

Задача А. Разрезание Строки

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

Дана строка s длины n , состоящая из строчных латинских букв. Будем обозначать i -й символ строки как s_i . Пусть $d(l, r)$ — количество различных символов среди $s_l, s_{l+1}, \dots, s_r$. Назовем *стоимостью* отрезка строки $[l, r]$ величину, равную $(d(l, r))^2$.

От вас требуется разбить строку на непересекающиеся отрезки $[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]$, чтобы были выполнены условия $l_1 = 1$, $r_i + 1 = l_{i+1}$, и $r_k = n$, и при этом сумма стоимостей отрезков, то есть величина $\sum_{i=1}^k (d(l_i, r_i))^2$, была минимальна.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 3 \cdot 10^5$) — длина строки s .

Далее следует строка s длины n , состоящая из строчных латинских букв.

Формат выходных данных

В единственной строке выведите одно число — минимальную стоимость разбиения строки на отрезки.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	потестовая	—
1	20	$1 \leq n \leq 100$	подзадача	—
2	30	$1 \leq n \leq 10^3$	подзадача	1
3	30	$1 \leq n \leq 10^5$	подзадача	1, 2
4	20	Дополнительных ограничений нет	подзадача	1, 2, 3

Примеры

стандартный ввод	стандартный вывод
4 abcd	4
3 aaa	1
13 abababcbababab	9

Замечание

В первом примере оптимальное разбиение выглядит так: $[1, 1], [2, 2], [3, 3], [4, 4]$. Стоимость этого разбиения равна $1^2 + 1^2 + 1^2 + 1^2 = 4$.

Во втором примере оптимальным является разбиение, состоящее из одного отрезка $[1, 3]$. Стоимость этого разбиения равна $1^2 = 1$.

В третьем примере можно разбить строку на три отрезка $[1, 6], [7, 7]$ и $[8, 13]$. Стоимость разбиения равна $2^2 + 1^2 + 2^2 = 9$.

Задача В. Расстояние по Левенштейну

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

Дана текстовая строка. С ней можно выполнять следующие операции:

1. Заменить один символ строки на другой символ.
2. Удалить один произвольный символ.
3. Вставить произвольный символ в произвольное место строки.

Например, при помощи первой операции из строки «СОК» можно получить строку «ТОК», при помощи второй операции — строку «ОК», при помощи третьей операции — строку «СТОК».

Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна.

Определите расстояние Левенштейна для двух данных строк.

Формат входных данных

Программа получает на вход две строки, длина каждой из которых не превосходит 1000 символов, строки состоят только из заглавных латинских букв.

Формат выходных данных

Требуется вывести одно число — расстояние Левенштейна для данных строк.

Пример

стандартный ввод	стандартный вывод
ABCDEFGH ACDEXGIH	3

Задача С. Почтальон

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

Биттаун – маленький, тихий, дружелюбный городок в Байтеландии, жителям которого приходит много писем. Здесь все N домов расположены по одну сторону одной длинной улицы и пронумерованы различными целыми числами от 1 до N в произвольном порядке, т.е. не обязательно по возрастанию их номеров.

Перед Днем Святого Байта в каждый дом приходит ровно одно поздравительное письмо.

Когда почтальон выходит из отделения почты, у него все письма сложены в стопку в порядке возрастания номеров домов, т.е. сверху лежит письмо, которое необходимо отнести в дом с номером 1, а снизу – в дом с номером N . В день перед праздником почтальон хочет доставить как можно больше писем, а те, которые он не доставит сразу, он сможет разнести на следующий день, когда отправлений будет меньше.

Почтальон начинает маршрут с дома, расположенного в начале улицы. Останавливаясь возле каждого дома, он смотрит на его номер. Далее он либо проходит мимо, либо откладывает некоторое количество писем сверху стопки в сумку, если наверху стопки оказывается письмо, адресованное в этот дом, то он вкладывает его в почтовый ящик и идет дальше. Почтальон никогда не берёт письма, отложенные в сумку, он отнесёт их в другой раз, возможно, на следующий день.

Например, если почтальон подошёл к дому с номером 5, а верхнее письмо в его стопке адресовано в дом с номером 2, то он либо уберёт 3 письма в сумку (адресованные в дома с номерами 2, 3 и 4) и опустит письмо в почтовый ящик дома с номером 5, либо пройдёт мимо этого дома, не перекладывая писем в сумку.

За один день почтальон проходит вдоль улицы K раз, поэтому, когда почтальон доходит до конца улицы, он разворачивается и идёт в обратную сторону, продолжая раскладывать письма по почтовым ящикам. Дойдя до начала улицы, он снова может развернуться.

Почтальон хочет разносить за день как можно больше писем, но не знает, как это сделать. Он просит вас подсказать ему оптимальное решение, т.е. определить наибольшее количество писем, которые он сможет разнести за день.

Формат входных данных

Первая строка входных данных содержит два числа N и K ($1 \leq N \leq 10^5, 1 \leq K \leq 100$) — количество домов в Биттауне и количество проходов почтальоном по улице за один день соответственно.

Вторая строка содержит перестановку N целых чисел — номера домов в порядке от начала к концу улицы в Биттауне.

Формат выходных данных

Выведите наибольшее количество писем, которое сможет разнести почтальон за один день.

Примеры

стандартный ввод	стандартный вывод
5 1 1 3 5 2 4	3
7 2 1 7 2 6 3 5 4	7

Задача D. НВП

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

Числовая последовательность задана рекуррентной формулой: $a_{i+1} = (k \cdot a_i + b) \bmod m$. Найдите её наибольшую возрастающую подпоследовательность. Если таких последовательностей несколько, можно вывести любую.

Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности n ($1 \leq n \leq 10^5$), начальный элемент последовательности a_1 , параметры k, b, m для вычисления последующих членов последовательности ($1 \leq m \leq 10^4, 0 \leq k < m, 0 \leq b < m, 0 \leq a_1 < m$).

Формат выходных данных

На первой строке выходного файла вы должны вывести количество чисел в найденной вами наибольшей возрастающей подпоследовательности. На следующей строке выведите элементы подпоследовательности, разделяя их пробелами.

Примеры

стандартный ввод	стандартный вывод
5 41 2 1 100	3 41 67 71
7 1 2 1 10	4 1 3 5 7
7 2 2 1 10	3 1 3 5

Задача Е. Миллиардная Функция Васи

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

Вася — начинающий математик — решил сделать вклад в развитие этой науки и прославиться на весь мир. Но как это сделать, когда самые интересные факты, типа теоремы Пифагора, давно уже доказаны? Правильно! Придумать что-то свое, оригинальное. Вот юный математик и придумал Теорию Функций Васи, посвященную изучению поведения этих самых функций. Функции Васи (ФВ) устроены довольно просто: значением N -й ФВ в точке S будет количество чисел от 1 до N , имеющих сумму цифр S . Вам, как крутым программистам, Вася поручил найти значения миллиардной ФВ (то есть ФВ с $N = 10^9$), так как сам он с такой задачей не справится. А Вам слабо?

Формат входных данных

В единственной строке записано целое число S ($1 \leq S \leq 81$).

Формат выходных данных

Выведите одно число — значение миллиардной Функции Васи в точке S .

Примеры

стандартный ввод	стандартный вывод
1	10
2	45
3	165

Задача F. Плавные числа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Назовем число *плавным*, если любые его две соседние цифры различаются не более, чем на 1. По данному натуральному n определите количество плавных натуральных чисел, имеющих длину n . Гарантируется, что ответ не превосходит $2^{31} - 1$.

У плавного числа нет лидирующих нулей.

Формат входных данных

Вводится натуральное число $n \leq 20$.

Формат выходных данных

Выведите количество плавных чисел.

Примеры

стандартный ввод	стандартный вывод
1	9
3	75

Задача G. Трипростые числа

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

Будем называть натуральное число трипростым, если в нем любые подряд идущие 3 цифры образуют трехзначное простое число. Требуется по данному N найти количество N -значных трипростых чисел.

Формат входных данных

На вход подаётся одно натуральное число N : ($3 \leq N \leq 10^4$).

Формат выходных данных

Ответ должен содержать количество N -значных трипростых чисел, которое следует вывести по модулю $10^9 + 9$.

Примеры

стандартный ввод	стандартный вывод
3	143
4	204
4793	851557205

Задача Н. Наибольшая общая возрастающая

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Даны две последовательности чисел — a и b . Нужно найти наибольшую общую возрастающую подпоследовательность. Более формально: такие $1 \leq i_1 < i_2 < \dots < i_k \leq a.n$ и $1 \leq j_1 < j_2 < \dots < j_k \leq b.n$, что $\forall t : a_{i_t} = b_{j_t}, a_{i_t} < a_{i_{t+1}}$ и k максимально.

Формат входных данных

На первой строке целые числа n и m от 1 до 3 000 — длины последовательностей. Вторая строка содержит n целых чисел, задающих первую последовательность. Третья строка содержит m целых чисел, задающих вторую последовательность. Все элементы последовательностей — целые неотрицательные числа, не превосходящие 10^9 .

Формат выходных данных

Выведите одно целое число — длину наибольшей общей возрастающей подпоследовательности.

Пример

стандартный ввод	стандартный вывод
6 5 1 2 1 2 1 3 2 1 3 2 1	2

Задача I. 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 $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
1 100	54

Задача J. Раскраска графа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дан граф из n вершин, раскрасьте его в минимально возможное число цветов так, чтобы никакие две вершины, соединенные ребром, не были одного цвета.

Формат входных данных

В первой строке содержится число t — количество тестовых примеров ($1 \leq t \leq 5$).

Далее содержится t тестовых случаев, заданных в следующем формате:

В первой строке записаны числа n и m — количество вершин и ребер соответственно ($1 \leq n \leq 17$, $0 \leq m \leq \frac{n \cdot (n-1)}{2}$).

Затем идет m строк, в которых содержится по два числа v_i u_i , что означает, что вершины v_i и u_i соединены ребром ($1 \leq v_i, u_i \leq n, v_i \neq u_i$).

Гарантируется, что все ребра в каждом тестовом случае различны.

Формат выходных данных

Для каждого тестового случая в первой строке выведите минимальное число цветов k .

Во второй строке выведите n чисел a_i — цвета вершин ($1 \leq a_i \leq k$).

Пример

стандартный ввод	стандартный вывод
3	3
3 3	3 2 1
1 2	2
2 3	1 2 2 1 1
3 1	3
5 3	1 3 1 1 2 1
2 1	
3 1	
4 2	
6 7	
1 2	
1 5	
2 5	
2 3	
2 4	
5 6	
5 4	

Задача К. Деловые встречи

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

Алексей — успешный предприниматель, и в течение одного дня у него бывает много встреч с разными деловыми партнёрами. К сожалению, встречи бывают разные и не все приносят ему радость, после других же настроение улучшается. Также, на многие встречи не стоит приходить в слишком плохом или хорошем настроении — результат таких встреч может быть не таким, какой хочется Алексею.

К счастью, недавно Алексей научился оценивать своё настроение с помощью целых чисел. После этого для каждой встречи он оценил, при каком максимальном и минимальном настроении стоит на неё приходить, а также как изменится его настроение после этой встречи. Теперь он хочет распланировать порядок встреч так, чтобы в течение дня совершить максимальное число встреч.

Ваша задача — написать программу, которая по информации о всех встречах и настроении Алексея в начале дня находит порядок проведения встреч такой, что их количество при этом максимально.

Формат входных данных

Первая строка входного файла содержит два целых числа n и k ($1 \leq n \leq 20$, $-100 \leq k \leq 100$) — количество встреч и настроение Алексея в начале дня.

Следующие n строк содержат по три целых числа a_i , b_i и c_i ($-100 \leq a_i, b_i, c_i \leq 100$) — минимальное и максимальное настроение, при котором встреча возможна, и изменение настроения по окончании встречи, соответственно.

Формат выходных данных

В первой строке выходного файла выведите число m — максимально возможно число встреч. В следующей строке выведите m целых чисел — номера встреч в порядке их проведения. Встречи пронумерованы в порядке описания во входном файле.

Если ответов с максимальным числом встреч несколько, выведите любой.

Примеры

стандартный ввод	стандартный вывод
3 0 1 3 3 0 1 2 1 3 1	3 2 3 1
3 1 -10 -5 3 -5 5 -2 -3 2 1	2 3 2