

G. H. Raison College of Engineering & Management, Wagholi, Pune – 412 207			
Department of Information and Technology Engineering			
SUBJECT :			
Name: Swayam Terode	Year: SY	Roll Number: 45	SEC: A
Registration No: 2020AIFT1101047		Date: 28/09/2021	

ASSIGNMENT NO: 2

PROBLEM DEFINITION: Implementation of Different Searching Algorithms Methods and Techniques.

1. LINEAR SEARCH

SOURCE CODE:

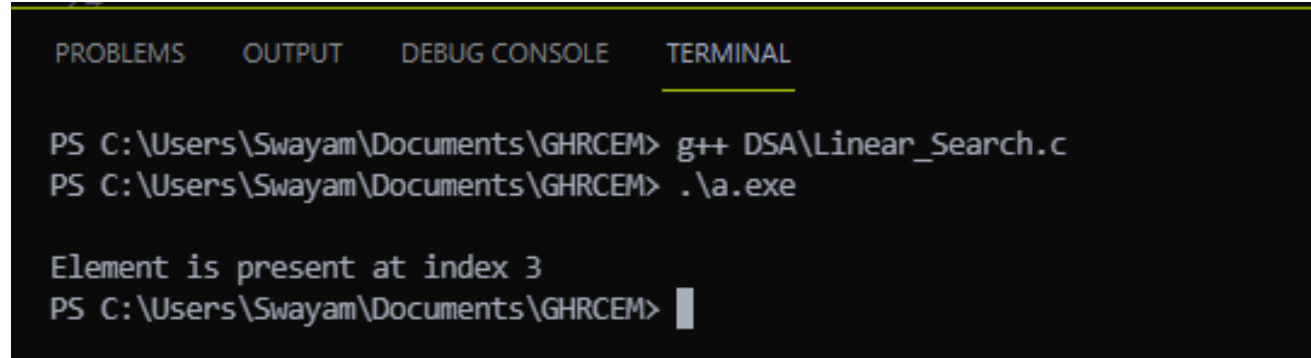
```
#include <stdio.h>
int linearSearch(int arr[], int n, int x)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
            return i;
    return -1;
}

int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;
    int n = sizeof(arr) / sizeof(arr[0]);

    int result = linearSearch(arr, n, x);
```

```
(result == -1) //Using ternary operator
    ? printf("\nElement is not present in array\n")
    : printf("\nElement is present at index %d\n", result);
return 0;
}
```

OUTPUT:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Swayam\Documents\GHRCEM> g++ DSA\Linear_Search.c
PS C:\Users\Swayam\Documents\GHRCEM> .\a.exe

Element is present at index 3
PS C:\Users\Swayam\Documents\GHRCEM> █
```

CONCLUSION:

The **time complexity** of the above algorithm is $O(n)$.
Linear search is rarely used practically because other search algorithms such as the binary search algorithm and hash tables allow significantly faster-searching comparison to Linear search.

B: BINARY SEARCH

SOURCE CODE:

```
// C program to implement recursive Binary Search
#include <stdio.h>

int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;

        // If the element is present at the middle
        // itself
        if (arr[mid] == x)
            return mid;

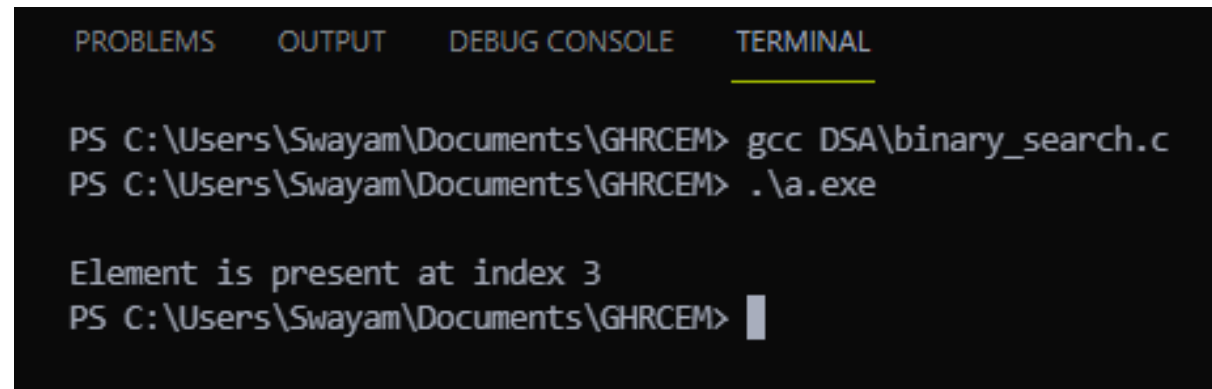
        // If element is smaller than mid, then
        // it can only be present in left subarray
        if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);

        // Else the element can only be present
        // in right subarray
        return binarySearch(arr, mid + 1, r, x);
    }
    return -1;
}

int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 10;
    int result = binarySearch(arr, 0, n - 1, x);
    (result == -1) ? printf("Element is not present in array\n")
        : printf("\nElement is present at index %d\n",
            result);
}
```

```
    return 0;  
}
```

OUTPUT:



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are four tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is selected and has a yellow underline. Below the tabs, the terminal shows the following commands and output:

```
PS C:\Users\Swayam\Documents\GHRCEM> gcc DSA\binary_search.c  
PS C:\Users\Swayam\Documents\GHRCEM> .\a.exe  
  
Element is present at index 3  
PS C:\Users\Swayam\Documents\GHRCEM> |
```