Содержание

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь примерами.

В некоторых задачах большой ввод и вывод. Пользуйтесь быстрым вводом-выводом.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу.

Обратите внимание на компилятор `GNU C++11 5.1.0 (TDM-GCC-64) inc`, который позволяет пользоваться дополнительной библиотекой. Под ним можно сдать вот это.

Разделяй и властвуй

Задача A. Деление многочленов [1 sec, 256 mb]

Даны два многочлена с коэффициентами из $\mathbb{Z}/7\mathbb{Z}$. Старший коэффициент обоих не равен нулю. Нужно поделить их с остатком.

Формат входных данных

Каждая из двух строк задаёт описание многочлена. Многочлен $a_k x^k + \cdots + a_2 x^2 + a_1 x + a_0$ описывается числом k ($0 \leqslant k \leqslant 50\,000$) и $k+1$ числами от 0 до 6: $a_k, \ldots, a_2, a_1, a_0$.

Формат выходных данных

На первой строке многочлен-частное. На второй строке многочлен-остаток. Выводите многочлены в том же формате. Если многочлен – тождественный ноль, для него $k = 0$.

Примеры

divpoly.in	divpoly.out
3 1 1 1 1 1 1 1	2 1 0 1 0 0
3 1 1 3 1 2 1 1 1	1 1 0 1 2 1
8 2 1 2 1 2 1 2 1 2 4 1 2 3 4 5	4 2 4 2 5 2 3 3 1 3 6

Задача A1. FFT и манная кашка [1 sec, 256 mb]

Даны два многочлена
$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$
$B(x) = b_m x^m + b_{m-1} x^{m-1} + \cdots + b_0$
Найти $C(x) = A(x)B(x)$

Формат входных данных

$n, a_n, a_{n-1}, \ldots, a_0$
$m, b_m, b_{m-1}, \ldots, b_0$
$0 \leqslant n, m < 2^{16}$, $|a_i|, |b_j| \leqslant 9$
$a_n \neq 0, b_m \neq 0$

Формат выходных данных

Выведите коэффициенты C в том же формате.

Примеры

mulpoly.in	mulpoly.out
2 1 0 2 1 2 3	3 2 3 4 6

Суффиксный автомат

Задача B1. Суффиксный автомат [1 sec, 256 mb]

Или зачёт, или автомат.

Ганнибал Ректор

Дана строка. Постройте её суффиксный автомат.

Формат входных данных

Строка длины от 1 до 100 000, состоящая из маленьких латинских букв.

Формат выходных данных

На первой строке число состояний автомата и число рёбер. Следующие строки содержат рёбра в формате “откуда” “куда” “символ на ребре”. Далее число терминальных состояний и строка, содержащая все терминальные состояния в произвольном порядке. Начальным состоянием автомата должно быть состояние номер один.

Примеры

automaton.in	automaton.out
ababb	7 9 1 2 a 1 7 b 2 3 b 3 4 a 3 6 b 4 5 b 5 6 b 7 4 a 7 6 b 3 6 7 1

Задача B2. Суффиксный пулемёт [0.5 sec, 256 mb]

Тут была легенда по мотивам произведений Роберта Асприна,
НЛО прилетело и забрало её...

Вам дан автомат и строка s. Проверьте, является ли данный автомат суффиксным пулемётом строки s, то есть, детерминированным конечным автоматом, принимающим ровно суффиксы строки s.

Формат входных данных

Во входном файле задан один или несколько тестовых наборов. В первой строке каждого набора заданы количество состояний автомата N, количество переходов M, а также количество принимающих состояний T ($1 \leqslant T \leqslant N \leqslant 50\,000$, $1 \leqslant M \leqslant 100\,000$). Во второй строке через пробел заданы T различных чисел в пределах от 1 до N — принимающие состояния автомата, в возрастающем порядке. В последующих M строках заданы переходы в виде a_i b_i c_i, где $1 \leqslant a_i, b_i \leqslant n$, а c_i — маленькая буква латинского алфавита. Переход производится из состояния a_i в состояние b_i по букве c_i. Из каждого состояния a_i есть не более одного перехода по символу c_i. Последняя строка описания набора — это строка S, для которой автомат должен являться суффиксным. Она состоит только из маленьких латинских букв, и ее длина лежит в пределах от 1 до 50 000 включительно. Кроме того, сумма всех N и суммарная длина всех строк, для которых необходимо произвести проверку, не превосходит 50 000, а сумма всех M не превосходит 100 000.

Файл заканчивается фиктивным набором, в котором $N = M = T = 0$.

Начальным состоянием автомата является первое. Если при интерпретации какой-то строки в автомате отсутствует соответствующий переход, то автомат вываливается по ошибке и строку не принимает. Таким образом, строка принимается, только если при её интерпретации были найдены все переходы, и по их завершении автомат оказался в принимающем состоянии (при этом неважно, были по пути принимающие состояния, или нет).

Формат выходных данных

Выведите в выходной файл, является ли данный автомат пулемётом, следуя формату примера.

Пример

suffix.in
2 1 2 1 2 1 2 a a 2 2 2 1 2 1 1 a 1 2 b ab 0 0 0
suffix.out
Automaton 1 is a machinegun. Automaton 2 is not a machinegun.

Задача B3. Помогите, спасите! [1 sec, 256 mb]

Дана строка. Найдите для каждого её префикса количество различных подстрок в нём.

Формат входных данных

В единственной строке входных данных содержится непустая строка S, состоящая из N ($1 \le N \le 2 \cdot 10^5$) маленьких букв английского алфавита.

Формат выходных данных

Выведите N строк, в i-й строке должно содержаться количество различных подстрок в i-м префиксе строки S.

Примеры

keepcounted.in	keepcounted.out
aabab	1 2 5 8 11
atari	1 3 5 9 14

Задача B4. Подстроки-4 [1 sec, 256 mb]

Даны K строк из маленьких латинских букв. Найдите их наибольшую общую подстроку.

Формат входных данных

В первой строке число K ($1 \leqslant K \leqslant 10$). Далее K строк длины от 1 до 200 000.

Формат выходных данных

Наибольшая общая подстрока.

Примеры

substr4.in	substr4.out
3 abacaba mycabarchive acabistrue	cab

Задача B5. LZSS encoding [2 sec, 256 mb]

Алиса хочет отправить сообщение Бобу. Она хочет зашифровать сообщение, используя оригинальный метод шифрования. Сообщение – строка S, состоящая из N строчных английских букв.

$S[a \dots b]$ означает подстроку S от $S[a]$ до $S[b]$ $(0 \leqslant a \leqslant b < N)$. Если первые i букв уже зашифрованы, Алиса найдёт такие (j, k): $s[j..j+k] = s[i..i+k], k \geqslant 0, 0 \leqslant j < i, k = \max$. Если несколько j дают максимальное k, Алиса выберет минимальное j. Если $k > 0$ Алиса добавит пару $\langle j, k \rangle$ в шифр и увеличит i на k, иначе Алиса добавит -1 и ASCII код буквы $S[i]$ в шифр и увеличит i на 1.

Очевидно шифр начнёт с -1, далее будет ASCII код символа $S[0]$. Помогите Алисе реализовать её метод шифрования.

Формат входных данных

Перрвая строка ввода содержит количество тестов T $(1 \leqslant T \leqslant 50)$. Следующие T строк содержат сообщения для шифровки, каждое длины от 1 до 10^5, состоящие из строчных английских букв. Гарантируется, что суммарная длина всех сообщений не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого теста на отдельной строке выведите "`Case #X:`", где X – номер теста, нумерация с 1. Далее выведите шифр, в каждой строке по два целых числа через пробел.

Примеры

lzss.in	lzss.out
2 aaaaaa aaaaabbbbbaaabbc	Case #1: -1 97 5 0 Case #2: -1 97 4 0 -1 98 4 5 5 2 -1 99

Паросочетания

Задача C1. Рабочее расписание [0.2 sec, 256 mb]

Дан неориентированный граф из N вершин и нескольких рёбер.
Найти максимальное паросочетание.
Здесь можно прочесть оригинальную легенду.

Формат входных данных

На первой строке число вершин N ($1 \leqslant N \leqslant 222$).
Далее строки, содержащие рёбра.

Формат выходных данных

На первой строке число вершин, покрытых максимальным паросочетанием.
На следующих строках рёбра паросочетания.
Если есть несколько максимальных паросочетаний, выведите любое.

Примеры

schedule.in	schedule.out
3 1 2 2 3 1 3	2 1 2

Подсказка по решению

Алгоритм Эдмондса сжатия соцветий. Реализация Габова.

Линейное программирование

Задача D1. Простая задача [0.2 sec, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + \ldots + a_{2n}x_n \leqslant b_2 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ x_1 \geqslant 0 \\ \ldots \\ x_n \geqslant 0 \\ c_1x_1 + \ldots + c_nx_n \to \max \end{cases}$$

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество переменных и количество уравнений ($1 \leqslant n, m \leqslant 5$). Следующие m строк содержат по $n+1$ целому числу: $a_{i1}, \ldots, a_{in}, b_i$. Следующая строка содержит n целых чисел: $c_1, \ldots, c_n$. Все числа во входном файле не превышают 100 по модулю.

Формат выходных данных

Выведите одно число: максимальное значение $c_1x_1+\ldots+c_nx_n$. Гарантируется, что решение существует и максимальное значение достигается.

Пример

simple.in	simple.out
2 2 1 2 3 2 1 3 1 1	2.0
1 1 1 2 -2	0.0
1 1 4 5 3	3.75

Задача D2. Простая задача [0.2 sec, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + \ldots + a_{2n}x_n \leqslant b_2 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ x_1 \geqslant 0 \\ \ldots \\ x_n \geqslant 0 \\ c_1x_1 + \ldots + c_nx_n \to \max \end{cases}$$

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество переменных и количество уравнений ($1 \leqslant n, m \leqslant 5$). Следующие m строк содержат по $n+1$ целому числу: $a_{i1}, \ldots, a_{in}, b_i$. Следующая строка содержит n целых чисел: $c_1, \ldots, c_n$. Все числа во входном файле не превышают 100 по модулю.

Формат выходных данных

Выведите одно число: максимальное значение $c_1x_1+\ldots+c_nx_n$. Если решения нет, выведите "`No solution`". Если можно получить сколь угодно большое значение, выведите "`Unbounded`".

Пример

simple2.in	simple2.out
2 2 1 2 3 2 1 3 1 1	2.0
2 1 -1 -1 0 1 1	Unbounded
2 1 1 1 -1 1 1	No solution

Задача D3. Простая задача с хорошими тестами [0.2 sec, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + \ldots + a_{2n}x_n \leqslant b_2 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ x_1 \geqslant 0 \\ \ldots \\ x_n \geqslant 0 \\ c_1x_1 + \ldots + c_nx_n \to \max \end{cases}$$

Формат входных данных

Входной файл содержит один или несколько тестов. На первой строке число тестов, далее сами тесты. Каждый тест описывается следующим образом. Первая строка теста содержит два целых числа: n и m — количество переменных и количество неравенств ($1 \leqslant n, m \leqslant 5$). Следующие m строк содержат по $n + 1$ целому числу: $a_{i1}, \ldots, a_{in}, b_i$. Следующая строка содержит n целых чисел: $c_1, \ldots, c_n$. Все числа во входном файле целые и не превышают 100 по модулю.

Формат выходных данных

Для каждого теста выведите одну строку.

Выведите максимальное значение $c_1x_1 + \ldots + c_nx_n$, пробел, двоеточие, пробел, сами числа $x_1x_2 \ldots x_n$. Все числа выводите с максимальной точностью. Если решения нет, выведите "`No solution`". Если можно получить сколь угодно большое значение, выведите "`Unbounded`".

Пример

simple3.in	simple3.out
4 2 2 1 2 3 2 1 3 1 1 2 1 -1 -1 0 1 1 2 1 1 1 -1 1 1 2 2 1 2 3 2 1 3 -10 1	2.0 : 1.0 1.0 Unbounded No solution 1.5 : 0 1.5