

Содержание

Задача А. Сосчитайте. . . [0.5 sec, 256 mb]	2
Задача В. Том Сойер и его друзья [0.5 sec, 256 mb]	3
Задача С. Главные точки [0.5 sec, 256 mb]	4
Задача D. Выпуклый многоугольник [0.5 sec, 256 mb]	5
Задача Е. Минимум и максимум [0.5 sec, 256 mb]	6
Задача F. Ядра [0.5 sec, 256 mb]	7
Задача G. Психотренинг [0.5 sec, 256 mb]	8
Задача H. Количество циклов [0.5 sec, 256 mb]	9
Задача I. Внутренняя точка [0.5 sec, 256 mb]	10
Задача J. Для любителей статистики [0.5 sec, 256 mb]	11
Задача K. Рюкзак [0.5 sec, 256 mb]	12
Задача L. King's Assassination [0.5 sec, 256 mb]	13
Задача M. Три типа скобок [0.5 sec, 256 mb]	14

Задача А. Сосчитайте... [0.5 sec, 256 mb]

Ваша задача — подсчитать количество неотрицательных целых решений неравенства

$$x_1 + x_2 + \dots + x_m \leq n,$$

где $1 \leq m \leq 30$, $0 \leq n \leq 30$.

Формат входных данных

Входной файл состоит из двух целых чисел m и n .

Формат выходных данных

В выходной файл необходимо вывести количество решений этого неравенства в неотрицательных целых числах.

Пример

stdin	stdout
3 5	56

Задача В. Том Сойер и его друзья [0.5 сек, 256 mb]

Друзья Тома Сойера по очереди красят забор разными красками. Каждый из них красит несколько идущих подряд секций забора в определенный цвет, при этом используемые цвета могут повторяться. Новая краска ложится поверх старой. Для каждой краски вычислите количество секций, которые будут покрашены этой краской после того, как все друзья закончат работу.

Формат входных данных

В первой строке входного файла содержатся два целых числа: N ($1 \leq N \leq 10^9$) и K ($1 \leq K \leq 50\,000$) — количество секций в заборе и количество различных красок соответственно.

Во второй строке содержится единственное число M ($0 \leq M \leq 50\,000$) — количество друзей Тома Сойера.

Далее следуют M строк: в i -ой строке содержится информация о работе друга, который красил забор i -ым по счету, а именно 3 целых числа c_i, l_i, r_i ($1 \leq c_i \leq K, 1 \leq l_i \leq r_i \leq N$) — номер краски, которую использовал i -й друг, номер первой и номер последней покрашенной секции соответственно.

Формат выходных данных

Выведите в единственную строку выходного файла K целых чисел: i -ое число должно быть равно количеству секций, покрашенных i -й краской.

Пример

stdin	stdout
5 3 4 1 3 4 2 4 5 3 2 3 1 5 5	1 1 2
5 3 3 1 1 5 2 2 4 1 3 3	3 2 0

Задача С. Главные точки [0.5 сек, 256 mb]

Даны точки на плоскости.

Требуется выделить минимальное подмножество точек S , такое что для любой исходной точки A существует точка $B \in S$, такая что $A_x \leq B_x$ и $A_y \leq B_y$.

Формат входных данных

В первой строке вводится число n ($1 \leq n \leq 500\,000$) — количество точек.

В следующих n строках задано по два числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты точки.

Формат выходных данных

В первой строчке вывести числа k — количество точек в множестве S .

Дальше должно следовать k чисел — индексы точек, которые вошли в ответ.

Точки нумеруются с единицы.

Если ответов несколько, вывести любой из них.

Пример

stdin	stdout
5	2
1 -1	3 5
0 0	
-1 3	
1 1	
1 1	

Задача D. Выпуклый многоугольник [0.5 sec, 256 mb]

Формат входных данных

В первой строке одно число N ($3 \leq N \leq 100\,000$). Далее в N строках по паре целых чисел — координаты очередной вершины простого многоугольника в порядке обхода по или против часовой стрелки.

Формат выходных данных

Одна строка «YES», если приведённый многоугольник является выпуклым, и «NO» в противном случае.

Пример

stdin	stdout
3 1 0 0 1 1 1	YES

Задача Е. Минимум и максимум [0.5 сек, 256 mb]

Пусть есть мультимножество целых чисел (множество с возможными повторениями). Необходимо реализовать структуру данных для их хранения, поддерживающую следующие операции: **GetMin** — извлечение минимума, **GetMax** — извлечение максимума, **Insert(N)** — добавление числа в множество.

Формат входных данных

В первой строке входного файла записано одно целое число N ($1 \leq N \leq 100\,000$) — число запросов к структуре. Затем в N строках следуют запросы по одному в строке: **GetMin**, **GetMax**, **Insert(A)** — извлечение минимума, максимума и добавление числа A ($1 \leq A \leq 2^{31} - 1$). Запросы корректны, то есть нет операций извлечения для пустого множества.

Формат выходных данных

Для каждого запроса **GetMin** или **GetMax** выведите то число, которое было извлечено.

Примеры

stdin	stdout
10	1
Insert(100)	100
Insert(99)	1
Insert(1)	2
Insert(2)	99
GetMin	
GetMax	
Insert(1)	
GetMin	
GetMin	
GetMax	

Задача F. Ядра [0.5 сек, 256 mb]

Капитан Вася всегда держит на своем корабле запас пушечных ядер для борьбы с пиратами. Так как он привык во всем поддерживать порядок, он хранит ядра в виде пирамид. Каждый из слоев одной пирамиды является равносторонним заполненным ядрами треугольником, сторона которого содержит ровно k ядер. Сторона основания пирамиды состоит из n ядер, в следующем слое сторона состоит из $n - 1$ ядра, и т.д., пока на вершину не будет положено одно ядро (которое является равносторонним треугольником со стороной 1).

Например, пирамида размера 3 состоит из трех уровней, выглядящих так (сверху вниз):

```
X

X
X X

X
X X
X X X
```

Ясно, что каждый из треугольников может содержать только 1, 3, 6, 10 и т.д. ядер. Таким образом, пирамида может содержать только 1, 4, 10, 20, и т.д. ядер.

Вася отправляется в плавание и берет с собой ровно m ядер. Какое минимальное число пирамид требуется ему сложить из них на своем корабле?

Формат входных данных

В первой строке входного файла записано количество тестов $1 \leq T \leq 20$. В последующих T строках задается количество ядер в i -м тесте $1 \leq m_i \leq 300\,000$.

Формат выходных данных

Для каждого из T тестов входного файла выведите в отдельной строке минимальное количество пирамид.

Пример

stdin	stdout
5	1
1	2
5	3
9	3
15	2
91	

Задача G. Психотренинг [0.5 sec, 256 mb]

На очередном психологическом тренинге n участников сборов играют в занимательную игру. Участники игры рассаживаются по кругу и получают номера от 1 до n против часовой стрелки. После этого главный психолог отсчитывает против часовой стрелки k -го участника игры, начиная с первого. Этот участник выходит из круга и может идти на ужин. А остальные продолжают участие в тренинге. Главный психолог отсчитывает еще k участников, начиная со следующего после выбывшего. Участник, который оказался k -ым, тоже покидает тренинг, и так далее.

Участники сборов решили сесть в круг таким образом, чтобы один вредный тип пошел ужинать последним. Для этого они хотят установить, какой номер он должен для этого получить. Помогите им.

Формат входных данных

Входной файл содержит два целых числа: n и k ($1 \leq n \leq 10^{18}$, $1 \leq k \leq 1000$).

Формат выходных данных

Выведите в выходной файл одно число — номер участника, который пойдет на ужин последним.

Пример

stdin	stdout
5 3	4

Задача Н. Количество циклов [0.5 sec, 256 mb]

Формально, *путь* в графе — это чередующаяся последовательность вершин и рёбер $u_1, e_1, u_2, e_2, u_3, \dots, u_k$, начинающаяся и заканчивающаяся вершиной и такая, что любые соседние вершина и ребро в ней инцидентны.

Цикл — это путь, начальная и конечная вершины которого совпадают. В цикле должно быть хотя бы одно ребро.

Простой путь отличается от обычного пути тем, что в нём не может быть повторяющихся вершин.

Простой цикл — это цикл, в котором нет повторяющихся вершин и рёбер.

Дан неориентированный граф. Посчитайте, сколько в нём различных простых циклов. Заметим, что циклы считаются одинаковыми, если они обходят одно и то же множество вершин в одном и том же порядке, возможно, начиная при этом из другой вершины, или если порядок обхода противоположный. Например, циклы с порядком обхода вершин 1, 2, 3, 1, 2, 3, 1, 2 и 1, 3, 2, 1 считаются одинаковыми, а циклы 1, 2, 3, 4, 1 и 1, 3, 4, 2, 1 — нет, поскольку порядок обхода вершин различен.

Формат входных данных

В первой строке входного файла заданы числа N и M через пробел — количество вершин и рёбер в графе, соответственно ($1 \leq N \leq 10$). Следующие M строк содержат по два числа u_i и v_i через пробел ($1 \leq u_i, v_i \leq N$, $u_i \neq v_i$); каждая такая строка означает, что в графе существует ребро между вершинами u_i и v_i . В графе нет кратных рёбер.

Формат выходных данных

Выведите одно число — количество простых циклов в заданном графе.

Примеры

stdin	stdout
3 2 1 2 2 3	0
4 5 1 2 2 3 3 4 4 1 1 3	3

Задача I. Внутренняя точка [0.5 sec, 256 mb]

Дан строго выпуклый N -угольник и K точек. Для каждой точки нужно определить, где она находится — внутри, на границе, или снаружи.

Формат входных данных

N ($3 \leq N \leq 10^5$). Далее N точек — вершины многоугольника.

K ($0 \leq K \leq 10^5$). Далее K точек — запросы.

Все координаты — целые числа по модулю не превосходящие 10^7 .

Формат выходных данных

Для каждого запроса одна строка — INSIDE, BORDER или OUTSIDE.

Примеры

stdin	stdout
4	INSIDE
0 0	BORDER
2 0	BORDER
2 2	OUTSIDE
0 2	
4	
1 1	
0 0	
0 1	
0 3	

Задача J. Для любителей статистики [0.5 сек, 256 mb]

Вы никогда не задумывались над тем, сколько человек за год перевозят трамваи города с десятиллионным населением, в котором каждый третий житель пользуется трамваем по два раза в день?

Предположим, что на планете Земля n городов, в которых есть трамваи. Любители статистики подсчитали для каждого из этих городов, сколько человек перевезено трамваями этого города за последний год. Из этих данных была составлена таблица, в которой города были отсортированы по алфавиту. Позже выяснилось, что для статистики названия городов несущественны, и тогда их просто заменили числами от 1 до n . Поисковая система, работающая с этими данными, должна уметь быстро отвечать на вопрос, есть ли среди городов с номерами от l до r такой, что за год трамваи этого города перевезли ровно x человек. Вам предстоит реализовать этот модуль системы.

Формат входных данных

В первой строке дано целое число n , $0 < n < 70\,000$. В следующей строке приведены статистические данные в виде списка целых чисел через пробел, i -е число в этом списке — количество человек, перевезенных за год трамваями i -го города. Все числа в списке положительны и не превосходят $10^9 - 1$. В третьей строке дано количество запросов q , $0 < q < 70\,000$. В следующих q строках перечислены запросы. Каждый запрос — это тройка целых чисел l , r и x , записанных через пробел ($1 \leq l \leq r \leq n$, $0 < x < 10^9$).

Формат выходных данных

Выведите строку длины q , в которой i -й символ равен 1, если ответ на i -й запрос утвердителен, и 0 в противном случае.

Пример

stdin	stdout
5 123 666 314 666 434 5 1 5 314 1 5 578 2 4 666 4 4 713 1 1 123	10101

Задача К. Рюкзак [0.5 sec, 256 mb]

У нас есть несколько предметов и рюкзак, который выдерживает вес C . Предмет с номером i имеет вес x_i . Определите число различных наборов предметов, которые можно унести в рюкзак. Два набора считаются различными, если существует хотя бы один предмет, который включён в один из наборов и не включён в другой.

Формат входных данных

В первой строке ввода записано целое число n — количество предметов ($1 \leq n \leq 30$). Во второй строке записано n целых чисел x_i ($1 \leq x_i \leq 10^9$). В третьей строке записано целое число C — вместимость рюкзака ($0 \leq C \leq 10^9$).

Формат выходных данных

Выведите единственное целое число — искомое число способов.

Примеры

stdin	stdout
1 1 1	2
1 1 2	2
2 2 2 1	1
2 1 1 2	4
2 1 1 1	3
30 1 30	1073741824

Задача L. King's Assassination [0.5 sec, 256 mb]

Дан граф из n вершин и m ребер. Граф ориентированный. Нужно определить число вершин, содержащихся на всех путях из s в t (сами s и t учитывать не нужно).

Формат входных данных

Первая строка содержит n , m , s и t ($2 \leq n \leq 100\,000$, $1 \leq m \leq 300\,000$, $1 \leq s, t \leq n$, $s \neq t$).

Следующие m строк содержат пары чисел x_i и y_i — индексы вершин от 1 до n . Это означает что есть дорога из вершины с номером x_i в вершину с номером y_i .

Формат выходных данных

Число вершин k . Далее k чисел — номера вершин в возрастающем порядке.

Примеры

stdin	stdout
4 3 1 4 1 2 2 3 3 4	2 2 3
4 4 1 4 1 2 2 3 3 4 1 3	1 3
4 5 1 4 1 2 2 3 3 4 1 3 2 4	0

Задача М. Три типа скобок [0.5 сек, 256 mb]

Определим по индукции множество \mathcal{T} *правильных скобочных последовательностей* из трёх типов скобок:

- $\varepsilon \in \mathcal{T}$ (пустая строка)
- $A \in \mathcal{T} \Rightarrow (A) \in \mathcal{T}$
- $A \in \mathcal{T} \Rightarrow [A] \in \mathcal{T}$
- $A \in \mathcal{T} \Rightarrow \{A\} \in \mathcal{T}$
- $A \in \mathcal{T}, B \in \mathcal{T} \Rightarrow AB \in \mathcal{T}$

Пусть теперь \mathcal{T}_n — это множество правильных скобочных последовательностей из $2n$ символов — n открывающих и n закрывающих скобок.

Упорядочим элементы множества \mathcal{T}_n лексикографически с некоторым порядком символов ‘(’, ‘)’, ‘[’, ‘]’, ‘{’ и ‘}’.

По данным числам n и p , а также порядку, заданному на скобках, найдите p -ый в этом порядке элемент множества \mathcal{T}_n .

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и p ($0 \leq n \leq 20$, $0 \leq p \leq 9 \cdot 10^{18}$). Скобочные последовательности нумеруются с нуля.

Во второй строке записаны шесть символов — ‘(’, ‘)’, ‘[’, ‘]’, ‘{’ и ‘}’ — в некотором порядке. Их порядок задаёт лексикографический порядок на множестве \mathcal{T}_n .

Формат выходных данных

В первой строке выходного файла выведите $2n$ символов без пробелов — p -ю правильную скобочную последовательность длины $2n$ из трёх типов скобок.

Если для данного n не существует p -я правильная скобочная последовательность, выведите в первой строке “N/A”.

Примеры

stdin	stdout
1 0 (){}[]	()
1 1 (){}[]	[]
1 2 (){}[]	{}
1 3 (){}[]	N/A