

1.IMPLEMENTATION OF LINEAR SEARCH:

```
#include <stdio.h>

void main()
{
int arr[50];
int i, key,pos,n;
printf("enter array size:");
scanf("%d", &n);
printf("enter array:");
for(i=0;i<n;i++)
{
scanf("%d",&arr[i]);
}
printf("enter key =");
scanf("%d", &key);
for(i=0;i<n;i++);
{
if(arr[i]==key)
{
pos=i;
flag=1;
break;
}
}
if(flag==1)
printf("element found at the position %d",i);
```

```
else  
printf("element not found");  
}
```

2.IMPLEMENTATION OF BINARY SEARCH:

```
#include <stdio.h>  
  
int binarySearch(int arr[], int l, int r, int x)  
{  
if (r >= l) {  
int mid = l + (r - l) / 2;  
if (arr[mid] == x)  
return mid;  
if (arr[mid] > x)  
return binarySearch(arr, l, mid - 1, x);  
return binarySearch(arr, mid + 1, r, x);  
}  
return -1;  
}  
  
void main()  
{  
int arr[50];  
int i, key, pos, n;  
printf("enter array size:");  
scanf("%d", &n);  
printf("enter array:");  
for(i=0; i<n; i++)  
{
```

```
scanf("%d",&arr[i]);
}
printf("enter key =");
scanf("%d", &key);
int found = binarySearch(arr,0, n-1, key);
if(found == -1)
printf("not found");
else
printf("Element is found at position %d",found+1);
}
```

3.IMPLEMENTATION OF BUBBLE SORT:

```
#include <stdio.h>
void swap(int *x, int *y)
{
int temp = *x;
*x = *y;
*y = temp;
}
void bubble_sort(int arr[], int n)
{
int i, j;
for (i = 0; i < n-1; i++)
{
for (j = 0; j < n-i-1; j++)
{
if (arr[j] > arr[j+1])
```

```
swap(&arr[j], &arr[j+1]);
}
}
}
void main()
{
int arr[50];
int i, key,pos,n;
printf("enter array size:");
scanf("%d", &n);
printf("enter array:\n");
for(i=0;i<n;i++)
{
scanf("%d",&arr[i]);
}
bubble_sort(arr, n);
printf("Sorted array: \n");
for(i=0;i<n;i++)
{
printf("%d, ",arr[i]);
}
}
```

4.IMPLEMENTATION OF INSERTION SORT:

```
#include<stdio.h>
void insertion_sort(int arr[], int n)
{
```

```
int i, key, j;
for (i = 1; i < n; i++)
{
key = arr[i];
j = i - 1;
/* Move elements of arr[0..i-1], that are greater than key, to one position ahead of their current
position */
while (j >= 0 && arr[j] > key)
{
arr[j + 1] = arr[j];
j = j - 1;
}
arr[j + 1] = key;
}
}

void main()
{
int arr[50];
int i, key, pos, n;
printf("enter array size:");
scanf("%d", &n);
printf("enter array:\n");
for(i=0;i<n;i++)
{
scanf("%d",&arr[i]);
}
}
```

```
insertion_sort(arr, n);  
printf("Sorted array: \n");  
for(i=0;i<n;i++)  
{  
printf("%d, ",arr[i]);  
}  
}
```

5.IMPLEMENTATION OF SELECTION SORT:

```
#include <stdio.h>  
  
void swap(int *x, int *y)  
{  
int temp = *x;  
*x = *y;  
*y = temp;  
}  
  
void selection_sort(int arr[], int n)  
{  
int i, j, min_idx;  
for (i = 0; i < n-1; i++)  
{  
min_idx = i;  
for (j = i+1; j < n; j++)  
if (arr[j] < arr[min_idx])  
min_idx = j;  
swap(&arr[min_idx], &arr[i]);  
}  
}
```

```
}  
void main()  
{  
int arr[50];  
int i, key,pos,n;  
printf("enter array size:");  
scanf("%d", &n);  
printf("enter array:\n");  
for(i=0;i<n;i++)  
{  
scanf("%d",&arr[i]);  
}  
selection_sort(arr, n);  
printf("Sorted array: \n");  
for(i=0;i<n;i++)  
{  
printf("%d, ",arr[i]);  
}  
}
```