a) Write a c program to perform linear Search

```
b) #include <stdio.h>
c) int linearSearch(int a[], int n, int val) {
d)   // Going through array sequencially
e)   for (int i = 0; i < n; i++)
f)     {
g)        if (a[i] == val)
h)        return i+1;
i)     }
j)   return -1;
k) }
l)  int main() {
m)   int a[] = {70, 40, 30, 11, 57, 41, 25, 14, 52}; // given array
n)   int val = 41; // value to be searched
o)   int n = sizeof(a) / sizeof(a[0]); // size of array
p)   int res = linearSearch(a, n, val); // Store result
q)   printf("The elements of the array are - ");
r)   for (int i = 0; i < n; i++)
s)   printf("%d ", a[i]);
t)   printf("\nElement to be searched is - %d", val);
u)   if (res == -1)
v)   printf("\nElement is not present in the array");
w)  else
x)   printf("\nElement is present at %d position of array", res);
y)   return 0;
 }
```

b) Write a c program to perform binary search

```
1. #include <stdio.h>
2. int binarySearch(int a[], int beg, int end, int val)
3. {
4.     int mid;
5.     if(end >= beg)
```

```
6.     {       mid = (beg + end)/2;
7. /* if the item to be searched is present at middle */
8.         if(a[mid] == val)
9.         {
10.                 return mid+1;
11.             }
12.                 /* if the item to be searched is smaller than middle, t
   hen it can only be in left subarray */
13.             else if(a[mid] < val)
14.             {
15.                 return binarySearch(a, mid+1, end, val);
16.             }
17.                 /* if the item to be searched is greater than middle, t
   hen it can only be in right subarray */
18.             else
19.             {
20.                 return binarySearch(a, beg, mid-1, val);
21.             }
22.           }
23.           return -1;
24.       }
25.       int main() {
26.         int a[] = {11, 14, 25, 30, 40, 41, 52, 57, 70}; // given array
27.         int val = 57; // value to be searched
28.         int n = sizeof(a) / sizeof(a[0]); // size of array
29.         int res = binarySearch(a, 0, n-1, val); // Store result
30.         printf("The elements of the array are - ");
31.         for (int i = 0; i < n; i++)
32.         printf("%d ", a[i]);
33.         printf("\nElement to be searched is - %d", val);
34.         if (res == -1)
35.         printf("\nElement is not present in the array");
36.         else
37.         printf("\nElement is present at %d position of array", res);
```

```
38.         return 0;
39.       }
```