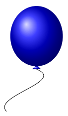
Problem E. Boxes and Flags

Ballon: 
Time limit: 1 seconds
Memory limit: 512 megabytes

This is an interactive problem.
An interactive problem require communication between submission and the interactor via their stdin/stdout. Submission's stdout is interactor stdin and vice versa.

Alice and **Bob** are partaking in a puzzle game with a prize. In this game, there will be 2 players **Alice** and **Bob** and there will be a Gamemaster. There are two types of room: the Waiting Room and the Playing Room. Initially, they are both in the Waiting Room, and the Gamemaster is in the Playing Room.

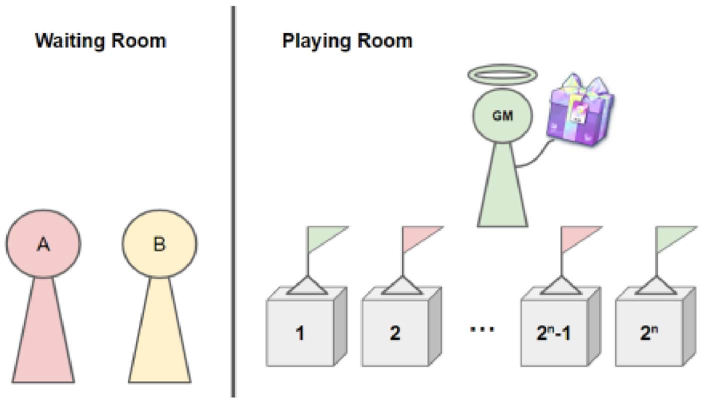


Figure 1: Initial setting of the game

The game goes as follow:

1. In the Playing Room, there are 2^n identical empty boxes, numbered from $1, 2, \ldots$ and so forth. On top of each box there is a flag with only two colors (of Green and Red). Initially, all flags have their color randomed.

2. Firstly, **Alice** will come in, see the state of all flags, see the Gamemaster put the reward in a random box. After that, **Alice** must pick one flag and change its color (from Green to Red, or Red to Green).

3. Then, **Alice** will leave the room without seeing **Bob**. And **Bob** will enter the Playing Room. He must somehow with only 1 guess, correctly guess the box which contains the prize, just by the colors of the flag.

After the Gamemaster finishes explaining the rules to **Alice** and **Bob**, they all think this game is a scam. But the Gamemaster keeps persisting that it is possible, you just need a good strategy beforehand. Would you help them discuss the strategy as well?

Also, the Game Master really does not want you to win by luck, so instead of 1 Playing Room, there will be T consecutive Playing Rooms. **Alice** after finish with Playing Room i will move out of that room, and wait for **Bob** to finish his turn at Room i. After that, if **Bob** guesses correctly, they will repeat the game in the next room, the only difference that could occur are the flags and the position of the prize.

How To Interact:

Please check the Example section for better understanding. From the problem statement, for better clarity, the interaction is divided into 4 phases:

1. **Phase 1**: The interactor will prepare 2^n boxes and flags, and output n for the participant to read. $(1 \le n \le 6)$.

2. **Phase 2**: Participant then prints 2^n lines. $i - th$ line contains a mathematical expression (details are below this section [1]). The $i - th$ expression describes to **Bob** that IF evaluation of that expression is true, the prize should be in $i - th$ box. (aka. This is the phase where **Alice** gives **Bob** the strategy before heading into the Playing Rooms. The strategy is only given once, and is used by **Bob** throughout T playing rooms).

3. **Phase 3**: The interactor output T, announce the amount of Playing Rooms there are. $(1 \le T \le 100)$

4. **Phase 4**: For each room i:

a. Firstly, the interactor output on the same line 2^n values of 0 and 1, $j - th$ value describe the colors of the $j - th$ flag. They are separated by a space.

b. Next, the interactor will output another line containing one integer, which is the position that the Gamemaster hid the prize. (aka. **Alice** enters Playing Room $i - th$ and observes where the gamemaster put the prize). Now, the participant must output an integer value *cpos*, represents the position of the flag that **Alice** must change its color. $(1 \le cpos \le 2^n)$

Finally, the interactor will check if the color change will help locate the prize according to the strategies given at the start of the interaction. If it's correct, the interactor will output the word **OK** on a single line. Otherwise, it will output **BAD** on a single line, at this point you should terminate your program to avoid undefined behaviors.

[1] For constraints of the mathematical expression:

- Length of expression should not exceed 2506 characters.

- For operands, you can use integers. At every point of calculation, absolute value must not exceed 10^18 to avoid undefined behaviors.

- Other than integer operands, you can also use $[x]$ with x is the index of the box $(1 \le x \le 2^n)$ to obtain the value of the flag on top of $x - th$ box. If the color is Green, $[x]$ is 0. If the color is Red $[x]$ is 1.

- For operators, you can use addition $+$, subtraction $-$, multiplication $*$, integer division $/$, and modulo $\%$. For the division behavior or precedence order, it is the same as defined in language C++.

- You can use parenthesis ( ).

- You can use comparisons such as $=$ (equality), $>$ (greater), $<$ (lesser). Which will output result 1 if the comparison was truthful, otherwise 0.

- 0 is considered a boolean value of false. Other integers are true. For every case, there should exist only 1 true and 2^{n-1} false from given expressions. Failure to evaluate expression (such as divided by zero) also leads to a WA.

- It can be proven that a solution exists within the given constraints. Attempts to break constraints will lead to undefined behaviors, WA is high probability.

For example, in your strategies your $25th$ line looks like this:

```
...
([1] + [2] + [3]) * 2 = 6
...
```

Then **Bob** will check the $25 - th$ box if the first, second, and the third flag are all Red.

Examples

Example 1

Interaction (Left is Participant, Right is Interactor)	**Notes**
1	Interactor output n
[1] - [2] $>$ 0 [2] $>$ [1]	Participant prints 2^n strategies
2	Interactor announce T
	Room#1
0 0 1	Flags state and prize position
1	**Alice** flips 1-st position. State 00 is now 10.
OK	Strategies evaluated as: - $1st$ expression: [1]>[2] is true - $2nd$ expression: [2]>[1] is false Then **Bob** should check the 1st box since the 1st expression is the only truthy one, and it's the correct answer.
	Room #2
1 1 2	State and new hidden position for prize for room 2.
1	State 11 is now 01
OK	Strategies evaluated as: - $1st$ expression: [1]>[2] is false - $2nd$ expression: [2]>[1] is true Then **Bob** should check the 2nd box. It is correct.

Example 2

Interaction (Left is Participant, Right is Interactor)	**Notes**
1	Phase 1, the interactor prepares 2^n boxes and outputs n for participants.
([2] = 0) ([2] = 1)	Participant prints 2^n strategies
2	Interactor announce T
	Room #1
0 0 1	- Interactor first print state of all flags. In this case, all have Green colors. - On $2nd$ line, it outputs the position of the prize, which is in the $1st$ box
1	- **Alice** flips colors of the $1st$ flag. State 00 is now 10
OK	- According to the strategies, 10 has $2nd$ flag is a 0, so the prize should be in the $1st$ box - Correct. Interactor outputs OK
	Room #2
0 1 2	State and new hidden position for prize for room 2.
1	State 01 is now 11
OK	$2nd$ cell has value 1, it matches the $2nd$ expression.

Example 3

Interaction (Left is Participant, Right is Interactor)	**Notes**
2	Interactor output n
(([1]+[2]) = 1) (([1]+[3]) = 1) (([1]+[4]) = 1) (([2]+[4]) = 1)	Participant prints 2^n strategies
3	Interactor announce T
Room #1	
1 1 1 1 2	
3	**Alice** flips the color of the $3rd$ flag. State 1111 is now 1101.
OK	According to strategies evaluation the $2nd$ strategy is truthy, it claims that the prize is in $2nd$ box and it was correct. Interactor outputs OK
Room #2	
0 0 0 0 1	State and new hidden position for prize for room 2.
1	State 0000 is now 1000
BAD	This state has more than 1 truthful expression, so **Bob** does not know which to open.
Terminal	