

## Задача А. Место встречи изменить нельзя

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

Даны  $N$  точек. Найдите такие две из них, что расстояние между ними минимально.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  ( $2 \leq N \leq 100\,000$ ) — количество точек. Каждая из следующих  $N$  строк содержит пару целых чисел  $X$  и  $Y$ , разделённых пробелом, — координаты ( $-1\,000\,000\,000 \leq X, Y \leq 1\,000\,000\,000$ ). Все точки различны.

### Формат выходных данных

Единственная строка выходного файла должна содержать координаты двух выбранных точек.

### Пример

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

## Задача В. Сокровища

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

Дочь короля Флатландии собирается выйти за прекрасного принца. Принц хочет подарить принцессе сокровища, но он не уверен какие именно бриллианты из своей коллекции выбрать.

В коллекции принца  $n$  бриллиантов, каждый характеризуется весом  $w_i$  и стоимостью  $v_i$ . Принц хочет подарить наиболее дорогие бриллианты, однако король умен и не примет бриллиантов суммарного веса больше  $R$ . С другой стороны, принц будет считать себя жадным всю оставшуюся жизнь, если подарит бриллиантов суммарным весом меньше  $L$ .

Помогите принцу выбрать набор бриллиантов наибольшей суммарной стоимости, чтобы суммарный вес был в отрезке  $[L, R]$ .

### Формат входных данных

Первая строка содержит число  $n$  ( $1 \leq n \leq 32$ ),  $L$  и  $R$  ( $0 \leq L \leq R \leq 10^{18}$ ). Следующие  $n$  строк описывают бриллианты и содержит по два числа — вес и стоимость соответствующего бриллианта ( $1 \leq w_i, v_i \leq 10^{15}$ ).

### Формат выходных данных

Первая строка вывода должна содержать  $k$  — количество бриллиантов, которые нужно подарить принцессе. Вторая строка должна содержать номера даримых бриллиантов.

Бриллианты нумеруются от 1 до  $n$  в порядке появления во входных данных.

Если составить подарок принцессе невозможно, то выведите 0 в первой строке вывода.

### Примеры

<code>dowry.in</code>	<code>dowry.out</code>
3 6 8	1
3 10	2
7 3	
8 2	

## Задача С. Соединение и разъединение

Имя входного файла:	connect.in
Имя выходного файла:	connect.out
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Вы когда-нибудь слышали про обход в глубину? Например, используя этот алгоритм, вы можете проверить является ли граф связным за время  $O(E)$ . Вы можете даже посчитать количество компонент связности за то же время.

А вы когда-нибудь слышали про систему непересекающихся множеств? Используя эту структуру, вы можете быстро обрабатывать запросы “Добавить ребро в граф” и “Посчитать количество компонент связности в графе”.

А вы когда-нибудь слышали о *динамической* задаче связности? В этой задаче вам необходимо обрабатывать три типа запросов:

1. Добавить ребро в граф.
2. Удалить ребро из графа.
3. Посчитать количество компонент связности в графе.

Можно считать, что граф является неориентированным. Изначально граф является пустым.

### Формат входных данных

В первой строке находятся два целых числа  $N$  и  $K$  — количество вершин и количество запросов, соответственно ( $1 \leq N \leq 300\,000$ ,  $0 \leq K \leq 300\,000$ ). Следующие  $K$  строк содержат запросы, по одному в строке. Каждый запрос имеет один из трех типов:

1.  $+ u v$ : Добавить ребро между вершинами  $u$  и  $v$ . Гарантируется, что такого ребра нет.
2.  $- u v$ : Удалить ребро между  $u$  и  $v$ . Гарантируется, что такое ребро есть.
3.  $?$ : Посчитать количество компонент связности в графе.

Вершины пронумерованы целыми числами от 1 до  $N$ . Во всех запросах  $u \neq v$ .

### Формат выходных данных

Для каждого запроса типа ‘?’, Выведите количество компонент связности в момент запроса.

### Примеры

connect.in	connect.out
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

## Задача D. Конфеты и Камни

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	50 мегабайт

Мальчик Геральд и его тренер Миша играют в интересную игру. В начале игры имеются куча из  $n$  конфет и куча из  $m$  камней. Геральд и Миша ходят по очереди, первым ходит Миша. Миша на своем ходу проверяет, сколько на данный момент Геральд съел конфет и камней. Пусть Геральд съел  $a$  конфет и  $b$  камней. Тогда Миша начисляет Геральду  $f(a, b)$  призовых очков. Геральд же на своем ходу съедает либо одну конфету из кучи с конфетами, либо один камень из кучи с камнями. Когда Миша обнаруживает, что Геральд съел все, кроме одной конфеты и одного камня, он последний раз начисляет очки и игра заканчивается. Опустошать ни ту, ни другую кучку Геральд не имеет права. Расскажите Геральду, как ему играть, чтобы получить наибольшее количество очков: требуется найти один из возможных оптимальных вариантов игры для Геральда.

### Формат входных данных

В первой строке содержатся три целых числа  $n, m, p$  ( $1 \leq n, m \leq 20000$ ,  $1 \leq p \leq 10^9$ ). Во второй строке находятся  $n$  целых чисел  $x_0, x_1, \dots, x_{n-1}$  ( $0 \leq x_i \leq 20000$ ). В третьей строке находятся  $m$  целых чисел  $y_0, y_1, \dots, y_{m-1}$  ( $0 \leq y_i \leq 20000$ ). Величина  $f(a, b)$  вычисляется, как остаток от деления суммы  $x_a + y_b$  на число  $p$ .

### Формат выходных данных

В первой строке выведите единственное число: максимальное количество призовых очков, которое может заработать Геральд. Во второй строке выведите строку из  $n + m - 2$  символов, каждый из которых — это «C» или «S»,  $i$ -ый символ должен быть «C», если на своем  $i$ -ом ходу Геральд должен съесть конфету, и «S», если камень.

### Примеры

стандартный ввод	стандартный вывод
2 2 10 0 0 0 1	2 SC
3 3 10 0 2 0 0 0 2	10 CSSC
3 3 2 0 1 1 1 1 0	4 SCSC

### Замечание

В первом тесте, если на первом ходу Геральд съест камень, то после него он получит одно очко, а если конфету — то ноль. Перед первым своим ходом Геральд получит в любом случае 0 очков, а после второго — в любом случае 1. Таким образом, максимальное количество очков, которое может получить Геральд равно 2, и для этого надо съесть сначала камень, потом конфету.

## Задача E. Work for robots

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	512 мегабайт

На планете PTZZZ живёт и работает  $n$  роботов. С незапамятных времён некоторые роботы дружат между собой, а некоторые — нет.

Ровно один раз в день некоторые роботы выходят на работу, а все остальные идут в парк развлечений и отдыхают. При этом на работу должен выйти хотя бы один робот. Определением того, кто будет работать, а кто отдыхать, занимается робот-директор. Работа настолько важна для роботов, что первый день, когда робот-директор принял решение, был назван Первым днём Мира.

Если окажется так, что в некоторый день на работу вышли в точности те же роботы, что и в какой-нибудь из предыдущих дней, то робот-директор заржавеет от горя. Кроме того, закон не позволяет роботу-директору отправлять на работу группу роботов, в которой какие-нибудь два робота не будут дружить между собой.

Робот-директор не хочет ржаветь от горя как можно дольше, поэтому он старается каждый раз отправлять на работу другую группу роботов. Однако, как легко понять, различных групп роботов всё же конечное число, поэтому роботу-директору в конце концов придётся заржаветь.

От Вас требуется найти лишь номер дня, в который это произойдёт.

### Формат входных данных

В первой строке записано целое число  $n$  — количество роботов на PTZZZ ( $1 \leq n \leq 50$ ).

В каждой из следующих  $n$  строк записано по  $n$  цифр 0 или 1.  $j$ -я цифра в  $i$ -й строке равна единице, если  $i$ -й и  $j$ -й роботы дружат между собой, и нулю в противном случае.

Гарантируется, что  $i$ -я цифра в  $i$ -й строке равна нулю, а  $j$ -я цифра в  $i$ -й строке совпадает с  $i$ -й цифрой в  $j$ -й строке.

### Формат выходных данных

Выведите номер дня, в который робот-директор вынужден будет заржаветь от горя.

### Примеры

стандартный ввод	стандартный вывод
6 011100 101100 110100 111000 000001 000010	19

## Задача F. Дружелюбные хомячки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

На плоскости живут  $n$  хомячков. Каждый в точке с целыми координатами. Хомячки дружат, если существует прямоугольник со сторонами, параллельными осям координат, содержащий этих двух хомячков и не содержащий никаких других.

Прямоугольник содержит хомячка, если точка, в которой он живет, лежит внутри прямоугольника или на его границе.

Сколько пар хомячков дружат?

### Формат входных данных

На первой строке число  $n$ ,  $1 \leq n \leq 100\,000$ .

Следующие  $n$  строк содержат по два целых числа — координаты точек, в которых живут хомячки.

Все точки различны, а координаты целые, по модулю не превосходят  $10^9$ .

### Формат выходных данных

Выведите одно целое число — количество пар дружащих хомячков.

### Примеры

стандартный ввод	стандартный вывод
5 0 0 0 2 2 0 2 2 1 1	8

## Задача G. Прибавления на отрезках [A']

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Когда-то Костя решал учебную задачу о поиске максимума с прибавлением на отрезке. Формально задача звучала так:

«Дан массив длины  $n$ , изначально состоящий из нулей. Элементы массива пронумерованы от 1 до  $n$ . К массиву было применено  $q$  операций.  $i$ -я операция задается тремя целыми числами  $l_i$ ,  $r_i$  и  $x_i$  ( $1 \leq l_i \leq r_i \leq n$ ), ( $1 \leq x_i \leq n$ ) и означает, что к элементам с номерами  $l_i, l_i + 1, \dots, r_i$  прибавили число  $x_i$ . Требуется найти максимум в массиве после применения всех этих операций.»

Много лет спустя Костя нашел в своем Github-профиле решение этой задачи. Костя решил, что хочет улучшить свой старый код. В том числе, он решил, что в целях экономии памяти уберет запретную строчку кода, которая превращает все 32-битные числа в 64-битные. Однако, он хочет удостовериться, что решение все еще будет корректным.

Вычисляя ограничения на максимум в рамках данной задачи, он задумался: «интересно, а какие значения может принять максимум в массиве после применения некоторого подмножества данных операций?».

Помогите Косте, найдите все такие целые числа  $y$  от 1 до  $n$ , что после применения некоторого (возможно, пустого) подмножества данных операций максимум в массиве равен  $y$ .

### Формат входных данных

В первой строке находятся два целых числа  $n$  и  $q$  ( $1 \leq n, q \leq 10^4$ ) — длина массива и количество запросов в исходной задаче.

В следующих  $q$  строках описаны запросы, по одному в строке.  $i$ -я из этих строк содержит три целых числа  $l_i$ ,  $r_i$  и  $x_i$  ( $1 \leq l_i \leq r_i \leq n$ ,  $1 \leq x_i \leq n$ ), что обозначает запрос на добавление числа  $x_i$  на отрезке с  $l_i$ -го по  $r_i$ -й элемент включительно.

### Формат выходных данных

В первую строку выведите единственное число  $k$ , обозначающее количество возможных целых чисел от 1 до  $n$ , которым может быть равен максимум в массиве после применения некоторого (возможно, пустого) подмножества данных операций.

В следующей строке выведите через пробел все  $k$  чисел от 1 до  $n$  — возможные значения максимума. Выводите эти числа в возрастающем порядке.

### Примеры

стандартный ввод	стандартный вывод
4 3 1 3 1 2 4 2 3 4 4	4 1 2 3 4
7 2 1 5 1 3 7 2	3 1 2 3
10 3 1 1 2 1 1 3 1 1 6	6 2 3 5 6 8 9

### Замечание

Если в первом тестовом примере оставить только первый запрос, то максимум будет равен 1. Если оставить только второй запрос, то максимум будет равен 2. Если оставить первые два запроса, то максимум будет равен 3. Если оставить только третий запрос, то максимум будет равен 4. Но

если оставить третий запрос и еще какой-то, максимум будет больше  $n$ , поэтому его выводить не требуется.

Во втором тестовом примере, оставив только первый запрос, можно получить 1. Оставив только второй, можно получить 2. А если оставить все запросы, максимум будет равен 3.

В третьем тестовом примере можно получить максимумы так:

- Можно получить максимум 2 оставив запросы: (1).
- Можно получить максимум 3 оставив запросы: (2).
- Можно получить максимум 5 оставив запросы: (1, 2).
- Можно получить максимум 6 оставив запросы: (3).
- Можно получить максимум 8 оставив запросы: (1, 3).
- Можно получить максимум 9 оставив запросы: (2, 3).