

Problem 7: The CALICasinO

5+6=11 Points

Problem ID: `drawcheck`

Rank: 2+3

Introduction

It's 2024, and with a generous donation from an alumnus and Codeforces grandmaster, CALICO has built a breathtaking casino in Vegas: the CALICasinO! However, the donation didn't cover the installation of slot machines, or the copyright for traditional casino games such as blackjack or poker, so the only game in the CALICasinO is this game called `drawcheck` that some problem writer with an obsession with interactive problems wrote in 2023.

After your mom found out about your gambling addiction and your use of her credit card, she kicked you out of the house. Instead of getting a real job, you decide to gamble your life savings away at the CALICasinO for a chance to win big! However, to actually make money in `drawcheck`, you must perform better in *drawcheck* than the other players *on average*.

Well, what are you waiting for? Your gambling addiction isn't going to fund itself!

Problem Statement

This is an interactive problem! The judge has a deck of **500** cards numbered from 1 to **500**. At the start of each test case, the judge removes **a card of their choice (may not be random)** without telling you. Find the number on this card by performing *draw* queries and *check* queries on the remaining deck while keeping the average number of queries per test case below a certain threshold.

Each interaction, you can perform one of the two following queries. You may query as many times as necessary until you find the missing number.

- *Draw* a hand of cards:
 - The judge shuffles the deck (**random permutation**) and draws **17** cards from the top.
 - The judge shows you the numbers on these cards before returning them to the deck.
- *Check* for a specific card number:
 - The judge searches the deck to see if there is a card with a number of your choice.

- If a card with your number is present, the judge will respond with `PRESENT`.
- If a card with your number is absent, the judge will respond with `ABSENT` and you pass the test case.

Each test file has exactly **500** tests. At the end of each test file, if the average number of queries per test case is no more than the threshold (specified in constraints below) you will receive a `CORRECT` verdict. Otherwise, you will receive a `WRONG_ANSWER` verdict.

Interaction Format

This is an interactive problem! Unlike regular problems, your program and the judge will run simultaneously. Please see the [contest guide](#) for more information. Please flush your buffer as instructed by [this post](#) when you output, or use our template code that handles it for you. If you run into technical issues with interaction, please let us know with a clarification request!

Begin by reading a single line containing an integer **T** denoting the number of test cases that follow. For each test case:

1. Start by making *draw* and *check* queries in any order until you find the missing number.
 - a. To make a *draw* query:
 - i. First, output a single line containing the word `draw`
 - ii. Then, read a single line containing **17** space-separated integers $x_1 x_2 \dots x_{17}$ denoting the numbers on the cards that were drawn.
 - b. To make a *check* query:
 - i. First, output a single line in the following format:
`check i`
where i is an integer between 1 and **500** denoting the number you want to check.
 - ii. Then, read a single line containing `PRESENT` or `ABSENT`.
2. When you *check* a number and find it's `ABSENT`, this test case ends and the next begins. After all test cases, read a single line containing your verdict: `CORRECT` or `WRONG_ANSWER`.

If at any point your program deviates from the interaction format (e.g. invalid query type, checking a number not between 1 and **500**), the judge will send `WRONG_ANSWER`. If your program reads `WRONG_ANSWER` at any point, you should exit to receive a wrong answer verdict.

Constraints

Time Limit: **4 seconds**

T = 500 – Note that **T** is equal to exactly **500**, not less than or equal to.

The deck will always begin with **500** cards.

When drawing, **17** cards are drawn at a time.

Main Test Set

The average number of queries per test case must be no more than **260**.

Bonus Test Set

The average number of queries per test case must be no more than **95**.

Sample Interaction

The line spacing here is to emphasize the order in which interaction takes place only. Do not expect or output blank lines between each line of interaction. **Also, note that the value for T has been adjusted for sample purposes.** Real test cases will always have T at 500.

Sample Input	Sample Output
	2
388 237 483 153 210 419 ...	draw
419 299 17 388 93 27 153 ...	draw
PRESENT	check 7
ABSENT	check 42
PRESENT	check 1
PRESENT	check 2
ABSENT	check 69
CORRECT	

Sample Explanations

The judge begins by outputting 2, the number of test cases.

For test case #1, the program begins by sending a *draw* query, and receives the numbers 388 237 483 153 210 419 ... (note that there should be 17 numbers, but this is omitted due to space) from the judge. The program then sends another *draw* query. Note that numbers may be repeated. Next, the program *checks* 7, and finds it is present. Finally, the program *checks* 42 and finds it is absent, completing the first test case.

For test case #2, the program *checks* 1 and 2, both of which are present. The program then *checks* 69, which is absent, completing the second test case.

Now that all test cases have completed, we see that the program took 4 queries to complete the first test case and 3 queries to complete the second test case. Thus, the program used $(4 + 3) / 2 = 3.5$ queries on average. As this value is below the required average of **260** (and also below **95** for the bonus), the judge gives a `CORRECT` verdict. Upon receiving this line, the program exits.