**Ticket Counter**

**Easy**

**N** people from 1 to N are standing in the queue at a movie ticket counter. It is a weird counter, as it distributes tickets to the **first K** people and then the **last K** people and again first K people and so on, once a person gets a ticket moves **out** of the queue. The task is to find the **last** person to get the ticket.

**Example 1**

**Input:**  
N = 9  
K = 3  
**Output:**  
6  
**Explanation:**  
Starting queue will like {1, 2, 3, 4, 5, 6, 7, 8, 9}. After the first distribution queue will look like {4, 5, 6, 7, 8, 9}. And after the second distribution queue will look like {4, 5, 6}. The last person to get the ticket will be 6.

**Example 2**

**Input:**  
N = 5  
K = 1  
**Output:**  
3  
**Explanation:**  
Queue start as {1, 2, 3, 4, 5} -> {2, 3, 4, 5} -> {2, 3, 4} -> {3, 4} -> {3}  
Last person to get ticket will be 3.

**Expected Time Complexity:** O(N)  
**Expected Space Complexity:**O(N)  
Try to solve it using O(1) space complexity.

**Constraints:**  
1 <= K<= N <= 105

**Topic Tags**

[**Greedy**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Greedy) [**Queue**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Queue)

//{ Driver Code Starts

//Initial Template for Java

import java.io.\*;

import java.util.\*;

class CodingMaxima{

public static void main(String args[]) throws IOException {

BufferedReader in=new BufferedReader(new InputStreamReader(System.in));

PrintWriter out=new PrintWriter(System.out);

int t = Integer.parseInt(in.readLine());

while (t-- > 0) {

String s[]=in.readLine().trim().split(" ");

int n=Integer.parseInt(s[0]);

int k=Integer.parseInt(s[1]);

Solution ob=new Solution();

out.println(ob.distributeTicket(n,k));

}

out.close();

}

}

// } Driver Code Ends

//User function Template for Java

class Solution {

public static int distributeTicket(int N,int K)

{

int k=K;

int low=1, high=N;

while(low< high){

k=K;

while(k-->0 && low<high){

low++;

}

k=K;

while(k-->0 && low <high){

high--;

}

}

return high;

}

}