

Задача A. Russian Dolls

Имя входного файла: `a.in`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Матрёшки — известный русский сувенир. Набор матрёшек состоит из нескольких полых деревянных кукол, последовательно вложенных друг в друга. Каждая из матрёшек имеет внешний объём out_i (объём, который она занимает) и внешний объём in_i — объём свободного места внутри матрёшки. Матрёшка A может быть вложена в матрёшку B , только если внешний объём матрёшки A строго меньше внутреннего объёма матрёшки B . При этом внутрь каждой матрёшки помещается непосредственно не более одной матрёшки.

Пусть для каждой матрёшки стоимость единицы пустого пространства непосредственно внутри i -й матрёшки (но не внутри вложенных) равна $cost_i$.

Вам дано некоторое количество матрёшек. Требуется вложить некоторые из них друг в друга согласно правилам таким образом, чтобы суммарная стоимость пустого пространства во всех матрёшках в результате была наименьшей.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 1000$) — количество матрёшек. Следующие N строк задают матрёшек, по одной на строку. i -я строка задаёт параметры i -й матрёшки и содержит три целых числа $out_i, in_i, cost_i$ ($1 \leq in_i < out_i \leq 1000, 1 \leq cost_i \leq 1000$) — внешний объём, внутренний объём и стоимость единицы пустого пространства соответственно.

Формат выходных данных

Выведите одно целое число — наименьшую суммарную стоимость пустого пространства для заданного набора матрёшек.

Примеры

<code>a.in</code>	<code>standard output</code>
3	7
5 4 1	
4 2 2	
3 2 1	

Задача В. Intercity

Имя входного файла: `b.in`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 mebibytes

Несколько лет назад железнодорожное расписание Украинских железных дорог было устроено так: между любыми двумя из N крупных городов Украины существовал ровно один прямой (безостановочный) поезд. Проезд в каждом из таких поездов стоил B гривен.

Перед Чемпионатом Европы по футболу некоторые поезда были заменены скоростными. Цена поездки на скоростном поезде равна A гривен. При этом по-прежнему между любыми двумя крупными городами Украины ходит ровно один поезд (скоростной или обычный). Все маршруты поездов являются двусторонними.

Требуется построить маршрут из города с номером 1 в город с номером N (города занумерованы последовательными целыми числами) так, чтобы суммарная стоимость поездки была минимальной.

Формат входных данных

В первой строке входного файла заданы четыре целых числа N , K , A и B ($2 \leq N \leq 5 \cdot 10^5$, $0 \leq K \leq 5 \cdot 10^5$, $1 \leq A, B \leq 5 \cdot 10^5$) — количество крупных городов, количество скоростных поездов, цена проезда в скоростном поезде и цена проезда в обычном поезде. Каждая из последующих K строк содержит по два целых числа u_i and v_i ($1 \leq u_i, v_i \leq N$, $u_i \neq v_i$) — номера городов, соединённых скоростным поездом. Гарантируется, что ни одна пара чисел не встретится более одного раза.

Формат выходных данных

Выведите одно целое число — минимальную стоимость пути между городами 1 и N .

Примеры

<code>b.in</code>	<code>standard output</code>
5 4 1 4 1 2 2 3 2 4 3 5	3

Задача C. Count (Binary Input!)

Имя входного файла: `c.in`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 512 mebibytes

Вам задана матрица, состоящая из целых положительных чисел, имеющая следующее свойство: все строки и все столбцы матрицы отсортированы по возрастанию (то есть $A[i, j] \geq A[i - 1, j]$ и $A[i, j] \geq A[i, j - 1]$ для всех i, j).

Требуется ответить на один или несколько запросов. Каждый запрос состоит из пары чисел (X, Y) таких, что $Y \geq X$. Для каждого запроса требуется выяснить, сколько элементов матрицы больше или равны X , но меньше или равны Y .

Формат входных данных

Входной файл является **бинарным** файлом, содержащим 32-битные знаковые целые числа (записанные в little-endian формате). Входной файл состоит из:

- 32-битного знакового целого положительного числа N , задающего количество строк (не более 10^4).
- 32-битного знакового целого положительного числа M , задающего количество столбцов (не более 10^4).
- $N \times M$ 32-битных знаковых целых положительных чисел, задающих значения матрицы, записанных по рядам.
- Некоторое количество 32-битных знаковых целых чисел, задающих пары (X, Y) соответственно. Гарантируется, что количество пар не больше 1 и не менее 1000, а также что в конце файла не будет «неполной» пары.

Формат выходных данных

Для каждой пары (X, Y) выведите в отдельной строке одно число — количество элементов матрицы, больших или равных X и меньших или равных Y .

Примеры

<code>c.in</code>	<code>standard output</code>
2 4	5
1 5 10 10	2
2 10 20 99	0
10 99	3
2 9	
100 1000	
10 10	

Замечание

В примере перечислены данные во входном файле, прочитанные как 32-битные целые (реально же входной файл, соответствующий примеру, текстовым не является и имеет размер $18 \cdot 4 = 72$ байта).

Задача D. Chess (Division 1 Only!)

Имя входного файла:	d.in
Имя выходного файла:	standard output
Ограничение по времени:	1 second
Ограничение по памяти:	256 mebibytes

На шахматной доске находятся две белые ладьи и один чёрный король. Белые ходят первыми. Требуется поставить мат чёрному королю как можно быстрее при том, что чёрные используют оптимальную стратегию (направленную на то, чтобы мат был поставлен как можно позже).

Используемая в данной задаче модификация правила шахмат такова:

1. В отличие от обычных шахмат (и шахматных задач), в данной задаче белый король на доске отсутствует.
2. Доска имеет размер 8×8 . Вертикали обозначены строчными латинскими буквами от *a* до *h*, а горизонтали — цифрами от 1 до 8.
3. Белые и чёрные делают ходы (двигают фигуру своего цвета в соответствии с правилами) по очереди. В контексте данной задачи считаются только ходы, сделанные белыми.
4. Ладья ходит по вертикали и горизонтали на любое количество незанятых полей. Ладья не может перепрыгивать через фигуру своего цвета (другую белую ладью) или вставать на занятую ей клетку.
5. Шах — это угроза взятия короля на следующем ходу, то есть позиция, в которой одна из ладей находится на одной горизонтали или вертикали с королём.
6. Король может двигаться на один квадрат в любом направлении (по горизонтали, по вертикали или по диагонали), если после его хода король не оказывается под шахом. При этом король может пойти на клетку, занятую одной из ладей. В этом случае ладья снимается с доски и больше в игре не участвует.
7. Мат — это позиция, в которой король находится под шахом и из которой любой ход короля ведёт в позицию, в которой он оказывается под шахом.
8. Пат — это позиция, в которой у игрока нет хода, не нарушающего правил, при этом король не находится под шахом. В этом случае фиксируется ничья (исход, неприемлемый для белых в контексте данной задачи).

Формат входных данных

Первая строка входного файла содержит количество позиций в файле T ($1 \leq T \leq 10^5$), каждая из последующих T строк задаёт одну позицию. Гарантируется, что все позиции во входном файле попарно различны.

Позиция задаётся перечислением трёх клеток доски (задаются буквой, соответствующей вертикали, и цифрой, соответствующей горизонтали) — расположением чёрного короля и двух ладей соответственно. Гарантируется, что все клетки попарно различны и что в начальной позиции король не находится под шахом.

Формат выходных данных

Для каждой позиции во входном файле выведите одно целое число в новой строке — минимальное количество ходов, за которое белые гарантированно поставят мат чёрным, начиная с соответствующей позиции. В случае, если из данной позиции мат не ставится, выведите 0.

Примеры

d.in	standard output
2	2
c7 f1 g6	1
h6 c3 g8	

Задача E. A+B (Division 1 Only!)

Имя входного файла: `e.in`
Имя выходного файла: `standard output`
Ограничение по времени: `0.5 seconds`
Ограничение по памяти: `256 mebibytes`

Дан компьютер с двумя ячейками памяти (обозначим эти ячейки буквами a и b). В каждой из этих ячеек в произвольный момент времени записано некоторое целое число. Компьютер может выполнять только две команды: $a+ = b$ и $b+ = a$. Первая команда увеличивает значение переменной a на значение, заданное переменной b . Вторая, соответственно, увеличивает значение переменной b на значение, заданное переменной a .

Программа для данного компьютера состоит из (возможно, пустой) последовательности соответствующих команд. Команды выполняются последовательно. Требуется по заданным начальным значениям a и b установить, существует ли какая-нибудь программа, в результате выполнения которой значение одной из ячеек будет равно заданному числу S .

Формат входных данных

Входной файл содержит три целых числа: начальное значение переменной a , начальное значение переменной b и число S , которое требуется построить ($0 \leq a, b, S \leq 10^{18}$).

Формат выходных данных

Выведите "YES", если S может получиться как результат выполнения некоторой программы при заданных начальных значениях, или "NO" в противном случае.

Примеры

<code>e.in</code>	<code>standard output</code>
1 2 3	YES
3 4 5	NO
3 4 17	YES

Задача F. Dragon Pattern (Division 1 Only!)

Имя входного файла: `f.in`
Имя выходного файла: `standard output`
Ограничение по времени: `0.5 seconds`
Ограничение по памяти: `256 mebibytes`

Кривая-дракон — это полиномиальная ломаная, составленная из отрезков длины 1 и определяемая рекурсивно. Выберем некоторую точку на плоскости и одно из четырёх направлений, коллинеарным координатным осям. Левый (соответственно, правый) дракон порядка n строится следующим образом:

- Если n равно нулю, мы строим отрезок длины 1 из текущей точки в текущем направлении, перемещаясь в конечную точку.
- В противном случае мы строим левого дракона порядка $n - 1$ от заданной точки в заданном направлении, поворачиваемся на 90 градусов влево (соответственно, вправо) в конечной точке и строим правого дракона порядка $n - 1$;

Построим левого дракона порядка n , в качестве стартовой точки выберем начало координат, а в качестве текущего направления — направление возрастания оси OX .

Вам задан некий шаблон как последовательность направлений перемещений единичной длины (направления параллельны координатным осям). Требуется найти, сколько раз данный шаблон встретится в построенной кривой-драконе.

Формат входных данных

Входной файл содержит целое число n ($0 \leq n \leq 10^9$) и строку S , состоящую из символов 'R', 'L', 'U' and 'D', определяющих шаблон. 'R' обозначает движение вправо (по возрастанию оси OX), 'L' — движение влево (по убыванию оси OX), 'U' — движение вверх (по возрастанию оси OY), 'D' — движение вниз (по убыванию оси OY). Длина строки S находится в промежутке от 1 до 10^6 .

Формат выходных данных

Выведите количество заданных шаблонов в левом драконе порядка n , построенном в направлении возрастания оси OX , по модулю $10^9 + 7$.

Примеры

<code>f.in</code>	<code>standard output</code>
2 RU	1
3 L	3
4 LDL	2

Задача G. Points

Имя входного файла: `g.in`
Имя выходного файла: `standard output`
Ограничение по времени: 1 second
Ограничение по памяти: 256 mebibytes

Джон является большим любителем различных аттракционов. На этой неделе он решил развлечься в тире. Мишени в этом тире расположены по прямой. Для каждой мишени заданы три бонуса. За поражение мишени с номером i (мишени занумерованы справа налево, начиная с единицы) прибавляется a_i баллов, если ни одна из соседних мишеней не поражена, b_i , если поражена ровно одна, и c_i , если поражены обе.

Какое наибольшее количество баллов может набрать Джон?

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n < 10^6$) — количество мишеней. i -я из последующих n строк содержит три целых числа — значения бонусов a_i , b_i и c_i для мишени с номером i ($0 \leq a_i, b_i, c_i \leq 1000$).

Формат выходных данных

Выведите одно целое число — наибольшее количество баллов, которое может набрать Джон.

Примеры

<code>g.in</code>	<code>standard output</code>
1 3 0 0	3
1 1 2 3	1

Задача Н. Red John Game

Имя входного файла: `h.in`
Имя выходного файла: `standard output`
Ограничение по времени: `0.5 seconds`
Ограничение по памяти: `256 mebibytes`

На бесконечной шахматной доске изначально стоят $n \times n$ фишек, расположенных в виде квадрата $n \times n$. Фишки могут двигаться горизонтально или вертикально, при этом фишка может сделать ход тогда и только тогда, когда в некотором направлении непосредственно рядом с ней есть другая фишка, а следующее за этой фишкой в том же направлении поле свободно. В этом случае фишка переносится на это поле (то есть её соответствующая координата изменяется на два), а фишка, через которую совершался «прыжок», удаляется с поля.

Выясните, существует ли для заданного n такая последовательность ходов, которая оставляет на доске только одну фишку?

Ниже приведена последовательность ходов для $n = 2$. Пешки обозначены буквами 'P', пустые поля — точками.

```
....      .P..      .PP.      ...P
.PP.      ..P.      ....      ....
.PP.      ..P.      ....      ....
....      ....      ....      ....
```

Формат входных данных

Входной файл содержит одно целое число n ($0 < n < 10^9$).

Формат выходных данных

Выведите 1 в случае, если можно сделать некоторое количество ходов так, чтобы в результате на доске осталась только одна фишка, и 0 в противном случае.

Примеры

<code>h.in</code>	<code>standard output</code>
3	0
4	1

Задача I. Influence

Имя входного файла: `i.in`
Имя выходного файла: `standard output`
Ограничение по времени: `0.5 seconds`
Ограничение по памяти: `256 mebibytes`

В любом обществе можно определить отношение влияния человека x на предпочтения человека y (иначе говоря, ситуации, при которой предпочтения y определяются мнением x). При этом считается, что если x влияет на y , то y не может влиять на x . В сообществе болельщиков популярного футбольного клуба отношение влияния ещё и транзитивно, то есть если x влияет на y и y влияет на z , то и x влияет на z . Таким образом, в этом обществе предпочтения x распространяются на всех, на кого влияет x .

Пусть среди болельщиков имеется группа из X активных фанатов. Председатель совета директоров этого клуба хочет пригласить на должность заместителя по связям с общественностью одного из фанатов. Причём фанат выбирается таким образом, чтобы количество болельщиков, на которое он влияет, было максимальным.

Вам поручено подобрать требуемую кандидатуру.

Формат входных данных

В первой строке входного файла заданы два целых числа, разделённых пробелом: n ($1 \leq n \leq 5000$) — количество болельщиков клуба (пронумерованных числами от 1 до n) и k — количество фанатов.

Следующая строка перечисляет фанатов и содержит $k > 0$ попарно различных целых положительных чисел, не больших n . Далее следуют n строк, описывающих структуру влияния среди болельщиков. i -я из этих строк содержит номера болельщиков, на которые влияет i -й болельщик. Общее количество чисел в этих строках меньше, чем $2.5 \cdot 10^5$.

Формат выходных данных

Выведите одно целое число — наименьший номер фаната, обладающего требуемыми свойствами.

Примеры

<code>i.in</code>	<code>standard output</code>
5 2 1 2 1 3 4 2 3 4 3 5 4 5 5	1
6 3 1 2 3 1 2 2 5 3 4 2 4 6 5 6	3

Задача J. An Idea of Mr. A

Имя входного файла: `j.in`
Имя выходного файла: `standard output`
Ограничение по времени: 0.5 seconds
Ограничение по памяти: 256 mebibytes

Mr. A задал своему сыну следующую задачу:

“Рассмотрим два целых числа n_1 и n_2 такие, что $1 \leq n_1 < n_2 \leq 10^4$. Используя функцию $p : N^* \rightarrow N^*$, $p_n = 2^n$ для всех $n \in N^*$ (N^* — множество целых положительных чисел) мы построим множество

$$S(n_1, n_2) = \{p(p(n)) + 1 \mid n_1 \leq n \leq n_2\}$$

Определим множество пар следующим образом:

$$T(n_1, n_2) = \{(m_1, m_2) \mid m_1, m_2 \in S(n_1, n_2), m_1 < m_2\}$$

Рассмотрим следующую формулу:

$$R(n_1, n_2) = \sum_{(m_1, m_2) \in T(n_1, n_2)} \gcd(m_1, m_2)$$

где $\gcd(m_1, m_2)$ — наибольший общий делитель m_1 и m_2 . Требуется найти число $R(n_1, n_2)$.
Решите задачу.

Формат входных данных

Входной файл состоит из одной строки, содержащей значения n_1 и n_2 , разделённые пробелом.

Формат выходных данных

Выведите вычисленное значение $R(n_1, n_2)$.

Примеры

<code>j.in</code>	<code>standard output</code>
1 34	561
15 147	8778
125 1000	383250

Задача К. Banking (Division 2 Only!)

Имя входного файла: `k.in`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 Mebibytes
Отображение результатов:

Крупный коммерческий банк «Раздербанк» в течение ряда лет собирал статистику ежедневного роста и падения акций банка. Основываясь на этих числах, владельцы банка решили вычислить «оптимальную стратегию» для акционера — в какой из дней было наиболее выгодно всего купить акции банка, а в какой — продать.

Ваша задача — по заданной последовательности чисел (ежедневным изменениям стоимости одной акции) найти «оптимальную тактику» при условии, что можно один раз купить акции банка и один раз полностью продать все купленные акции. Докупать акции или продавать акции частично не разрешается.

Формат входных данных

Входной файл состоит из двух строк. Первая строка содержит целое число N ($1 \leq N \leq 10^6$) — время, в течение которого собиралась статистика. Вторая строка содержит N целых чисел p_i ($-1000 \leq p_i \leq 1000$) — изменение стоимости акции в i -й день. Гарантируется, что хотя бы одно из чисел p_i положительно.

Формат выходных данных

В выходной файл выведите два целых числа k и l ($1 \leq k \leq l \leq N$) — номер дня (нумерация начинается с единицы), когда лучше всего будет купить и продать акции соответственно. В случае, если решение не единственно, выберите решение с наименьшим k , а при совпадении k — с наименьшим l .

Пример

<code>k.in</code>	<code>standard output</code>
11 -3 1 -1 2 3 1 -1 2 -3 -5 7	2 8
9 1 -2 3 -1 -1 3 -2 2 -4	3 6

Задача L. Dimensions (Division 2 Only!)

Имя входного файла: `l.in`
Имя выходного файла: `standard output`
Ограничение по времени: `2 seconds`
Ограничение по памяти: `256 Mebibytes`
Отображение результатов:

На уроке математики в начальных классах ученикам была роздана таблица, в каждой строке которой были записаны высота, ширина, длина и объём некоторого прямоугольного параллелепипеда. При этом ровно одно значение в каждой строке было зачёркнуто. Ученики должны были восстановить исходный вид таблицы. Напишите программу, которая решает аналогичную задачу.

Формат входных данных

Входной файл состоит из нескольких тестовых примеров, каждый из которых состоит из одной строки, содержащей 4 целых числа w , h , l and v — ширину, высоту, длину и объём прямоугольного параллелепипеда соответственно ($0 < l, w, h < 100, 0 < v < 10^5$). Зачёркнутые числа представлены нулями, то есть в каждой строке одно из чисел заменено нулём. Входной файл завершается строкой из четырёх нулей, которая не должна обрабатываться.

Формат выходных данных

Для каждого тестового примера в отдельной строке выведите 4 целых числа — ширину, высоту, длину и объём соответствующего параллелепипеда, заменив во входном примере 0 на вычисленное значение соответствующего параметра. Гарантируется, что вычисленное значение всегда будет целым.

Пример

<code>l.in</code>	<code>standard output</code>
<code>1 0 2 6</code>	<code>1 3 2 6</code>
<code>5 5 5 0</code>	<code>5 5 5 125</code>
<code>0 2 2 80</code>	<code>20 2 2 80</code>
<code>8 0 9 576</code>	<code>8 9 9 576</code>
<code>0 0 0 0</code>	

Задача M. Greatest Product (Division 2 Only!)

Имя входного файла: `m.in`
Имя выходного файла: `standard output`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 Mebibytes
Отображение результатов:

Для каждого целого положительного числа x определим функцию $P(x)$, равную произведению цифр в десятичной записи числа x . По заданному N вычислить максимальное значение $P(x)$ при $x \leq N$.

Формат входных данных

Во входном файле содержится одно целое число N ($1 \leq N \leq 2 \cdot 10^9$).

Формат выходных данных

Выведите максимальное значение функции $P(x)$ на промежутке от 1 до N .

Примеры

<code>m.in</code>	<code>standard output</code>
1	1
101090000	43046721
28994	10368
4876	2268
2789	1008