

**Practical No. 1**

**Aim of Practical:** Implement Binary Search using Iterative algorithm design approach.

**Software Required:** C Compiler

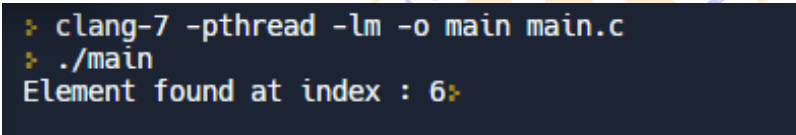
**Algorithm:**

```
BinarySearch(arr, x, low, high)
Step 1: Repeat Step 2 till low = high
Step 2: mid = (low + high)/2
        if (x == arr[mid])
            return mid
        else if (x > arr[mid])
            low = mid + 1
        else
            high = mid - 1
Step 3: Exit
```

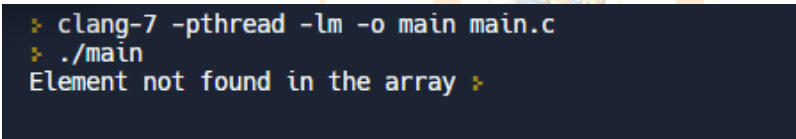
**Code:**

```
#include <stdio.h>
int iterativeBinarySearch(int array[], int start_index, int
end_index, int element)
{
    while (start_index <= end_index){
        int middle = start_index + (end_index- start_index )/2;
        if (array[middle] == element)
            return middle;
        if (array[middle] < element)
            start_index = middle + 1;
        else
            end_index = middle - 1;
    }
    return -1;
}
```

```
int main(void){
    int array[] = {1, 4, 7, 9, 16, 56, 70};
    int n = 7;
    int element = 1;
    int found_index = iterativeBinarySearch(array, 0, n-1, element);
    if(found_index == -1 ) {
        printf("Element not found in the array ");
    }
    else {
        printf("Element found at index : %d",found_index);
    }
    return 0;
}
```

**Output/ Result & Analysis:**

```
❯ clang-7 -pthread -lm -o main main.c
❯ ./main
Element found at index : 6❯
```



```
❯ clang-7 -pthread -lm -o main main.c
❯ ./main
Element not found in the array ❯
```

Conclusion: Thus we have implemented Binary Search using Iterative algorithm design approach.