

Задача А. Звездные треугольники

Имя входного файла: `triangles.in`
Имя выходного файла: `triangles.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов: полное

Жомарт любит наблюдать за звездами и создавать из них различные геометрические фигуры. Небо предоставляется в виде декартовой системы координат, а звезды на ней точками. На этот раз Жомарта интересует вопрос, сколько различных прямоугольных треугольников, у которого катеты параллельны осям координат, можно составить с помощью звезд на небе.

Формат входных данных

В первой строке задается N — количество звезд на небе ($3 \leq N \leq 300000$). В каждой из следующих N строк заданы целые X, Y ($|X, Y| \leq 10^9$) — координаты соответствующей звезды.

Формат выходных данных

Выведите ответ к задаче.

Примеры

<code>triangles.in</code>	<code>triangles.out</code>
3 0 0 1 0 0 1	1
4 0 0 1 0 0 1 1 1	4

По этой задаче будет полное отображение результатов.

В 30% тестов $N \leq 100$.

Задача В. Гипер-минимум

Имя входного файла: `hyper.in`
Имя выходного файла: `hyper.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов: нет

Имеется 4-мерный массив X , каждый индекс которого может принимать значения от 1 до N . Вы должны построить новый 4-мерный массив Y , элементы которого должны принимать следующие значения: $Y[i_1, i_2, i_3, i_4] = \min(X[j_1, j_2, j_3, j_4])$, где $1 \leq i_k \leq N - M + 1, i_k \leq j_k \leq i_k + M - 1$, а M — заданное число.

Формат входных данных

В первой строке входного файла задаются N и M ($1 \leq M \leq N$). Остальные строки файла содержат элементы массива X . Количество элементов не будет превышать 1500000 и сами они будут целыми числами, не превышающими по абсолютному значению 10^9 . Они расположены в таком порядке, что считать их можно с помощью псевдокода:

```
for i = 1 to N:  
  for j = 1 to N:  
    for k = 1 to N:  
      for l = 1 to N:  
        read X[i, j, k, l]
```

Формат выходных данных

Выведите искомый массив Y в том же формате, в котором был дан массив X .

Примеры

hyper.in	hyper.out
1 1 1	1
3 2 3 1 4 -4 0 4 0 0 -3 0 -2 -5 5 3 5 -4 4 -3 -5 -4 -4 5 -1 0 -3 -2 -1 2 -5 -5 -1 1 1 -4 3 5 3 -3 -3 3 0 1 4 -1 -2 3 -2 5 4 -1 -5 3 -4 0 -3 -1 3 -1 4 4 -1 -5 -3 4 -4 5 1 5 -4 3 2 2 -2 -2 4 2 -4 -3 1 3 1	-5 -5 -4 -3 -5 -5 -4 -5 -5 -5 -5 -5 -4 -5 -4 -5

Задача С. Энергичная черепаха

Имя входного файла: `turtle.in`
Имя выходного файла: `turtle.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов: нет

Дана сетка с $N + 1$ рядами и $M + 1$ столбцами. Черепаха находится на клетке $(0, 0)$ и хочет попасть в клетку (N, M) . Черепаха может идти только вверх или вправо. На сетке в K клетках находятся ловушки. Если черепаха пойдет в одну из этих клеток, то она перевернется. У черепашки есть силы для того, чтобы встать не более чем T раз. Посчитайте, сколькими различными путями черепаха может попасть в клетку (N, M) . Так как это число может быть очень большим, выведите остаток от его деления на Z .

Формат входных данных

В первой строке входного файла задается 5 целых чисел: N, M, K, T и Z ($1 \leq N, M \leq 300000$, $0 \leq K, T \leq 20$, $1 \leq Z \leq 1000000000$). В каждой из следующих K строк расположены координаты соответствующей клетки с ловушкой X, Y ($0 \leq X \leq N$, $0 \leq Y \leq M$). Гарантируется, что все клетки с ловушками различные и в клетках $(0, 0)$ и (N, M) ловушек нет.

Формат выходных данных

Выведите требуемое число.

Примеры

<code>turtle.in</code>	<code>turtle.out</code>
1 1 1 0 1000 0 1	1
2 2 0 0 10	6

В 40% тестов $N, M \leq 1000$.

Задача D. Взвешивание камней

Имя входного файла: `stones.in`
Имя выходного файла: `stones.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт
Отображение результатов: нет

Джек нашел N камней и упорядочил их в порядке возрастания их массы. Массы всех камней различны. Самый легкий камень получил номер 1, следующий — 2 и так далее, самый тяжелый получил номер N .

У Джека есть чашечные весы и он решил положить все камни на них в каком-то порядке. Известен порядок, в котором он будет класть камни, и какой камень на какую чашу попадет.

Ваша задача — определить состояние весов после добавления каждого камня. Точные массы камней не известны — даются только их номера.

Формат входных данных

Первая строка содержит целое число N ($1 \leq N \leq 100000$).

Каждая из следующих N строк содержит по два целых числа: R ($1 \leq R \leq N$) и S ($1 \leq S \leq 2$). R — номер камня, который будет положен на чашу S . Все R будут различны.

Формат выходных данных

Выведите N строк — по одной для каждого камня. Если после добавления соответствующего камня чаша 1 тяжелее, выведите “<”. Если сторона 2 тяжелее, выведите “>”. Если невозможно определить, в каком состоянии будут весы, выведите “?”.

Примеры

<code>stones.in</code>	<code>stones.out</code>
5	<
1 2	>
3 1	>
2 1	?
4 2	>
5 1	

Задача Е. Чет-нечет

Имя входного файла: `oddeven.in`
Имя выходного файла: `oddeven.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов: полное

Дана возрастающая последовательность целых чисел 1, 2, 4, 5, 7, 9, 10, 12, 14, 16, 17, ... Она сформирована следующим образом: берется одно нечетное число, затем два четных, затем три нечетных и так далее. Выведите N -й элемент этой последовательности.

Формат входных данных

Одно целое число N ($1 \leq N \leq 10^{100}$).

Формат выходных данных

Выведите одно целое число — N -й элемент последовательности.

Примеры

<code>oddeven.in</code>	<code>oddeven.out</code>
1	1
4	5

В 50% тестов $N \leq 10^{18}$.

Задача F. Скайлайн

Имя входного файла: `skyline.in`
Имя выходного файла: `skyline.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов: нет

Вы хотите, чтобы небоскребы в вашем городе имели красивый вид. Решено построить N небоскребов в ряд. У небоскреба с номером i должно быть ровно $h[i]$ этажей.

У Вас есть предложения от различных строительных компаний. Первая из них предлагает строить один этаж в любом из небоскребов за 3 миллиона евро. Вторая предлагает строить по одному этажу в каждом из двух соседних небоскребов за 5 миллионов евро. Заметим, что не имеет значения, находятся ли эти этажи на одинаковой высоте или нет. Третья компания предлагает строить по одному этажу в каждом из трех последовательных небоскребов за 7 миллионов евро.

Вы можете построить этажи в любом порядке. Вычислите минимальную необходимую сумму денег для строительства.

Формат входных данных

Первая строка содержит одно целое число N ($1 \leq N \leq 300$). Вторая строка содержит N целых чисел $h[1], h[2], \dots, h[N]$, $1 \leq h[i] \leq 200$.

Формат выходных данных

В единственной строке выведите одно целое число: минимальную сумму денег для строительства, в миллионах.

Примеры

skyline.in	skyline.out
3 2 2 2	14
4 1 3 1 1	15

Задача G. Коллайдер

Имя входного файла: `collider.in`
Имя выходного файла: `collider.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов: нет

Физики проводят эксперимент для исследования частиц трёх типов: x , y и z . Они запускают в коллайдер пронумерованный ряд из n частиц. Во время эксперимента происходит воздействие на одну конкретную частицу, после чего частица исчезает с i -ого места ряда и моментально появляется на месте j . После её исчезновения номера частиц, стоящих правее, уменьшаются на 1, а после появления, номера частиц, стоящих правее, увеличиваются на 1. После определенного числа воздействий физики интересуются какая частица стоит на месте k . Напишите программу, которая поможет физикам.

Формат входных данных

В первой строке файла два целых числа: n — количество частиц и m — общее количество воздействий и вопросов ($1 \leq n \leq 1000000$, $1 \leq m \leq 15000$). Во второй строке — последовательность из символов x , y и z длиной n . На каждой из следующих m строк ($1 \leq m \leq 15000$) описано воздействие или вопрос. Строка, в которой описано воздействие, начинается символом a и после пробела дается два целых числа из интервала $[1; n]$. Первое из них показывает начальное, а второе — конечное местоположение частицы во время воздействия. Строка, в которой описан вопрос, начинается символом q и после пробела дается одно целое число из интервала $[1; n]$. Оно указывает позицию, которая интересует физиков.

Формат выходных данных

Выведите столько строк, сколько вопросов во входном файле. В строке номер i надо записать ответ на вопрос i — название соответствующей частицы x , y или z .

Примеры

collider.in	collider.out
15 6	y
xzxуyzzxxzxуyzyx	z
a 2 10	y
a 15 4	
q 3	
a 12 2	
q 14	
q 2	

Пояснение: последовательность после первого воздействия — ххууyzzxxzxуyzyx, последовательность после второго воздействия — ххухуyzzxxzxуyzy, последовательность после третьего воздействия — хухухуyzzxxzxуzy,

Задача Н. К-й путь

Имя входного файла: `kthpath.in`
Имя выходного файла: `kthpath.out`
Ограничение по времени: 2 секунд
Ограничение по памяти: 256 мегабайт
Отображение результатов: нет

У вас есть таблица с N строками и M столбцами. В каждой ячейке таблицы записана одна строчная буква английского алфавита. Рассмотрим все возможные пути от левого верхнего угла до правого нижнего угла, если вам разрешено идти только вправо и вниз. Конкатенация букв в порядке обхода составляют строку. Скажем, что эта строка — значение пути. Теперь рассмотрим все такие пути и отсортируем их значения в алфавитном порядке. Ваша задача найти значение K -го пути в этом отсортированном листе.

Формат входных данных

В первой строке задается два целых числа N — количество рядов и M — количество столбцов заданной таблицы ($1 \leq N, M \leq 30$). Каждая из следующих N строк содержит ровно M строчных букв английского алфавита. Последняя строка входного файла содержит целое число K ($1 \leq K \leq 10^{18}$). Гарантируется, что для K ответ всегда существует.

Формат выходных данных

Первая и последняя строка выходного файла должна содержать одну строку — ответ к задаче.

Примеры

<code>kthpath.in</code>	<code>kthpath.out</code>
3 4 abcd efdg hijk 4	abfdgk

abcdgk, abcdgk, abcdjk, abfdgk, abfdjk, abfijk, aefdgk, aefdjg, aefijk, aehijk