

Задача А. 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. Но «Моногон» требует одно интересное правило: номера задач в контесте должны образовывать 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 В. Распил брусьев

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

Вам нужно распилить деревянный брус на несколько кусков в заданных местах. Распиловочная 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

<code>bars.in</code>	<code>bars.out</code>
10 3 2 4 7	20
100 3 15 50 75	200

Задача С. Игра

Имя входного файла: `game.in`
Имя выходного файла: `game.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

На листке записано в одну строку N целых положительных чисел. Каждое число не превышает 200. Играют двое. За каждый ход можно зачеркивать крайнее число либо слева, либо справа. Зачеркнутое число добавляется к очкам игрока. N – четное. Игру начинает первый игрок. Необходимо вывести максимально возможную сумму очков для первого игрока при условии, что противник играет наилучшим образом.

Формат входных данных

В первой строке входного файла содержится одно целое число N ($2 \leq N \leq 10000$). В следующих N строках записан исходный ряд чисел, по одному числу в строке.

Формат выходных данных

Выходной файл должен содержать единственное число – максимально возможную сумму очков для первого игрока при наилучшей игре второго игрока.

Примеры

<code>game.in</code>	<code>game.out</code>
4 4 7 2 9	16

Задача D. Удаление скобок - 2

Имя входного файла: `erase2.in`
Имя выходного файла: `erase2.out`
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

<code>erase2.in</code>	<code>erase2.out</code>
<code>([]]</code>	<code>[]</code>
<code>{([({})])}</code>	<code>([]{})</code>
<code>]{}[</code>	

Задача Е. 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 F. Покраска забора

Имя входного файла: `paint.in`
Имя выходного файла: `paint.out`
Ограничение по времени: 0.7 секунда
Ограничение по памяти: 256 мегабайт

Мэр города Многоярославца решил построить перед своим домом забор из n деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время. Оставить какую-то из досок непокрашенной, естественно, запрещается.

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 300$), где n количество досок в заборе. Вторая строка содержит строку из n символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

Формат выходных данных

В первой строке выходного файла выведите m - наименьшее возможное время покраски забора в часах. Следующие m строк должны содержать программу покраски для робота. Каждая строка должна содержать два числа l_i и r_i , а также заглавную букву латинского алфавита, задающую цвет c_i и означает, что робот должен покрасить участок забора с l_i по r_i - доску в цвет c_i (если длина забора n , должно выполняться $1 \leq l_i \leq r_i \leq n$).

Примеры

<code>paint.in</code>	<code>paint.out</code>
5 ABBCA	3 1 5 A 4 4 C 2 3 B
2 AA	1 1 2 A

Задача G. Дубы

Имя входного файла: `oaks.in`
Имя выходного файла: `oaks.out`
Ограничение по времени: 1.5 секунды
Ограничение по памяти: 256 мегабайт

На аллее перед зданием Министерства Обороны в ряд высажены n дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

Формат входных данных

Первая строка входного файла содержит целое число n — количество дубов, растущих на аллее ($2 \leq n \leq 200$). Вторая строка содержит n чисел — высоты дубов, приведённые слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

Формат выходных данных

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число -1 .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число m — минимальное количество дубов, которые необходимо срубить. В следующие m строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до n .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

Примеры

<code>oaks.in</code>	<code>oaks.out</code>
5	2
3 2 4 8 5	2
	4

Задача Н. Выбери поменьше

Имя входного файла: `minimal.in`
Имя выходного файла: `minimal.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Задано число k . Рассматривается множество всех натуральных чисел, имеющих сумму попарных произведений цифр, равную k (такие $m = \overline{a_l \dots a_1 a_0}$, что $\sum_{0 \leq i < j \leq l} a_i a_j = k$). Нужно найти минимальное число в этом множестве.

Формат входных данных

Во входном файле содержится одно целое число k ($1 \leq k \leq 5000$) — требуемая сумма попарных произведений цифр.

Формат выходных данных

В выходной файл должно быть выведено минимально возможное число (без лидирующих нулей и пробелов), имеющее сумму попарных произведений цифр, равную k . Если такого числа не существует, выведите -1 .

Примеры

<code>minimal.in</code>	<code>minimal.out</code>
3	13
58	2236