

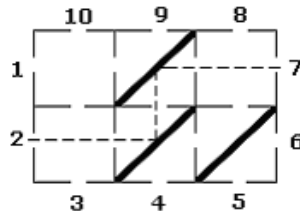
Задача А. Ящик с зеркалами

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

Один математик очень любит разные головоломки, его самая любимая головоломка — «ящик с зеркалами». Если мы посмотрим на сечение этого ящика горизонтальной плоскостью, то увидим, что он состоит из $n \times m$ квадратных ячеек (n строк и m столбцов). В каждой ячейке может быть зеркало, оно расположено по диагонали из левого нижнего в правый верхний угол. Обе стороны зеркала отражают свет. В бортах ящика в каждой строке и каждом столбце проделаны отверстия, через которые луч света можно направить в ящик и через которое он может выйти из ящика. Через каждое такое отверстие можно направить луч по единственному направлению — горизонтально, перпендикулярно борту ящика, в котором проделано отверстие. Когда луч света попадает в зеркало, он меняет направление на 90 градусов. Когда он проходит через пустую ячейку, его направление не меняется. Отверстия пронумерованы от 1 до $2(n + m)$ против часовой стрелки, начиная с верхнего левого угла. Поскольку зеркала внутри ящика не видны, то единственный способ выяснить их расположение — это запускать луч света в различные отверстия и смотреть, откуда он выйдет.



Вид снаружи на ящик



Горизонтальное сечение ящика

Напишите программу, которая по серии экспериментов восстанавливает некоторое возможное расположение зеркал в ящике.

Формат входного файла

Первая строка входного файла содержит числа n и m ($1 \leq n, m \leq 100$). Затем следует $n + m$ чисел, $i + 1$ -е число представляет собой номер отверстия, через которое выходит луч, запущенный в отверстие с номером i .

Формат выходного файла

Выведите в выходной файл n строк по m чисел, j -е число в i -й строке должно быть 1, если в соответствующей ячейке ящика имеется зеркало и 0 в противном случае.

Пример

<code>box.in</code>	<code>box.out</code>
2 3	0 1 0
9 7 10 8 6 5 2 4 1 3	0 1 1

Задача В. Максимальная сумма

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

Задан ориентированный граф, имеющий n вершин и m рёбер. В каждой вершине i графа записано некоторое целое число c_i . Рассмотрим следующую игру на этом графе. Сначала игрок ставит фишку в вершину 1 и получает c_1 очков. Затем он делает k ходов; каждый ход состоит в перемещении фишки по ребру, исходящему из текущей вершины, и после каждого хода игрок получает количество очков, равное числу, записанному в вершине, куда он перешёл.

Зная, что граф таков, что игрок сможет сделать k ходов из начальной вершины, найдите максимальную сумму очков, которую он сможет при этом набрать.

Формат входного файла

В первой строке входного файла заданы три числа — n , m и k ($2 \leq n \leq 20$, $1 \leq m \leq 400$, $1 \leq k \leq 10^{15}$).

Во второй строке записаны n чисел $c_1 c_2 \dots c_n$ через пробел ($1 \leq c_j \leq 100$). Это числа, записанные в вершинах.

Следующие m строк содержат по два числа u_i и v_i каждая ($1 \leq u_i, v_i \leq n$); эти числа обозначают, что в графе есть ребро из u_i в v_i .

Все числа во входном файле — целые.

Формат выходного файла

В первую строку выходного файла выведите число S — максимальную сумму, которую можно получить, сделав k ходов на графе.

Примеры

<code>maxsum.in</code>	<code>maxsum.out</code>
3 4 5	16
1 3 4	
1 2	
2 2	
2 1	
1 3	

Задача С. Художник

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

Не успев дорисовать свой гениальный футуристический шедевр, М. Калевич увлекся рисованием одномерных черно-белых картин. Он пытается найти оптимальное местоположение и количество черных участков картины. Для этого он проводит на прямой белые и черные отрезки, и после каждой из таких операций хочет знать количество черных отрезков на получившейся картине и их суммарную длину.

Изначально прямая — белая. Ваша задача — написать программу, которая после каждой такой операции выводит в выходной файл интересные художника данные.

Формат входного файла

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leq N \leq 100\,000$). В последующих N строках содержится описание операций. Каждая операция описывается строкой вида $c\ x\ l$, где c — цвет отрезка (W для белых отрезков, B для черных), а сам отрезок имеет вид $[x; x + l]$, причем координаты обоих концов — целые числа, не превосходящие по модулю 500 000. Длина задается положительным целым числом.

Формат выходного файла

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество черных отрезков на картине и их суммарную длину, разделенные одним пробелом.

Пример

<code>painter.in</code>	<code>painter.out</code>
7	0 0
W 2 3	1 2
B 2 2	1 4
B 4 2	1 4
B 3 2	2 6
B 7 2	3 5
W 3 1	0 0
W 0 10	