# Строки. Индексация

# Игра в города: один раунд

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

При игре в города игроки по очереди называют названия городов (или, возможно, любые другие слова) так, чтобы первая буква каждого нового слова совпадала с последней буквой предыдущего.

Напишите программу, которая считывает подряд две строки, после чего выводит «ВЕРНО», если последний символ первой строки совпадает с первым символом второй, и «НЕВЕРНО» в противном случае.

## Формат ввода

Два слова — каждое в своей строке.

## Формат вывода

Одно сообщение — «ВЕРНО» или «НЕВЕРНО».

## Пример

| Ввод | Вывод |
| --- | --- |
| париж  житомир | ВЕРНО |

# Игра в города

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

Пользователь (или несколько пользователей за одним компьютером) вводит слова. Начиная со второго введённого слова, программа проверяет, совпадает ли первая буква свежевведённого слова с последней буквой предыдущего. Если да, то программа работает дальше (считывает очередное слово). Если нет — выводит последнее на этот момент введённое слово и завершает работу.

## Формат ввода

Вводятся несколько строк подряд.

## Формат вывода

Выводится одна строка.

## Пример

| Ввод | Вывод |
| --- | --- |
| новгород  дублин  новгород  дублин  тула | тула |

## Примечания

В данной задаче гарантируется, что будет введено как минимум два слова.

# Какая-то там буква

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

Напишите программу, которая считывает сообщение, затем номер. После этого программа выводит букву из сообщения с таким номером, причём считается, что номера букв отсчитываются с единицы.  
Если введённое число не является правильным номером буквы, вывести «ОШИБКА».

## Формат ввода

В первой строке записано сообщение, во второй — номер буквы.

## Формат вывода

Одна буква или сообщение «ОШИБКА».

### Пример 1

| Ввод | Вывод |
| --- | --- |
| привет  2 | р |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| привет  -100 | ОШИБКА |

# Цезарь его знает

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

Как известно, Цезарь тоже пользовался шифрованием сообщений, причем у него был свой способ. Сначала выбирается шаг шифрования (число), а затем все буквы послания заменяются на буквы, отстоящие от них в алфавите на шаг шифрования. Например, при шаге шифрования 3 (таким чаще всего пользовался Цезарь), буква А заменяется на букву Г, буква Б – на букву Д.

Обратите внимание, что алфавит «зациклен», то есть при сдвиге буквы Я на шаг 3 получится буква В.

Напишите программу, которая будет зашифровывать послание с помощью шифра Цезаря с заданным шагом шифрования.

## Формат ввода

Две строки. Первая содержит шаг шифрования, вторая – послание.

## Формат вывода

Строка с зашифрованным посланием.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 3  АБВ | ГДЕ |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| 5  На дворе трава, на траве дрова! | Те йзухк чхезе, те чхезк йхузе! |

## Примечания

Символы русского алфавита расположены в стандартной для Python таблице кодировки подряд, то есть номера, выдаваемые функцией ord(symbol), идут подряд. Буква «ё» идёт в таблице кодировки отдельно от основного алфавита. При решении задачи считайте, что буквы «ё» в русском алфавите нет.

# Ползём вниз

Kлассная работа

макс. 3 балл.

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

Изобразите извилистый спуск улитки по стене по заданной траектории (см. примеры).

## Формат ввода

Вводится одна строка.

Начальный символ этой строки — символ рисования пути улитки.

Далее следует последовательность символов “<”, “>” и “V”, которые означают движение улитки, соответственно, влево, вправо и вниз.

Гарантируется, что по любой горизонтали движение происходит только в одну сторону (иными словами, между идущими в любом порядке “<” и “>” всегда есть хотя бы один “V”) и что путь не пройдёт левее начального положения.

## Формат вывода

Выводятся несколько строк, состоящих из символов рисования пути и пробелов. Верхний левый угол соответствует начальному положению улитки. Каждый последующий символ входной строки означает, что нужно дорисовать символ рисования пути левее, правее или ниже предыдущего. Пробелы используются для отступов.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| .>>>VV<<V | ....  .  ...  . |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| @VVV>>>>>>>>V<<VVVV<<<VV>>> | @  @  @  @@@@@@@@@  @@@  @  @  @  @@@@  @  @@@@ |