Problem M
MofK's Mysterious Money Making Machine

This is an interactive problem.

MofK has invented an arcade game machine. Inside MofK's machine, there are n secret coins. The states of the coins are represented by a binary string s of length n, where $s_i = 0$ denotes that the i-th coin is tails, and $s_i = 1$ denotes that i-th coin is heads. Since this is a secret coin sequence, no one except MofK knows how the string s looks like.

The player can interact with the machine by inserting coins into it, and if the player can guess the state of the secret coin sequence inside the machine, the machine will reward the player with all the coins inside, including the n secret coins. An interaction with the machine consists of the following steps:

1. First, the player inserts n coins into the machine. The states of these inserted coins are represented by a binary string x, where $x_i = 0$ denotes that the i-th coin is tails, and $x_i = 1$ denotes that the i-th coin is heads.
2. Then, the machine calculates the following values:
   - $cnt_{\texttt{correct}}$ is the number of positions i where $s_i = x_i$.
   - $cnt_{\texttt{incorrect}}$ is the number of positions i where $s_i \neq x_i$.
3. Next, the player presses one of two buttons: the "`?`" button to ask, or the "`!`" button to guess the secret coin sequence:
4. After either of the two buttons is pressed, the machine reacts as below:

If "`?`" button is pressed	The machine will release a **heads** coin if $cnt_{\texttt{correct}} < cnt_{\texttt{incorrect}}$. The machine will release a **tails** coin if $cnt_{\texttt{correct}} > cnt_{\texttt{incorrect}}$.
If "`!`" button is pressed	The machine will release all the coins inside if $cnt_{\texttt{correct}} = n$. The machine will be locked if $cnt_{\texttt{correct}} < n$. The player will not be able to interact with the machine anymore.

You have coins to interact with the machine at most 1024 times, including both asking and guessing interactions. Can you come up with a strategy to retrieve all the coins from MofK's arcade game machine?

Interaction

In this problem, the jury plays the role of MofK's arcade game machine, and your program plays the role of the player of the arcade game machine.

Before the interaction process begins, you need to read an odd integer n ($1 \leq n \leq 999$) – the number of secret coins that MofK has placed into the machine. Then you can interact with the Jury's program as follows:

If you decide to insert n coins into the machine and then press the "`?`" button to ask the machine:

- First, print a single line "`?` x" (without quotation mark), where x is a binary string of length n that describes the state of the coins you will insert into the machine in the above specified format.
- Then, you need to read a number c ($c \in \{0, 1\}$) that describes the state of the coin that the machine releases. $c = 1$ means that the machine releases a **heads** coin, and $c = 0$ means that the machine releases a **tails** coin.

If you decide to insert a coin into the machine and press the "`!`" button to guess the coin sequence:

- First, print a single line "`!` x" (without quotation marks), where x is a binary string of length n that describes the state of the coins you will insert into the machine in the above specified format.
- Then, the interaction process stops immediately. Your program must terminate.

The state sequence of the secret coins s in the machine is fixed before the interaction process begins and will not change during the interaction.

Example Interaction

Standard Input	Standard Output	Explanation
3		There are $n = 3$ secret coins in the machine.
	? 000	You ask the machine with the coin state sequence 000.
1		The machine releases a **heads** coin, which means the number of positions where the hidden string s and 000 match is **less than** the number of positions where they do not match.
	? 001	You ask the machine with the coin state sequence 001.
0		The machine releases a **tails** coin, which means the number of positions where the hidden string s and 001 match is **greater than** the number of positions where they do not match.
	? 010	You ask the machine with the coin state sequence 010.
1		The machine releases a **heads** coin, which means the number of positions where the hidden string s and 010 match is **less than** the number of positions where they do not match.
	? 011	You ask the machine with the coin state sequence 011.
1		The machine releases a **heads** coin, which means the number of positions where the hidden string s and 011 match is **less than** the number of positions where they do not match.
	? 100	You ask the machine with the coin state sequence 100.
0		The machine releases a **tails** coin, which means the number of positions where the hidden string s and 100 match is **greater than** the number of positions where they do not match.
	? 101	You ask the machine with the coin state sequence 101.
0		The machine releases a **tails** coin, which means the number of positions where the hidden string s and 101 match is **greater than** the number of positions where they do not match.
	? 110	You ask the machine with the coin state sequence 110.
1		The machine releases a **heads** coin, which means the number of positions where the hidden string s and 110 match is **less than** the number of positions where they do not match.
	? 111	You ask the machine with the coin state sequence 111.
0		The machine releases a **tails** coin, which means the number of positions where the hidden string s and 111 match is **greater than** the number of positions where they do not match.
	! 101	You guess the secret coin states as *heads up, tails up, heads up* respectively. Your program then terminates.

Note

After printing a line do not forget to output end of line and flush the output. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- see documentation for other languages.