Linear search

```
#include<stdio.h>

#include<stdlib.h>

int main()

{

int a[5]={3,2,5,6,7},i,s;

printf("Enter search data: ");

scanf("%d",&s);

for(i=0;i<5;i++)

if(a[i]==s)

{

printf("Data found at index: %d",i);

exit(0);

}

printf("Data not found");

return 0;

}
```

Binary Search

```
#include<stdio.h>

#include<stdlib.h>

int main()

{

int a[9]={1,2,5,6,7,9,11,16,20},s,l=0,r=8,m;

printf("Enter search data: ");

scanf("%d",&s);
while(l<=r)
{
    m=(l+r)/2;
    printf("%d---%d---%d\n",l,m,r);
    if(s==a[m])
    {
    printf("Data found at index: %d",m);
    exit(0);
    }
    else if(s>a[m])
        l=m+1;
        else
        r=m-1;
}
printf("Data not found");
return 0;
}
```

Bubble sort

```
#include<stdio.h>
#include<stdlib.h>
#define n 5
int main()
{
int a[n]={1,4,5,8,7},i,j,temp;
for(i=0;i<n-1;i++)
for(j=0;j<n-1-i;j++)
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
printf("The bubble sorted array is: ");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
return 0;
}
```

//Optimized bubble sort

```
#include<stdio.h>
#include<stdlib.h>
#define n 5
int main()
{
int a[n]={1,4,5,8,7},count=0,i,j,temp,flag;
for(i=0;i<n-1;i++)
{
count++;
flag=0;
for(j=0;j<n-1-i;j++)
if(a[j]>a[j+1])
{
flag=1;
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
if(flag==0)
break;
}
printf("Number of passes: %d\n",count);
printf("The bubble sorted array is: ");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
return 0;
}
```

//Insertion sort

```
#include<stdio.h>
#include<stdlib.h>
#define n 5
int main()
{
    int a[n]={5,9,7,8,17}, i, j, temp;
    for(i=1;i<n;i++)
    {
    temp=a[i];
    j=i-1;
    while(j>=0 && a[j]>temp)
    {
a[j+1]=a[j];
j--;
    }
    a[j+1]=temp;
}
printf("The result of insertion sort is: ");
 for(i=0;i<n;i++)
 printf("%d\t",a[i]);
 return 0;
}
```

//Selection sort

```
#include<stdio.h>

#include<stdlib.h>

#define n 5

int main()

{

    int a[n]={12,19,7,1,17}, i, j, min, temp;

    for(i=0;i<n-1;i++)

    {

        min=i;

    for(j=i+1;j<n;j++)

    if(a[j]<a[min])

    min=j;

    if(min!=i)

    {

        temp=a[min];

        a[min]=a[i];

        a[i]=temp;

    }

    }

printf("The result of selection sort is: ");

 for(i=0;i<n;i++)

 printf("%d\t",a[i]);

 return 0;

}
```

// Quick Sort

```
#include<stdio.h>
# define n 8
int a[n]={5,7,12,9,1,19,27,37};
void quicks(int h, int t){
    int l,r,temp,k;
    if(h<t){
        l=h;
        k=h;
        r=t;
        while(l<r){
        while(a[l]<=a[k] && l<t)
        l++;
        while(a[r]>a[k] && r>h)
        r--;
        if(l<r){
            temp=a[l];
            a[l]=a[r];
            a[r]=temp;
        }
        }
    temp=a[k];
            a[k]=a[r];
            a[r]=temp;
            quicks(h,r-1);
            quicks(r+1,t);
    }
}
int main(){
    int i;
    quicks(0,n-1);
    printf("The sorted list is: ");
    for(i=0;i<n;i++)
    printf("%d\t",a[i]);
    return 0;
}
```

//Merge Sort

```
#include<stdio.h>
#include<stdlib.h>
#define n 8
int a[n]={1,12,15,92,17,9,6,20},s,m,b[n];
void merg(int l, int m, int r) {
   int i = l, j = m +  1, k = l;
   while(i<=m && j<=r) {
      if(a[i] <= a[j])
      {
         b[k] = a[i];
      i++;
      }
      else
      {
         b[k] = a[j];
         j++;
   }
   k++;
   }
   if (i>m)
       while(j <= r)   {
      b[k] = a[j];
      k++;
      j++;
   }
   if(j>r)
          while(i <= m)
          {
      b[k] = a[i];
      k++;
      i++;
}
   for(i = l; i < k; i++)
      a[i] = b[i];
}
void sort(int l, int r) {
   int m;
   if(l < r) {
      m = (l + r) / 2;
      sort(l, m);
      sort(m+1, r);
```

```
        merg(l, m, r);
    }
}
int main() {
    int i;
    sort(0, n-1);
    printf("\nResult of merge sort is: \n");
    for(i = 0; i < n; i++)
        printf("%d ", a[i]);
}
```