

## Задача А. 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.

### Пример

dowry.in	dowry.out
3 6 8	1
3 10	2
7 3	
8 2	

## Задача В. Соединение и разъединение

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

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.

### Пример

<code>connect.in</code>	<code>connect.out</code>
5 11	5
?	1
+ 1 2	1
+ 2 3	2
+ 3 4	
+ 4 5	
+ 5 1	
?	
- 2 3	
?	
- 4 5	
?	

## Задача С. Дружелюбные хомячки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
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.

### Пример

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

## Задача D. Потерянный массив

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Маленькому Леше улыбнулась удача: ему удалось раздобыть очень редкий и необычный массив  $A$ , состоящий из  $n$  целых чисел. Заполучив в свои руки такой ценный предмет, мальчик сразу же отнес его в свою подземную лабораторию, намереваясь постичь тайны устройства массива. Как опытный исследователь, Леша решил не спешить с проведением жутких экспериментов над бедным массивом, а сперва понаблюдать за ним. Он записал на листик бумаги результаты  $m$  наблюдений за массивом. Каждое наблюдение описывается тремя числами  $l_i$ ,  $r_i$  и  $s_i$ , которые означают, что сумма элементов по модулю 2 на подотрезке  $[l_i, r_i]$  массива равна  $s_i$ . Леша подготовил все необходимое для проведения  $m + 1$ -го наблюдения, но тут он услышал, что мама зовет его обедать. По возвращении в свою лабораторию, мальчик увидел страшное: массив исчез. Совсем. Все надежды мальчика на получение Нобелевской премии в области исследования массивов рассыпались на глазах. Но потом Леша вспомнил, что у него все же осталась некоторая информация о массиве. Основываясь на записанных фактах, он решил построить компьютерную модель массива и проводить эксперименты над ней. Конечно, мальчик понимал, что он не сможет однозначно восстановить утерянный массив, потому что существует бесконечно много массивов, удовлетворяющих записанным фактам. Все мы знаем, что операции с большими числами занимают много времени, а если еще к тому же числа становятся отрицательными... Чтобы сделать свою компьютерную модель как можно более быстрой, Леша решил построить массив, удовлетворяющий известным фактам, элементы которого неотрицательны, а их сумма минимально возможная. Помогите ему сделать это!

### Формат входных данных

Первая строка входных данных содержит числа  $n$  и  $m$  - количество элементов в массиве и количество фактов. Следующие  $m$  строк содержат числа  $l_i$ ,  $r_i$  и  $s_i$  - описание  $i$ -го факта.

$$1 \leq n \leq 40$$

$$0 \leq m \leq 100$$

$$1 \leq l_i \leq r_i \leq n$$

$$0 \leq s_i \leq 1$$

$$0 \leq a_i \leq 10^9$$

### Формат выходных данных

Вывод должен содержать  $n$  целых чисел  $a_i$  — элементы восстановленного массива. Если существует несколько возможных ответов, выведете лексикографически минимальных среди них. Массив  $X$  лексикографически меньше массива  $Y$  если существует такой индекс  $k \leq |X|$ , что  $x_i = y_i$  для всех  $i < k$  и  $x_k < y_k$ .

### Пример

стандартный ввод	стандартный вывод
3 3	0 1 0
2 2 1	
3 3 0	
2 3 1	

### Замечание

Полуфинал БГУИР 2016

## Задача Е. Место встречи изменить нельзя

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

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 F. Прибавления на отрезках

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

Гриша пришел на констест и увидел там следующую задачу.

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.

Но Гриша не из глупых! Он решил эту задачу очень быстро.

Однако что-то в нем переклинило, и он задумался: «интересно, а какие значения может принять максимум в массиве после применения некоторого подмножества данных операций?».

Помогите Грише, 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).

## Задача G. Тандемные повторы

Имя входного файла: `tandems.in`  
Имя выходного файла: `tandems.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Дана строка  $s$  длины  $n$ .

Тандемным повтором в ней называются два вхождения какой-либо подстроки подряд. Иными словами, тандемный повтор описывается парой индексов  $i < j$  такими, что подстрока  $s[i \dots j]$  — это две одинаковые строки, записанные подряд.

От вас требуется посчитать количество пар индексом  $i < j$  таких, что подстрока  $s[i \dots j]$  является тандемным повтором.

### Формат входных данных

Во входном файле находятся не более 30 тестов. Каждый тест состоит из единственной непустой строки, состоящей из символов **A,C,G,T**. Длина строки не превосходит  $10^5$ . Входной файл заканчивается строкой **0**.

### Формат выходных данных

Для каждого теста выведите единственное число — количество тандемных повторов. Числа разделяйте переводами строк.

### Пример

<code>tandems.in</code>	<code>tandems.out</code>
AGGA	1
AGAG	1
ATTCGATTCGATTCG	9
AAAA	4
0	



## Задача Н. Максимальная подпоследовательность

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 256 мегабайт

Вам дан массив  $a$ , состоящий из  $n$  целых чисел и целое число  $m$ . Выберите последовательность позиций  $b_1, b_2, \dots, b_k$  ( $1 \leq b_1 < b_2 < \dots < b_k \leq n$ ) такую, чтобы значение  $\sum_{i=1}^k a_{b_i} \bmod m$  было максимально. Выбранная подпоследовательность может быть пустой.

Подсчитайте максимальное возможное значение  $\sum_{i=1}^k a_{b_i} \bmod m$ .

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$  ( $1 \leq n \leq 35$ ,  $1 \leq m \leq 10^9$ ).

Во второй строке записаны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите максимальное возможное значение  $\sum_{i=1}^k a_{b_i} \bmod m$ .

### Примеры

стандартный ввод	стандартный вывод
4 4 5 2 4 1	3
3 20 199 41 299	19

### Замечание

В первом примере можно выбрать последовательность  $b = 1, 2$ , чтобы сумма  $\sum_{i=1}^k a_{b_i}$  была равна 7 (равна 3 по модулю 4).

Во втором примере можете выбрать последовательность  $b = 3$ .

## Задача I. Большие Диаграммы Юнга

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

256 мегабайт

Яша и Саша — два роскошных брата, которые любят дарить друг другу подарки на день рождения. И вот, на очередной день рождения, Яша подарил Саше на день рождения Диаграмму Юнга. Саша решил посчитать число клеток в этой диаграмме. Но Саша опаздывал на матч по футболу, поэтому посчитал только число клеток в первой строке и первом столбце. И теперь ему интересно, какое максимальное по модулю число клеток может быть в этой диаграмме.

### Формат входных данных

В первой строке входных данных находится одно целое число — размер первой строки диаграммы. Во второй строке находится другое целое число — размер первого столбца диаграммы. Длины этих чисел не превосходит двухсот пятидесяти тысяч символов. Размеры строк этой диаграммы могут быть отрицательными.

### Формат выходных данных

В единственной строке выведите единственное число — максимальный по модулю размер диаграммы.

### Пример

<code>multiply.in</code>	<code>multiply.out</code>
2	4
2	