

Simple Maths Equations

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

We hid an equation consisting of exactly 8 characters. A character can be a digit or an operator (+, -, *, /) or an equal sign. You want to guess the equation according to the following rules.

- Each guess is a correct equation consisting of 8 characters.
- In your guesses (and the correct answer), the equations must only contain the following characters:
0 1 2 3 4 5 6 7 8 9 + - * / or =.
- Each guess (and the correct answer) must contain exactly one =.
- Each guess (and the correct answer) must not contain +, -, * or / to the right of the =.
- Standard order of operations applies, so calculate * and / before + and -.
- In each guess (and the correct answer) + **cannot** be used as a positive sign, - **cannot** be used as a negative sign.
- It is OK to have non-integer values in the middle of calculations (for example $9 / 12 * 4 = 3$ is acceptable).
- Leading zeroes are not allowed in the final answer and are invalid guesses.

Following are examples of valid and invalid guesses:

- $4*5+4=24$, $12+34=46$, $9/12*4=3$, $1-2+4=3$ are valid guesses.
- $1*1=1$ is too short, so it is an invalid guess.
- $1+2*2=69$ is not correct, so it is an invalid guess.
- $73=73=73$ contains two =, so it is an invalid guess.
- $37=13+24$ contains a + to the right of =, so it is an invalid guess.
- $13+-2=11$ is an invalid guess as - is used as a negative sign.
- $13-+2=11$ is an invalid guess as + is used as a positive sign.
- $-5*-4=20$ is an invalid guess as - is used as a negative sign.
- $73-74=-1$ is an invalid guess as - is used as a negative sign.

After making a guess, you will receive feedback: a string consisting of 8 characters. Each character is either G, P or B. Let *ans* be the correct answer string, *guess* be your guess string, *feedback* be the feedback string. For each distinct character *c*,

- Let cnt_0 be the number of positions *i* such that $ans_i = guess_i = c$.
- Let cnt_1 be the number of positions *i* such that $ans_i = c$ but $guess_i \neq c$.
- Let cnt_2 be the number of positions *i* such that $ans_i \neq c$ but $guess_i = c$.
- For all the cnt_0 positions *i* such that $ans_i = guess_i = c$, $feedback_i = G$.

- Among all cnt_2 positions i such that $ans_i \neq c$ but $guess_i = c$, choose $\min(cnt_1, cnt_2)$ of them **arbitrarily**, such positions have $feedback_i = P$.
- Other positions with $ans_i \neq c$ but $guess_i = c$ have $feedback_i = B$.

Guessing too many times is not fun, so we restricted on the number of guesses you can make. Refer to the scoring section for details.

Interaction Protocol

To make a guess, output your guess equation on a single line (remember to print a new line and flush the output).

Then, we will return the feedback string. If the feedback string is **GGGGGGGG** (which means you have correctly guessed the equation), you should **return immediately**, else you may receive an unexpected verdict (WA/TLE/ILE).

Note that if your guess is invalid, you may receive an unexpected verdict (WA/TLE/ILE).

Scoring

Q is the limit on the number of guesses you can make. All guesses, including the final guess, is counted towards the value of Q .

If $Q \leq 6$, you score 100 points.

If $Q > 1006$, you score 0 points.

Else, your score is $\left\lfloor 100 \cdot \left(1 - \sqrt{1 - \frac{1006-Q}{1000}}\right) \right\rfloor$ points.