

Задача А. 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 В. Топологическая сортировка

Имя входного файла: `topsort.in`
Имя выходного файла: `topsort.out`
Ограничение по времени: 2 секунды
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 С. Слонёнок

Имя входного файла: elephant-sum.in
Имя выходного файла: elephant-sum.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Шёл 2034 год. ЛКШ продолжалась вечно. Тигрёнок (Илья Матюхин) вырос и привёз в ЛКШ своего сына, которого здесь все называют Слонёнком. Конечно же, он живёт в ГК и каждый день ест вечерку.

Исторически в ЛКШ сложилась следующая традиция поедания вечерки в ГК:

- По заранее заказанному списку, который определяется один раз на всю смену, все напитки выставляются в ряд слева направо. К 2034 году помимо привычных вам снежка, молочка, ряженки и кефирчика появились другие, не менее вкусные напитки.
- Преподаватели берут напитки с правого края и не берут очередной напиток, если ещё можно допить один из стоящих справа.
- Школьники действуют аналогично, но выбирают напитки слева направо.

Когда все школьники уходят спать, а преподаватели начинают их укладывать, в ГК появляется Слонёнок и наслаждается оставшейся вечеркой.

Так как школьники употребляют напитки слева направо, а преподаватели справа налево, то для Слонёнка остаётся какой-то подотрезок напитков.

От каждой вечерки Слонёнок получает определённое количество радости. Радость, полученная им от вечерки в определённый день может быть вычислена по следующей формуле:

$$H = \sum_{p=1}^{10000} p \cdot cnt(p)$$

Здесь p — вид вечерки (кефирчик, ряженка, сок, чай и другие), а $cnt(p)$ — количество напитка p среди тех, которые остались для Слонёнка. Как видно из формулы, удовольствие получаемое Слонёнком от данного вида вечерки пропорционально его номеру.

Смена в ЛКШ длится t дней. В день с номером i Слонёнку оставят напитки с номерами от l_i до r_i включительно.

От вас требуется посчитать, сколько радости получит Слонёнок от вечерки в каждый из дней.

Формат входных данных

В первой строке ввода записаны два числа n и t ($1 \leq n, t \leq 100\,000$) — количество напитков на вечерке в ГК и количество дней в смене соответственно.

Во второй строке ввода содержатся n чисел. i -е число описывает вид напитка в вечерке на позиции i . Все числа в этой строке целые положительные и не превосходят 10 000.

В следующих t строках записаны пары чисел, описывающие оставшуюся вечерку в каждый из дней смены. В i -й из данных строк содержится пара чисел l_i и r_i , которая означает, что в день с номером i Слонёнку оставят напитки из вечерки с номерами от l_i до r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Выведите t строк. В i -й из них выведите количество радости, полученной Слонёнком от вечерки в день с номером i .

Примеры

elephant-sum.in	elephant-sum.out
19 2	20
3 3 1 9 3 3 1 5 5 5 5 9 9 1 3 9 5 1 5	5
2 7	
10 10	

Замечание

В первый день Слонёнку достанутся 6 напитков, среди которых будут присутствовать виды 1, 3 и 9, встречающиеся 2, 3 и 1 раз соответственно. Удовольствие Слонёнка будет равно $1 \cdot 2 + 3 \cdot 3 + 9 \cdot 1 = 20$.

Во второй день Слонёнок сможет насладиться только одним напитком типа 5. Удовольствие Слонёнка будет равно $5 \cdot 1 = 5$.

Задача D. Задача Иосифа

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

N мальчиков стоят по кругу. Они начинают считать себя по часовой стрелке, счет ведется с единицы. Как только количество посчитанных достигает p , последний посчитанный (p -й) мальчик покидает круг, а процесс счета начинается со следующего за ним мальчика и вновь ведется с единицы.

Последний оставшийся в кругу выигрывает.

Можете ли вы посчитать, номер выигравшего мальчика в исходном кругу? (мальчики нумеруются числами от 1 до N по часовой стрелке, начиная с того самого мальчика, с которого начинался счет).

Формат входных данных

Во входном файле два целых числа — N и P ($1 \leq N, P \leq 10^5$).

Формат выходных данных

Выведите номер выигравшего мальчика.

Примеры

	joseph.in	joseph.out
	3 4	2

Задача Е. Сумма подмножества

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

Дано множество из N натуральных чисел. Существует ли в нем такое подмножество, сумма элементов в котором равна S ?

Формат входных данных

В первой строке вводятся числа N (натуральное, не превышает 12) и S (натуральное, не превосходит 10000), в следующей строке вводятся элементы исходного множества — натуральные числа, не превосходят 100.

Формат выходных данных

Выведите слово 'YES', если существует такое подмножество, сумма элементов которого равна S , или вывести 'NO' в противном случае.

Примеры

<code>subset-sum.in</code>	<code>subset-sum.out</code>
1 5 10	NO
3 8 5 4 3	YES

Задача F. Транзитивное замыкание

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

Незвешенный ориентированный граф задан своей матрицей смежности. Требуется построить его транзитивное замыкание, то есть матрицу, в которой в i -й строке и j -м столбце находится 1, если от вершины i можно добраться до вершины j , и 0 — иначе.

Формат входных данных

В первой строке дано число N ($1 \leq N \leq 100$) — число вершин в графе. Далее задана матрица смежности графа: в N строках даны по N чисел 0 или 1 в каждой. i -е число в i -й строке всегда равно 1.

Формат выходных данных

Необходимо вывести матрицу транзитивного замыкания графа в формате, аналогичным формату матрицы смежности.

Примеры

<code>closure.in</code>	<code>closure.out</code>
4	1 1 1 0
1 1 0 0	1 1 1 0
0 1 1 0	1 1 1 0
1 0 1 0	1 1 1 1
0 0 1 1	