/*A hospital maintains a patient queue using a singly linked list. Each node contains:

• Patient ID (int)

• Patient Name (String)

Patients are added at the end of the list as they arrive. Patients can leave the queue in three ways:

1. Front Deletion: A patient completes consultation and leaves (deletion from front).

2. Position-based Deletion: A patient is transferred to another department (delete at a given position).

3. End Deletion: A patient leaves without consultation (delete from end).

• Create a node class for Patient with ID and name.

• Implement the following operations:

o insertAtEnd() – to add a new patient.

o deleteFromFront() – when consultation is done.

o deleteAtPosition(pos) – transferred to another department.

o deleteFromEnd() – leaves without consultation.

• Display the patient queue after each operation.*/

```
import java.util.*;
class Node
{
   int id;
   String name;
   Node next;
}
public class july5ht
{
   Node head;

   void insertAtEnd(String value,int id)
   {
      Node newNode=new Node();
      newNode.id=id;
      newNode.name=value;
```

```
    newNode.next=null;
    if(head==null)
    {
       head=newNode;
    }
    else
    {
       Node temp=head;
       while(temp.next!=null)
       {
          temp=temp.next;

       }
       temp.next=newNode;

    }
}
void deleteAtFront()
{
    if (head==null)
    {
       System.out.println("****No patient found******");
       return;

    }
    System.out.println("Patient who treated:"+"Id:"+head.id+"  Name:"+head.name);
    head=head.next;


}
void deleteAtEnd()
```

```
{
   if(head==null)
   {
      System.out.println("*******No patient*******");
      return;
   }
   if(head.next==null)
   {
      System.out.println("Patient treated:"+"ID:"+head.id+"  Name:"+head.name);
      head=null;
      return;
   }
   Node temp=head;
   while(temp.next.next!=null)
   {
      temp=temp.next;
   }
   System.out.println("Patient treated:"+"ID:"+temp.next.id+"  Name:"+temp.next.name);
   temp.next=null;
}
void deleteAtPosition(int position)
{
   if(position<1 || head==null)
   {
      System.out.println("Invalid position");
      return;
   }
   if(position==1)
   {
      System.out.println("Patient deleted:"+"ID:"+head.id+"  Name:"+head.name);
      head=head.next;
```

```
        return;
    }
    Node temp=head;
    for (int i=1;i<(position-1)&&temp!=null;i++)
    {
        temp=temp.next;

    }
    if(temp==null || temp.next==null)
    {
        System.out.println("Patient is out of range");

    }
    System.out.println("Patient treated:"+"ID:"+temp.next.id+"  Name:"+temp.next.name);
    temp.next=temp.next.next;
}
void display()
{
    Node temp=head;
    if(temp==null)
    {
        System.out.println("No patient is there");
        return;

    }
    while(temp!=null)
    {
        System.out.println(temp.id+":"+temp.name+"->");
        temp=temp.next;
    }
    System.out.println("Last patient is reached");
```

```
}
public static void main(String[]args)
{
    july5ht pl=new july5ht();
    Scanner sc=new Scanner(System.in);
    int choice;

    do
    {
        System.out.println("\n1.Book appointment\n2.See consulted patient\n3.Transfer to another department\n4.Leave without consultation\n5.Patient list\n6.Exit");
        System.out.println("Enter the choice you want to view");
        choice=sc.nextInt();
        sc.nextLine();
        switch(choice)
        {
        case 1:

            System.out.println("Enter the patient id:");
            int id=sc.nextInt();
            sc.nextLine();
             System.out.println("Enter the name of the patient who want to book appointment");
            String patient_name=sc.nextLine();
            pl.insertAtEnd(patient_name,id);
            break;
        case 2:
            System.out.println("See the patient name who is consulted ");
            pl.deleteAtFront();
            System.out.println("The patient is consulted----view patient list now");
            break;
```

```
        case 3:
            System.out.println("Enter the patient token number who has been transfered to another department");
            int position=sc.nextInt();
            sc.nextLine();
            pl.deleteAtPosition(position);
            break;
        case 4:
            System.out.println("See the patient who leaves without consultation");
            pl.deleteAtEnd();
            System.out.println("Sorry to hear that the patient left without consultation----view patient list");
            break;
        case 5:
            System.out.println("View patient list");
            pl.display();
            break;
        case 6:
            System.out.println("---Exiting----");
            break;
        }

    }while(choice!=6);
  }

  }
```