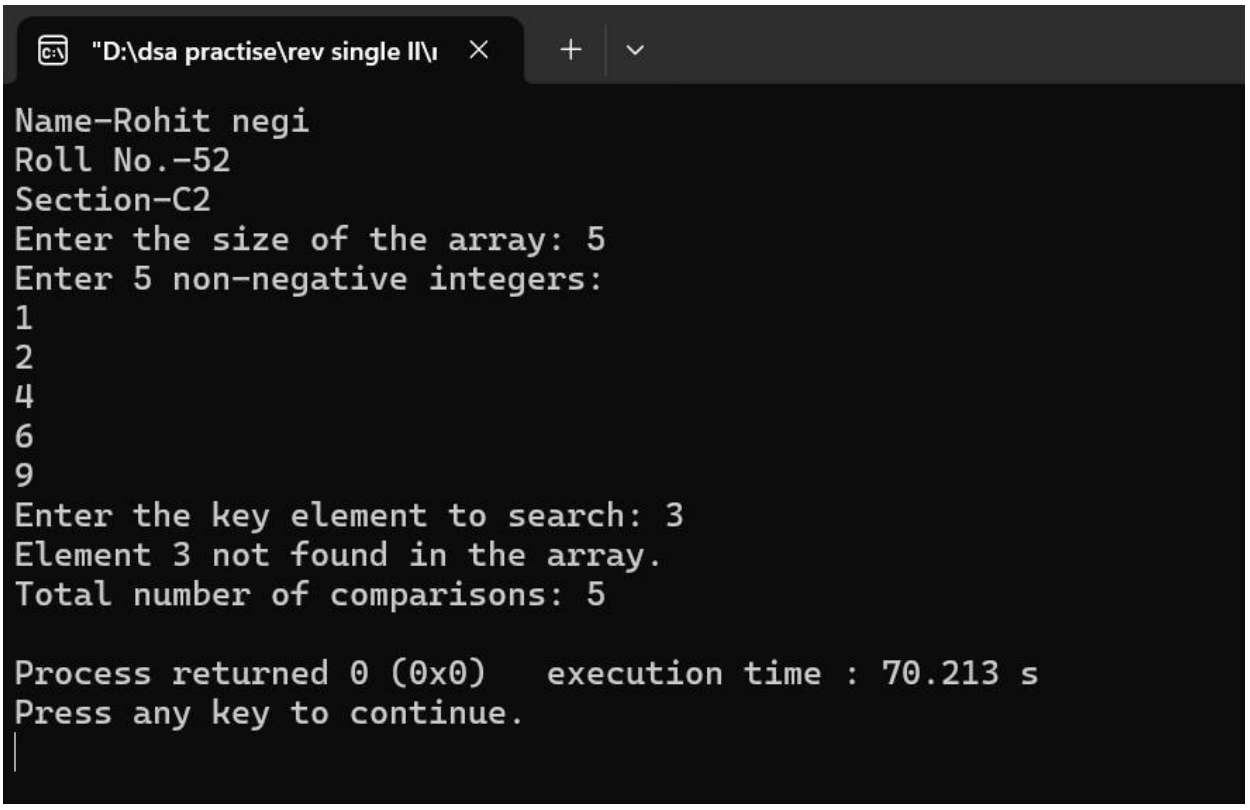
1. **Given an array of non- negative integers , design a linear algorithm and implement it using a program to find whether given key element is present in the array or not. Also , find total number of comparisons for each input case. (Time complexity = O(n) , where n is the size of input).**

Solution:

```
#include <stdio.h>
int linearSearch(int arr[], int n, int key, int *comparisons) {
for (int i = 0; i < n; i++) {
(*comparisons)++;
if (arr[i] == key) {
return i;
            }
            }
return -1;
}
int main() {
int n;
printf("Name-Rohit Negi\nRoll
No.52\nSection-C2\n");
printf("Enter the size of the array: ");
scanf("%d", &n);
int arr[n];
printf("Enter %d non-negative integers:\n", n);
for (int i = 0; i < n; i++) {
scanf("%d", &arr[i]);
 }
 int key;
```

```
 printf("Enter the key element to search: ");

 scanf("%d", &key);

 int comparisons = 0;
 int result = linearSearch(arr, n, key, &comparisons);

 if (result != -1) {

 printf("Element %d found at index %d.\n", key, result);
 } else {
printf("Element %d not found in the array.\n", key);
}
printf("Total number of comparisons: %d\n", comparisons);

return 0;

}
```

OUTPUT:


"D:\dsa practise\rev single ll\
Name-Rohit negi
Roll No.-52
Section-C2
Enter the size of the array: 5
Enter 5 non-negative integers:
1
2
4
6
9
Enter the key element to search: 3
Element 3 not found in the array.
Total number of comparisons: 5
Process returned 0 (0x0)   execution time : 70.213 s
Press any key to continue.

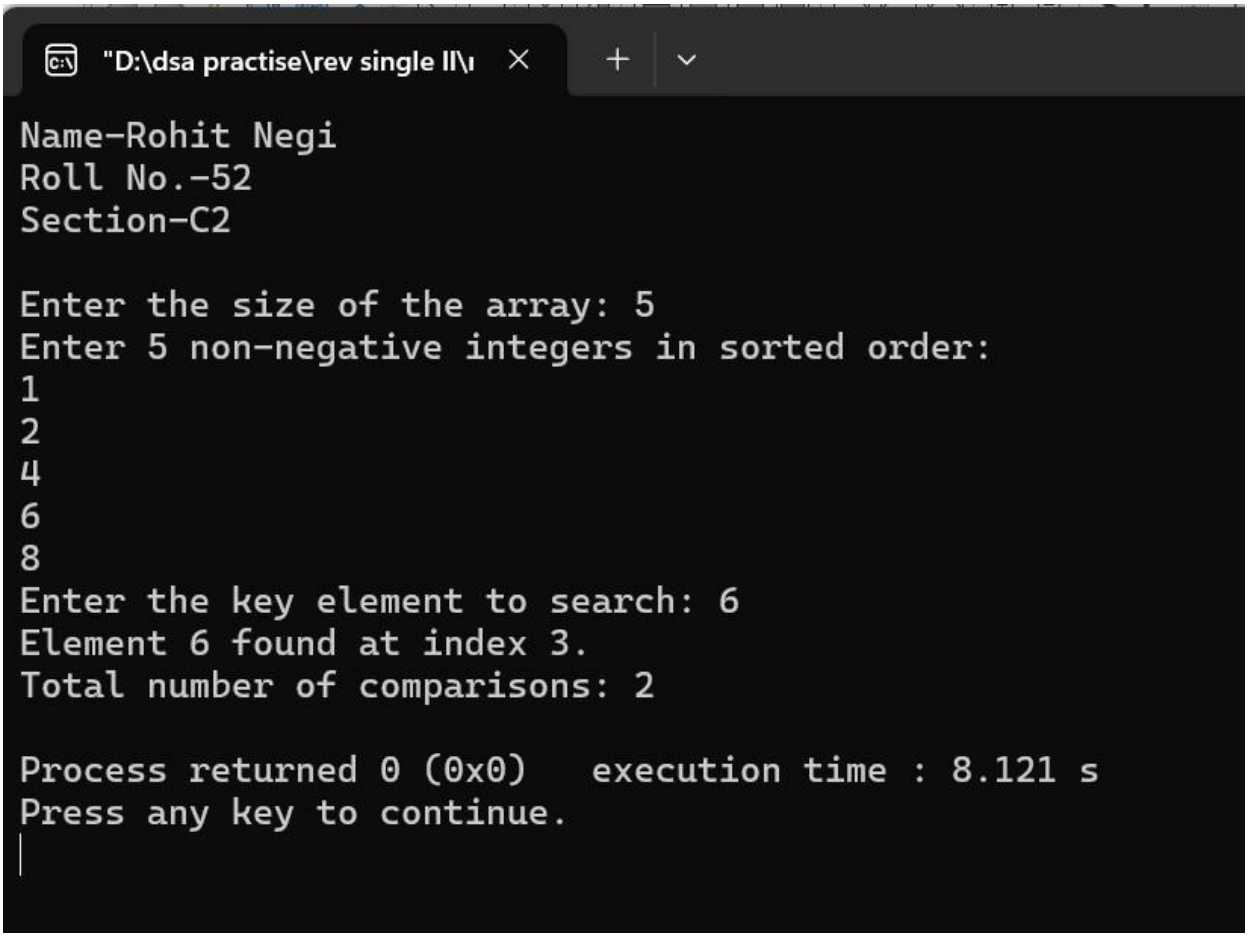
2. Given an array of non-negative integers, design a linear algorithm and implement it using a program in c programming to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time complexity = O(nlogn), where n is the size ofinput)

Solution:

```
#include <stdio.h>
#include <stdlib.h>
int binarySearch(int arr[], int low, int high, int key, int *comparisons) {
while (low <= high) {
      int mid = low + (high - low) / 2;
      (*comparisons)++;
     if (arr[mid] == key) {
         return mid;
      } else if (arr[mid] < key) {
         low = mid + 1;
      } else {
         high = mid - 1;
      } }
     return -1;
 }
 int main() {
 int n;
 printf("Name-Rohit Negi\nRoll No.-
   52\nSection-C2\n");
 printf("\nEnter the size of the array: ");
   scanf("%d", &n);
   int arr[n];
   printf("Enter %d non-negative integers in sorted order:\n", n);
   for (int i = 0; i < n; i++) {
      scanf("%d", &arr[i]);
   }
   int key;
   printf("Enter the key element to search: ");
   scanf("%d", &key);
   int comparisons = 0;
   int result = binarySearch(arr, 0, n - 1, key, &comparisons);

   if (result != -1) {
```

```
        printf("Element %d found at index %d.\n", key, result);
    } else {
        printf("Element %d not found in the array.\n", key);
    }
    printf("Total number of comparisons: %d\n", comparisons);
    return 0;
}
```

OUTPUT:


"D:\dsa practise\rev single ll\
Name-Rohit Negi
Roll No.-52
Section-C2
Enter the size of the array: 5
Enter 5 non-negative integers in sorted order:
1
2
4
6
8
Enter the key element to search: 6
Element 6 found at index 3.
Total number of comparisons: 2
Process returned 0 (0x0) execution time : 8.121 s
Press any key to continue.

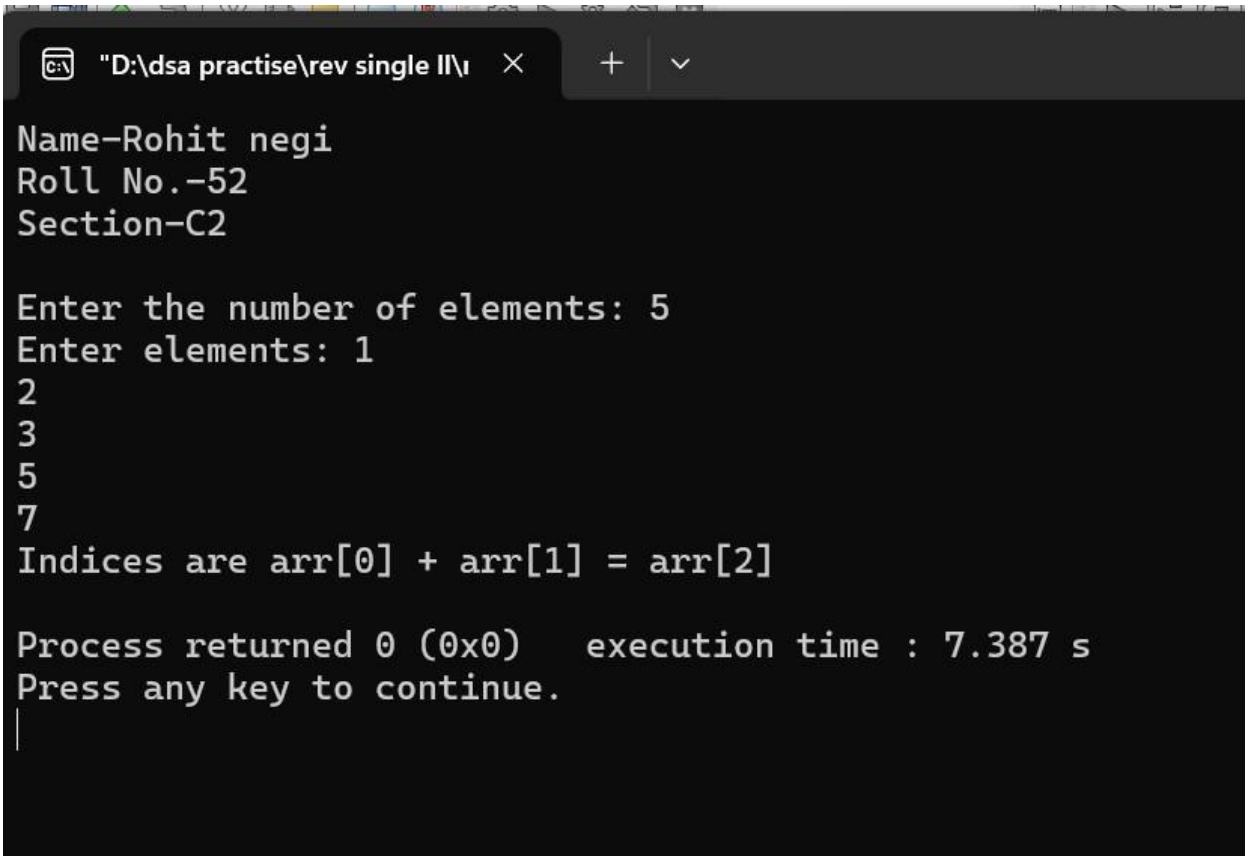
3. Given a sorted array of positive integers , design an algorithm and implement it using a program to find three indices i,j,k such that arr[i]+arr[j]=arr[k].

Solution:

```
#include<stdio.h>
void indices(int arr[], int n){
int flag = 1;
for(int i = 0; i < n - 2; i++) {
        for(int j = i + 1; j < n - 1; j++) {
          for(int k = j + 1; k < n; k++) {
            if(arr[i] + arr[j] == arr[k]) {
               printf("Indices are arr[%d] + arr[%d] = arr[%d]\n", i, j, k);
               flag = 0;
               return;
            }
          }
       }
   }
if(flag == 1){
      printf("No sequence found\n");
   }
}
int main(){
int n;
printf("Name-Rohit Negi\nRoll No.-
52\nSection-C2\n");
printf("\nEnter the number of elements: ");
scanf("%d", &n);
int arr[n];
printf("Enter elements: ");
```

```
for(int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
 indices(arr, n);
 return 0;
 }
```

OUTPUT:


"D:\dsa practise\rev single ll\
Name-Rohit negi
Roll No.-52
Section-C2
Enter the number of elements: 5
Enter elements: 1
2
3
5
7
Indices are arr[0] + arr[1] = arr[2]
Process returned 0 (0x0)   execution time : 7.387 s
Press any key to continue.