

# Codeforces Marathon Round 1

## A2. Online Exam

time limit per test: 5 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Felix the Robot is preparing for a probability theory exam. Unfortunately, during the semester, he took a course of the belles-lettres instead of studying the subject, so now he does not know the answer to any of the upcoming exam's questions. One thing is for sure: Felix needs help!

The exam for robots is an online event. It consists of  $n = 5000$  questions each of which has only two possible answers: "yes" and "no". A robot can attempt to pass the exam at most  $x = 100$  times. The questions, their order and right answers don't change from one attempt to another. Once the exam starts, Felix will get just a few seconds for all his attempts, and he can't learn the right answers so fast.

The robot will try to pass the exam in the following way. First, Felix fixes the answers for all questions. The result is a string of  $n$  bits: answers for the first, second, ...,  $n$ -th questions. In this string, 0 means that the answer is "no", and 1 is for "yes". Then he answers the questions according to this string, spending one attempt. After that, Felix can fix another string of  $n$  bits and make another attempt, and so on until there are no more attempts.

In the online system for exam, the following optimization is implemented: if at some moment of time, the examinee already got  $k = 2000$  answers wrong, the attempt is immediately terminated with the corresponding message. For Felix, this means that the remaining bits in the string he fixed are ignored. If there were strictly less than  $k$  wrong answers for all  $n$  questions, the exam is considered passed.

The result of an attempt is a number from  $k$  to  $n$  inclusive: the number of questions after which the attempt was terminated. If the exam is passed, this number is considered to be  $n + 1$ . The exam result is the highest result among all attempts. If there were no attempts, the exam result is zero.

Your task is to write a program which will determine the bit strings for all attempts Felix makes. After each attempt, your program will get its result immediately. Help Felix get the highest exam result you can!

### Interaction Protocol

Your solution can make from 0 to  $x$  attempts inclusive. To make an attempt, print a string to the standard output. The string must consist of exactly  $n$  binary digits without spaces and end with a newline character.

To prevent output buffering, after printing a string, insert a command to flush the buffer: for example, it can be `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, `flush (output)` in Pascal or `sys.stdout.flush ()` in Python.

After each attempt you make, you can immediately read its result from the standard input. The result is an integer from  $k$  to  $n + 1$  inclusive, followed by a newline character.

### Scoring System

A test is defined by a string of  $n$  binary digits: the right answers to  $n$  questions. This string is kept secret from the solution. Each test is evaluated separately.

If a solution followed the interaction protocol and terminated correctly on a test, it gets a score of  $\max(0, S - 4000)$  where  $S$  is the exam result. Otherwise, the solution gets zero score for the test.

### Testing

Your solution will be checked on sets of tests generated in advance. Each test is created using a pseudo-random number generator. You can consider that the answers are uniformly distributed (the probabilities of digits 0 and 1 are the same) and mutually independent (the probabilities of all  $2^n$  possible strings are the same). A solution gets the score which is the sum of its score on all the tests.

During the main phase of the contest, there are two ways to send a solution for checking.

- The first one is to check on examples. There are 10 example tests which are also available for local testing. As soon as the solution is checked, you can see reports for all examples by clicking on the submission result.
- The second way is to check on preliminary tests. There are 100 preliminary tests which are generated in advance but kept secret. The score for preliminary tests (but not for example tests) is used in the preliminary scoreboard. This score does not affect the final results, but nevertheless allows to roughly compare a solution with others.

After the main phase ends, for each participant, the system chooses the final solution:

- consider all solutions sent for **preliminary testing**;
- choose the ones which got a total score strictly greater than zero;
- define the final solution as the one of chosen solutions which has **the latest** submission time.

Note that the solutions sent only to be checked on examples are not considered when choosing the final solution.

During the final testing, all final solutions will be checked on the same large set of a large number ( $\approx 1000$ ) of final tests. The score for final tests determines the final scoreboard. The winner is the contestant whose solution gets the highest total score. In case two or more participants have equal total score, the contestants with such score tie for the same place.

A package for local development is available on GitHub at the following address: <https://github.com/GassaFM/online-exam>. You can download sources or the latest release: <https://github.com/GassaFM/online-exam/releases>.

### Example

To have an example which fits into the problem statement, let  $n = 10$ ,  $k = 2$ , and  $x = 3$  (recall that in the real problem,  $n = 5000$ ,  $k = 2000$ , and  $x = 100$ , so this example is not a correct test for the problem). Let the right answers be defined by the string `1010001111`. Before any attempts are made, the exam result is zero.

Consider a solution making three attempts.

Let the first attempt be defined by the string `0100100100`. The result of this attempt is the number 2: the first wrong answer is the answer to the first question, and the second is to the second question. The exam result at this moment is 2.

Let the second attempt be defined by the string `1010101010`. The result of this attempt is the number 8: the first wrong answer is the answer to the fifth question, and the second is to the eighth question. The exam result at this moment is 8.

Let the second attempt be defined by the string `1001011001`. The result of this attempt is the number 4: the first wrong answer is the answer to the third question, and the second is to the fourth question. The exam result at this moment is still 8.

As  $x = 3$  in our example, further attempts are impossible, so if the solution terminates correctly, the exam result is 8.

Now consider another solution making two attempts.

Let the first attempt be defined by the string `1010001110`. Its result is the number 11: the first and only wrong answer is the answer to the tenth question,  $k = 2$ , so the exam is considered passed.

Let the first attempt be defined by the string `0000011111`. Its result is the number 3: the first wrong answer is the answer to the first question, and the second one is to the third question.

If the solution terminates correctly after the above two attempts, the exam result is 11.