Exercise 11 Searching Methods

11. Write a C++ Program for searching integer number using following methods

a) Sequential Searching b) Binary Searching.

Objective: The objective of this exercise enables you to perform searching integer number using sequential search and binary search methods

Procedure and description:

a) Sequential Search: In the sequential search, each element of the array is compared to the key (searching element), in the order it appears in the array, until the element of matching key is found.

Algorithm:

list: elements of array
last: last element of array
target: key element (searching element)
Location: index position of array

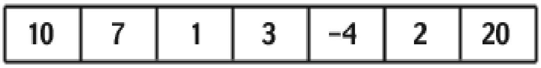
The steps for Sequential searching are given below:

```
Seqsearch (list, last, target, location)
Step 1:   set current to 0
Step 2:   set found to FALSE
Step 3:   loop (current <end List AND list [current<>target)
           Increment current
Step 4:   end loop
Step 5:   set location to current
Step 6:   if (list [current] =target)
             Set found to TRUE
Step 7: end if
End Seqsearch
```

Expected output:

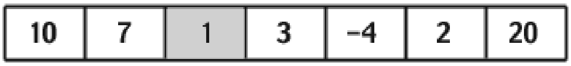
After execute the program. First enter Size of array and enter integer numbers and then enter searching number (key), for better understanding example as shown below

Input: The list of integer numbers



10	7	1	3	-4	2	20

And enter the key value suppose 1 is the key element, then output is “element is found”



10	7	1	3	-4	2	20

b) Binary search

Binary search is one of the fastest ways to search the element in a sorted array. The idea is to look at the element in the middle. If the key is equal to that element, then the search is finished. If the key is less than the middle element, do a binary search on the first half. If it's greater than the middle element, do a binary search of the second half.

Algorithm:

A [ ]: array elements

n: size of array

x: key element (searching element)

low: starting index value of array

high: highest index value of array

mid :(low +high)/2

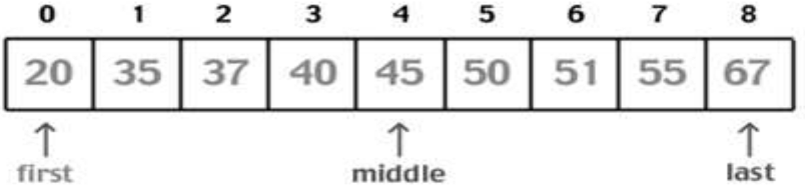
The steps for binary searching are given below:

```
     Binary-Search [A, n, x]
Step 1:   low ← 1
Step 2:   high ← n
Step 3:   while low ≤ high
Step 4:   do mid← [(low + high)/2]
Step 5:   if A[mid] = x
Step 6:   then return mid
Step 7:   else if A[mid] < x
Step 8:   then low← mid + 1
Step 9:   else high   mid − 1
          return “x not found
```

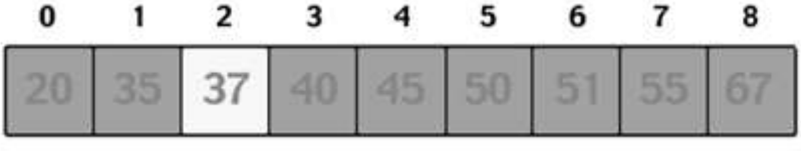
Expected output:

After execute the program. First enter input size of array and enter integer numbers and then enter searching number (key), as shown below

Input: The list of integer numbers in sorted order



As specified in algorithm it verifies searching element by dividing list into two parts. Suppose the key value is 37 then Output is “Key element is found”


0
1
2
3
4
5
6
7
8
20
35
37
40
45
50
51
55
67