LINEAR SEARCH

```
#include <stdio.h>
int RecursiveLS(int arr[], int value, int index, int n)
{
  int pos = 0;

  if(index >= n)
  {
     return 0;
  }

  else if (arr[index] == value)
  {
     pos = index + 1;
     return pos;
  }

  else
  {
     return RecursiveLS(arr, value, index+1, n);
  }
  return pos;
}

int main()
{
  int n, value, pos, m = 0, arr[100];
  printf("Enter the total elements in the array  ");
  scanf("%d", &n);

  printf("Enter the array elements\n");
  for (int i = 0; i < n; i++)
  {
     scanf("%d", &arr[i]);
  }

  printf("Enter the element to search  ");
  scanf("%d", &value);
```

```
  pos =  RecursiveLS(arr, value, 0, n);
  if (pos != 0)
  {
     printf("Element found at pos %d ", pos);
  }
  else
  {
     printf("Element not found");
  }
  return 0;
}
```

BINARY SEARCH

```
#include <stdio.h>

void binary_search(int [], int, int, int);
void bubble_sort(int [], int);

int main()
{
  int key, size, i;
  int list[25];

  printf("Enter size of a list: ");
  scanf("%d", &size);
  printf("Enter elements\n");
  for(i = 0; i < size; i++)
  {
     scanf("%d",&list[i]);
  }
  bubble_sort(list, size);
  printf("\n");
  printf("Enter key to search\n");
  scanf("%d", &key);
  binary_search(list, 0, size, key);
```

```
}

void bubble_sort(int list[], int size)
{
  int temp, i, j;
  for (i = 0; i < size; i++)
  {
     for (j = i; j < size; j++)
     {
        if (list[i] > list[j])
        {
           temp = list[i];
           list[i] = list[j];
           list[j] = temp;
        }
     }
  }
}

void binary_search(int list[], int lo, int hi, int key)
{
  int mid;

  if (lo > hi)
  {
     printf("Key not found\n");
     return;
  }
  mid = (lo + hi) / 2;
  if (list[mid] == key)
  {
     printf("Key found\n");
  }
  else if (list[mid] > key)
  {
     binary_search(list, lo, mid - 1, key);
  }
  else if (list[mid] < key)
```

```
  {
    binary_search(list, mid + 1, hi, key);
  }
}
```