Binary Search Program

```
#include <stdio.h>

int main()

{

int i, low, high, mid, n, key, array[100];

printf("Enter number of elements n");

scanf("%d",&n);

printf("Enter %d integersn", n);

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

scanf("%d",&array[i]);

printf("Enter value to findn");

scanf("%d", &key);

low = 0;

high = n - 1;

mid = (low+high)/2;

while (low <= high) {

if(array[mid] < key)
```

```
low = mid + 1;
else if (array[mid] == key) {
printf("%d found at location %d.n", key, mid+1);
break;
}
else
high = mid - 1;
mid = (low + high)/2;
}
if(low > high)
printf("Not found! %d isn't present in the list.n", key);
return 0;
}
```

Linear Search Program

```
#include<stdio.h>
int main()
```

```
{
    int a[20],i,x,n;
    printf("How many elements?");
    scanf("%d",&n);
    printf("\n Enter array elements:n");
    for(i=0;i<n;++i)
        scanf("%d",&a[i]);
    printf("\n Enter element to search:");
    scanf("%d",&x);
    for(i=0;i<n;++i)
        if(a[i]==x)
            break;
       if(i<n)
        printf("Element found at index %d",i);
        else
        printf("Element not found");
    return 0;
```

```
}
```

Queue Program

```
#include <stdio.h>
#include<stdlib.h>
#define MAX 50
void insert();
void delete();
void display();
int queue_array[MAX];
int rear = - 1;
int front = - 1;
int main()
{
int choice;
while (1)
{
```

```
printf("1.Insert element to queue n");
printf("2.Delete element from queue n");
printf("3.Display all elements of queue n");
printf("4.Quit n");
printf("Enter your choice : ");
scanf("%d", &choice);
switch(choice)
{
case 1:
     insert();
      break;
case 2:
     delete();
     break;
case 3:
    display();
    break;
```

```
case 4:
     exit(1);
default:
    printf("Wrong choice n");
}
}
}
void insert()
{
int item;
if(rear == MAX - 1)
printf("Queue Overflow n");
else
{
if(front== - 1)
front = 0;
printf("Inset the element in queue : ");
```

```
scanf("%d", &item);
rear = rear + 1;
queue_array[rear] = item;
}
}
void delete()
{
if(front == - 1 || front > rear)
{
printf("Queue Underflow n");
return;
}
else
{
printf("Element deleted from queue is : %dn",
queue_array[front]);
front = front + 1;
```

```
}
}
void display()
{
int i;
if(front == - 1)
printf("Queue is empty n");
else
{
printf("Queue is : n");
for(i = front; i <= rear; i++)
printf("%d ", queue_array[i]);
printf("n");
}
}
```

Stack Program

```
#include<stdio.h>
#include<conio.h>
#define MAX 3
int a[MAX], top = -1;
void push();
void pop();
void peep();
void change();
void display();
void main()
{
  int ch;
  clrscr();
 while(1)
{
  printf("\n 1. PUSH or INSERT");
```

```
printf("\n 2. POP or DELETE");
printf("\n 3. PEEP or SEARCH");
printf("\n 4. CHANGE or UPDATE");
printf("\n 5. Display");
printf("\n 6. End program");
printf("\n Enter Choice : ");
scanf("%d",&ch);
clrscr();
 switch(ch)
  {
    case 1:
          {
            push();
            break;
          }
    case 2:
          {
```

```
        pop();
        break;
       }
case 3:
      {
        peep();
        break;
      }
case 4:
       {
         change();
         break;
       }
case 5:
       {
         display();
         break;
```

```
            }
      case 6:
            {
              exit(0);
            }
      default:
           printf("\n invalid choice !!!");
            }
    getch();
 }
}
void push()
 {
         int data;
         if(top==MAX-1)
            {
               printf("\n overflow or stack is full !!!");
```

```
            }
        else
            {
                printf("\n Enter the element : ");
                scanf("%d",&data);
                top++;
                a[top]=data;
            }
}
void pop()
{
            if(top==-1)
            {
            printf("\ nunder flow STACK or STACK is empty");
            }
            else
            {
```

```
            printf("\n POP or DELETE element %d",a[top]);
             top--;
           }
}
void display()
{
      int i;
      if(top>=0)
         {
            printf ("\n Elemets : ");
           for(i=top; i>=0; i--)
              {
                printf ("\n %d",a[i]);
               }
         }
       else
           {
```

```
                printf("\n The STACK is Empty");
            }
}
void peep()
{
  int p;
  printf ("\n Enter the position : ");
  scanf ("%d",&p);
  if (top-p<=-1)
  {
    printf("\nSTACK is overflow !!!");
  }
  else
  {
    printf("\nThe Elements is : %d",a[top-p]);
  }
}
```

```
void change()
{
  int v1,v2;
  printf("\nEnter Position for change : ");
  scanf("%d",&v1);
  printf("\nEneter the Number for change : ");
  scanf("%d",&v2);
  if(top-v1<=-1)
  {
     printf ("\nSTACK is overflow !!!");
  }
  else
  {
    a [top-v1]=v2;
    printf ("\n CHANGE successful !!!");
  }
}
```

Bubble sort code

```
#include <stdio.h>

int main()

{

  int a[100], n, i, j, temp;

  printf("Enter number of elements\n");

  scanf("%d", &n);

  printf("Enter %d integers\n", n);

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

    scanf ("%d", &a[i]);

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

  {

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

    {

      if (a[j] > array[j+1])

      {

        temp = a[j];
```

```
        a[j]   = a[j+1];
        a[j+1] = temp;
      }
    }
  }
  printf ("Sorted list in ascending order: \n");
  for (i = 0; i < n; i++)
     printf ("%d\n", a[i]);
  return  0;
}
```