**Методы списков и строк**

# Комментарии в программе

Классная работа

макс. 1 балл.

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

Напишите программу, которая удаляет из программы на языке Python комментарии, а также символы пустого пространства на конце строк. Считайте, что символ решётки всегда означает начало комментария (проверьте себя: в каком случае это не так?).

## Формат ввода

На первой строке вводится символ решётки и сразу же натуральное число N — количество строк в программе, не считая эту первую.  
Далее следует N строк.

## Формат вывода

Нужно вывести те же строки, но удалить комментарии и символы пустого пространства в конце строк. Пустую строку вместо первой строки ввода выводить не надо.

## Пример

| Ввод | Вывод |
| --- | --- |
| #2  name = input()  print('Привет,', name) # здороваемся | name = input()  print('Привет,', name) |

# Польский калькулятор

Kлассная работа

макс. 2 балл.

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

Напишите программу, которая производит вычисление выражения, записанного в обратной польской нотации (ОПН).

В ОПН нет ни скобок, ни приоритета операторов («умножение раньше сложения»). Чтобы прочитать выражение, записанное в ОПН, нужно просматривать выражение строго последовательно. Вводимые значения последовательно добавляются в стек, а когда встречается символ операции, то из стека извлекаются последние положенные туда значения, с ними проделывается эта операция, и результат возвращается в стек. Если для операции важен порядок значений, с которыми она производится, то первым идёт число, лежавшее в стеке глубже; в частности, если операция — вычитание, то из предпоследнего числа в стеке вычитается последнее, а не наоборот. Изначально стек пустой, в результате полного вычисления выражения в нём должно остаться одно значение — результат вычислений.

Первый пример следует читать так: в стек последовательно добавляются значения 7, 2, 3, затем встречаем знак операции \*, поэтому значения 2 и 3 извлекаются, перемножаются, результат (6) кладётся обратно в стек, и следующий знак - извлекает из стека два оставшихся в нём значения 7 и 6, вычитает одно из другого и кладёт результат снова в стек. Выражение закончилось, в стеке одно число 1, это и есть результат вычисления.

## Формат ввода

Вводится одна строка, содержащая разделённые пробелами целые числа и знаки операций +, -, \*, вместе составляющие корректное выражение в обратной польской нотации.

## Формат вывода

Выводится одно целое число — результат вычисления выражения.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 7 2 3 \* - | 1 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| 10 15 - 7 \* | -35 |

# Маленький частотный анализ

Kлассная работа

макс. 3 балл.

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

Частотный анализ — это подсчёт, какие символы чаще встречаются в тексте. Это важнейший инструмент взлома многих классических шифров — от шифра Цезаря до шифровальной машины «Энигма». Выполним простой частотный анализ: выясним, какой символ чаще всего встречается в данном тексте.

## Формат ввода

Вводится одна строка.

## Формат вывода

Выводится один символ в нижнем регистре — наиболее часто встречающийся во введённой строке, кроме пробела, без учёта регистра, если таких несколько — выводится первый по алфавиту.

## Пример

| Ввод | Вывод |
| --- | --- |
| Баобаб | б |

# Знаков без пробелов

Kлассная работа

макс. 1 балл.

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

Часто бывает нужно сосчитать, сколько в тексте печатаемых знаков. Напишите программу, которая это делает.

## Формат ввода

Вводится одна строка.

## Формат вывода

Выводится одно целое число — количество символов в этой строке, не являющихся ни пробелами, ни табуляциями.

## Пример

| Ввод | Вывод |
| --- | --- |
| Раз два три | 9 |

# Пристраиваемся в очередь

Kлассная работа

макс. 3 балл.

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

Ещё одна не очень достоверная модель очереди.

Изначально в очереди никого нет. Далее в каком-то порядке происходят события трёх типов:

* фраза «<фамилия> встал(а) в очередь.» означает, что пришёл новый человек и встал в конец очереди;
* фраза «Привет, <фамилия1>! <фамилия2> будет за тобой.» означает, что пришёл новый человек и встал в середину очереди за другим человеком, который там уже стоял;
* фраза «<фамилия>, хватит тут стоять, пошли отсюда.» означает, что кто-то покинул очередь.

Нужно определить, кто остался в очереди в конце концов и в каком порядке.

**Формат ввода**

На первой строке вводится натуральное число N — количество событий.

Далее следуют N строк описанного в условии формата.

Гарантируется, что фамилии людей все разные и что в середину очереди пристраиваются только за кем-то, кто там уже действительно стоит, и отзывают из очереди тоже только тех, кто там уже стоит.

Фамилии могут выглядеть как угодно, в том числе состоять из нескольких слов и включать любые знаки препинания, кроме восклицательного знака.

**Формат вывода**

Выводится список людей, оставшихся в очереди в конце концов.

**Пример**

| Ввод | Вывод |
| --- | --- |
| 5  Иванова встала в очередь.  Кузнецов встал в очередь.  Поливанов встал в очередь.  Привет, Кузнецов! Зорина будет за тобой.  Иванова, хватит тут стоять, пошли отсюда. | Кузнецов  Зорина  Поливанов |