

Задача 1. День недели

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

В единственной строке файла записано три буквы, обозначающие день недели на английском языке. Требуется вывести номер этого дня недели.

В файле нет пробелов. Первая буква заглавная, остальные две маленькие. Гарантируется, что с трёх записанных букв начинается название дня недели на английском языке.

Пример

<code>input.txt</code>	<code>output.txt</code>
Wed	3

Пояснение к примеру

Среда в английском называется Wednesday.

Задача 2. Слова из трех букв

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Ваши друзья пытаются заполнить кроссворд, нужно подобрать слово из трех букв, так, чтобы на втором месте стояла буква **A**. Они неоднократно предлагают друг другу слова, надеясь, что они подходят. Кто-то предлагает написать программу, которая может быстро определить, соответствует ли данное слово приведенному выше описанию.

Ваша задача — написать такую программу.

Формат входных данных

Во входном файле записано одно слово, состоящее из прописных латинских букв. Его длина не превышает 12 символов. После слова записан символ «точка».

Формат выходных данных

В выходной файл необходимо вывести текст, который обозначает, подходит или нет введенное слово. Если слово состоит из трех букв, и на втором месте стоит буква **A**, то нужно вывести **FITS**. В противном случае, ваша программа должна вывести **DOES NOT FIT**.

Примеры

<code>input.txt</code>	<code>output.txt</code>
CAT.	FITS
FATE.	DOES NOT FIT
TEN.	DOES NOT FIT

Задача 3. Что бы ты ни сделал, я могу сделать лучше

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

У вас есть друг, который слишком любит соревноваться, всегда старается вас «переплюнуть». Если вы говорите, что ваша машина быстрая, то ваш товарищ обязательно скажет, что его машина еще быстрее. Если вы скажете, что ваша машина быстрее, то он заявит, что его – самая быстрая. После нескольких таких разговоров вы понимаете, что всегда можно предугадать, что ваш товарищ скажет в следующий момент.

Чтобы продемонстрировать, как это все надоедает, вы решили написать программу, которая аккуратно предсказывает ответы вашего друга. В частности, для любого английского прилагательного, ваша программа будет возвращать его же в сравнительной форме путем добавления **er** к нему. Заметим, что если прилагательное уже заканчивается на **e**, вы должны только добавить букву **r**. Если же программе на вход дано прилагательное уже в сравнительной форме, то программа должна вернуть это прилагательное в превосходной форме путем простой замены **er** на **est**.

Формат входных данных

Во входном файле записано одно слово, после которого идет символ точка. Слово представляет собой прилагательное или сравнительную форму прилагательного. Слово состоит только из строчных латинских букв. Его длина не превосходит 30

Формат выходных данных

В выходной файл необходимо вывести преобразованное прилагательное в сравнительной или в превосходной форме, как описано в условии.

Примеры

<code>input.txt</code>	<code>output.txt</code>
<code>warm.</code>	<code>warmer</code>
<code>smaller.</code>	<code>smallest</code>
<code>rare.</code>	<code>rarer</code>

Задача 4. Делимость на 15

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

На вход подается число. Нужно определить делится ли оно на 15.

Формат входных данных

Входной файл из последовательности цифр, за которыми следует символ «точка». Длина числа не превышает 500.

Формат выходных данных

Если введенное число делится нацело на 15, то в выходной файл необходимо вывести слово YES. В противном случае необходимо вывести слово NO.

Примеры

<code>input.txt</code>	<code>output.txt</code>
12345.	YES
67.	NO

Задача 5. Количество слов

Источник:	основная
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

В единственной строке файла записан набор слов, разделённых символами точки. Каждое слово состоит из букв латинского алфавита. Между словами находится один или несколько символов точки. До первого слова и после последнего точки могут быть, а могут не быть. В строке может быть записано всего одно слово, а может и вовсе не быть слов. Длина заданной строки находится в диапазоне от 1 до 10 000.

Требуется вывести одно целое число — количество слов в строке.

Пример

input.txt	output.txt
..ko..Privet.kreved....ko...	4

Пояснение к примеру

В примере записано четыре слова: ko, Privet, kreved и ko.

Комментарий

В данной задаче нужно читать символы из входного файла по одному, сохраняя их в переменную типа `char`. Примерно так:

```
char symbol;  
scanf("%c", &symbol);
```

Гарантируется, что после строки имеется символ перевода строки `'\n'`.

Задача 6. Игра в алфавит

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Два товарища играют в слова. Игроки по очереди приводят примеры слов из некоторой предметной области. Начинает Игрок 1. На каждом ходу игрок должен привести пример, который содержит указанную букву алфавита, которая изменяется на следующем шаге. В частности, Игрок 1 должен сначала привести пример слова, содержащего букву 'а'. Игрок 2 должен затем привести пример слова, содержащего букву 'б'. Игрок 1 должен на следующем шаге привести пример слова, содержащего 'с', и так далее. Если игрок не может назвать соответствующее слово, то он проигрывает, а другой игрок, соответственно, выигрывает.

Для заданной последовательности слов, которые по очереди произносили игроки, вам нужно написать программу, которая бы определяла, кто победил. Проигравшим считается тот игрок, который первым не смог привести соответствующий пример слова.

Формат входных данных

Во входном файле записана непустая строка, содержащая последовательность слов, разделенных одним пробелом. Все слова содержат маленькие латинские буквы. Количество слов не превышает 26. Длина одного слова может изменяться от 1 до 20.

Формат выходных данных

В выходной файл необходимо вывести одну из трёх возможных комбинаций слов. Если выигрывает первый игрок, то в выходной файл нужно вывести `PLAYER 1`. Если же выигрывает второй игрок, нужно вывести `PLAYER 2`. Если никто не выигрывает, то вывести `NO WINNER`.

Примеры

<code>input.txt</code>	<code>output.txt</code>
<code>alpha beta gamma delta epsilon</code>	<code>PLAYER 2</code>
<code>january february march april may</code>	<code>PLAYER 1</code>
<code>cab cab cab ford chevy ford dodge honda</code>	<code>NO WINNER</code>

Задача 7. Смешивание цветов

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Известно, что можно смешивать различные цвета красок, чтобы получать другие цвета. В частности, для того чтобы получить зеленый цвет, нужно смешать желтый и синий, оранжевый может получиться смешиванием красного и желтого, а фиолетовый — из красного и синего.

В магазине есть автоматическая линия для смешивания красок. В нее заряжены баллончики с красками разных цветов. Она может смешивать любые два баллончика. Ваша задача по заданным требуемым цветам (`green`, `orange` или `purple`) определить, какие баллончики смешивать.

Формат входных данных

Во входном файле записана строка, состоящая из двух слов, разделенных одним пробелом.

Первое слово, записанное маленькими буквами, определяет требуемый цвет, один из трех: `green`, `orange` или `purple`.

Второе слово состоит из последовательности больших латинских букв, каждая буква представляет какой-то цвет. Цвета красный, желтый и синий будут представлены буквами `R`, `Y` и `B`, соответственно. Другие буквы тоже могут в этом слове встречаться. Буквы `R`, `Y` и `B` встречаются во втором слове только один раз. Его длина может изменяться от 3 до 26.

Формат выходных данных

В выходной файл необходимо вывести два целых числа, разделенных пробелом. Эти числа задают номера баллончиков с краской, которые нужно смешать, чтобы получить нужный цвет. Целые нужно выводить в возрастающем порядке. Нумерация баллончиков начинается с 1.

Примеры

input.txt	output.txt
purple KWBYNXRS	3 7
green BARCYZ	1 5
orange GSWXBRY	6 7

Задача 8. Сумма чисел

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

В заданном тексте нужно посчитать сумму входящих в него десятичных целых чисел.

Формат входных данных

Во входном файле записана непустая последовательность символов, среди которых могут встречаться цифры. Эта последовательность заканчивается символом «точка». Длина последовательности не превосходит 10^4 . Числа, которые в ней встречаются, имеют не более 5 знаков.

Формат выходных данных

В выходной файл необходимо вывести целое число – сумму чисел, которые встречаются в заданной последовательности.

Пример

<code>input.txt</code>	<code>output.txt</code>
This is example: three (3) + 15 is equal to 18.	36

Задача 9. Разность множеств

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Дано два множества символов $S1$ и $S2$. Вам нужно вычислить их разность ($S1 - S2$) – новое множество, в котором присутствуют только те элементы $S1$, которых нет в $S2$.

Формат входных данных

В единственной строке входного файла через пробел записано две непустые последовательности латинских букв в верхнем регистре. Эти последовательности задают два множества – $S1$ и $S2$, соответственно. Каждый элемент множества встречается в последовательности не более одного раза. Количество элементов в каждом множестве не более 26.

Формат выходных данных

В выходной файл необходимо вывести последовательность символов, которые будут представлять разность заданных множеств. Элементы искомого множества выдавать в возрастающем порядке. Если результирующее множество – пусто, то вывести `Empty Set`.

Примеры

<code>input.txt</code>	<code>output.txt</code>
AEIOU AEIOUY	Empty Set
PQRSTUV UVWXY	PQRST
TEXAS COMPUTER	ASX

Задача 10. Слова

Источник: повышенной сложности
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Перечислить все слова из заданной строки, не содержащие букв, из которых состоит первое слово.

Формат входных данных

Во входном файле записана строка, состоящая из слов, разделенных одним пробелом. Каждое слово состоит из маленьких латинских букв.

Формат выходных данных

В выходной файл необходимо вывести все слова, по одному на строку, обладающие заданным свойством, в порядке их встречаемости во входном файле.

Пример

input.txt	output.txt
to be or not to be that is the question	be be is