

Задача А. Спуск с горы

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В одном из горнолыжных курортов Италии проводятся соревнования по горнолыжному спуску. Каждому спортсмену предстоит скатиться с горы на лыжах. На любом этапе спуска участник получает определённое число очков. После прохождения трассы очки суммируются. Участник, набирающий наибольшее количество очков, выигрывает. Гора представляет собой треугольник, в качестве элементов которого выступают целые числа — очки за прохождение этапа. На каждом уровне спортсмену предоставляется выбор — двигаться вниз влево или вниз вправо. Начало спуска — в самой высокой точке горы, конец — в любой из самых низких.

```
  1
 4 3
5 6 7
8 9 0 9
```

Требуется найти максимальное количество очков, которое может набрать спортсмен.

Формат входных данных

В первой строке содержится целое число n — количество этапов ($1 \leq n \leq 100$). Далее следуют n строк, каждая из которых характеризует один уровень. В i -й из этих строк содержится ровно i целых чисел: a_1, a_2, \dots, a_i — количество очков в каждой из позиций ($-100 \leq a_k \leq 100$ для всех $1 \leq k \leq i$).

Формат выходных данных

Выведите одно целое число: максимальное количество очков, которое может набрать спортсмен.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4	20
1	
4 3	
5 6 7	
8 9 0 9	

Задача В. Поле

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Отряду нужно пересечь прямоугольное поле размера $m \times n$ квадратов, двигаясь из левого верхнего угла в правый нижний и перемещаясь между соседними квадратами только в двух направлениях — вправо и вниз. Поле не очень ровное, но у отряда есть карта, на которой отмечена высота каждого квадрата. Опасность перехода с квадрата высоты h_1 на соседний квадрат высоты h_2 оценивается числом $|h_2 - h_1|$; опасность всех переходов в пути суммируется. Выясните, какова минимальная опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Формат входных данных

В первой строке заданы два числа m и n через пробел ($1 \leq m, n \leq 100$). В следующих n строках записано по m чисел в каждой; i -е число j -й из этих строк соответствует высоте квадрата (i, j) . Все высоты — целые числа в диапазоне от 1 до 100, включительно.

Формат выходных данных

Выведите одно число — минимальную опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2 1 1 1 1	0
4 2 1 2 3 5 3 8 4 7	6
2 3 1 2 2 3 3 1	4

Задача С. Один козёл

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Лишь один козёл остался в живых после истории с кровной мстью. Он успешно пересёк болото, но теперь он хочет вернуться домой. Для этого ему придется пересечь болото опять! Посчитайте число последовательностей ходов, которые приводят к его смерти, а также число последовательностей ходов, которые приводят к его успешному возвращению домой.

Козёл начинает свое движение из клетки $(1, 1)$ и может увеличить одну или обе своих координаты на единицу (то есть передвинуться вниз, вправо или по диагонали вниз-вправо). Болото представляет собой прямоугольную доску $m \times n$, некоторые квадраты которой покрыты непроходимой трясинной. Если козёл попадает на квадрат с трясинной, он тут же тонет в болоте и умирает. Дом козла находится в клетке (m, n) , после того, как козёл встанет на эту клетку, он считается выжившим и успешно достигшим своего дома.

Болото со всех сторон ограничено лесом, поэтому козёл не может выйти за пределы прямоугольника $m \times n$.

Формат входных данных

Во вводе заданы один или несколько наборов входных данных. Каждый набор начинается строкой, содержащих три целых числа m, n и k ($1 \leq m, n \leq 25$, $0 \leq k \leq mn - 2$). Следующие k строчек содержат координаты клеток, покрытых трясинной. Гарантируется, что клетки $(1, 1)$ и (m, n) трясинной не покрыты. Имейте в виду, что каждая строка завершается переводом строки, а первый тест совпадает с тестом из примера!

Формат выходных данных

Для каждого из наборов входных данных выведите два числа в одной строке. Первое число — количество корректных последовательностей ходов козла, приводящих к его смерти; второе — количество последовательностей, которые приводят к тому, что козёл успешно вернулся домой живым.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2 1	1 2
1 2	2 1
2 2 2	
1 2	
2 1	

Задача D. Грибы

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Маша решила навестить свою бабушку. Она взяла с собой две корзинки — одну с пирожками, а другую — пустую, для грибов, которые она хочет собрать по пути.

Для того, чтобы попасть к бабушке, Маше необходимо пройти через лес, который представляет собой прямоугольник размером $m \times n$, в некоторых клетках которого растут деревья, а в некоторых — грибы. Маша выходит из клетки $(1, 1)$ и идёт к бабушке в деревню, расположенную в клетке (m, n) . Каждым своим ходом Маша может пойти вправо или вниз (то есть увеличить одну и только одну из своих координат на 1), если в клетке, в которой она после этого окажется, не стоит дерево. Если в обеих клетках и справа, и снизу, находятся деревья, то Маша считается заблудившейся.

Вам необходимо по данному лесу выяснить, может ли Маша дойти до бабушки, не заблудившись, и если может, то посчитать максимальное количество грибов, которое она может при этом собрать.

Формат входных данных

В первой строке находятся четыре числа m, n, g, t ($2 \leq m, n \leq 100$, $0 \leq g, t \leq g + t \leq m \cdot n - 2$). В следующих g строках расположены по два числа в каждой — x и y -координаты i -го гриба. За ними следуют t строк с описаниями деревьев в аналогичном формате. Ни в какой клетке не может расти больше одного гриба, гриб и дерево одновременно, или больше одного дерева. Кроме того, в клетках $(1, 1)$ и (m, n) ничего не растёт.

Формат выходных данных

Если Маша может дойти до бабушки, то в первой строке необходимо выдать максимальное количество грибов, которое она сможет при этом собрать, а в последующих $m + n - 1$ строках нужно выдать координаты клеток, последовательно посещаемых Машей, в формате $x_i y_i$, для пути, на котором достигается максимальное количество грибов. Если таких путей несколько, то разрешается выдавать любой из этих путей.

В противном случае нужно вывести единственное число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 4 3 2	2
1 4	1 1
2 3	1 2
4 3	1 3
2 2	2 3
3 4	3 3
	4 3
	4 4
2 2 0 2	-1
1 2	
2 1	

Задача Е. Редакционное расстояние

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В информатике *редакционным расстоянием* между двумя строками называется минимальное количество добавлений, удалений и замен символов, при помощи которых можно из одной строки получить другую. К примеру, редакционное расстояние между строками «ab» и «ab» равно нулю, так как строки равны между собой безо всяких изменений; расстояние между строками «short» и «ports» равно трём: в слове «short» нужно удалить из начала букву «s», заменить «h» на «p» и добавить в конец букву «s». Редакционное расстояние также называют *расстоянием Левенштейна*.

Найдите редакционное расстояние между двумя заданными строками.

Формат входных данных

В первой строчке задана одна строка, во второй — другая. Длины обеих строк — от 1 до 100 символов, включительно.

Формат выходных данных

Выведите единственное число — редакционное расстояние между двумя заданными строками.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
ab ab	0
short ports	3

Задача F. Малое редакционное расстояние

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим две строки S и T , состоящие из символов с ASCII-кодами от 33 до 126 включительно. *Редакционное расстояние* между строками — минимальное количество вставок, удалений и замен символов, которые необходимы для того, чтобы преобразовать одну из этих строк в другую. Вам необходимо найти редакционное расстояние между S и T или сообщить, что оно больше заданного числа k .

Формат входных данных

В первой строке задана строка S , во второй — строка T , а в третьей — целое число k ($0 \leq |S|, |T| \leq 100\,000$, $0 \leq k \leq 50$).

Формат выходных данных

Если искомое редакционное расстояние больше k , то выведите в первой строке слово «Infinity». В противном случае выведите редакционное расстояние.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
abacabadabacaba abacabadabacaba 24	0
abcdef acdefu 7	2
aaaaaaaaa bbbbbbbbbb 5	Infinity

Задача G. Наибольшая общая подпоследовательность Задача H. Три последовательности

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычёркиванием некоторых элементов).

Формат входных данных

В первой строке записано число N — длина первой последовательности ($1 \leq N \leq 1000$).

Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 1000$).

В четвёртой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

Выведите единственное целое число: длину наибольшей общей подпоследовательности, или же число 0, если такой не существует.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 3 4 2 1 3 5	2

Имя входного файла: *стандартный ввод*
 Имя выходного файла: *стандартный вывод*
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 512 мегабайт

Даны три последовательности целых чисел. Ваша задача — найти их наибольшую общую подпоследовательность.

Формат входных данных

Входные данные содержат описание трёх последовательностей. Каждая последовательность задаётся в двух строках. Первая из них содержит длину последовательности n ($1 \leq n \leq 100$), а вторая — её элементы (32-битные целые числа).

Формат выходных данных

Первая строка должна содержать длину максимальной общей подпоследовательности. Саму подпоследовательность необходимо вывести во второй строке. Если таких подпоследовательностей несколько, можно вывести любую из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 3 3 2 1 3 3 1 3 5	2 1 3
3 1 2 3 3 4 5 6 3 1 3 5	0

Задача I. Длина подпоследовательности

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Вам требуется написать программу, которая по заданной последовательности находит длину её максимальной невозрастающей подпоследовательности (то есть такой последовательности чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длины $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать число k — длину максимальной невозрастающей подпоследовательности.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 5 8 10 4 1	3

Задача J. Невозрастающая подпоследовательность

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (то есть такую последовательность чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длины $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке число k — длину максимальной невозрастающей подпоследовательности. В последующих строках должны быть выведены (по одному числу в каждой строке) все номера элементов исходной последовательности i_j , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 5 8 10 4 1	3 2 4 5