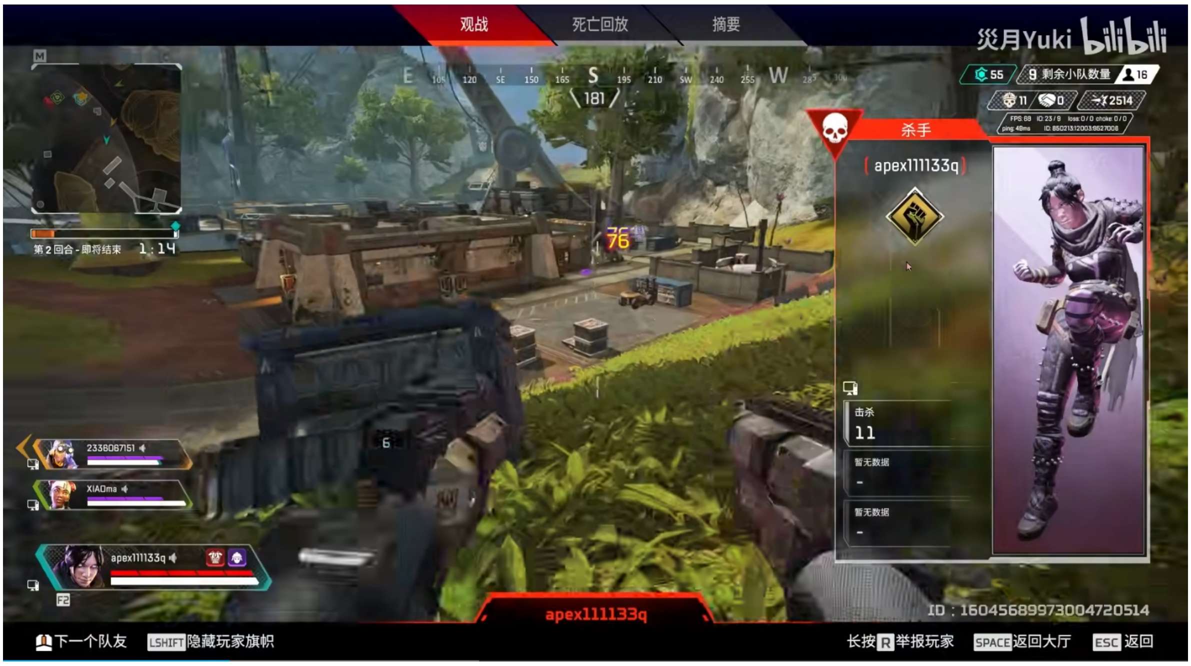
Gua!

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are playing the game Apex Legends, which is a Battle Royale game famous for, uh, a large number of cheaters, a.k.a. Mr. Gua. (‘Gua’ means cheating in Chinese)

One type of cheating is changing the parameters of guns. For example, someone can even hold two guns parallelly like the following screenshot, which makes you wonder if she were the Lucian in the game League of Legends.



Bilibili Video: `BV1V3411j7QW`

After losing a huge amount of rating points in the ranking games, you are angry because of the arrogance of Mr. Gua and the ignorance of the game development company, Respawn. You decide to write a complaint letter to teach Respawn detecting cheaters.

Someone killed you with a gun that each bullet will deal **at maximum** B damages. This gun can shoot R Rounds Per Minute (RPM), meaning that after shooting a bullet the gun must rest for at least $1/R$ **minute**. There is a special case $R = 0$, which means that the player is without weapon or out of ammo, so no damage can be dealt.

In the replay, the player dealt D damages in S **seconds**, counting from the first bullet to the last bullet. If we say that this guy must be cheating by changing the parameters of guns, it means that he/she dealt more damage than he/she could. Here, we consider the maximum possible damage by only B, R, S, ignoring the other aspects in the games like magazine size, reload time and bugs.

We take the gun Wingman used in the above screenshot as an example. This gun can deal 38 damage when hit Fortified legends on the body, and the cheater dealt $152 = 38 \times 4$ damages to the enemy in just 1 second. Since the gun's rate of fire is only 156 RPM, it's only possible to wait $156/60 = 2.6$ cool-downs in a second, which means shoot 3 bullets at maximum. Because $4 > 3$, we can ensure that this guy is really cheating.

Write a demo program to determine whether a player must be cheating by B, R, D, S.

Input

The first line contains a single integer $T(1 \le T \le 10^3)$, denoting the number of test cases.

For each test case, there are four integers $B, R, D, S(0 \le B, R, D, S \le 2000)$ in a single line, denoting maximum single bullet damage, rate of fire in RPM, damage dealt by the player, the time window from the first bullet to the last bullet in seconds.

Output

For each test print a single line,

- if the player must be cheating, print "`gua!`" (without quotes),
- otherwise print "`ok`" (without quotes).

Example

standard input	standard output

7
38 156 152 1
280 25 280 0
99 51 9 10
0 0 1 1
99 0 1 1
11 1080 209 1
11 1080 210 1
	gua!
	ok
	ok
	gua!
	gua!
	ok
	gua!

Note

The first sample case is explained above.

For the second sample case, it's possible to use Kraber to do a head shot, in which we count the time window as 0. We are not sure whether it is really cheating, although it's possible that the player use some aiming aid cheating.

The real Bronze player plays normally in the third case, and it's a reminder that the player can deal 'partial' damage using shotgun like Peacekeeper. As long as the total amount of damage does not exceed the maximum possible damage, we can't determine whether it is really cheating or not.

The fourth and fifth are the cases when one player has no weapon or has out of ammo. Therefore, no damage can be dealt without help of the 'high technique'.