

# Гипотеза Коллатца и случайные увеличения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

*Это интерактивная задача.*

Определим функцию Коллатца  $\text{collatz}(x)$ , действующую на целых числах, так: если  $x$  чётное, то  $\text{collatz}(x) = \frac{x}{2}$ , а иначе  $\text{collatz}(x) = 3x + 1$ . Знаменитая гипотеза Коллатца гласит, что, если начать с любого целого положительного  $x_0$  и построить последовательность  $x_1 = \text{collatz}(x_0)$ ,  $\dots$ ,  $x_{i+1} = \text{collatz}(x_i)$ , то в этой последовательности обязательно встретится единица.

Невероятная сложность, из-за которой гипотеза до сих пор ни доказана, ни опровергнута, заключается в очень хаотичном поведении последовательности  $\{x_i\}$  до момента, когда она достигает единицы. Даже для очень небольших чисел единица может достигаться довольно поздно: так, если стартовать с  $x_0 = 9$ , получим

$9 \rightarrow 28 \rightarrow 14 \rightarrow 7 \rightarrow 22 \rightarrow 11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ ,

а если начать с  $x_0 = 27$ , то получим единицу лишь после 111 применений  $\text{collatz}(x)$ !

Вы сидите перед автоматом, у которого есть экран и две кнопки: красная и синяя. На экране отображается целое положительное число (гарантируется, что это случайное число от 2 до  $10^7$  включительно), и его вам надо превратить в единицу. Красная кнопка называется **collatz**, она заменяет число  $x$  на экране на  $\text{collatz}(x)$ . Синяя кнопка называется **random**, она заменяет  $x$  на случайное целое число от  $3x + 1$  до  $6x$  включительно. Нажатия на кнопки не бесплатны: после каждого нажатия кнопки, когда на экране отображается число  $x_{i+1}$ , требуется внести в автомат столько жетонов, сколько цифр в десятичной записи числа  $x_{i+1}$ . Например, в вышеприведённом процессе, стартующем с девятки, нужно заплатить за все цифры чисел 28, 14, 7,  $\dots$ , 2, 1; на это уйдёт 32 жетона.

Если нажимать только на красную кнопку, вы достигнете единицы, потратив в среднем 707 жетонов. Несмотря на то, что синяя кнопка всегда увеличивает число на экране (причём значительно), можно ускорить процесс получения единицы, если в правильные моменты нажимать синюю кнопку! Ваша задача — написать программу, которая тратит на одно число в среднем не более 600 жетонов. Чтобы проверка меньше зависела от случайности, мы в каждом тесте дадим программе  $t \leq 50$  случайных стартовых чисел, и вам надо будет превратить их все в единицы за  $600 \cdot 50 = 30\,000$  жетонов.

## Протокол взаимодействия

Для начала прочитайте строку, содержащую целое число  $t$  — число стартовых чисел ( $1 \leq t \leq 50$ ). У жюри есть  $t$  стартовых целых чисел  $x_0$ , выбранных случайно, независимо и равномерно распределённых на отрезке  $[2; 10^7]$ , и вам требуется по очереди превратить их все в единицу.

В начале  $i$ -го раунда прочитайте строку, содержащую одно число  $x_0$  — очередное стартовое число ( $2 \leq x_0 \leq 10^7$ ). Далее вам требуется преобразовывать число. После очередного преобразования, если на экране число  $x_j$ , выведите в отдельной строке либо слово «**collatz**» (буквы могут быть в любом регистре), если вы хотите заменить число на  $x_{j+1} = \text{collatz}(x_j)$ , либо слово «**random**» (буквы могут быть в любом регистре), если вы хотите заменить число на  $x_{j+1}$ , равное случайному целому числу, равномерно распределённому на отрезке  $[3x_j + 1; 6x_j]$ . После вывода каждой строки не забывайте очищать буфер вывода — иначе, скорее всего, вы получите ошибку **Idleness Limit Exceeded**:

- `std::cout.flush()` в C++;
- `fflush(stdout)` в C;
- `stdout.flush()` в Python и D;
- `flush(output)` в Pascal и Delphi;
- `System.out.flush()` в Java;
- для помощи по остальным языкам обратитесь к документации.

После этого прочитайте строку, содержащую одно целое число  $x$ . Возможны следующие варианты:

- если  $x = 0$ , то у вас кончились жетоны;
- если  $x = 1$ , то  $x_{j+1} = 1$  — вы успешно завершили раунд, переходите к следующему раунду (для чего начните со считывания следующего стартового числа  $x_0$ );
- если  $x > 1$ , то  $x_{j+1} = x$  — число на экране изменилось после нажатия кнопки, продолжайте выводить строки «collatz» и «random». Обратите внимание: в процессе взаимодействия число  $x$  может превысить и  $2^{64}$ , и  $2^{128}$ .

Также жюри передаст в вашу программу число 0, если вы нарушите формат вывода и напечатаете любую строку, кроме «collatz» и «random». Прочитав 0, ваша программа должна сразу завершить работу, чтобы получить вердикт **Wrong Answer**. В противном случае вердикт может быть каким угодно (кроме **Accepted**).

Считав в  $t$ -й раз единицу, завершите работу программы, чтобы получить вердикт **Accepted**.

## Пример

стандартный ввод	стандартный вывод
2	
4	Collatz
2	cOLLaTZ
1	
3	RANDOM
16	collatz
8	COLlatz
4	cOLLATz
2	CoLlAtZ
1	

## Замечание

Пример — единственный тест, где стартовые числа  $x_0$  выбраны не случайно, а вручную.