

Задача А. Подготовка к экзамену

Имя входного файла: a.in
Имя выходного файла: a.out
Максимальное время работы на одном тесте: 2 секунды
Максимальный объем используемой памяти: 64 мегабайта

Вам предстоит сдать экзамен по теории графов. Известно, что задачи на экзамене будут распределены по двум вариантам. Тема задач тоже известна — независимые подмножества в графах. Сверившись с Википедией, вы вспомнили несколько определений:

Подмножество вершин графа называется **независимым**, если никакие две вершины этого подмножества не соединены ребром.

Независимое подмножество вершин графа называется **наибольшим**, если количество вершин в нём максимально.

В задаче **первого варианта** требуется найти максимальный размер наибольшего независимого подмножества, который может встречаться в графах, содержащих N вершин и M рёбер.

В задаче **второго варианта** требуется найти минимальный размер наибольшего независимого подмножества, который может встречаться в графах, содержащих N вершин и M рёбер.

К сожалению, вы не знаете, какой именно вариант вам достанется, поэтому решили написать программу, решающую оба типа задач.

Формат входных данных

В первой строке входного файла находится номер варианта, задачу из которого вашей программе предстоит решить. Далее, во второй строке расположены числа N и M ($1 \leq N \leq 10^6$, $0 \leq M \leq N(N - 1)/2$), задающие количество вершин и рёбер графа, соответственно.

Формат выходных данных

В первой строке выходного файла выведите искомое число.

Примеры

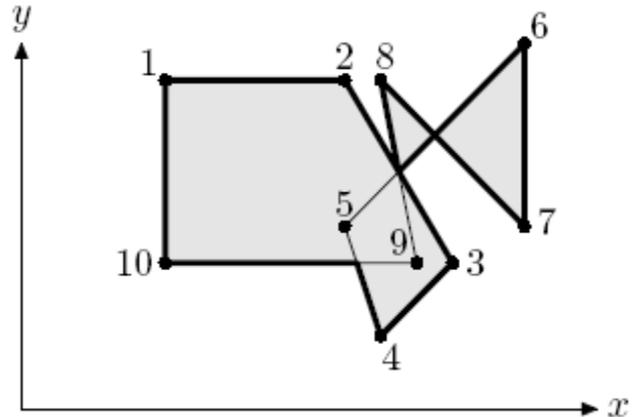
a.in	a.out
1 4 2	3
2 4 2	2

Задача В. Ломаная

Имя входного файла:
 Имя выходного файла:
 Максимальное время работы на одном тесте:
 Максимальный объем используемой памяти:

b.in
 b.out
 2 секунды
 64 мегабайта

Замкнутая ломаная (возможно, имеющая самопересечения) разбивает плоскость на несколько областей. Одна из областей не ограничена — это область, находящаяся снаружи ломаной. Все ограниченные регионы вместе с самой ломаной задают многоугольник (заштрихован на рисунке). Граница многоугольника (на рисунке выделена жирным) также является ломаной. Эта ломаная задает тот же многоугольник, что и заданная ломаная. Ваша задача состоит в том, чтобы найти границу многоугольника, задающегося данной ломаной.



Чтобы гарантировать уникальность ломаной, представляющей границу, необходимо, чтобы выполнялись следующие требования:

- ломаная не имеет самопересечений (возможно, имеет самокасания);
- соседние вершины ломаной не совпадают;
- соседние ребра ломаной не коллинеарны;
- при обходе ломаной по ребрам, ее внутренняя часть находится слева.

Формат входных данных

Первая строка входного файла содержит целой число N ($3 \leq N \leq 100$) – количество вершин в ломаной. Следующие N строк содержат по два целых числа X_i и Y_i ($0 \leq X_i, Y_i \leq 100$) — координаты вершин ломаной. Все вершины различны и никакая вершина не лежит на ребре между двумя другими вершинами. Соседние ребра ломаной не коллинеарны.

Формат выходных данных

Выведите в выходной файл целое число M — количество вершин в ломаной, задающей границу. Затем выведите M строк с координатами вершин. Координаты необходимо выводить с точностью до 4 знаков после точки.

Пример

b.in	b.out
10	13
4 9	9.3333 4
9 9	10 2
12 4	12 4
10 2	10.5 6.5
9 5	11.5 7.5
14 10	14 5
14 5	14 10
10 9	11.5 7.5
11 4	10 9
4 4	10.5 6.5
	9 9
	4 9
	4 4

Задача С. Шахтеры

Имя входного файла: c.in
 Имя выходного файла: c.out
 Максимальное время работы на одном тесте: 2 секунды
 Максимальный объем используемой памяти: 64 мегабайта

На двух угольных шахтах работают шахтеры. Добыча угля — тяжелая работа, так что шахтерам необходимо доставлять еду прямо в шахты. Каждый раз, когда очередная порция продовольствия доставляется в шахту, шахтеры производят некоторое количество угля. Существует три типа еды для шахтеров: мясо, фрукты и бублики. Шахтеры любят разнообразие в еде и будут работать продуктивнее, если их диета будет разнообразной. Точнее, каждый раз, когда в шахту доставляется новая порция продовольствия, необходимо посмотреть на две предыдущие поставки (или меньше чем две, если их еще не было) и действовать по следующему правилу:

- если вся еда была одинаковой — будет произведена одна тонна угля;
- если еды была двух типов — будет произведено две тонны угля;
- если трех типов — три тонны угля.

Тип и порядок поставок еды известен. Однако, можно выбрать, на какую из двух шахт какую поставку отправить. Поставка не может быть разделена и обязательно должна быть доставлена на первую или вторую шахту. Две шахты не обязательно должны получить одинаковое количество поставок (например, все поставки могут идти на одну шахту).

По известному порядку и типу поставок необходимо написать программу, определяющую максимально возможное количество тонн угля, которое можно добыть на обеих шахтах, при наилучшем распределении поставок между первой и второй шахтой.

Формат входных данных

Первая строка содержит число N ($1 \leq N \leq 100000$) — количество поставок еды.

Вторая строка содержит N символов — типы поставок в том порядке, в котором они будут осуществляться. Каждый из символов будет большой латинской буквой М (мясо), F (фрукты) или В (бублики).

Формат выходных данных

Выведите единственное число — количество тонн угля, которое можно добыть.

Примеры

c.in	c.out
6 MBMFFB	12
16 MMBMBVVVMMMMBMB	29

В примере 1, поставки можно распределить так: 1, 1, 2, 2, 1, 2. В результате будет получено 1, 2, 1, 2, 3, 3 тонн угля соответственно. Существуют и другие способы наилучшего распределения