=p[i+1])

ci++;

}

if(ci==(n-1))

return 1;

else

{

int cd=0;

for(int i=0;i<(n-1);i++)

{

if(p[i]>=p[i+1])

cd++;

}

if(cd==(n-1))

return 1;

else

return 0;

}

}

int Read\_Element\_to\_Search() // Read the element you want to search from keyboard and return

{

int ele;

scanf("%d",&ele);

return ele;

}

// Write the code for Binary search

// l is the first element position

// h is the last element position-1

int Binary\_Search(int \*p, int l, int h, int s)

{

h--;

if(p[l]<p[h])

{

int ind=-1;

while(l<=h)

{

int mid=(l+h)/2;

if(p[mid]==s)

{

ind=mid+1;

break;

}

else if(p[mid]>s)

h=mid-1;

else

l=mid+1;

}

return ind;

}

else if(p[l]>p[h])

{

int ind=-1;

while(l<=h)

{

int mid=(l+h)/2;

if(p[mid]==s)

{

ind=mid+1;

break;

}

else if(p[mid]<s)

h=mid-1;

else

l=mid+1;

}

return ind;

}

else

{

if(p[l]==s)

return -1;

else

return 1;

}

}

**Output Screenshot:**

