```
#include<iostream>
using namespace std;
typedef struct dequeue
{
      int data[30];
      int rear,front;
}dequeue;

void initialize(dequeue *p)
{
      p ->rear=-1;
      p->front=-1;
}

int isEmpty(dequeue *p)
{
      if(p->rear==-1)
      {
            return 1;
      }return 0;
}

int isFull (dequeue *p)
{
      if((p->rear+1)%30==p->front)
      {
            return 1;
      }return 0;
}

void enqueueRear(dequeue *p,int x)
{
      if(isEmpty(p))
      {
            p->rear=0;
            p->front=0;
            p->data[0]=x;
      }
      else
      {
            p->rear=(p->rear+1)%30;
            p->data[p->rear]=x;
      }
}

void enqueueFront(dequeue *p,int x)
{
      if(isEmpty(p))
      {
            p->rear=0;
            p->front=0;
            p->data[0]=x;
      }
      else
      {
            p->front=(p->front-1+30)%30;
            p->data[p->front]=x;
      }
```

```
}

int dequeueFront(dequeue *p)
{
      int x;
      x=p->data[p->front];
      if(p->rear==p->front)
      {
            initialize(p);
      }
      else
      {
            p->front=(p->front+1)%30;
            return(x);
      }
}

int dequeueRear (dequeue *p)
{
      int x;
      x=p->data[p->rear];
      if(p->rear==p->front)
      {
            initialize(p);
      }
      else
      {
            p->rear=(p->rear-1+30)%30;
            return(x);
      }
}

void display(dequeue *p)
{
      if(isEmpty(p))
      {
            cout<<"Queue is empty";
            exit(0);
      }
      int i;
      i=p->front;
      while(i!=p->rear)
      {
            cout<<""<<p->data[i];
            i=(i+1)%30;
      }
      cout<<""<<p->data[p->rear];
}

int main()
{
      int x,ch,n;
      dequeue q;
      initialize(&q);
      do
      {
            cout<<"********************Menu*****************";
            cout<<"\n1.Create";
```

```
cout<<"\n2.Insert(rear)";
cout<<"\n3.Insert(front)";
cout<<"\n4.Delete(rear)";
cout<<"\n5.Delete(front)";
cout<<"\n6.Display";
cout<<"\n7.Exit";
cout<<"\nEnter your choice :";
cin>>ch;
switch(ch)
{
      case 1:
            int i;
            cout<<"Enter the number of the elements :";
            cin>>n;
            for(i=0;i<n;i++)
            {
                  cout<<"Enter the element :";
                  cin>>x;
                  if(isFull(&q))
                  {
                        cout<<"The Queue is full";
                        exit(0);
                  }
                  enqueueRear(&q,x);
            }
            break;
      case 2:
            cout<<"Enter the element to be inserted :";
            cin>>x;
            if(isFull(&q))
            {
                  cout<<"Queue is full";
                  exit(0);
            }
            enqueueRear(&q,x);
            break;
      case 3:
            cout<<"Enter the element to be inserted :";
            cin>>x;
            if(isFull(&q))
            {
                  cout<<"Queue is full";
                  exit(0);
            }
            enqueueFront(&q,x);
            break;
      case 4:
            if(isEmpty(&q))
            {
                  cout<<"Queue is empty";
                  exit(0);
            }
            x=dequeueRear(&q);
            cout<<"Element deleted";
            break;
      case 5:
            if(isEmpty(&q))
            {
```

```
                        cout<<"Queue is empty";
                        exit(0);
                    }
                    x=dequeueFront(&q);
                    cout<<"Element is deleted";
                    break;
               case 6:
                    display(&q);
                    break;
               case 7:
                    exit(0);
                    break;
          }
     }while(ch!=7);
     return 0;
}
```