Problem I. Is It Rated?

Time limit:	3 seconds
Memory limit:	512 megabytes

The popular improv website *Interpretation Impetus* hosts regular improv contests and maintains a rating of the best performers. However, since improv can often go horribly wrong, the website is notorious for declaring improv contests *unrated.* It now holds a wager before each improv contest where the participants try to predict whether it will be rated or unrated, and they are now more popular than the improv itself.

Izzy and n other participants take part in each wager. First, they each make their prediction, expressed as `1` ("rated") or `0` ("unrated"). Izzy always goes last, so she knows the predictions of the other participants when making her own. Then, the actual competition takes place and it is declared either rated or unrated.

You need to write a program that will interactively play as Izzy. There will be m wagers held in 2021, and Izzy's goal is to have at most $1.3 \cdot b + 100$ wrong predictions after all those wagers, where b is the *smallest* number of wrong predictions that any other wager participant will have after all those wagers.

The number b is not known in advance. Izzy also knows nothing about the other participants — they might somehow always guess correctly, or their predictions might be correlated. Izzy's predictions, though, do not affect the predictions of the other participants and the decision on the contest being rated or not — in other words, in each test case, your program always receives the same inputs, no matter what it outputs.

Interaction Protocol

First, a solution must read two integers n ($1 \leq n \leq 1000$) and m ($1 \leq m \leq 10\,000$). Then, the solution must process m wagers. For each of them, the solution must first read a string consisting of n `0`s and `1`s, in which the i-th character denotes the guess of the i-th participant. Then, the solution must print Izzy's guess as `0` or `1`. Don't forget to flush the output after printing it! Then, the solution must read the actual outcome, also as `0` or `1`, and then proceed to the next wager, if this wasn't the last one.

Your solution will be considered correct if it makes at most $1.3 \cdot b + 100$ mistakes, where b is the smallest number of mistakes made by any other participant. Note that if a solution outputs anything except `0` or `1` for a wager, it will be considered incorrect even if it made no other mistakes.

There are 200 test cases in this problem.

Example

standard input	standard output
3 4	
000	
	0
1	
100	
	0
1	
001	
	1
0	
111	
	1
1	

Note

In the example, the participants made 1, 2, and 3 mistakes respectively, therefore $b = 1$ (the smallest of these numbers). Izzy made 3 mistakes, which were not more than $1.3 \cdot b + 100 = 101.3$, so these outputs are good enough to pass this test case (as are any other valid outputs).