### Stack using two queues

Implement a Stack using two queues**q1** and**q2**.

**Example 1:**

**Input:**

push(2)

push(3)

pop()

push(4)

pop()

**Output:** 3 4

**Explanation:**

push(2) the stack will be {2}

push(3) the stack will be {2 3}

pop() poped element will be 3 the

  stack will be {2}

push(4) the stack will be {2 4}

pop()   poped element will be 4

**Example 2:**

**Input:**

push(2)

pop()

pop()

push(3)

**Output:** 2 -1

**Constraints:**  
1 <=Number of queries <= 100  
1 <= values of the stack <= 100

**Company Tags**

[**Accolite**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Accolite) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**OYO Rooms**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=OYO%20Rooms)[**Snapdea l**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Snapdeal)[**D-E-Shaw**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=D-E-Shaw) [**Oracle**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Oracle) [**Adobe**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Adobe) [**Cisco**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Cisco) [**Grofers**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Grofers) [**Coupon Dunia**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=CouponDunia) [**Kritikal Solutions**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Kritikal%20Solutions)

**Topic Tags**

[**Stack**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Stack) [**Queue**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Queue) [**Design-Pattern**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Design-Pattern) [**Data Structures**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Data%20Structures)

Java code

import java.util.\*;

class StackUsingQueues

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while(t>0)

{

Queues g = new Queues();

int q = sc.nextInt();

while(q>0)

{

int QueryType = sc.nextInt();

if(QueryType == 1)

{

int a = sc.nextInt();

g.push(a);

}

else if(QueryType == 2)

System.out.print(g.pop()+" ");

q--;

}

System.out.println();

t--;

}

}

}

class Queues

{

Queue<Integer> q1 = new LinkedList<Integer>();

Queue<Integer> q2 = new LinkedList<Integer>();

//Function to push an element into stack using two queues.

void push(int a)

{

q1.add(a);

}

//Function to pop an element from stack using two queues.

int pop()

{

if(q1.isEmpty())

return -1;

if(q1.size()==1){

return q1.poll();

}

int i=0;

while(i<=q1.size()-2){

q2.add(q1.poll());

}

int n=q1.poll();

while(!q2.isEmpty()){

q1.add(q2.poll());

}

return n;

}

}