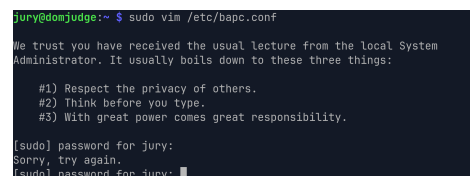


E Expected Error

Time limit: 1s

You are typing your password at a `sudo` prompt, but suddenly, one of your fingers slipped onto a wrong key. Because the terminal hides the characters that you type, you are uncertain whether you have typed an extra character. To finish typing your password, you consider three possible strategies.



```
jury@demjudge:~$ sudo vim /etc/bapc.conf
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for jury:
Sorry, try again.
[sudo] password for jury: 
```

1. *Continue* to type the rest of your password.
2. Press *backspace* to delete the last character and type the rest of your password.
3. *Restart* typing your password from scratch.

To determine the optimal strategy, make use of the following typing speed assumptions.

- Typing any character of your password takes 0.1 seconds.
- Pressing backspace or submitting your password also takes 0.1 seconds.
- To restart typing your password, you delete all characters, which takes 0.3 seconds.
- If you submit the wrong password, it takes an additional 0.3 seconds to realize and start typing your password at a new, empty prompt.

You are given the number of characters in your password n , the number of correctly typed characters k before your finger slipped, and a probability of $p\%$ indicating the likelihood that you pressed a wrong key and ended up with $k + 1$ characters. Assuming you make no further errors, determine which strategy yields the lowest expected time to finish typing your password.

Input

The input consists of:

- One line with three integers n , k , and p ($1 \leq n \leq 1000$, $1 \leq k \leq n$, $0 \leq p \leq 100$), the number of characters in your password, the number of characters you correctly typed before your finger slipped, and the probability percentage that you pressed a wrong key.

Output

Output one of the strings “continue”, “backspace”, or “restart” indicating the optimal strategy. It is guaranteed that the expected time of the optimal strategy is at least one millisecond shorter than the other strategies.

Sample Input 1

10 8 20

Sample Output 1

continue

Sample Input 2

10 8 80

Sample Output 2

backspace

Sample Input 3

10 2 50

Sample Output 3

restart

Sample Input 4

10 4 55

Sample Output 4

restart