

Задача А. Сумма двух

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Найдите сумму двух заданных чисел.

Формат входных данных

В первой строке входных данных расположены два целых числа A и B , не превосходящие 1000 по модулю.

Формат выходных данных

Ваша программа должна выдавать одно число — сумму чисел A и B .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3	5
17 -18	-1

Задача В. Сумма двух наоборот

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Конечно, вы уже сталкивались с такой задачей: даны два числа, требуется вычислить их сумму. Решите ту же задачу, только наоборот. А именно: дана сумма двух целых чисел, найдите два числа, дающие такую сумму.

Формат входных данных

В первой строке входных данных задано одно целое число c .

Формат выходных данных

Выведите два целых числа a и b , разделив их пробелом. Выведенные числа не должны превышать по модулю 10^9 . Кроме того, для них должно выполняться равенство $a + b = c$. Гарантируется, что заданное c таково, что существует ответ, удовлетворяющий этим ограничениям. Если ответов несколько, разрешается выводить любой из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	3 -2

Задача С. Стирание чисел

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На доске выписаны целые числа от 1 до N , включительно. Сотрём некоторые из них так, чтобы сумма оставшихся чисел отличалась от суммы стёртых на как можно меньшую величину. Чему равна эта величина?

Формат входных данных

В первой строке входного файла задано число N ($1 \leq N \leq 100$).

Формат выходных данных

Выведите единственное число L в первой строке выходного файла — минимальную разность между суммой оставшихся чисел и суммой стёртых чисел.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	1
3	0

Задача D. Палиндром

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Палиндром — это строка, которая читается в обратном порядке так же, как в прямом. Проверьте, является ли данная непустая строка палиндромом.

Формат входных данных

Входные данные состоят из одной строки. Строка состоит из символов с ASCII-кодами от 33 до 126 включительно. Длина строки — от одного до 10 000 символов включительно. Строка завершается переводом строки (ASCII-символы с кодами 13 и 10).

Формат выходных данных

Выведите «YES», если данная строка является палиндромом, или «NO», если не является. Регистр букв имеет значение.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
abacaba	YES
abacabadaba	NO

Задача Е. Число

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана последовательность из восьми цифр, каждая из которых равна 1, 2 или 3. Требуется выбрать из них четыре цифры так, чтобы число, полученное при выписывании этих цифр в том же порядке, в каком они были в последовательности, оказалось как можно меньше.

Формат входных данных

Первая строка входного файла содержит восемь цифр из набора {1, 2, 3}, записанных подряд без пробелов.

Формат выходных данных

В первой строке выходного файла выведите четыре цифры, образующие искомое минимальное число, также без пробелов.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
12333333	1233
32111111	1111
12131211	1111
33333323	3323
32333313	2313

Задача F. Уголок

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Дана прямоугольная доска $M \times N$, некоторые клетки в которой вырезаны. Сколькими способами можно поставить на неё «уголок» из трёх клеток так, чтобы все три клетки уголка находились внутри доски и не были вырезаны?

Формат входных данных

В первой строке входного файла даны два числа M и N ($1 \leq M, N \leq 100$), разделённые пробелом. В следующих M строках содержится по N символов в каждой; i -ый символ j -ой из этих строк равен 'X' (большая буква икс), если клетка вырезана, и '.' (точка) в противном случае.

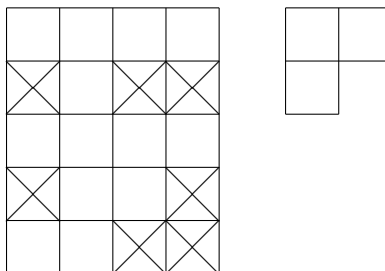
Формат выходных данных

Выведите одно число — сколько существует способов поставить уголок на данную доску.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2	4
2 3 ..X .X.	1
5 4 X.XX X..X ..XX	12

Доска из второго примера и уголок проиллюстрированы на рисунке.



Задача G. Круг на асфальте

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Маленькие девочки Маша и Катя недавно выучили весь латинский алфавит! По этому случаю они придумали себе игру. Сначала они нарисовали на асфальте круг, поделили его на сектора, и в каждом секторе написали какую-то букву. Игра заключается в том, чтобы, войдя в круг, прыгать по буквам либо по, либо против часовой стрелки, попутно называя букву, на которую прыгаешь. Девочки сохраняют направление движения по кругу и всегда прыгают в соседний сектор.

Скоро выяснилось, что Маше играть сложнее, поскольку она путает право и лево и, входя в круг, не знает, по часовой стрелке она станет прыгать или против. Поэтому Маше сложнее запомнить буквы в той последовательности, в которой она по ним прыгает.

Проходя мимо, Машин старший брат Петя заметил, что, если входить в круг в таком месте, что при перемещении как по часовой стрелке, так и против получается одна и та же последовательность букв, то девочки будут в равных условиях. Теперь он хочет выяснить, сколько таких позиций в данном круге. Помогите ему это сделать!

Формат входных данных

В первой строке входного файла задана строка, состоящая из строчных букв латинского алфавита — буквы, написанные в круге в порядке по часовой стрелке; после последней буквы строки в круге идёт первая. Длина этой строки находится в пределах от 1 до 100, включительно.

Формат выходных данных

В первой строке выходного файла выведите одно число — количество таких букв в круге, что, начав с этой буквы и перемещаясь как по часовой стрелке, так и против, мы получим одинаковую бесконечную последовательность букв.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
aba	1
baab	0
ab	2

Задача Н. Поиск

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче нужно уметь выяснять, содержится ли число в последовательности.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и k ($1 \leq n \leq 5000$, $1 \leq k \leq 300\,000$). Во второй строке задана последовательность из n целых чисел a_1, a_2, \dots, a_n , записанных через пробел ($1 \leq a_i \leq 10^9$). В третьей строке записаны запросы — k целых чисел b_1, b_2, \dots, b_k через пробел ($1 \leq b_j \leq 10^9$).

Формат выходных данных

В выходной файл выведите k строк. В j -ой строке выведите "YES", если число b_j содержится в последовательности $\{a_i\}$, и "NO" в противном случае.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 3 2 3 5 1 2 3	NO YES YES
3 4 2 1 2 2 4 1 5	YES NO YES NO
5 1 11111 1111 111 11 1 12345	NO

Задача I. Число Фибоначчи

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Числа Фибоначчи $F_0, F_1, F_2, \dots, F_n$ определяются следующим образом: $F_0 = 0, F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите число Фибоначчи F_n .

Формат входных данных

В первой строке входного файла задано единственное число n ($0 \leq n \leq 92$).

Формат выходных данных

Выведите число F_n в первой строке выходного файла.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	1
2	1
3	2
5	5

Задача J. Числа Фибоначчи по модулю

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Числа Фибоначчи $F_0, F_1, F_2, \dots, F_n$ определяются следующим образом: $F_0 = 0, F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите остаток от деления числа Фибоначчи F_n на m .

Формат входных данных

В первой строке ввода заданы через пробел два целых числа n и m ($0 \leq n \leq 10^9, 1 \leq m \leq 10^9$).

Формат выходных данных

В первой строке выведите одно число: остаток от деления F_n на m .

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
8 12345	21
100 1000000000	261915075

Пояснение к примеру

Первые несколько чисел Фибоначчи таковы: $F_0 = 0, F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5, F_6 = 8, F_7 = 13, F_8 = 21$.

Число Фибоначчи с номером 100 равно 354 224 848 179 261 915 075. Остаток от деления этого числа на 1 000 000 000 – это последние девять цифр его десятичной записи.

Задача К. Закон композиции

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Законом композиции на множестве M называется отображение $f : M \times M \rightarrow M$, сопоставляющее каждой упорядоченной паре элементов множества некоторый его элемент.

- Закон композиции называется *ассоциативным*, если $\forall x, y, z \in M f(x, f(y, z)) = f(f(x, y), z)$.
- *Единицей* называется такой элемент $e \in M$, что $\forall x \in M f(e, x) = f(x, e) = x$.
- *Обратным* к элементу $x \in M$ называется такой элемент $x^{-1} \in M$, что $f(x, x^{-1}) = f(x^{-1}, x) = e$.
- Если $\forall x, y \in M f(x, y) = f(y, x)$, то закон композиции называется *коммутативным*.

Требуется распознать одну из следующих алгебраических структур множества M с заданным на нём законом композиции:

- Если закон композиции ассоциативен, то структура называется *полугруппой*.
- Если закон композиции коммутативен — *коммутативным группоидом*.
- Если закон и ассоциативен, и коммутативен — *коммутативной полугруппой*.
- Если в полугруппе есть единица, то она называется *моноидом*.
- Если в моноиде выполнено условие коммутативности, то он называется *коммутативным моноидом*.
- Если в моноиде для каждого элемента найдётся обратный к нему, то он называется *группой*.
- Если структура является группой, и закон композиции удовлетворяет условию коммутативности, то группа называется *абелевой*.
- Если структура не является ни одной из вышеперечисленных, то она называется *группоидом*.

Формат входных данных

В первой строке входного файла задано количество элементов во множестве $1 \leq |M| \leq 128$. Далее следуют $|M|^2$ строк, в которых записан результат применения функции f к парам элементов множества M в порядке лексикографического возрастания пар аргументов (если для удобства занумеровать элементы множества числами от 1 до $|M|$, то получится, что первой идёт пара (1, 1), затем (1, 2), ..., (1, $|M|$), (2, 1), ..., ($|M|$, $|M|$)).

Формат выходных данных

В выходной файл нужно вывести название структуры на английском языке:

название структуры	вывод программы
группоид	groupoid
полугруппа	semigroup
коммутативная полугруппа	commutative semigroup
коммутативный группоид	commutative groupoid
моноид	monoid
коммутативный моноид	commutative monoid
группа	group
абелева группа	abelian group

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	abelian group
1	
2	
2	
1	
1	

Задача I. Словарь

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан некоторый текст на английском языке. Вам необходимо составить словарь слов, встречающихся в этом тексте.

Словом называется последовательность символов английского алфавита, ограниченная с двух сторон *ограничителями*.

Ограничителем является любой символ, отличный от букв английского алфавита, цифр, а также знака подчёркивания «_». Кроме того, ограничителями являются также перевод строки, начало и конец файла.

Например, рассмотрим входной файл:

```
kitten abc123e a_b_c, , , ,eats
```

```
!Ъmice
```

В нём словами являются «kitten», «eats» и «mice».

Формат входных данных

Входной файл содержит текст (не более 10 000 символов). В нём могут встречаться произвольные символы, отличные от символа конца файла (EOF).

Формат выходных данных

В выходной файл должен быть выведен искомый словарь, по одному слову на строке, в алфавитном порядке. Все слова должны быть приведены к нижнему регистру. Если слово встречается больше одного раза, то после него через пробел должно быть выведено количество раз, которое это слово встретилось. Слова, совпадающие побуквенно, но отличающиеся регистром, считаются одинаковыми.

Пример

<i>стандартный ввод</i>
kitten abc123e a_b_c, , , ,eats !Ъmice KITten does not eat OTHER_KITTENS
<i>стандартный вывод</i>
does eat eats kitten 2 mice not

Задача М. Стресс-тестирование

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Витю пригласили в жюри межпланетной олимпиады по программированию. Это большая честь для него, и он постарался не ударить в грязь лицом. Он реализовал технологию, которая позволяет находить интересные тесты для задач по имеющимся решениям жюри. Эта технология называется *стресс-тестирование*.

Поиск теста осуществляется по следующей схеме. Сначала пишется генератор общего случая теста к задаче, то есть такой генератор, который с достаточно большими вероятностями выдает тесты на все случаи, которые необходимо разобрать. Он не обязательно является генератором полностью случайного теста, так как зачастую на таких тестах реализуются далеко не все случаи решения.

Генератор, естественно, использует случайные числа для создания теста. Он инициализируется с помощью функции `randomize`, которая берет случайное значение для инициализации генератора случайных чисел из внешнего источника (как правило, в качестве такого источника используется системное время). Для того чтобы можно было потом восстановить тест, значение, которым был проинициализирован генератор случайных чисел, выдается в файл протокола. Всегда гарантируется, что при одной и той же инициализации генератор случайных чисел будет выдавать одну и ту же последовательность.

После генерации теста на нем запускаются два различных решения жюри. Специальная программа `run` замеряет время работы каждого из решений. Естественно, вся информация о времени работы, параметрах запуска и используемой памяти также выводится в файл протокола работы системы.

Затем происходит сравнение выходных файлов обеих программ. К сожалению, в связи с особенностями используемой суперсовременной операционной системы `Doogs`, информацию о сравнении не так-то просто вывести в файл протокола, поэтому Витя ограничился лишь прерыванием тестирования в случае, если одно из решений выдало неверный ответ.

Весь этот процесс запускается в цикле, пока не будет найден тест, на котором одна из программ выдает неверный ответ, или же член жюри, занимающийся тестированием, не сочтет, что уже было проверено достаточное количество различных тестов.

В первом случае, очевидно, интересным является тест, на котором завалилось решение жюри. Во втором же случае Витя считает интересными тесты, на которых каждое из двух решений работало максимальное время. К сожалению, исходный файл протокола не очень удобен для получения этой информации.

Необходимо написать программу, которая бы преобразовывала файл протокола в более удобный формат, а также находила бы тесты, на которых каждое из решений работало максимальное время.

Формат входных данных

Во входном файле содержится исходный файл протокола. Его размер не превосходит двух мегабайт. Он состоит из одного или нескольких блоков, каждый из которых соответствует запуску обоих решений на одном тесте. Каждый из этих блоков начинается и заканчивается непустой строкой, состоящей из одних символов `'-'`.

Внутри блока содержится информация об инициализации генератора случайных чисел, а также о времени работы каждого из решений. Инициализация генератора задается строкой в формате `randseed = x`, где x — целое неотрицательное число, не превосходящее 10^9 . За ней следуют протоколы запуска решений. Время работы каждого из решений выведено в формате `Work time: x_i ms`, где x_i — вещественное число, в котором в качестве разделителя целой и дробной части используется запятая — так уж вывела операционная система `Doogs`. Целая часть этого числа не превосходит 10^6 . Витя знает, что операционная система `Doogs` все равно производит измерения таким образом, что дробные части совпадают при совпадении целых частей, поэтому его интересует только целая часть этого числа.

Кроме указанных строк, в файле протокола может содержаться также другая информация, которую Витя пока не хочет анализировать.

Всегда сначала запускается первое решение жюри, а затем второе. Кроме того, запуск решений происходит только после генерации теста, а, следовательно, и вывода в файл протокола информации об инициализации генератора случайных чисел.

Формат выходных данных

Для каждого из блоков в порядке их следования должны быть выведены три строки информации. В первой строке выводится значение, которым был проинициализирован генератор случайных чисел, в формате `At randseed = x`. Во второй и третьей строках выводится время работы первого и второго решений соответственно, в виде `First: x_1 ms` и `Second: x_2 ms`, где x_i — время работы каждого из решений с отброшенной дробной частью.

Кроме этого, необходимо собрать информацию о том, на каких тестах каждое из решений работало максимальное время. Эта информация выводится после всей информации о блоках. Если какое-то решение работало одинаково долго на нескольких тестах, то выбирается самый первый из блоков, на котором было достигнуто такое время.

Проверка производится автоматически, поэтому вывод вашей программы должен по возможности максимально совпадать с эталонным выводом!

Пример

<i>стандартный ввод</i>
<pre>----- randseed = 43794135 Testing ..\b_mb, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (80 ms) Work time: 80,11520 ms Maximal memory usage: 10336 Kb Testing ..\b_al, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (391 ms) Work time: 390,56160 ms Maximal memory usage: 2680 Kb ----- randseed = 43903502 Testing ..\b_mb, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (110 ms) Work time: 110,15840 ms Maximal memory usage: 10328 Kb Testing ..\b_al, in=bring.in, out=bring.out, ext=, tl=100000000, ml=100000000 on bring.in Successful termination (310 ms) Work time: 310,44640 ms Maximal memory usage: 2680 Kb -----</pre>
<i>стандартный вывод</i>
<pre>At randseed = 43794135 First: 80 ms Second: 390 ms At randseed = 43903502 First: 110 ms Second: 310 ms Maximal work time for first: 110 at randseed = 43903502 Maximal work time for second: 390 at randseed = 43794135</pre>