Misdelivered letters

Dan's career as a postman isn't going very well. Yesterday, he was tasked with delivering electricity bills to each of the N houses in Brno. However, he somehow managed to mix up the order of the bills yet again and ended up delivering most of them to the wrong houses. Nevertheless, he is sure that he delivered at least one letter correctly. Can you help him find it?

The houses in Brno are numbered from 1 to N. There are N bills, each addressed to a different house. Every house received exactly one bill. Dan can go to any house, ring the bell and ask the residents which houses letter they ended up with. Dan doesn't want to disturb too many people, so he would like to minimize the number of houses he checks.

Communication

This task is interactive.

Your program will be executed on a number of test cases. As part of one test case, and therefore during one execution of your program, you will have to solve T different problems. The value of N is the same across all problems, but the assignment of letters to houses will (most probably) be different in each problem.

You can either implement the required protocol yourself, or use the provided template. You can find the template in CMS as a task attachment called `template.cpp`.

The protocol

First, your program should read a line from the standard input containing two space-separated integers N and T — the number of houses in Brno, and the number of problems you need to solve.

After that, your program can perform queries about the first problem.

Each query consists of a single containing `Q`, a space, and a number i. The number i is the number of the house that Dan should go to, and therefore $1 \leq i \leq N$. Your program should print such a line to the standard output and flush it afterwards. After that, you should read a single line containing a number j — the house number on the the bill received by house i.

When you're done performing queries, you can answer by printing a single line containing `A`, a space, and a number i to standard output. The number i should be the number of any house that received a letter addressed to it, and therefore $1 \leq i \leq N$. After you print the answer and flush the standard output, you should read a single line which will contain either `W` or `C`.

If the line contains `C`, then your answer was correct. In that case the program may start performing queries about the next problem, or, if this was your T-th answer, exit.

If the line contains `W`, then your answer wasn't correct. In that case the program should exit immediately.

Please note that exiting after `W` is important for getting the right feedback from CMS. If your program continues running, it may crash or receive some other unsuccessful verdict.

The template

If you are using the implementation of the protocol in `template.cpp`, you need to implement the function `int find_correctly_delivered_letter()`. This function will be called once for each problem. It should return an integer between 1 and N, the number of any house that received a letter addressed to it.

You can use the function `int ask_house(int house_id)` to find such a house. Its only argument is an integer i, the number of the house that Dan should go to. It returns the house number on the the bill received by house i.

You can also use the global variable `N` containing N from the statement. You may do any initialization in the `main` function after the first call to `scanf`.

Constraints

- $N = 1000$
- $T = 1000$
- At least one letter was delivered to the correct house.

Grading

This task uses partial scoring.

If your answer on any problem is wrong, you will receive zero points. Otherwise, the number of points for a given test case will be determined based on the average number of queries per problem. Let Q denote the average number of queries used by your program across all problems, rounded down to one decimal place. The score on a given test case will be computed according to the following rules:

- If $Q \geq 2\,400$ you will receive 0 points (wrong answer).
- If $400 < Q < 2\,400$, the number of points will be determined by the following formula:

$$100 \cdot \left(1 - \frac{Q - 400}{2\,000}\right)$$

- If $Q \leq 400$, you will receive 100 points.

Your solution will be graded on several test cases. The total number of points you'll receive will be the minimum number of points across all test cases.

The judge for the task is not adaptive, which means that the assignment of letters to houses has been determined prior to running your program.

The judging system will provide feedback for every test case. This feedback will include the value of Q of your solution on each test case where you reach a non-zero score.

Sample interaction

Here is a sample interaction with the grader (using values N and T smaller than the ones in real test data). Do not forget to flush the output after every line.

Your input	Your output
`10 2`	
	`Q 1`
`2`	
	`Q 2`
`3`	
	`Q 4`
`4`	
	`A 4`
`C`	
	`A 4`
`W`	

The program solved the first problem correctly, as house 4 indeed received the bill addressed to house 4.

These were numbers on the bills delivered to houses 1 to N:

```
2 3 1 4 5 6 9 8 7 10
```

It didn't solve the second one, as the correct solution was 5 (which the program couldn't have known, as it didn't perform any queries). Even if there were another problem, the program should immediately terminate.

This time houses 1 to N ended up with these bills:

```
10 9 8 7 5 6 4 3 2 1
```