

Задача А. Минимум на стеке

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

Вам требуется реализовать структуру данных, выполняющую следующие операции:

1. Добавить элемент x в конец структуры.
2. Удалить последний элемент из структуры.
3. Выдать минимальный элемент в структуре.

Формат входных данных

В первой строке входного файла задано одно целое число n — количество операций ($1 \leq n \leq 10^6$). В следующих n строках заданы сами операции. В i -ой строке число t_i — тип операции (1, если операция добавления, 2, если операция удаления, 3, если операция минимума). Если задана операция добавления, то через пробел записано целое число x — элемент, который следует добавить в структуру ($-10^9 \leq x \leq 10^9$). Гарантируется, что перед каждой операцией удаления или нахождения минимума структура не пуста.

Формат выходных данных

Для каждой операции нахождения минимума выведите одно число — минимальный элемент в структуре. Ответы разделяйте переводом строки.

Примеры

<code>stack.in</code>	<code>stack.out</code>
8	-3
1 2	2
1 3	2
1 -3	
3	
2	
3	
2	
3	

Задача В. НОП-2

Имя входного файла: `lcs2.in`
Имя выходного файла: `lcs2.out`
Ограничение по времени: 0.25 секунда
Ограничение по памяти: 256 мегабайт

Даны две строки. Найдите их наибольшую общую подпоследовательность.

Формат входных данных

Во входном файле находятся две строки, состоящие только из маленьких букв английского алфавита. Длина каждой из строк не превышает 1000.

Формат выходных данных

Выведите одну строку — ответ на задачу.

Примеры

<code>lcs2.in</code>	<code>lcs2.out</code>
<code>abacaba</code> <code>dacabc</code>	<code>acab</code>
<code>sislksh</code> <code>lkshsis</code>	<code>lksh</code>

Задача С. Топологическая сортировка

Имя входного файла: `topsort.in`
Имя выходного файла: `topsort.out`
Ограничение по времени: 0.5 секунда
Ограничение по памяти: 256 мегабайт

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входных данных

В первой строке входного файла даны два целых числа N и M ($1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$) — количества вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходных данных

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

Примеры

<code>topsort.in</code>	<code>topsort.out</code>
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

Задача D. Отрезок с максимальной суммой

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана последовательность из n целых чисел: a_1, \dots, a_n . Сумма на подотрезке с l по r ($l \leq r$) равна $a_l + \dots + a_r$. Найдите максимальную сумму среди всех возможных подотрезков данной последовательности.

Формат входных данных

В первой строке ввода записано одно целое положительное число n ($1 \leq n \leq 500\,000$).

Во второй строке ввода записано n целых чисел, разделенных пробелами — элементы последовательности a_i ($-1000 \leq a_i \leq 1000$).

Формат выходных данных

Выведите одно целое число — максимально возможную сумму среди всех подотрезков данной последовательности.

Примеры

stdin	stdout
3 -1 -2 -3	-1
5 2 -1 2 -1 2	4

Задача E. Расстояния от точки

Имя входного файла: `distance1.in`
Имя выходного файла: `distance1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайта

Даны три точки A, B и C . Необходимо подсчитать расстояния от точки C до прямой, луча и отрезка, образованного точками A и B .

Формат входных данных

В первой строке входного файла даны два целых числа — координаты точки C . Во двух следующих строках в таком же формате заданы точки A и B ($A \neq B$).

Все числа во входном файле по модулю не превосходят 10 000.

Формат выходных данных

В первой строке выходного файла выведите одно вещественное число — расстояние от точки C до прямой. В следующих двух строках выведите соответственно расстояния до луча AB (A — начало луча) и до отрезка AB . Все числа выводить с точностью не менее 10^{-6} . Луч строится по направлению от точки A к точке B .

Примеры

<code>distance1.in</code>	<code>distance1.out</code>
3 0	1.000000000
1 1	1.000000000
2 1	1.414213562

Задача F. Транспортировка

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

Компания “Яндекс” решила подарить всем школьникам и преподавателям ЛКШ оригинальные кружки. К сожалению, количество необходимых кружек оказалось столь велико, что изготовитель доставил кружки в офис Яндекса в самый последний момент. До открытия смены в “Берендеевых полянах” осталось всего 24 часа.

О плачевном состоянии дорог по пути на базу ходят легенды — в частности, на многих разбитых дорогах действует ограничение на вес автомобиля. Соответственно, от нагруженности машины зависит возможность воспользоваться тем или иным маршрутом, тяжёлой машине может потребоваться ехать в обход.

Уже совершенно очевидно, что все кружки не успеют к открытию. Чтобы спасти ситуацию, отвезите первым рейсом максимально возможное количество кружек успев до начала открытия смены.

Формат входных данных

В первой строке находятся целые числа n ($2 \leq n \leq 500$) и m — количество городов и количество двусторонних дорог, соответственно.

В следующих m строках описываются дороги.

В каждой строке находятся целые числа a_i, b_i, t_i, w_i — соответственно два города, ею соединяемые, время на проезд по ней в минутах и ограничение на вес автомобиля в граммах ($t_i \leq 1440$, $w_i \leq 10^9$, $1 \leq a_i, b_i \leq n$)

Между каждой парой городов есть не более одной дороги.

Кроме того, известно, что офис Яндекса имеет номер 1, а “Берендеевы поляны” — номер n , одна кружка весит 100 грамм, а пустой грузовик — 3 тонны.

Формат выходных данных

Выведите одно число — максимальное количество кружек, которое можно привезти, потратив не более 24 часов.

Примеры

<code>cups.in</code>	<code>cups.out</code>
3 3	2
1 2 10 3000220	
2 3 20 3000201	
1 3 1 3000099	