Implement a Queue using two stack**s1** and**s2**.

**Example 1:**

**Input:**

enqueue(2)

enqueue(3)

dequeue()

enqueue(4)

dequeue()

**Output:** 2 3

**Explanation:**

enqueue(2) the queue will be {2}

enqueue(3) the queue will be {3 2}

dequeue() the poped element will be 2

the stack will be {3}

enqueue(4) the stack will be {4 3}

dequeue() the poped element will be 3.

**Example 2:**

**Input:**

enqueue(2)

dequeue()

dequeue()

**Output:** 2 -1

//{ Driver Code Starts

import java.util.\*;

class Solution

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while(t>0)

{

Queue g = new Queue();

int q = sc.nextInt();

while(q>0)

{

int QueryType = sc.nextInt();

if(QueryType == 1)

{

int a = sc.nextInt();

g.enqueue(a);

}

else if(QueryType == 2)

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

q--;

}

System.out.println();

t--;

}

}

}

// } Driver Code Ends

class Queue

{

Stack<Integer> input = new Stack<Integer>();

Stack<Integer> output = new Stack<Integer>();

/\*The method pop which return the element poped out of the stack\*/

int dequeue()

{

if(input.isEmpty())

return -1;

while(!input.isEmpty()){

output.push(input.pop());

}

int temp=output.pop();

while(!output.isEmpty())

input.push(output.pop());

return temp;

}

/\* The method push to push element into the stack \*/

void enqueue(int x)

{

input.push(x);

}

}