

Содержание

1	Задача А. Делители	2
2	Задача В. Интересное число	3
3	Задача С. Path. Кратчайший путь	4
4	Задача D. Минимизируй его!	5
5	Задача Е. Сложение и вычитание	6
6	Задача F. Обратная инверсия-2	7

Гробовые задачи

1 Задача А. Делители

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

Дано число K . Найти минимальное положительное число N такое, что количество натуральных делителей N равно K .

Формат входных данных

Входной файл содержит единственное целое число K . ($1 \leq K \leq 10^5$).

Формат выходных данных

Выведите искомое число N . Если такого N не существует, выведите 0.

Пример

<code>formation.in</code>	<code>formation.out</code>
4	6

2 Задача В. Интересное число

Имя входного файла: `number.in`
Имя выходного файла: `number.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Для заданного числа n найдите наименьшее положительное целое число с суммой цифр n , которое делится на n .

Формат входных данных

Во входном файле содержится целое число n ($1 \leq n \leq 1000$).

Формат выходных данных

Выходной файл должен содержать искомое число. Ведущие нули выводить не разрешается.

Пример

<code>number.in</code>	<code>number.out</code>
1	1
10	190

3 Задача C. Path. Кратчайший путь

Имя входного файла: path.in
Имя выходного файла: path.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 megabytes

Дан взвешенный ориентированный граф и вершина s в нем. Требуется для каждой вершины u найти длину кратчайшего пути из s в u .

Формат входных данных

Первая строка входного файла содержит n , m и s — количество вершин, ребер и номер выделенной вершины соответственно ($2 \leq n \leq 2000$, $1 \leq m \leq 5000$).

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее 10^{15} по модулю. В графе могут быть кратные ребра и петли.

Формат выходных данных

Выведите n строк — для каждой вершины u выведите длину кратчайшего пути из s в u , '*' если не существует путь из s в u и '-' если не существует кратчайший путь из s в u .

Пример

path.in	path.out
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	

4 Задача D. Минимизируй его!

Имя входного файла: `minimize.in`
Имя выходного файла: `minimize.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Саша устал играть со своими палочками и нашел более интересное занятие. Он написал на доске все числа от 1 до N в одну строчку (так он получил очень большое число) и теперь хочет стереть ровно M цифр таким образом, чтобы оставшиеся написанное число было как можно меньше. Помогите ему!

Формат входных данных

Входной файл содержит два целых числа M и N ($0 \leq M \leq 30000$, $1 \leq N \leq 10000$). Входные данные корректны, поэтому по крайней мере одна цифра будет содержаться в выходных данных.

Формат выходных данных

Выходной файл должен содержать минимальное число, которое Саша может получить на доске, вычеркнув M цифр. Лидирующие нули допустимы.

Пример

<code>minimize.in</code>	<code>minimize.out</code>
12	0111114151617181920
20	

5 Задача Е. Сложение и вычитание

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

Выведите значение заданного арифметического выражения, состоящего из чисел, скобок и знаков сложения и вычитания.

Формат входных данных

В первой строке входного файла задано выражение, состоящее из чисел, скобок и знаков бинарных операций. Каждое число в выражении это — целое неотрицательное число в промежутке от 0 до 10 000, включительно, записанное без ведущих нулей. Скобки бывают открывающие ('(') и закрывающие (')'). Операции задаются символами '+' и '-'. Гарантируется, что заданное выражение математически корректно, и результаты всех промежуточных операций — целые числа, не превышающие по модулю 10 000. Выражение не содержит каких-либо других символов, в частности, пробелов. Длина выражения не меньше 1 и не больше 1000 символов.

Учтите, что операции при отсутствии скобок выполняются слева направо. Например, выражение $a - b - c$ вычисляется как $(a - b) - c$.

Формат выходных данных

В первой строке выходного файла выведите одно число — значение заданного выражения.

Примеры

<code>evalpm.in</code>	<code>evalpm.out</code>
48-13	35
5-(52+3)	-50

6 Задача F. Обратная инверсия-2

Имя входного файла: `invers2.in`
Имя выходного файла: `invers2.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Таблицей инверсий для перестановки $A = (a_1, a_2, \dots, a_n)$ чисел $\{1, 2, \dots, N\}$ называется массив $X = (x_i)_{1 \leq i \leq N}$, в котором на i -м месте стоит количество элементов, больших i , но стоящих левее, чем i , т.е. $x_i = \text{число таких } j', \text{ что } j' < j, a_{j'} > a_j = i$.

Например, таблицей инверсий для перестановки $(2, 5, 1, 3, 4)$ будет $(2, 0, 1, 1, 0)$, а для перестановки $(6, 1, 3, 7, 5, 4, 2) = (1, 5, 1, 3, 2, 0, 0)$.

Обратной перестановкой A^{-1} к перестановке A называется такая перестановка чисел, что на i -м месте в A^{-1} стоит номер места, на котором стоит элемент, равный i , в перестановке A .

Например, для перестановки $(2, 5, 1, 3, 4)$ обратной будет $(3, 1, 4, 5, 2)$ (т.к. 1 стоит на третьем месте, 2 — на первом, 3 — на четвертом, 4 — на пятом, а 5 — на втором), а для перестановки $(2, 7, 3, 6, 5, 1, 4)$ обратной будет $(6, 1, 3, 7, 5, 4, 2)$.

Ваша задача — по таблице инверсий перестановки A посчитать таблицу инверсий обратной перестановки A^{-1} .

Формат входных данных

Файл состоит ровно из N чисел, разделенных пробелами и переводами строки, задающих таблицу инверсий перестановки A . Число N находится в пределах от 1 до **262 144**.

Формат выходных данных

Выведите N целых чисел, разделенных пробелами — таблицу инверсий для обратной перестановки.

Пример

<code>invers2.in</code>	<code>invers2.out</code>
2 0 1 1 0	1 3 0 0 0
5 0 1 3 2 1 0	1 5 1 3 2 0 0