1. Write a program to implement linear search & Binary Search in array.

```
#include<stdio.h>
#include<conio.h>

void linear(int a[],int n,int item)
{
 int ele,i;
 printf("Enter element to be searched");
    scanf("%d",&ele);
    for(i=0;i<m;i++)
    {if(a[i]==ele)
     printf("Element is found at position %d",i+1); }
}

void binary(int a[],int m)
{
 int ele,i,mid,first,last;
    printf("Enter element to be searched");
    scanf("%d",&ele);
 first=0;
 last=m-1;
 while(first<=last)
 {
mid=(first+last)/2;
       if(a[mid]<ele)
       first=mid+1;
       if(a[mid]>ele)
   last=mid-1;
   else
if(a[mid]==ele)
     printf("Element is found at position %d",mid+1);
   else
   printf("Element not in array");
      }
}

void display(int a[],int n)
{int i;
 printf("Your array is");
 for(i=0;i<n;i++)
 printf("%d",a[i]);
}

void main()
{
 int a[40],n,ch;
 printf("Enter no of elements of array:");
 scanf("%d",&n);
 printf("Enter elements of array\n");
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 while(1)
 {
  printf("\n Enter your choice");
  printf("\n1.Linear Search");
  printf("\n2.Bubble search");
  printf("\n3.Exit");
  scanf("%d",&ch);
  switch(ch)
  {
  case 1: linear(a,n,);
```

```
                  display(a,n);
                  break;
  case 2:
binary(a,n);
                  display(a,n);
       break;
  case 3:
printf("Press Enter to Exit");
       break;
  default: printf("Sorry!!!, Wrong choice entered");
 }
getch();
}
```

2. Write a program to implement linear search in linked list.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
 int info;
 struct node *next;
}*p,*q;
typedef struct node node;

node *create(node *start)
{
 char ch;
 do{
  p=(node*)malloc(sizeof(node));
  if(p==NULL)
  printf("Overeflow");
  else{
   printf("Enter the info:");
   scanf("%d",&p->info);
   if(start!=NULL)
   {
    q->next=p;
    q=p;
   }
   else
   start=q=p;
   printf("wanna insert another node?(y/n):");
   fflush(stdin);
   ch=getchar();
  }
 }while(ch!='n');
 q->next=NULL;
 return start;
}

void display(node *start)
{
 node *ptr;
 printf("Elements of link list are....\n");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
 printf("%d\n",ptr->info);
}
```

```
void Lsearch(node *start,int item)
{
 node *ptr=start;
 int n=1;
 while(ptr!=NULL && ptr->info!=item)
 {
  ptr=ptr->next;
  n++;
  }
  if(ptr==NULL)
  printf("not present");
  else
  printf("item found at %dth place",n);
}

void main()
{
 int item;
 node *start=NULL;
 start=create(start);
 display(start);
 printf("enter item to be searched:");
 scanf("%d",&item);
 Lsearch(start,item);
    getch();
}
```

3. Write a program to implement sorting in linked list.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
 int info;
 struct node *next;
}*p,*q;
typedef struct node node;

node *create(node *start)
{
 char ch;
 do{
  p=(node*)malloc(sizeof(node));
  if(p==NULL)
  printf("Overflow");
  else{
   printf("Enter the info:");
   scanf("%d",&p->info);
   if(start!=NULL)
   {
    q->next=p;
    q=p;
   }
   else
   start=q=p;
   printf("Want to insert another node?(y/n):");
   fflush(stdin);
```

```
   ch=getchar();
  }
 }while(ch!='n');
 q->next=NULL;
 return start;
}

void display(node *start)
{
 node *ptr;
 printf("Elements of link list are....\n");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
 printf("%d\n",ptr->info);
}

     //SELECTION_SORTING

void sort(node *start)
{
 node *ptr1,*ptr2;
 int tmp;
 for(ptr1=start;ptr1->next!=NULL;ptr1=ptr1->next)
 {
  for(ptr2=ptr1->next;ptr2!=NULL;ptr2=ptr2->next)
   if(ptr1->info>ptr2->info)
   {
    tmp=ptr1->info;
    ptr1->info=ptr2->info;
    ptr2->info=tmp;
   }
 }
}

void main()
{
 node *start=NULL;
 start=create(start);
 display(start);
 sort(start);
 printf("after sorting...\n");
 display(start);
    getch();
}
```

4. Write a program to implement Bubble sort in array.

```
#include<stdio.h>
#include<conio.h>
void sort(int a[],int n)
{
 int i,j,temp=0;
 for(i=0;i<n-1;i++)
 {for(j=0;j<n-i-1;j++)
  {if(a[j]>a[j+1])
   {
  tmp=a[j];
  a[j]=a[j+1];
  a[j+1]=tmp;
     }
```

```
}}
printf("Your Sorted Array is ");
for(i=0;i<n;i++)
printf("%d",a[i]);
}

void main()
{
 int b[40],m,i;
 printf("Enter number of elements");
 scanf("%d",&n);
 printf("Enter elements\n");
 for(i=0;i<m;i++)
 scanf("%d",&a[i]);
 sort(b,m);
    getch();
}
```

5. Write a program to implement Selection sort in array.

```
#include<stdio.h>
#include<conio.h>
void Ssort(int a[],int n)
{
 int i,j,tmp,min,loc;
 for(i=0;i<n-1;i++)
 {
  min=a[i];
  for(j=i+1;j<n;j++)
  if(min>a[j])
  {
   min=a[j];
   loc=j;
  }
  tmp=a[i];
  a[i]=a[loc];
  a[loc]=tmp;
 }
}
main()
{
 int a[25],n,i;
 printf("enter number of item to be inserted in array:");
 scanf("%d",&n);
 printf("insert items in array...\n");
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 printf("original(may be unsorted) array is...\n");
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 printf("\nsorted array is...\n");
 Ssort(a,n);
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 getch();
}
```

6. Write a program to implement Insertion sort in array.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void Isort(int a[],int n)
{
 int i,ptr,tmp;
 for(i=0;i<n;i++)
 {
  tmp=a[i];
  ptr=i-1;
  while(ptr>=0 && tmp<a[ptr])
  {
   a[ptr+1]=a[ptr];
   ptr=ptr-1;
  }
  a[ptr+1]=tmp;
 }
}
main()
{
 int a[25],n,i;
 printf("enter number of item to be inserted in array:");
 scanf("%d",&n);
 printf("insert items in array...\n");
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 printf("original(may be unsorted) array is...\n");
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 printf("\nsorted array is...\n");
 Isort(a,n);
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 getch();
}
```

7. Write a program to implement Quick sort in array.

```
#include<stdio.h>
#include<conio.h>
int partition(int a[],int lb,int ub)
{
 int x=a[ub],tmp,j;
 int i=lb-1;
 for(j=lb;i<ub;j++)
  if(a[j]<=x)
  {
   i=i+1;
   tmp=a[i];
   a[i]=a[j];
   a[j]=tmp;
  }
 tmp=a[ub];
 a[ub]=a[i+1];
 a[i+1]=tmp;
 return i+1;
}
```

```
void Quicksort(int a[],int lb,int ub)
{
 int p;
 if(lb<ub)
 {
  p=partition(a,lb,ub);
  Quicksort(a,lb,p-1);
  Quicksort(a,p+1,ub);
 }
}

void main()
{
 int a[25],n,i,lb,ub;
 printf("enter number of item to be inserted in array:");
 scanf("%d",&n);
 printf("insert items in array...\n");
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 printf("original(may be unsorted) array is...\n");
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 lb=0;
 ub=n-1;
 printf("\nsorted array is...\n");
 Quicksort(a,lb,ub);
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 getch();
}
```

8. Write a program to implement Merge sort in array.

```
#include<stdio.h>
#include<conio.h>
void merge(int a[],int p,int r,int q)
{
 int i,j,k;
 int n=q-p+1;
 int m=r-q;
 int L[25],R[25];
 for(i=0;i<n;i++)
 L[i]=a[p-i+1];
 for(j=0;j<m;j++)
 R[j]=a[q+j];
 L[n]=32768;
 R[m]=32768;
 i=j=1;
 for(k=p;k<=r;k++)
 {
  if(L[i]<=R[i])
  {
   a[k]=L[i];
   i++;
  }
  else
  {
   a[k]=R[j];
   j++;
```

```
 }
 }
}

void mergesort(int a[],int p,int r)
{
 int q;
 if(p<r)
 {
  q=(p+r)/2;
  mergesort(a,p,q);
  mergesort(a,q+1,r);
  merge(a,p,r,q);
 }
}

main()
{
 int a[25],n,i,lb,ub;
 printf("enter number of item to be inserted in array:");
 scanf("%d",&n);
 printf("insert items in array...\n");
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 printf("original(may be unsorted) array is...\n");
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 lb=0;
 ub=n-1;
 printf("\nsorted array is...\n");
 mergesort(a,lb,ub);
 for(i=0;i<n;i++)
 printf("\t%d",a[i]);
 getch();
}
```

9. Write a program to implement Heap sort in array.

```
#include<stdio.h>
#include<conio.h>
buildheap(int data[],int n,int item)
{
 n=n+1;
 int par,ptr=n;
 while(ptr>1)
 {
  par=ptr/2;
  if(item<=data[par])
  {data[ptr]=item;
  }
  data[ptr]=data[par];
  ptr=par;
 }
 data[1]=item;
 return n;
}
void heapsort(int data[],int n)
{
 int i;
```

```
 for(i=0;i<n;i++)
 buildheap(data,i,data[i]);
 /*while(n>0)
 {
  deleteheap(data,n);
  data[n]=item;
 }*/

}

void display(int data[],int n)
{
 int i;
 for(i=0;i<n;i++)
 printf("%d\t",data[i]);
 printf("\n");
}

deleteheap(int tree[],int n,int item)
{item=tree[1];
 int ptr=1,last,left=2,right=3;
 last=tree[n];
 n=n-1;
 while(right<=n)
 {if((last>=tree[left])&&(last>=tree[right]))
  {tree[ptr]=last;}
  if(tree[right]=<tree[left])
  {tree[ptr]=tree[left];
   ptr=left;}
  else
  {tree[ptr]=tree[right];
   ptr=right;}
  left=2*ptr;
  right=left+1;
  }
 if((left==n)&&(last=<tree[left]))
 {ptr=left; }
 tree[ptr]=last;
 return;
}

void main()
{
 int data[25],n,i;
 printf("enter number of items:");
 scanf("%d",&n);
 printf("insert items...\n");
 for(i=0;i<n;i++)
 scanf("%d",&data[i]);
 printf("original(may be unsorted) array is...\n");
 display(data,n);
 printf("\nsorted array is...\n");
 heapsort(data,n);
 display(data,n);
 getch();
}
```