

Задача 1. Площадь полигона

Имя входного файла: polygon.in
Имя выходного файла: polygon.out
Ограничения по времени: 1 секунда
Ограничения по памяти: 256 МБ

Полигоном (он же многоугольник) будем называть либо замкнутую ломаную, либо часть плоскости, ограниченную такой ломаной. В задаче будем рассматривать простые полигоны, то есть такие, у которых стороны не соприкасаются (за исключением соседних — в вершинах) и не пересекаются. Полигон задан последовательностью своих вершин.

Требуется найти площадь полигона.

Формат входных данных

В единственной строке входного файла находится последовательность пар целых координат вершин, гарантировано образующая простой полигон. Числа разделены пробелом. Признаком окончания ввода — повторение координат первой вершины. Числа не превышают 10^5 , последовательность пар не превышает 256.

Формат выходных данных

Выходной файл должен содержать одно число — площадь многоугольника. Его следует округлить до ближайшего целого числа.

Примеры

polygon.in	polygon.out
0 0 1 0 1 1 0 1 0 0	1
5 0 0 5 -5 0 0 -5	50

Задача 2. Белые и черные кубики

Имя входного файла: bw.in
Имя выходного файла: bw.out
Ограничения по времени: 1 секунда
Ограничения по памяти: 256 МБ

Два друга любили играть в следующую игру. Бросают K_1 белых и K_2 черных кубиков. На каждом выпадает число от 1 до 6. Подсчитывают суммы чисел, выпавших на белых и на черных кубиках, и полученные суммы перемножают. Тот, у кого результат получается наибольшим, побеждает. А так как друзья увлекаются математикой, то им стало интересно узнать количество способов получения произведения A . Помогите друзьям решить эту проблему.

Кубики считаются различными, то есть получить на первом белом кубике 2 и на втором белом кубике 5 – не то же самое, что 5 на первом и 2 на втором. Суммарное количество кубиков не превосходит 100, а максимальное произведение не превышает 10^6 .

Формат входных данных

Входной файл содержит числа K_1 , K_2 и A в одной строке в указанном порядке через пробел. $K_1+K_2 \leq 100$, $A \leq 10^6$.

Формат выходных данных

Выходной файл должен содержать одно число — количество способов.

Примеры

bw.in	bw.out
2 2 8	6
2 3 29	0

Задача 3. Сокровища гробницы фараона

Имя входного файла: treasury.in
Имя выходного файла: treasury.out
Ограничения по времени: 1 секунда
Ограничения по памяти: 256 МБ

После долгих исканий Индиана Джонс нашел древнюю гробницу фараона. Его задача заключается в том, чтобы вынести из сокровищницы гробницы как можно больше предметов. При этом в сокровищницу можно войти один раз. Сокровищница прямоугольная, состоящая из отдельных комнат - «клеток». В каждой комнате хранится определенное количество предметов, которое Индиане Джонсу необходимо забрать. Он может начать с любой комнаты вдоль внешней северной стены сокровищницы (выбор комнат – за Индиана Джонс). Переходить можно в одну из трех «южно-соседних» комнат: южную, юго-восточную, юго-западную. Из комнат, граничащих с восточной или западной внешней стеной, возможны только два направления выхода. Закончить путь можно только в любой из комнат на южной стороне сокровищницы.

У Индиана Джонс есть план сокровищницы – прямоугольная таблица, в которой обозначены количества предметов в каждой комнате. Направлению с севера на юг соответствует движение сверху вниз.

Помогите Индиану Джонсу собрать как можно больше предметов.

Формат входных данных

Входной файл содержит два числа M, N — неотрицательные целые числа, обозначающие «длину» и «ширину» сокровищницы. $M, N \leq 80$. В каждой из следующих M строк содержится N чисел — количество вещей в комнатах гробницы. Количество вещей в каждой комнате не превышает 300.

Формат выходных данных

В первой строке выходного файла указывается номер (в порядке слева направо, с запада на восток) комнаты северного ряда, из которой нужно начать движение, во второй – строке символы – направление движения из комнаты в комнату (S – юг, E – юго-восток, W – юго-запад); в третьей – максимально возможное количество предметов, которое можно вынести из гробницы. Если возможно несколько путей то можно вывести любой из них.

Примеры

treasury.in	treasury.out
4 5	2
0 12 10 0 5	SWE
0 20 10 5 2	49
7 5 2 3 0	
9 10 10 2 0	

Задача 4. Найти дорогу

Имя входного файла: path.in
Имя выходного файла: path.out
Ограничения по времени: 1 секунда
Ограничения по памяти: 256 МБ

Дана система односторонних дорог, определяемая набором пар городов. Каждая такая пара (i, j) указывает, что из города i можно проехать в город j , но это не значит, что можно проехать и в обратном направлении.

Необходимо определить, можно ли проехать из заданного города A в заданный город B таким образом, чтобы посетить город C и не проезжать ни по какой дороге более одного раза.

Формат входных данных

Входной файл в первой строке содержит натуральное число N ($N \leq 50$) – количество городов (города нумеруются от 1 до N). Во второй строке находится натуральное M ($M \leq 100$) – количество дорог. В каждой из следующих строк находятся пары номеров городов, которые связывает дорога. В последней, $(M+3)$ -й, строке находятся номера городов A , B и C .

Формат выходных данных

Первая строка выходного файла должна содержать количество городов в последовательности. Начиная со второй строки, в каждой из строк должен содержаться номер города из последовательности городов. Последовательность начинается городом A и заканчивается городом B и удовлетворяет условиям задачи. При отсутствии возможного пути необходимо записать в первую строку файла число -1 .

Примеры

path.in	path.out
3	3
2	1
1 2	2
2 3	3
1 3 2	