



॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Constituent of Symbiosis International (Deemed University), Pune

Assignment No.: 06	
Course Name	Programming in C Lab
Name of Student	Faheemuddin Sayyed
PRN No.	23070122196
Branch	CSE
Class	C-1
Academic Year & Semester	2023-2024 & Semester 2
Date of Performance	16/02/2024
Assignment Title (Full):	Write a C program of binary search for a user-given element in a set of 'n' numbers.
<p>Theory: (Note: According to the assignment title, please write the background information as an introduction, then write the steps/logic/process/algorithm of the C program in the Journal Notebook, and add its screenshot in the below theory response.)</p>	
<p>Theory Response:</p> <ol style="list-style-type: none">1. Accept the size of the array (size) and allocate memory dynamically.2. Input sorted elements into the array.3. Accept the element to search (x).4. Implement a binary search function (BinarySearch) to find the index of the element in the array.5. Display whether the element is found and its index.	
<p>Output: (Note: Execute the C program as per the assignment title, take an input code and output result screenshot with the date and time from your computer, and add its screenshot in the below output response.)</p>	

Output Response:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int BinarySearch(int arr[], int length, int x){
5      int l=0, h=length-1;
6      while (l<=h){
7          int mid = (l+h)/2;
8          if (arr[mid] == x){
9              return mid;
10         } else if (arr[mid] < x){
11             l = mid+1;
12         } else {
13             h = mid-1;
14         }
15     }
16     return -1;
17 }
18
19 int main(){
20     int size,*arr,x;
21     printf("\nEnter the size of the array: ");
22     scanf("%d", &size);
23     arr = (int *)malloc(size * sizeof(int));
24     printf("\nEnter the elements of the array (sorted): ");
25     for (int i=0; i<size; i++){
26         scanf("%d", &arr[i]);
27     }
28     printf("\nEnter the element to search: ");
29     scanf("%d", &x);
30     int result = BinarySearch(arr, size, x);
31     if (result == -1){
32         printf("Element not found.\n");
33     } else {
34         printf("Element found at index %d.\n", result);
35     }
36     free(arr);
37     return 0;
38 }
```

```
Enter the size of the array: 10
```

```
Enter the elements of the array (sorted): 1 2 3 4 5 6 7 8 9 10
```

```
Enter the element to search: 6
```

```
Element found at index 5.
```

```
(base) fahee@Faheems-MacBook-Pro Programming_in_C % █
```

```
Enter the size of the array: 10
```

```
Enter the elements of the array (sorted): 1 2 3 4 5 6 7 8 9 10
```

```
Enter the element to search: 11
```

```
Element not found.
```

```
(base) fahee@Faheems-MacBook-Pro Programming_in_C % █
```

Conclusion: (Note: Write the key findings or outcome from this assignment, enlist their potential real-world applications in Journal Notebook, and add its screenshot in the below conclusion response.)

Conclusion Response:

The C code demonstrates a binary search on a sorted array. It dynamically allocates memory for the array, takes user input for sorted elements, and performs a binary search for a specified element. The code efficiently handles memory allocation and deallocation while providing a clear and functional binary search implementation. The result informs the user about whether the element is found and, if so, at which index in the array.

Please note that assignment content can be readable.

Faculty Name:

Dr. Kanhaiya Sharma
Prof. Mahesh Arse
Prof. Sachin R. Gaikwad
Prof. Surabhi Thatte