# True и false, break и continue

# 1984

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Вы — сотрудник Министерства Правды в тоталитарной сверхдержаве Океании, которая то воюет, то заключает мир с двумя другими тоталитарными сверхдержавами, Евразией и Остазией. Ваша задача — информационное обеспечение войны в соответствии с указаниями правительства.

Изначально идёт война с Евразией, мир с Остазией.

## Формат ввода

На первой строке указывается натуральное число N — количество команд от правительства.

Далее следует N команд. Каждая команда вводится на отдельной строке и представляет собой одну из трёх строк:  
«С кем война?» означает, что нужно вывести название государства, с которым идёт война;  
«С кем мир?» — что нужно вывести название государства, с которым мир;  
«Меняем» означает, что с тем государством, с которым шла война, заключён мир, а с тем государством, с которым был мир, началась война; выводить по этой команде ничего не нужно.

## Формат вывода

Несколько строк «Евразия» и «Остазия» в соответствии с поступившими командами.

## Пример

| Ввод | Вывод |
| --- | --- |
| 6  С кем война?  С кем мир?  Меняем  С кем война?  Меняем  С кем война? | Евразия  Остазия  Остазия  Евразия |

# Найди кота (break)

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите программу, которая находит кота. Пользователь вводит сначала количество строк, потом сами строки. Если хотя бы в одной введённой строке нашлось сочетание букв «Кот» или «кот», программа выводит «МЯУ», иначе программа выводит «НЕТ».

При обнаружении кота цикл должен немедленно прерываться командой break.

## Формат ввода

В первой строке записано число n.  
Далее следует n строк.

## Формат вывода

Напечатайте нужное сообщение в зависимости от того, есть во введенных строчках кот или нет.

## Пример

| Ввод | Вывод |
| --- | --- |
| 4  Как устроен типичный фрукт:  кожура;  мякоть;  косточки. | МЯУ |

## Примечания

Данная задача дополнительно проверяется преподавателем.

# Найди кота — 2 (break)

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите программу, которая находит кота.

Пользователь вводит строки до тех пор, пока он не введёт «СТОП». Программа выводит номер строки, на которой впервые был упомянут кот (наличие в строке сочетаний букв «Кот» или «кот»), или -1 (минус один), если кот не был упомянут.

При обнаружении кота цикл должен немедленно прерываться командой break.

## Формат ввода

Несколько строк.  
Сигнал остановки — строка «СТОП».

## Формат вывода

Одно число — номер первой строчки, в которой появился кот, или -1, если кота нет.

## Пример

| Ввод | Вывод |
| --- | --- |
| Как устроен типичный фрукт:  кожура;  мякоть;  косточки.  СТОП | 3 |

## Примечания

Обратите внимание, что программа должна прекратить свою работу после считывания строки «мякоть;», а остальные строчки, поданные на вход программы, считывать не нужно.  
Данная задача дополнительно проверяется преподавателем.

# Найди кота — 3

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите программу, которая находит кота.

Пользователь вводит строки до тех пор, пока он не введёт «СТОП». Программа выводит, во-первых, общее количество строк, в которых были упомянуты коты, во-вторых, номер строки, на которой впервые был упомянут кот (в том же смысле, что и в предыдущих задачах), или -1 (минус один), если кот не был упомянут.

## Формат ввода

Несколько строк.  
Сигнал остановки — строка «СТОП».

## Формат вывода

Всегда два числа — общее количество строк с котом и номер первой такой строки (или -1, если такой строки нет). Числа должны быть разделены пробелом.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| Как устроен типичный фрукт:  кожура;  мякоть;  косточки.  СТОП | 1 3 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| Животное такое.  С усами, хвостом.  Мяукать умеет.  Мышей ловит.  (Если настроение подходящее.)  Кто бы это мог быть?  СТОП | 0 -1 |

# Биржевой робот

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Напишите робота для автоматической торговли акциями на бирже.

Вводится цена акций в первый, второй и т. д. дни, ноль — сигнал остановки. Возможно, сначала цена уменьшается. В какой-то момент цена начинает расти. Мы покупаем акции в первый день, когда их цена превышает цену в предыдущий день. После этого в какой-то момент цена акций начинает уменьшаться. Мы продаём акции в первый же день, как только их цена становится меньше цены в предыдущий день. Возможно, после этого цены как-то ещё меняются.

Гарантируется, что среди введенных цен точно будет день, когда цена начнет расти, а после день, когда цена начнет падать. После продажи акций робот больше не участвует в торгах на бирже.

Программа должна вывести цену акций, по которой мы их купили, цену, по которой продали, и выгоду с каждой акции (возможно, отрицательную).

Не следует пользоваться советами этого робота в реальной жизни.

## Формат ввода

Несколько целых чисел — цены акций в последовательные дни.  
Число 0 — сигнал прекращения ввода цен.

## Формат вывода

Три целых числа — цена покупки, цена продажи, выгода.

## Пример

| Ввод | Вывод |
| --- | --- |
| 32  30  31  34  38  37  39  0 | 31 37 6 |

# Школа танцев

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

В танце очень важно чувствовать ритм музыки. Напишите программу, которая проверяет, правильно ли ученик отсчитывает: раз, два, три, четыре, раз, два, три, четыре... При этом считается, что у учителя есть некоторый ограниченный запас терпения, и после определённого числа ошибок он заканчивает занятие.

## Формат ввода

На первой строке вводится натуральное число n — запас терпения учителя.

Далее следуют строки с отсчётами.

## Формат вывода

Пока в вводе повторяются по очереди строки «раз», «два», «три», «четыре», программа не выводит ничего. Как только выводится что-то иное, чем ожидалось, программа выводит: «Правильных отсчётов было <количество правильных отсчётов>, но теперь вы ошиблись.» (Количество правильных отсчётов после этого считается заново, и сами отсчёты снова должны начинаться с «раз».) Если это произошло в n-й раз, выводится «На сегодня хватит.», и дальнейший ввод игнорируется.

## Пример

| Ввод | Вывод |
| --- | --- |
| 2  раз  два  три  четыре  раз  двыа  раз  два  три  три | Правильных отсчётов было 5, но теперь вы ошиблись.  Правильных отсчётов было 3, но теперь вы ошиблись.  На сегодня хватит. |

# Проверка блокчейна

Kлассная работа

макс. 3 балл.

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 64Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Блокчейн (blockchain) переводится как «цепочка блоков». Это способ хранения данных, защищённый от подделки, используемый, в частности, криптовалютой биткоин.

Блокчейн действительно представляет собой последовательность блоков. Каждый блок представляет собой некоторую полезную информацию (в частности, в случае **биткоина** это список транзакций за определённый период времени — кто кому когда сколько денег передал), снабжённую случайным числом и некоторыми служебными данными, в том числе **хэшем** — числом, которое по определённой формуле зависит от остальной части блока и хэша предыдущего блока.

Хэш должен быть меньше определённого числа. При этом формула, по которой вычисляется хэш, устроена так, что невозможно получить достаточно маленький хэш иначе, чем перебирая различные значения случайного числа. Поэтому если злоумышленник решит подделать блокчейн (и, допустим, вставить в его середину блок с записью о том, что все люди передают ему все свои деньги), то ему придётся подобрать новое случайное число в новое поддельном блоке и всех последующих (ведь хэш каждого следующего блока зависит от хэша предыдущего), что потребует невозможно больших вычислительных мощностей. Поэтому блокчейн в целом защищён от подобных атак.

Напишите программу, которая проводит проверку правильности хэшей в модельном блокчейне с простой хэш-функцией.

Блок bn с номером n  представлен одним числом

bn = hn + rn×256 + mn×2562,

где mn, – полезная информация, представленная натуральным числом,

rn – случайное число от 0 до 255,

hn – хеш (целое число от 0 до 255), вычисляемое по формуле

hn = 37×(mn+rn+hn-1) (по модулю 256).

При вычислении хэша начального блока h0 вместо хэша предыдущего блока берётся ноль. При этом требуется, чтобы хэш hn был меньше 100.

## Формат ввода

На первой строке вводится натуральное число N — количество блоков.  
Далее следуют N чисел bn, каждое на отдельной строке.

## Формат вывода

Следует вывести номер первого блока, у которого неправильный хэш (не меньше 100 или не совпадает с вычисленным по указанной в условии формуле), или -1, если все хэши в блокчейне правильные. Нумерация блоков идёт с нуля, т. е. они имеют номера от 0 до N-1.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 5  6122802  14406496  15230209  2541121  1758741 | -1 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| 5  1865535  13479687  16689153  1839958  5214020 | 3 |