

Задача А. Количество ПСП

Имя входного файла: `quant.in`
Имя выходного файла: `quant.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Посчитайте количество правильных скобочных последовательностей длины $2n$ (n открывающих скобок и n закрывающих), составленных из круглых и квадратных скобок так, что внутри любой пары круглых скобок нет квадратных скобок.

Формат входных данных

В единственной строке через пробел записано целое неотрицательное число n , не превосходящее 1000.

Формат выходных данных

Выведите остаток от деления количества искомых правильных скобочных последовательностей на $10^9 + 7$.

Примеры

<code>quant.in</code>	<code>quant.out</code>
1	2
2	7

Задача В. Разбиение на слагаемые

Имя входного файла: `part.in`
Имя выходного файла: `part.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано число N . Ваша задача — посчитать количество способов разбить число N на слагаемые так, чтобы все слагаемые были различны между собой.

Формат входных данных

В единственной строке входных данных содержится число N ($1 \leq N \leq 2000$).

Формат выходных данных

Выведите искомое количество способов разложить N на слагаемые. Так как ответ может быть очень большим, выведите его по модулю 1 000 000 007.

Примеры

<code>part.in</code>	<code>part.out</code>
7	5

Задача С. Разбиения на K слагаемых

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 128 мегабайт

Для данных натуральных чисел N и K определите количество способов представить число N в виде суммы K натуральных слагаемых, если способы, отличающиеся только порядком слагаемых считать одинаковыми.

Формат входных данных

Программа получает на вход два натуральных числа N и K , не превосходящих 400. Гарантируется, что ответ не превосходит $2^{64} - 1$.

Формат выходных данных

Выведите ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
6 3	3
5 2	2

Задача D. Восстановление

Имя входного файла: `recover.in`
Имя выходного файла: `recover.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дарья обнаружила ошибку в своей программе, которая удаляет все символы из строки кроме “(” и “)”. Оказывается, некоторые символы заменяются на что-то нечитаемое.

Теперь её заинтересовал вопрос, сколько различных правильных скобочных последовательностей длины $2n$ могут являться результатом исправленного алгоритма, то есть не будут противоречить данным, которые она таки не потеряла.

Формат входных данных

Единственная строка входного файла содержит строку из круглых скобок и знаков вопроса, где вопросами обозначены утраченные символы. Длина строки не превосходит 10 000.

Формат выходных данных

Выведите одно число — количество различных скобочных последовательностей, удовлетворяющих шаблону Дарьи, по модулю $10^9 + 7$.

Примеры

<code>recover.in</code>	<code>recover.out</code>
<code>(??()?)</code>	2

Замечание

Вместо `python3` при сдаче этой задачи используйте `pyru3`.

Задача Е. Число возрастающих подпоследовательностей

Имя входного файла: `subseq.in`
Имя выходного файла: `subseq.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задана последовательность из n чисел $a_1, a_2, \dots, a_n$. Подпоследовательностью длины k этой последовательности называется набор индексов $i_1, i_2, \dots, i_k$, удовлетворяющий неравенствам $1 \leq i_1 < i_2 < \dots < i_k \leq n$. Подпоследовательность называется возрастающей, если выполняются неравенства $a_{i_1} < a_{i_2} < \dots < a_{i_k}$.

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности $a_1, \dots, a_n$. Так как это число может быть достаточно большим, необходимо найти остаток от его деления на $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 1000$). Вторая строка входного файла содержит n целых чисел: $a_1, a_2, \dots, a_n$. Все a_i не превосходят 10^9 по абсолютной величине.

Формат выходных данных

В выходной файл выведите ответ на задачу.

Примеры

subseq.in	subseq.out
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8

Задача F. Контест-палиндром

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Как вы знаете, Хет и Юра очень любят готовить контесты и готовы заниматься этим целыми часами. Вот и сейчас Хет перебирает задачи, которые он хочет добавить в предстоящий контест. Он считает, что контест будет **успешным** только в том случае, если первые буквы названий задач, которые в него входят, образуют палиндром (контест, который не содержит ни одной задачи, по логике Хета, тоже успешный!).

Система подготовки контестов «Моногон» содержит в себе N задач, которые пронумерованы последовательно от 1 до N . Все их названия начинаются с заглавной латинской буквы. Хет хочет выбрать некоторые из них. При этом, «Моногон» требует, чтобы задачи в контесты были отсортированы по порядковому номеру. Т. е. чтобы номера задач в контесте образовывали возрастающую последовательность.

Хет выписал первые буквы названий задач с номера 1 по N и принёс вам. Помогите Хету определить, сколько всего **успешных** контестов он сможет подготовить.

Так как количество контестов может быть очень большим, ответ требуется вывести по модулю $10^9 + 7$.

Формат входных данных

Единственная строка представляет из себя последовательность длины N ($1 \leq N \leq 5000$) из латинских заглавных букв.

Формат выходных данных

Выведите кол-во успешных контестов по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
АВАСАВА	42
ІТМО	5

Замечание

В первом примере название задачи №1 начинается с буквы «А», №2 с буквы «В», №3 с буквы «А» и т. д.

Задача G. Интересные числа

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

Софья считает число интересным, если его цифры идут в неубывающем порядке. Например, числа 123, 1111 или 888999 – интересные.

Софья заинтересовалась, сколько существует интересных положительных чисел, лежащих в диапазоне от L до R включительно. Это число может оказаться довольно большим для больших L и R , поэтому Софья хочет найти остаток от деления этого числа на $10^9 + 7$.

Требуется написать программу, которая по заданным L и R определяет количество интересных чисел, лежащих в диапазоне от L до R включительно, и выводит остаток от деления этого числа на $10^9 + 7$.

Формат входных данных

Входной файл содержит две строки. Первая строка содержит число L , вторая строка содержит число R ($1 \leq L \leq R \leq 10^{100}$).

Формат выходных данных

Выходной файл должен одно целое число — остаток от деления количества интересных чисел, лежащих в диапазоне от L до R включительно, на $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
1 100	54

Задача Н. Распил брусьев

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам нужно распилить деревянный брус на несколько кусков в заданных местах. Распилочная компания берет k рублей за распил одного бруска длиной k метров на две части.

Понятно, что различные способы распила приводят к различной суммарной стоимости заказа. Например, рассмотрим брус длиной 10 метров, который нужно распилить на расстоянии 2, 4 и 7 м, считая от одного конца. Это можно сделать несколькими способами. Можно распилить сначала на отметке 2 м, потом 4 и, наконец, 7 м. Это приведет к стоимости $10 + 8 + 6 = 24$, потому что сначала длина бруса, который пилили, была 10 м, затем она стала 8 м, и, наконец, 6 м. А можно распилить иначе: сначала на отметке 4 м, затем 2, затем 7 м. Это приведет к стоимости $10 + 4 + 6 = 20$, что лучше.

Определите минимальную стоимость распила бруса на заданные части.

Формат входных данных

Первая строка входных данных содержит целое число L ($2 \leq L \leq 10^6$) - длину бруса и целое число N ($1 \leq N \leq 100$) - количество распилов. Во второй строке записано N целых чисел C_i ($0 < C_i < L$) в строго возрастающем порядке - места, в которых нужно сделать распилы.

Формат выходных данных

Выведите одно натуральное число - минимальную стоимость распила.

Примеры

stdin	stdout
10 3 2 4 7	20
100 3 15 50 75	200

Задача I. Удаление скобок - 2

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

Формат входных данных

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

Формат выходных данных

Выведите строку максимальной длины, являющуюся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов. Если возможных ответов несколько, выведите любой из них.

Примеры

stdin	stdout
<code>([])</code>	<code>[]</code>
<code>{([({})])}</code>	<code>([]{})</code>
<code>]{}[</code>	

Задача J. Кваквариум

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Лягушонок Кваки живёт на большом болоте и очень любит петь. Свою песню он записывает как строку S длины N , состоящую из маленьких букв латинского алфавита — каждая буква обозначает отдельный квак.

Однако не всякий квак получается чистым: пока Кваки распевается, в его песню затёсываются лишние звуки. Настоящую «лягушачью речь» понимают только тогда, когда песню можно без остатка разбить на известные всем обитателям болота слова из кваквариума — словаря, содержащего M слов. Каждое слово словаря можно использовать сколько угодно раз (в том числе ни разу).

Чтобы песня зазвучала правильно, Кваки готов вычеркнуть из неё несколько лишних кваков. Помогите ему определить наименьшее количество букв, которое нужно вычеркнуть из строки S , чтобы оставшуюся строку можно было представить как последовательность слов из кваквариума.

Формат входных данных

В первой строке входных данных находятся два целых числа N и M , разделённых пробелом — длина песни и количество слов в кваквариуме ($1 \leq N, M \leq 100$).

Во второй строке находится строка S длины N — песня лягушонка.

В каждой из следующих M строк находится одно слово кваквариума. Длина каждого слова не превышает N .

Строка S и все слова кваквариума состоят только из маленьких букв латинского алфавита.

Формат выходных данных

Выведите одно целое число — минимальное количество вычеркиваний, после которых песня Кваки может быть представлена в виде последовательности слов из кваквариума.

Пример

стандартный ввод	стандартный вывод
11 5 abafchtdsya aba a bach dsy zero	2

Замечание

В примере песня после вычеркивания лишних кваков **f** и **t** примет вид «**abachdsya**» (было сделано два вычеркивания: «**abafchtdsya**») и может быть представлена как последовательность слов кваквариума: «**a**», «**bach**», «**dsy**», «**a**».

Задача К. Оптимальная триангуляция

Имя входного файла: `triangulation.in`
Имя выходного файла: `triangulation.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Триангуляцией N -угольника называется набор из $N - 3$ непересекающихся (кроме как в вершинах многоугольника) диагоналей, разбивающих N -угольник на $N - 2$ треугольника.

Для заданного выпуклого N -угольника найдите триангуляцию, у которой длина самой большой диагонали, входящей в триангуляцию, минимальна.

Формат входных данных

Первая строка входных данных содержит целое число N ($4 \leq N \leq 100$) — количество вершин в многоугольнике. Далее идёт N пар целых чисел x_i, y_i , не превосходящих 10 000 по абсолютной величине — координаты выпуклого N -угольника в порядке обхода.

Формат выходных данных

В первой строке выходного файла выведите целое число — квадрат длины минимальной диагонали в триангуляции.

В последующих $N - 3$ строках выведите по два целых числа — вершины очередной диагонали триангуляции.

Диагонали можно выводить в любом порядке.

Вершины многоугольника нумеруются начиная с единицы в том же порядке, в котором они даны во входных данных.

Если оптимальных ответов несколько, разрешается вывести любой из них.

Примеры

triangulation.in	triangulation.out
4	2
0 0	2 4
0 1	
1 1	
1 0	

Задача L. Два станка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

На фабрике «Дельта» работает конвейер, по которому едет длинная лента с напечатанными на ней цифрами. Лента содержит ровно n цифр (возможно, начинающихся с нулей). В конце конвейера стоят два упаковочных станка — левый и правый.

Оператор должен направить каждую цифру ленты ровно на один из двух станков, не меняя порядок цифр. Каждый станок склеивает попавшие к нему цифры (в исходном порядке ленты) в одно длинное число. Настройки станков таковы, что:

- число, собранное левым станком, обязано делиться на A ;
- число, собранное правым станком, обязано делиться на B .

Каждый станок должен получить хотя бы одну цифру — пустые числа не допускаются. Оба собранных числа, как и сама лента, могут содержать ведущие нули.

Инженеры хотят, чтобы нагрузка на станки была как можно более равномерной. Поэтому среди всех допустимых способов распределения цифр нужно выбрать такой, при котором величина $|l - r|$ минимальна, где l — количество цифр, ушедших на левый станок, а r — количество цифр, ушедших на правый станок.

Помогите оператору: для каждой ленты определите подходящее распределение цифр или сообщите, что его не существует.

Формат входных данных

В первой строке записано одно целое число t ($1 \leq t \leq 10$) — количество лент (наборов входных данных). Далее следуют описания t лент.

Каждая лента описывается двумя строками. В первой строке записаны три целых числа n , A и B ($2 \leq n \leq 40$, $1 \leq A, B \leq 40$) — количество цифр на ленте и делители для левого и правого станков соответственно. Во второй строке записана последовательность из ровно n цифр — содержимое ленты (возможны ведущие нули).

Формат выходных данных

Для каждой ленты выведите в отдельной строке:

- число -1 , если требуемого распределения цифр не существует;
- иначе — строку s из n символов, каждый из которых равен «L» или «R». Если i -я цифра ленты направляется на левый станок, то i -й символ строки должен быть «L», иначе — «R». Число из цифр, отмеченных «L», должно делиться на A , число из цифр, отмеченных «R» — на B , оба станка должны получить хотя бы одну цифру, а величина $|l - r|$ должна быть минимальной. Если подходящих распределений несколько, выведите любое из них.

Пример

стандартный ввод	стандартный вывод
4	RRLRL
5 3 13	-1
02165	RRRRLLLL
4 2 1	RL
1357	
8 1 1	
12345678	
2 7 9	
90	

Замечание

В первой ленте ($n = 5$, $A = 3$, $B = 13$, цифры 02165) распределение RRLRL направляет на левый станок цифры 1 и 5 (число 15 делится на 3), а на правый — цифры 0, 2, 6 (число 026 делится на 13). При этом $|l - r| = |2 - 3| = 1$, и меньшего значения добиться нельзя.

Во второй ленте среди цифр нет ни одной чётной, поэтому собрать число, делящееся на 2, невозможно — ответ -1.

В третьей ленте $A = B = 1$, поэтому любое число делится на нужный делитель. Достаточно отдать по 4 цифры каждому станку, тогда $|l - r| = 0$.

В четвёртой ленте подходящее распределение единственно.