Have you ever wanted to prove that you knew something without revealing any information about what you know?

That's the idea behind zero knowledge proofs.

I have a 3SAT Solver and a bunch of 3SAT problems. I won't always use my solver to get to a valid assignment of literals, though, so it's your job to check if I'm bluffing or not.

But I don't want to just give you the 3SAT formula and my entire list of assignments, because then you'd have the answer to the problem, and everyone would know for sure whether I'm bluffing or not. So, I, wanting to be discreet, will only give you one clause at a time for you to verify.

You will be able to ask me one at a time for clauses you want to check, and I will gladly provide you with the truth values for the literals in those clauses. However, I will shuffle the order of the clauses each time, so you really can't tell anything about what my solution might be.

Your goal is to always have at most a $1-p$ chance of being wrong. I won't let you ask for too many clauses, though, because my discreet self doesn't want you to have too much certainty in being right.

You will have T trials, each with their own number of literals (N) and clauses (M). Once a trial has starts, I will confirm with you that I have permuted the clauses, and then you can ask me to reveal any one of them. I'll send you back the values for that clause for you to verify. You can then decide to either (1) ask for another clause or (2) tell me whether you think I'm bluffing or not. If you choose (1), I will permute the clauses again and the process repeats. If you choose (2), I will take note of whether you were right or not, and then proceed with another trial.

At the end of T trials, I will calculate your success rate. If it is at least p, I will give you the flag. If not, then tough luck (literally).

--

THE VALUE OF p IS 0.75 FOR THIS CHALLENGE.

Interaction details:

1. The first line of input will be the number of trial runs, T.
2. After that, the first trial will start.
3. The next line will be N.
4. The following line will be M.
5. Then I will print "permuted", and then prompt ": " so you can enter a clause number (from 1 to M).
6. I will then print an array with the values of the clauses according to my variable assignments.
7. Then you will be prompted to indicate whether you will like to continue asking for choices:
- "next" leads to another clause
- "true" will indicate you believe I have a valid solution
- "false" will indicate you believe I do not have a valid solution
8. "next" will repeat from step 5. "true" and "false" will log your correctness and repeat from step 3 until T trials have passed.
* Note that if "next" is called too many times, I will kick you out.
9. At the end of T trials, I will output the flag if your success rate is at least *p*.

Sample interaction: (blue text is user input)

```
110
109
411
permuted
: 200
[True, True, True]
next
permuted
: 410
[False, False, True]
next
permuted
: 310
[False, True, True]
next
permuted
: 45
[False, True, False]
next
permuted
: 45
```

```
[False, True, False]
next
permuted
: 45
[False, True, True]
next
permuted
: 45
[True, False, False]
next
permuted
: 67
[True, False, True]
next
permuted
: 101
[False, True, False]
true
105
408
permuted
: 404
[True, True, False]
next
permuted


[... etc.]
```