|  |  |
| --- | --- |
| **Magic Field** | **e.in** |
| Имя выходного файла: | e.out |
| Максимальное время работы на одном тесте: | 5 секунд |
| Максимальный объем используемой памяти: | 4 мегабайта |
| Максимальная оценка за задачу: | 60 баллов |

Для игры в "Поле чудес" используется круглый барабан, разделенный на сектора, и стрелка. В каждом секторе записано некоторое число. В различных секторах может быть записано одно и то же число.

Однажды ведущий решил изменить правила игры. Он сам стал вращать барабан и называть игроку (который барабана не видел) все числа подряд в том порядке, в котором на них указывала стрелка в процессе вращения барабана. Получилось так, что барабан сделал целое число оборотов, то есть последний сектор совпал с первым.

После этого ведущий задал участнику вопрос: какое наименьшее число секторов может быть на барабане? Напишите программу, отвечающую на этот вопрос.

Формат входных данных

Во входном файле записано сначала число *N* — количество чисел, которое назвал ведущий (2≤*N*≤30000). Затем записано *N* чисел, на которые указывала стрелка в процессе вращения барабана. Первое число всегда совпадает с последним (в конце стрелка указывает на тот же сектор, что и в начале). Числа, записанные в секторах барабана, — натуральные, не превышающие 32000.

Формат выходных данных

Выведите минимальное число секторов, которое может быть на барабане.

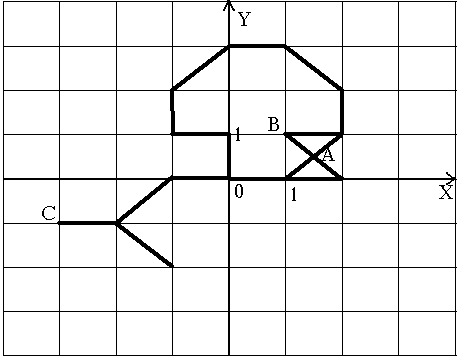
Примеры

|  |  |
| --- | --- |
| e.in | e.out |
| 13  5 3 1 3 5 2 5 3 1 3 5 2 5 | 6 |
| 4  1 1 1 1 | 1 |
| 4  1 2 3 1 | 3 |

Young Firestarter

|  |  |
| --- | --- |
| Имя входного файла: | f.in |
| Имя выходного файла: | f.out |
| Максимальное время работы на одном тесте: | 5 секунд |
| Максимальный объем используемой памяти: | 4 мегабайта |
| Максимальная оценка за задачу: | 120 баллов |

На клеточном поле введена система координат так, что центр координат находится в точке пересечения линий сетки и оси направлены вдоль линий сетки.



На этом поле выложили связную фигуру, состоящую из спичек. Использовались спички двух типов:

* Спички длины 1 выкладывались по сторонам клеток.
* Спички длины  выкладывались по диагоналям клеток.

Ребенок хочет сжечь фигуру. При этом он может поджечь ее в одной точке, имеющей целочисленные координаты (например, в точке A на рисунке поджигать фигуру нельзя, а в точках B и C — можно).

Известно, что огонь распространяется вдоль спички равномерно (но по каждой спичке — со своей скоростью). Спичка может гореть в нескольких местах (например, когда она загорается с двух концов; или когда в середине диагональной спички огонь перекидывается с одной спички на другую — огонь расползается по вновь подожженной спичке в обе стороны).

Напишите программу, которая определит, в какой точке нужно поджечь фигуру, чтобы она сгорела за минимальное время.

Формат входных данных

Во входном файле записано сначала число *N* — количество спичек (1≤*N*≤40). Затем идет *N* пятерок чисел вида *X*1, *Y*1, *X*2, *Y*2, *T*, задающих координаты концов спички и время ее сгорания при условии, что она будет подожжена с одного конца (гарантируется, что каждая спичка имеет длину 1 или , все спички образуют связную фигуру, и положение никаких двух спичек не совпадает). Все координаты — целые числа, по модулю не превышающие 200, время сгорания — натуральное число, не превышающее 107.

Формат выходных данных

Выведите координаты целочисленной точки, в которой нужно поджечь фигуру, чтобы она сгорела за наименьшее время, а затем время, за которое в этом случае фигура сгорит. Время должно быть выведено с точностью не менее 2-х знаков после десятичной точки. Если решений несколько, выведите любое из них.

Примеры

|  |  |
| --- | --- |
| f.in | f.out |
| 1  0 0 1 1 1 | 0 0  1.00 |
| 5  0 0 0 1 1  1 0 0 1 10  0 0 1 0 1  0 0 1 1 1  2 2 1 1 1 | 0 0  3.25 |
| 3  1 1 1 2 10  1 2 2 2 10  1 1 2 2 50 | 2 2  35.00 |
| 16  0 0 0 1 1 -2 –1 –3 –1 1 -2 –1 –1 0 1 -2 –1 –1 –2 1  -1 0 0 0 1 0 3 1 3 1 1 3 2 2 1 2 2 2 1 1  2 1 1 0 1 2 0 1 1 1 2 0 1 0 1 2 1 1 1 1  0 0 1 0 1 0 1 –1 1 1 -1 1 –1 2 1 0 3 –1 2 1 | 0 0  4.50 |

Примечание

Частичные решения для случая, когда время сгорания каждой из спичек равно 1 (вне зависимости от ее длины), будут оцениваться приблизительно половиной баллов.

**CASINO**

|  |  |
| --- | --- |
| *Имя входного файла:* | casino.in |
| *Имя выходного файла:* | casino.out |
| *Ограничение по времени:* | *2 секунды* |
| *Ограничение по памяти:* | *32 мегабайта* |
| *Максимальный балл:* | *100 баллов* |

В интеллектуальном казино «Что? Где? Когдя?» разыгрываются *N* (1 ≤ *N* ≤ 50) писем с вопросами для знатоков. В начале игры письма кладутся на круглый стол, разделенный на *N* секторов, по одному письму на сектор. Игра состоит из нескольких раундов, количество которых не превосходит *N*. В начале каждого раунда выбирается письмо, которое будет играть, по следующему правилу. Запускается волчок, стоящий в центре стола; он останавливается в некотором секторе (мы считаем, что волчок никогда не останавливается на границе секторов). Если в этом секторе лежит письмо, то оно и будет играть. Иначе волчок поворачивают против часовой стрелки до первого сектора, в котором лежит письмо, и играющим объявляют именно это письмо. Играющее письмо убирают со стола.

Игра закончилась. Вам известно, какие письма остались на столе. Требуется определить, сколько возможных последовательностей выпадения волчка приводят к такой конфигурации в конце игры.

***Формат входных данных***

В первой строке входного файла записано число *N*. В следующей строке записано, на каких секторах остались письма в конце игры — последовательность из *N* чисел 0 (нет письма) или 1 (есть письмо) для каждого сектора. Секторы занумерованы против часовой стрелки.

***Формат выходных данных***

В первой строке выходного файла выведите без лидирующих нулей одно число — ответ на задачу.

***Примеры***

|  |  |
| --- | --- |
| **casino.in** | **casino.out** |
| 3  0 1 0 | 3 |
| 6  1 0 1 0 0 0 | 64 |