

## 11. Файлы.

Во всех задачах ввод-вывод осуществляется только посредством чтения-записи файлов. Файл со входными данными называется `input.txt`, с выходными — `output.txt`.

Во входных данных всех задач можно рассчитывать на то, что **все** строки файла, включая последнюю, заканчиваются переносом строки (символом `'\n'`).

Словом во всех задачах называется последовательность прописных или строчных латинских букв, ограниченная слева и справа символом не-буквой или началом (концом) строки.

Если в файле есть русские буквы (и вообще любой символ с кодом, большим 127), то необходимо при открытии файла указывать кодировку:

```
data = open('input.txt', 'r', encoding='utf-8')
```

### Открытие файла

Для каждого файла, с которым необходимо производить операции ввода-вывода, нужно связать специальный объект - поток. Открытие файла осуществляется функцией `open`, которой нужно передать два параметра. Первый параметр (можно также использовать именованный параметр `file`) имеет значение типа `str`, в котором записано имя открываемого файла. Второй параметр (можно также использовать именованный параметр `mode`) — это значение типа `str`, которое равно `"r"`, если файл открывается для чтения данных (`read`), `"w"`, если на запись (`write`), при этом содержимое файла очищается, и `"a"` — для добавления данных в конец файла (`append`). Если второй параметр не задан, то считается, что файл открывается в режиме чтения.

Функция `open` возвращает ссылку на файловый объект, которую нужно записать в переменную, чтобы потом через данный объект использовать методы ввода-вывода. Например:

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
```

### Чтение данных из файла

Для файла, открытого на чтение данных, можно вызывать следующие методы, позволяющие читать данные из файла.

Метод `readline()` считывает одну строку из файла (до символа конца строки `'\n'`, возвращается считанная строка вместе с символом `'\n'`. Если считывание не было успешно (достигнут конец файла), то возвращается пустая строка. Для удаления символа `'\n'` из конца файла удобно использовать метод строки `rstrip()`. Например: `s = s.rstrip()`.

Метод `readlines()` считывает все строки из файла и возвращает список из всех считанных строк (одна строка — один элемент списка). При этом символы `\n` остаются в концах строк.

Метод `read()` считывает все содержимое из файла и возвращает строку, которая может содержать символы `\n`. Если методу `read` передать целочисленный параметр, то будет считано не более заданного количества символов.

Например, считывать файл побайтово можно при помощи метода `read(1)`.

### Вывод данных в файл

Данные выводятся в файл при помощи метода `write`, которому в качестве параметра передается одна строка. Этот метод не выводит символ конца строки `'\n'` (как это делает функция `print` при стандартном выводе), поэтому для перехода на новую строку в файле необходимо явно вывести символ `'\n'`.

Также можно выводить данные в файл при помощи функции `print`, если передать ей еще один именованный параметр `file`, равный ссылке на открытый файл. Например:

```
output = open('output.txt', 'w')
print(a, b, c, file = output)
```

### Заккрытие файла

После окончания работы с файлом необходимо закрыть его при помощи метода `close()`.

## Пример

Следующая программа считывает все содержимое файла `input.txt`, записывает его в переменную `s`, а затем выводит ее в файл `output.txt`.

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
s = input.read()
output.write(s)
input.close()
output.close()
```

Так можно читать файл построчно:

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
line = input.readline()
while line:
    output.write(line)
    line = input.readline()
input.close()
output.close()
```

Существует и другой способ обработать файл построчно, более удобный:

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
for line in input:
    output.write(line)
input.close()
output.close()
```

А вот аналогичная программа, но читающая данные уже посимвольно:

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
c = input.read(1)
while c:
    output.write(c)
    c = input.read(1)
input.close()
output.close()
```

A. Сумма нескольких чисел

Во входном файле записано некоторое количество чисел, каждое в отдельной строке. Выведите их сумму.

Input	Output
1	6
2	
3	

B. Поиск символа

Определите, есть ли во входном файле символ @. Выведите слово YES или NO.

Входной файл может быть очень большим, поэтому считывать файл нужно посимвольно.

Input	Output
Valid email: Bilbo.Baggins@bagend.hobbiton.shire.me	YES

Input	Output
Hello, world!	NO

C. Шифр Цезаря

Зашифруйте данный текстовый файл шифром Цезаря, при этом символы первой строки файла должны циклически сдвигаться на 1, второй строки — на 2, третьей строки — на 3 и т.д.

Символы, отличные от латинских букв, изменять не требуется.

В этой задаче удобно считывать файл построчно, шифруя каждую строку в отдельности.

Input	Output
Hello	Ifmmp
Hello	Jgnnq
Hello	Khoor
Hello	Lipps

D. Сумма чисел в файле - I

Дан файл, каждая строка которого может содержать одно или несколько целых чисел, разделенных одним или несколькими пробелами. Символов, отличных от цифр, пробелов и знаков переноса строки, в файле нет.

Вычислите сумму чисел в каждой строке и выведите эту сумму (для каждой строки выводится сумма чисел в этой строке).

В данной задаче удобно считывать данные построчно.

Input	Output
2 2	4
3 4	7
1 5	6

E. Сумма чисел в файле - II

В файле могут быть записаны десятичные цифры и все, что угодно. Числом назовём последовательность цифр, идущих подряд (т.е. число всегда неотрицательно).

Вычислите сумму всех чисел, записанных в файле. В данной задаче удобно считывать данные посимвольно.

Input	Output
123 aaa456 1x2y3 4 5 6	600

### Ф. Статистика по файлу

Дан файл. Определите сколько в нём букв (латинского алфавита), слов, строк. Под словом понимается последовательность букв латинского алфавита, ограниченная слева и справа символами-не-буквами или началом (концом) строки.

Выведите три найденных числа в формате, приведённом в примере.

Для экономии памяти читайте файл посимвольно, то есть не сохраняя целиком в памяти файл или отдельные его строки.

Input	Output
Beautiful is better than ugly. Explicit is better than implicit. Simple is better than complex. Complex is better than complicated.	Input file contains: 108 letters 20 words 4 lines

### Г. Максимальный балл по классам

В олимпиаде по информатике принимало участие несколько человек. Информация о результатах олимпиады записана в файле, каждая строка которого имеет вид: фамилия имя класс балл.

Фамилия и имя — текстовые строки, не содержащие пробелов. Класс — одно из трех чисел 9, 10, 11. Балл — целое число от 0 до 100.

Победителем олимпиады становится человек, набравший больше всех баллов. Победители определяются независимо по каждому классу.

Определите количество баллов, которое набрал победитель в каждом классе. Гарантируется, что в каждом классе был хотя бы один участник.

Выведите три числа: баллы победителя олимпиады по 9 классу, по 10 классу, по 11 классу.

В этой задаче файл необходимо считывать построчно, не сохраняя содержимое файла в памяти целиком.

Решения, содержащие разбор трёх случаев вида

```
if form == 9:  
  
elif form == 10:  
  
else:
```

приниматься не будут.

Input	Output
Иванов Сергей 9 90 Сергеев Петр 10 91 Петров Василий 11 92 Васильев Иван 9 93	93 91 92

### Н. Средний балл по классам

В условиях задачи Г определите и выведите средние баллы участников олимпиады в 9 классе, в 10 классе, в 11 классе.

Input	Output
Иванов Сергей 9 90 Сергеев Петр 10 91 Петров Василий 11 92 Васильев Иван 9 93	91.5 91.0 92.0

I. *Количество победителей по классам*

В условиях задачи G определите количество школьников, ставших победителями в каждом классе. Победителями объявляются все, кто набрал наибольшее число баллов по данному классу. Гарантируется, что в каждом классе был хотя бы один участник.

Выведите три числа: количество победителей олимпиады по 9 классу, по 10 классу, по 11 классу.

Input	Output
Иванов Сергей 9 80 Сергеев Петр 10 80 Петров Василий 11 81 Васильев Андрей 9 81 Андреев Александр 10 80 Александров Роман 9 81 Романов Иван 11 80	2 2 1

J. *Победитель олимпиады*

Зачёт в олимпиаде проводится без деления на классы. Выведите фамилию и имя победителя олимпиады. Если таких несколько — выведите только их количество.

Input	Output
Иванов Сергей 9 90 Сергеев Пётр 10 95 Петров Иван 11 85	Сергеев Пётр

Input	Output
Иванов Сергей 9 90 Сергеев Пётр 10 85 Петров Иван 11 90	2

K. *Максимальный балл не-победителя*

Зачёт проводится отдельно в каждом классе. Победителями олимпиады становятся школьники, которые набрали наибольший балл среди всех участников в данном классе.

Для каждого класса определите максимальный балл, который набрал школьник, не ставший победителем в данном классе.

Выведите три целых числа.

Input	Output
Иванов Сергей 9 80 Сергеев Петр 10 82 Петров Василий 11 82 Васильев Андрей 9 81 Андреев Александр 10 81 Александров Роман 9 81 Романов Иван 11 83	80 81 82

L. *Максимальный балл призёра и их количество*

Результаты олимпиады подводятся без деления на классы. Победителем олимпиады становятся те, кто набрал больше всего баллов. Призёрами олимпиады становятся участники, следующие за победителями.

Определите наибольший балл, который набрали призёры олимпиады и количество участников олимпиады, набравших такой балл. Выведите два числа: наибольший балл призёра и количество участников, имеющий такой балл.

Input	Output
Иванов Сергей 9 92 Сергеев Петр 10 91 Петров Василий 11 92 Васильев Иван 9 93	92 2

М. *Имя наилучшего не-победителя*

В условиях предыдущей задачи выведите фамилию и имя участника олимпиады, набравшего наибольший балл, но не ставшего победителем. Если таких школьников несколько — выведите их количество.

Input	Output
Иванов Сергей 9 93 Сергеев Петр 10 91 Петров Василий 11 92 Васильев Иван 9 93	Петров Василий

Input	Output
Иванов Сергей 9 92 Сергеев Петр 10 91 Петров Василий 11 92 Васильев Иван 9 93	2

Н. *Школы с наибольшим числом участников олимпиады*

В олимпиаде по информатике принимало участие  $N$  человек. Информация о результатах олимпиады записана в файле, каждая из строк которого имеет вид:

фамилия имя школа балл

Фамилия и имя — текстовые строки, не содержащие пробелов. Школа — целое число от 1 до 999. Балл — целое число от 0 до 100.

Определите школы, из которых в олимпиаде принимало участие больше всего участников.

Выведите номера этих школ в порядке возрастания.

В этой задаче необходимо считывать данные построчно, не сохраняя в памяти данные обо всех участниках, а только подсчитывая число участников для каждой школы.

Input	Output
Иванов Сергей 14 56 Сергеев Петр 23 74 Петров Василий 3 99 Васильев Андрей 3 56 Андреев Роман 14 75 Романов Иван 27 68	3 14

О. *Школы с наименьшим числом участников олимпиады*

В условиях задачи N определите школы, из которых в олимпиаде принимало участие меньше всего участников (но был хотя бы один участник). Выведите номера этих школ в порядке возрастания.

Input	Output
Иванов Сергей 14 56 Сергеев Петр 23 74 Петров Василий 3 99 Васильев Андрей 3 56 Андреев Роман 14 75 Романов Иван 27 68	23 27

P. Школы, в которых есть победители олимпиады

В условиях задачи N выведите в порядке возрастания номера школ, в которых есть хотя бы один победитель олимпиады.

Input	Output
Иванов Сергей 13 80 Сергеев Петр 26 70 Сергеев Андрей 35 80 Петров Василий 13 80 Иванов Роман 35 70 Иванов Иван 26 70	13 35

Q. Школы с высоким средним баллом

В условиях задачи N выведите в порядке возрастания номера школ, средний балл учащихся которых выше, чем средний балл всех участников олимпиады (то есть необходимо вычислить средний балл для каждой школы и средний балл по всем участникам).

Input	Output
Иванов Сергей 13 45 Сергеев Петр 13 45 Сергеев Андрей 20 55 Петров Василий 20 55 Иванов Роман 70 40 Иванов Иван 70 60	20

R. Школы с наибольшим средним баллом

В условиях задачи N выведите в порядке возрастания номера школ, средний балл учащихся которых максимален (то есть необходимо вычислить средний балл для каждой школы и вывести те школы, средний балл для которых максимален).

Input	Output
Иванов Сергей 13 45 Сергеев Петр 13 45 Сергеев Андрей 20 55 Петров Василий 20 55 Иванов Роман 70 40 Иванов Иван 70 60	20

S. Школы с наибольшим числом победителей

В условиях задачи N выведите в порядке возрастания номера школ, из которых наибольшее количество участников стало победителями олимпиады.

Input	Output
Иванов Сергей 13 70 Сергеев Петр 13 60 Сергеев Андрей 20 70 Петров Василий 20 70 Иванов Роман 70 60 Иванов Иван 70 60	20

### Г. Проходной балл

Для поступления в вуз абитуриент должен предъявить результаты трех экзаменов в виде ЕГЭ, каждый из них оценивается целым числом от 0 до 100 баллов. При этом абитуриенты, набравшие менее 40 баллов (неудовлетворительную оценку) по любому экзамену из конкурса выбывают. Остальные абитуриенты участвуют в конкурсе по сумме баллов за три экзамена.

В конкурсе участвует  $N$  человек, при этом количество мест равно  $K$ . Определите проходной балл, то есть такое количество баллов, что количество участников, набравших столько или больше баллов не превосходит  $K$ , а при добавлении к ним абитуриентов, набравших наибольшее количество баллов среди непринятых абитуриентов, общее число принятых абитуриентов станет больше  $K$ .

Программа получает на вход количество мест  $K$ . Далее идут строки с информацией об абитуриентах, каждая из которых состоит из имени (текстовая строка содержащая произвольное число пробелов) и трех чисел от 0 до 100, разделенных пробелами.

Программа должна вывести проходной балл в конкурсе. Выведенное значение должно быть минимальным баллом, который набрал абитуриент, прошедший по конкурсу.

Также возможны две ситуации, когда проходной балл не определён.

- Если будут зачислены все абитуриенты, не имеющие неудовлетворительных оценок, программа должна вывести число 0.
- Если количество абитуриентов, имеющих равный максимальный балл больше чем  $K$ , программа должна вывести число 1.

Решение должно использовать  $O(1)$  памяти, то есть нельзя использовать списки, длина которых зависит от длины входных данных.

Input	Output
5 Иванов Сергей 70 70 70 Сергеев Петр 100 100 0 Петров Василий 70 60 70 Васильев Андрей 70 60 70 Андреев Денис 100 30 100 Денисов Роман 50 50 50 Романов Иван 60 70 70 Ким Чен Ир 50 50 50 Ким Ир Сен 40 40 40	200

Input	Output
1 Иванов Сергей 40 40 40 Сергеев Петр 100 100 39	0

Input	Output
1 Иванов Сергей 60 60 60 Сергеев Петр 100 40 40	1

### У. Полупроходной балл

В условиях задачи Г определите полупроходной балл, то есть такое значение балла, что количество абитуриентов, набравших балл выше полупроходного, меньше  $K$ , а количество абитуриентов, набравших балл выше или равный полупроходному, больше  $K$ .

Программа должна вывести значение полупроходного балла, если полупроходного балла не существует, программа должна вывести одно число 0.

Решение должно использовать  $O(1)$  памяти, то есть нельзя использовать списки, длина которых зависит от длины входных данных.

Input	Output
1 Иванов Сергей 60 60 60 Сергеев Петр 100 40 40	1



## V. Призёры олимпиады

В олимпиаде участвовало  $N$  человек, каждый из которых мог набрать от 0 до 100 баллов. По положению об олимпиаде жюри может наградить не более 45% от числа участников, округляя их число до целого при необходимости вниз.

При этом если последний участник, попавший в 45% набирает столько же баллов, сколько первый участник, не попавший в 45%, то решение по этим участникам, и всем участникам, набравшим такой балл принимается следующим образом:

Все данные участники объявляются призерами, если набранный ими балл больше половины от максимально возможного балла.

Все эти участники не объявляются призерами, если набранный ими балл не больше половины от максимально возможного.

Программа получает на вход информацию об участниках олимпиады (один участник - в одной строке). Строка содержит имя участника (текстовая строка с произвольным числом пробелов) и набранный данным участником балл через пробел.

Программа должна вывести минимальный балл, который получил участник олимпиады, ставший ее призёром.

Решение должно использовать  $O(1)$  памяти, то есть нельзя использовать списки, длина которых зависит от длины входных данных

Input	Output
Иванов Сергей 70 Сергеев Петр 30 Петров Василий 40 Васильев Андрей 80 Андреев Денис 50 Денисов Роман 90 Романов Иван 70 Ким Чен Ир 60 Ким Ир Сен 100	70

Input	Output
Иванов Сергей 50 Сергеев Петр 70 Петров Василий 40 Васильев Андрей 10 Андреев Денис 50 Денисов Роман 20 Романов Иван 30 Ким Чен Ир 70 Ким Ир Сен 100	70

Input	Output
Иванов Сергей 30 Сергеев Петр 60 Петров Василий 20 Васильев Андрей 100 Андреев Денис 30 Денисов Роман 80 Романов Иван 20 Ким Чен Ир 40 Ким Ир Сен 10	40

## W. Гистограмма

Вовочка ломает систему безопасности. Для этого ему понадобилось узнать, какие символы в секретных зашифрованных посланиях употребляются чаще других. Для удобства изучения Вовочка хочет получить графическое представление встречаемости символов. Поэтому он хочет построить гистограмму количества символов в сообщении. Гистограмма — это график, в котором каждому символу, встречающемуся в сообщении хотя бы один раз, соответствует столбик, высота которого пропорциональна количеству этих символов в сообщении.

Входной файл содержит зашифрованный текст сообщения. Он содержит строчные и прописные латинские буквы, цифры, знаки препинания, пробелы и переводы строк. Текст содержит хотя бы один непробельный символ. Все строки входного файла не длиннее 200 символов.

Для каждого символа с кроме пробелов и переводов строк выведите столбик из символов «#», количество которых должно быть равно количеству символов с в данном тексте. Под каждым столбиком напишите символ, соответствующий ему. Отформатируйте гистограмму так, чтобы нижние концы столбиков были на одной строке, первая строка и первый столбец были непустыми. Не отделяйте столбики друг от друга. Отсортируйте столбики в порядке увеличения кодов символов.

Input	Output
Hello, world!	# ## ##### !,Hdelorw

Input	Output
Twas brillig, and the slithy toves Did gyre and gimble in the wabe;  All mimsy were the borogoves, And the mome raths outgrabe.	# # # # # # # # # # ### #### ## ##### #### ##### ##### ## # # ##### ## ##### ,.;ADTabdeghilmnorstuvw

## X. Форматирование текста

Многие системы форматирования текста, например TeX или *Wiki*, используют для разбиения текста на абзацы пустые строки. Текст представляет собой последовательность слов, разделенных пробелами, символами перевода строк и следующими знаками препинания: «,», «.», «?», «!», «-», «:» и «'» (ASCII коды 44, 46, 63, 33, 45, 58, 39). Каждое слово в тексте состоит из заглавных и прописных букв латинского алфавита и цифр. Текст может состоять из нескольких абзацев. В этом случае соседние абзацы разделяются одной или несколькими пустыми строками. Перед первым абзацем и после последнего абзаца также могут идти одна или несколько пустых строк.

Дальнейшее использование исходного текста предполагает его форматирование, которое осуществляется следующим образом. Каждый абзац должен быть разбит на строки, каждая из которых имеет длину не больше  $w$ . Первая строка каждого абзаца должна начинаться с отступа, состоящего ровно из  $b$  пробелов. Слова внутри одной строки должны быть разделены ровно одним пробелом. Если после слова идет один или несколько знаков препинания, они должны следовать сразу после слова без дополнительных пробелов. Если очередное слово вместе со следующими за ним знаками препинания помещается на текущую строку, оно размещается на текущей строке. В противном случае, с этого слова начинается новая строка. В отформатированном тексте абзацы не должны разделяться пустыми строками. В конце строк не должно быть пробелов.

Требуется написать программу, которая по заданным числам  $w$  и  $b$  и заданному тексту выводит текст, отформатированный описанным выше образом.

Input	Output
20 4 Yesterday, All my troubles seemed so far away, Now it looks as though they're here to stay, Oh, I believe in yesterday.  Suddenly, I'm not half the man I used to be, There's a shadow hanging over me, Oh, yesterday came suddenly...	Yesterday, All my troubles seemed so far away, Now it looks as though they' re here to stay, Oh, I believe in yesterday. Suddenly, I' m not half the man I used to be, There' s a shadow hanging over me, Oh, yesterday came suddenly...