SEARCHING IN C LANGUAGE

What is Searching?

Searching is an operation or a technique that helps finds the place of a given element or value in the list. Any search is said to be successful or unsuccessful depending upon whether the element that is being searched is found or not. Some of the standard searching technique that is being followed in the data structure is listed below:

- Linear Search or Sequential Search
- Binary Search

What is Linear Search?

This is the simplest method for searching. In this technique of searching, the element to be found in searching the elements to be found is searched sequentially in the list. This method can be performed on a sorted or an unsorted list (usually arrays). In case of a sorted list searching starts from 0^{th} element and continues until the element is found from the list or the element whose value is greater than (assuming the list is sorted in ascending order), the value being searched is reached.

As against this, searching in case of unsorted list also begins from the 0^{th} element and continues until the element or the end of the list is reached.

One-Dimensional Array having 7 Elements

Example:

The list given below is the list of elements in an unsorted array. The array contains ten elements. Suppose the element to be searched is '46', so 46 is compared with all the elements starting from the 0^{th} element, and the searching process ends where 46 is found, or the list ends.

The performance of the linear search can be measured by counting the comparisons done to find out an element. The number of comparison is 0(n).

// Linear Search Algorithm

Start

Step 1: Select the first element as the current element.

Step 2: Compare the current element with the target element. If matches, then go to step 5.

Step 3: If there is a next element, then set current element to next element and go to Step 2.

Step 4: Target element not found. Go to Step 6.

Step 5: Target element found and return location.

Step 6: Exit process.

//Implementation of Linear Search Technique in C Language

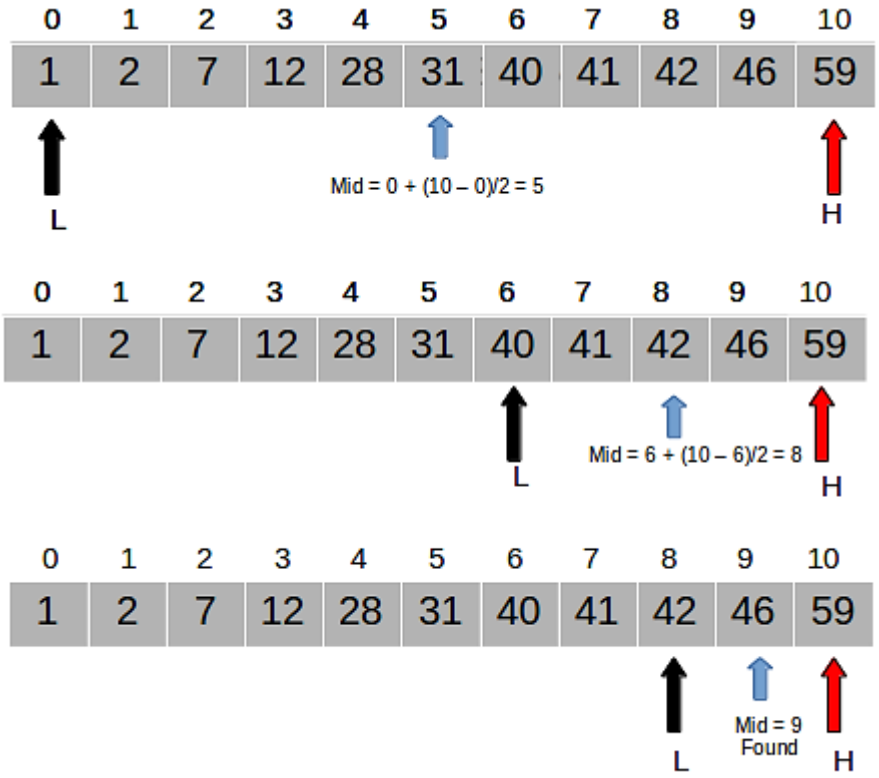
```
#include <stdio.h>
#include <conio.h>
void main()
{
        intarr[50], search, c, num;             //variable declaration

        printf("Enter the number of elements in array\n");
        scanf("%d",&num);

        printf("Enter %d integer in the Array \n", num);

        for (c = 0; c <num; c++)
        scanf("%d", &arr[c]);

        printf("Enter the number to search \n");
        scanf("%d", &search);

        for (c = 0; c <num; c++)
        {
                if (arr[c] == search)           /* if required element found */
                {
                        printf("%d is present at location %d.\n", search, c+1);
                        break;
                }
        }
        if (c == num)
        printf("%d is not present in array.\n", search);
        getch();
}
```

What is Binary Search?

Binary search is a very fast and efficient searching technique. It requires the list to be in **sorted order**. In this method, to search an element you can compare it with the present element at the center of the list. If it matches, then the search is successful otherwise the list is divided into two halves: one from the 0^{th} element to the middle element which is the center element (first half) another from the center element to the last element (which is the 2^{nd} half) where all values are greater than the center element.

The searching mechanism proceeds from either of the two halves depending upon whether the target element is greater or smaller than the central element. If the element is smaller than the central element, then searching is done in the first half, otherwise searching is done in the second half.

Binary Search to find the element "T=46" in a given list of numbers



//Binary Search Algorithm

Let us consider a Sorted LIST of size N and Target Value is T

BEGIN

Step 1. MAX = N
MIN = 1
FOUND = false

Step 2. WHILE (FOUND is false) and MAX > = MIN)
MID = (MAX + MIN) / 2
If T = LIST [MID]
I=MID
FOUND = true
Else If T <LIST[MID]
MAX = MID-1
Else
MIN = MD+1

END

//Implementation of Binary Search Technique in C Language

```
#include <stdio.h>
#include <conio.h>
void main()
{
        int c, first, last, middle, n, search, array[100];
        printf("Enter number of elements\n");
        scanf("%d", &n);
        printf("Enter %d integers in the Array\n", n);
        for (c = 0; c < n; c++)
        {       scanf("%d", &array[c]);         }
        printf("Enter the Number to Search\n");
        scanf("%d", &search);
        first = 0;
        last = n - 1;
        middle = (first+last)/2;
        while (first <= last)
        {
                if (array[middle] < search)
                {       first = middle + 1;}
                else if (array[middle] == search)
                {
                        printf("%d found at location %d.\n", search, middle+1);
                        break;
                }
                else
                {       last = middle - 1;}
                middle = (first + last)/2;
        }
        if (first > last)
        {       printf("The Number %d Not found \n", search);}
        getch();
}
```