//TASK 3

class Queue

{

private int arr[];

private int front;

private int rear;

private int capacity;

private int count;

Queue(int size)

{

arr=new int[size];

capacity=size;

front=0;

rear=-1;

count=0;

}

public void dequeue()

{

if(isEmpty())

{

System.*out*.println("Queue is underflow and cannot dequeued");

}

else

{

System.*out*.println("Dequeuing: "+arr[front]);

front=(front+1) % capacity;

count--;

}

}

public void enqueue(int item)

{

if(isFull())

{

System.*out*.println("Queue is full it is overflow");

return;

}

else

{

System.*out*.println("Enqueuing:"+item);

rear=(rear+1) % capacity;

arr[rear]=item;

count++;

}

}

public int peek()

{

if(isEmpty())

{

System.*out*.println("Queue is Empty cannot have peek");

return -1;

}

return arr[front];

}

public int size()

{

return count;

}

public boolean isEmpty()

{

return count==0;

}

public boolean isFull()

{

return count==capacity;

}

}

public class Main

{

public static void main(String args[])

{

Task4 q=new Task4(5);

q.*enqueue*(1);

q.*enqueue*(2);

q.*enqueue*(3);

System.*out*.println("Front element is: "+ q.*peek*());

q.*dequeue*();

System.*out*.println("Front element is: "+q.*peek*());

System.*out*.println("Queue size is:"+q.size());

q.*dequeue*();

q.*dequeue*();

if(q.*isEmpty*())

{

System.*out*.println("Queue is Empty");

}

else

{

System.*out*.println("Queue is not Empty");

}

}

}