

Задача 1. В лабиринте

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов:

Александр Ивановичу Корейко приснился очередной кошмарный сон — на этот раз он попал в лабиринт гигантов. Корейко долго бегал по лабиринту, в результате чего заблудился и окончательно вымотался. Теперь он хочет прислониться к стене, чтобы отдохнуть. Какое минимальное время в среднем потребуется ему, чтобы добежать до стены?

Лабиринт имеет простую квадратно-клеточную структуру. Его план можно легко нарисовать на клеточном поле. На этом плане каждая клетка будет либо полностью свободной, либо полностью заполнена стеной. Сторона каждой клетки имеет длину один километр. Будем считать, что на плане есть оси X и Y , которые направлены параллельно сторонам клеток.

Пространство во сне Корейко неевклидово. За бесконечно малое время δt он может сместиться на произвольный вектор $(\delta x, \delta y)$, такой что

$$\max(|\delta x|, |\delta y|) \leq \delta t,$$

считая, что время измеряется в секундах, а расстояние — в метрах.

Известно, что Корейко добежит до стены за минимально возможное время T_{min} . Однако неизвестно, в какой точке лабиринта он находится в начальный момент. Требуется найти математическое ожидание времени T_{min} . Начальная позиция Корейко может с равной вероятностью находиться в любой точке внутри свободных клеток лабиринта.

Формат входных данных

В первой строке входного файла задано два целых числа R и C — количество строк и количество столбцов в плане лабиринта ($1 \leq R, C \leq 10^3$). В следующих R строках описан план лабиринта. Каждая из этих строк содержит ровно C символов. Символ `.` обозначает свободную клетку, а символ `#` — стену.

Гарантируется, что на плане есть хотя бы одна свободная клетка. Все клетки за пределами плана являются стенами.

Формат выходных данных

В выходной файл необходимо вывести среднее время бега в секундах. Абсолютная или относительная погрешность вашего ответа не должна превосходить 10^{-9} .

Примеры

input.txt	output.txt
3 4 #### #.#. ####	166.6666666666666666
3 3 #.# ... #..	305.5555555555555555

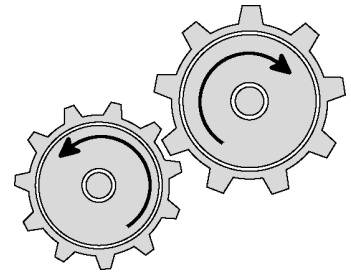
Задача 2. Шестерёнки

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт
Отображение результатов:

Штатному инженеру в НИИ поступило задание наладить работу вышедшей из строя лабораторной установки. После вскрытия устройства он обнаружил, что в основе его функционирования лежат соединённые между собой зубчатые колёса типа «шестерня».

Из руководства по эксплуатации лабораторной установки инженеру стало известно, что вся система приводится в движение при помощи усилия двух лаборантов-исследователей, которые вращают Главное колесо устройства по часовой стрелке. После осмотра оборудования инженер понял, что с задачей он справится достаточно быстро, поскольку не обнаружил циклически соединённых шестерёнок. Это означает, что в установке отсутствуют такие колёса, что их вращательный момент посредством зубчатой передачи возвращается обратно через несколько шестерёнок.

Для того чтобы устранить дефект, инженеру необходимо знать направление вращения концевых колес. Зубчатое колесо называется концевым, если оно соединено не более, чем с одним другим колесом.



Формат входных данных

В первой строке входного файла записаны два целых числа N — количество зубчатых колёс и G — номер Главного колеса ($1 \leq N \leq 10^5$, $1 \leq G \leq N$). Все шестерёнки пронумерованы числами от 1 до N .

В следующих N строках описаны зубчатые соединения в порядке возрастания номеров, по одному на строке. В каждой строке первым записано число g_i — количество шестерёнок, с которыми соединено колесо с номером i ($1 \leq i \leq N$, $0 \leq g_i < N$). Далее в этой же строке через пробел перечисляются номера зубчатых колёс, с которыми соединена описываемая шестерня.

Формат выходных данных

В выходной файл для каждого концевых колес в порядке увеличения номеров необходимо вывести “CW”, “CCW” или символ “-”, в зависимости от того, вращается ли шестерня по часовой стрелке, против часовой стрелки или не вращается соответственно.

Пример

input.txt	output.txt
6 2	CW
3 6 4 5	-
1 5	CCW
0	CCW
1 1	
2 2 1	
1 1	

Замечание

В рамках условия словосочетание «зубчатое колесо» и слова «колесо», «шестерня» и «шестерёнка» являются синонимами.

Задача 3. Дерево каталогов

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт
Отображение результатов:	

Содержимое произвольного каталога на диске можно представить в виде дерева. Корнем дерева является собственно этот каталог. Листья дерева — это файлы, внутренние узлы дерева — это каталоги. У каждого каталога сыновьями являются каталоги и файлы, которые лежат непосредственно в нём.

Для того, чтобы пользователь мог легко ориентироваться в файловой системе, в Безвестной Операционной Системе дерево каталогов представляется в удобном виде. Представление состоит из набора строк, в каждой строке сначала идёт некоторый префикс, а после него записано имя файла или каталога. Префикс состоит из нескольких отступов вида “| ” (ASCII: 124, 32) и флага. Отступы отражают уровень вложенности каталога или файла. Флаг равен ‘-’ (ASCII: 45) для каталога с раскрытым содержимым и ‘.’ (ASCII: 46) для файла.

В первой строке записано имя корневого каталога без отступов. В остальных строках отображено с отступом всё, что находится внутри него. Каталоги и файлы, непосредственно лежащие в корневом каталоге, упорядочиваются произвольным образом с тем условием, что сначала идут описания содержимого всех каталогов, а затем — файлы. Описания каталогов представляются аналогично описанию корневого каталога. Образец форматирования можно посмотреть в примере входного и выходного файлов.

Каталог не может быть пустым. Имена всех файлов и каталогов уникальны.

Разработчики Безвестной Операционной Системы решили добавить функциональность в свою систему. Теперь дерево каталогов можно распечатывать на принтере. При печати на одну страницу входит не более W строк. Поскольку полное описание дерева каталогов может занимать существенно больше строк, то в общем случае нужно разбивать дерево на несколько страниц. Вас, как известного алгоритмиста, просят минимизировать расход бумаги.

Чтобы уместить дерево каталогов на одну страницу, некоторые поддеревья разрешается «закрывать». Если какой-то каталог в дереве «закрывается», то его внутренности в представлении не отображаются. Флаг закрытого каталога — ‘+’ (ASCII: 43).

Пользователь должен на распечатках видеть всю структуру дерева полностью, поэтому содержимое каждого закрытого каталога нужно распечатать на отдельной странице. При этом на отдельной странице закрытый каталог полагается корневым, и распечатывается представление его поддеревья. Какие-то каталоги в нём можно снова «закрывать», а их содержимое распечатать отдельно, и так далее.

На каждой странице может быть только один корневой каталог. Например, нельзя распечатать содержимое двух закрытых каталогов на одной странице, даже если суммарно они занимают меньше W строк. Представление каждого каталога должно входить на одну страницу целиком, нельзя разрывать представление на несколько страниц.

Требуется минимизировать количество используемых страниц и вывести получающееся при этом распечатки.

Формат входных данных

В первой строке входного файла записано одно целое число W — количество строк на странице ($1 \leq W \leq 100$).

В остальных строках входного файла находится полное представление корневого каталога. Гарантируется, что представление соответствует описанному в условии формату, и что все каталоги в нём раскрыты.

Имена всех каталогов и файлов состоят только из строчных букв латинского алфавита. Их длины лежат в диапазоне от 1 до 50.

Количество строк в полном представлении корневого каталога лежит в диапазоне от 2 до 10 000.

Суммарное количество символов во всём представлении не превосходит 10^6 , не считая переводы строк.

Формат выходных данных

Если невозможно распечатать всё дерево согласно правилам, то выведите в единственную строку выходного файла слово “IMPOSSIBLE”.

В противном случае в первую строку требуется вывести одно целое число K — минимальное количество страниц. Далее необходимо вывести содержимое K распечатанных страниц. После содержимого каждой страницы нужно выводить строку с 20 символами “=” (ASCII: 61).

Если оптимальных решений несколько, выведите любое из них. Дерево на каждой странице требуется выводить согласно описанному в условии формату. Строго соблюдайте форматирование: лишние пробелы или переводы строк в данной задаче считаются ошибкой.

Примеры

input.txt	output.txt
5 -root -data -documents -docs .readme .reference -games -adom .adomexe .sudoku .tetris -programs -cpp .helloworld .testlib .trash .deleteme	5 -root +data .trash .deleteme ===== -documents -docs .readme .reference ===== -data +documents +games +programs ===== -games -adom .adomexe .tetris .sudoku ===== -programs -cpp .helloworld .testlib =====

Задача 4. Коварный изоморфизм

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов:

В этом году Вася поступил на первый курс механико-математического факультета. С первых же дней учебы он осознал всю тяжесть своего выбора. Пожалуй, самые сложные определения, которые он узнал этой осенью — это алгебраические системы и их изоморфизм. Сегодня вечером Вася, как всегда, сидит в комнате для занятий и пытается в очередной раз разобраться с этими сложными математическими понятиями.

Используя тетрадь с лекциями, он попробовал выписать определения в частном случае, когда алгебраическая система имеет ровно одну унарную операцию, и вот что у него получилось:

- Унарной операцией на множестве A называется произвольная функция, действующая из A в само множество A .
- Алгебраической системой с одной унарной операцией называется пара $\langle A, \star \rangle$, где A — некоторое множество, а \star — это унарная операция на A .
- Пусть есть две алгебраические системы $M = \langle A, \star \rangle$ и $N = \langle B, \circ \rangle$ с унарными операциями \star и \circ , кроме того, A и B равномощны. Тогда взаимнооднозначное отображение ϕ из A в B называется изоморфизмом алгебраических систем M и N , если $\phi(\star(a)) = \circ(\phi(a))$ для любого $a \in A$.

Чтобы лучше понять, как устроен подобный изоморфизм, Вася решил выписать все различные алгебраические системы вида $\langle A, \star \rangle$, где A — n -элементное множество, \star — унарная операция на A , учитывая, что изоморфные системы считаются одинаковыми. Помогите Васе посчитать общее количество таких алгебраических систем с точностью до изоморфизма, ведь ему сегодня еще надо успеть разобраться с машинами Тьюринга...

Формат входных данных

В первой строке входного файла записаны через пробел целое число n и простое число p ($1 \leq n \leq 50$, $53 \leq p \leq 10^9$).

Формат выходных данных

В выходной файл необходимо вывести одно целое число — остаток от деления на p числа, равного количеству алгебраических систем с точностью до изоморфизма, состоящих из n -элементного множества с одной унарной операцией.

Примеры

<code>input.txt</code>	<code>output.txt</code>
2 53	3
3 83	7

Задача 5. Инопланетные носки

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт
Отображение результатов:

Жена инопланетянина постирала его носки в стиральной машине. Теперь необходимо проверить, для всех ли носков имеются «парные».

Формат входных данных

В первой строке входного файла записано через пробел два целых числа N и L , где N — это количество носков, а L — количество ног у инопланетянина ($1 \leq N \leq 10^6, 1 \leq L \leq 10^4$).

В следующих N строках описываются носки. Каждому носку соответствует одна строка, в которой записано положительное целое число, не превышающее 10^5 — тип этого носка.

Формат выходных данных

В выходной файл необходимо вывести слово “OK”, если все носки можно разбить на группы по L штук так, чтобы каждую из групп составляли носки только одного типа.

В противном случае нужно вывести слово “ARGH!!1”.

Примеры

<code>input.txt</code>	<code>output.txt</code>
4 2 1 2 1 2	OK
6 3 10 15 7 15 10 15	ARGH!!1

Задача 6. Скобки

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов:

Как многим известно, основу синтаксиса LISP-подобных языков составляют скобочки. Круглые, фигурные, квадратные — на любой вкус. Конечно, там применяют и другие символы, но главное — скобочки.

В лабораториях Ну Гляди-какого-умного Университета (НГУ) придумали новый язык программирования — `Uncommon LISP`. Он впитал в себя самую суть всех функциональных языков. В нём кроме скобочек нет ничего...

В первой версии языка корректной программой на `Uncommon LISP` считалась правильная скобочная последовательность из четырех видов скобок: “()”, “{}”, “[]” и “<>”. Правильная скобочная последовательность определяется следующим образом:

- пустая строка — правильная скобочная последовательность;
- правильная скобочная последовательность, взятая в скобки одного типа — правильная скобочная последовательность;
- правильная скобочная последовательность, к которой приписана слева или справа правильная скобочная последовательность — тоже правильная скобочная последовательность.

К сожалению, эти правила были слишком сложными, так что во второй версии языка (`Uncommon LISP v2`) корректными программами стали считаться также те программы, из которых можно получить правильную скобочную последовательность, переставляя скобки определённым образом. В последовательности можно переставлять местами скобки, стоящие рядом, если они обе открывающие или обе закрывающие. Такую операцию можно выполнять сколько угодно раз, в том числе по несколько раз переставляя одни и те же скобки.

Необходимо определить для данной скобочной последовательности, является ли она корректной программой для `Uncommon LISP v2`.

Формат входных данных

В первой строке входного файла записано целое число N — количество тестов, следующих ниже. Каждый тест — это непустая строка, состоящая из скобок. Суммарная длина всех строк не превышает 1 000 000.

Формат выходных данных

В выходной файл необходимо для каждого теста в отдельную строку вывести ответ “T”, если заданную скобочную последовательность можно привести к правильному виду, и “NIL”, если нельзя.

Примеры

<code>input.txt</code>	<code>output.txt</code>
2	T
([])	NIL
)([]	

Задача 7. Шифр Джеймса Бонда

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт
Отображение результатов:

Агент 007 любит вести двойную игру со своими противниками. Они, конечно, об этом догадываются. Недавно в руки некоторого Агентства попала часть переписки Джеймса Бонда. Все письма зашифрованы.

У сотрудников Агентства есть предположение об используемом шифре. При написании писем 007 использует n различных символов: буквы, знаки препинания, пробелы и т.д. Для того, чтобы зашифровать письмо, он заменяет каждый символ c на соответствующее ему кодовое слово A_c . То есть, зашифрованное письмо представляет собой конкатенацию кодовых слов.

Каждому символу соответствует ровно одно кодовое слово, которое не зависит от его позиции в тексте и состоит только из цифр, строчных и прописных букв латинского алфавита. Все кодовые слова различны.

Агентству удалось вычислить кодовые слова для всех используемых символов, тем самым расшифровав письма Джеймса Бонда. Однако у них есть подозрение, что Бонд их дурачит. Ведь если какой-то зашифрованный текст можно расшифровать двумя различными способами, то это помогло бы Бонду подсовывать противникам неверную информацию, тем самым вводя их в заблуждение.

Противники агента 007 наняли вас для того, чтобы проверить, могло ли так произойти, что Бонд их одурачил. Иными словами, существует ли две такие различные последовательности символов c_1, c_2, \dots, c_t и d_1, d_2, \dots, d_s , что $A_{c_1}A_{c_2}\dots A_{c_t} = A_{d_1}A_{d_2}\dots A_{d_s}$.

Формат входных данных

В первой строке входного файла записано целое положительное число $n \leq 100$ — количество различных символов, которые Джеймс Бонд использует при написании писем. В каждой из следующих n строк записано по одному кодовому слову для каждого из символов. Суммарная длина кодовых слов не превышает $5 \cdot 10^4$.

Формат выходных данных

В выходной файл необходимо вывести “YES”, если Бонд дурачит Агентство, и “NO” — если нет.

Примеры

<code>input.txt</code>	<code>output.txt</code>
4 a aB B5 5	YES
3 aB c abc	NO

Задача 8. Генеральная уборка

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт
Отображение результатов:

В одно прекрасное воскресное утро Серёжа решил сделать уборку в своей комнате. Для начала он заглянул под кровать. Там он обнаружил коробку со старыми фотографиями. Одни фотографии были чёрно-белыми, другие — цветными. Так как все фотографии были перемешаны, то Серёжа решил их аккуратно заново сложить. Сначала он вытащил все фотографии из коробки и разложил их на N стопок на столе. Он решил, что в коробку положит фотографии в одну стопку так, чтобы сверху шли все цветные фотографии, а потом уже следовали чёрно-белые.

Поскольку уборка предстояла очень долгая, то Серёжа хотел потратить на фотографии как можно меньше времени. Помогите ему убрать фотографии в коробку за минимальное количество переключений.

За одно переключением разрешается взять несколько фотографий сверху одной стопки и положить их наверх другой стопки, не переворачивая и не перемешивая. При этом нужно выполнять следующие правила:

- старая стопка может исчерпаться, на её месте можно образовывать новую;
- образовывать новые стопки на новом месте нельзя;
- коробка в начальный момент пуста, туда можно только класть фотографии и нельзя вынимать обратно;
- на новом месте фотографии должны располагаться в том же порядке, что и в стопке, откуда их взяли, если их рассматривать сверху вниз.

Формат входных данных

В первой строке входного файла записано одно целое число N — количество стопок с фотографиями ($1 \leq N \leq 1000$). В следующих N строках описаны стопки с фотографиями. Каждая строка начинается с целого числа n_i — количество фотографий в i -ой стопке ($1 \leq i \leq N, 1 \leq n_i \leq 1000$). Далее через пробел перечисляются типы n_i фотографий в i -ой стопке в порядке сверху вниз. Каждый тип обозначается либо “**bw**”, если фотография чёрно-белая, либо “**c**”, если соответствующая фотография цветная.

Формат выходных данных

В выходной файл необходимо вывести одно целое число — наименьшее количество переключений, за которые можно перенести, соблюдая правила, все фотографии в коробку, так, что все цветные фотографии окажутся сверху чёрно-белых. Если добиться этого невозможно, то нужно вывести число -1 .

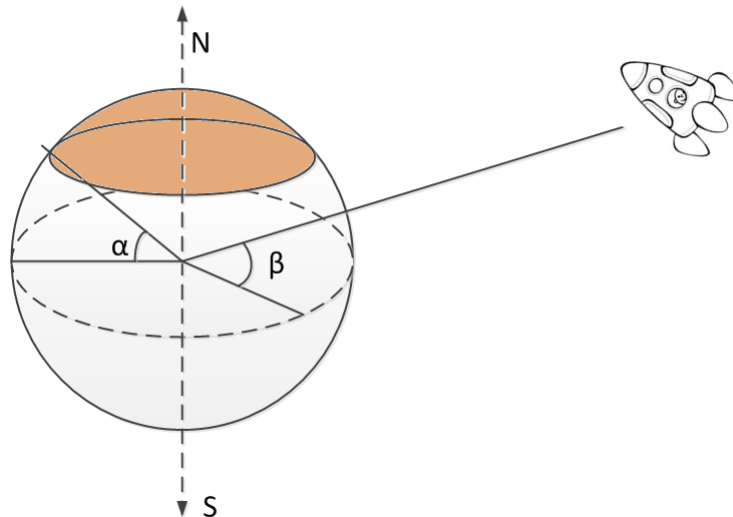
Примеры

<code>input.txt</code>	<code>output.txt</code>
2 2 bw bw 1 c	2
3 1 c 2 bw c 1 c	4

Задача 9. Пятно на планете

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт
Отображение результатов:

Киса и Ося на звездолете «Бриллиантовый дым» открыли в пятой тентуре весьма интересную планетную систему. Вокруг северного полюса планеты имеется круглое пятно, которое заполняет собой всю область планеты, расположенную севернее некоторой параллели α . Корабль же путешественников висит на бесконечном удалении над некоторой точкой планеты на широте β .



Астронавтов весьма беспокоит вопрос — какую долю видимого диска планеты занимает это пятно. Помогите им.

Формат входных данных

Во входном файле через пробел записано два целых числа α и β ($0 \leq \alpha \leq 89$, $-90 \leq \beta \leq 90$). Широта измеряется в градусах от -90 (южный полюс) до 90 (северный полюс).

Формат выходных данных

В выходной файл необходимо вывести одно вещественное число — ответ на поставленную задачу с абсолютной или относительной точностью не менее 10^{-8} .

Пример

<code>input.txt</code>	<code>output.txt</code>
60 0	0.02883444

Задача 10. Сельское хозяйство

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт
Отображение результатов:	

В провинции Хазельбрайт снова беспокойно. Форы развязали очередную междоусобную войну. Дело приняло настолько серьёзный оборот, что в конфликт вмешался император. Один из участников конфликта, благородный лорд Форвин, решил завоевать доверие императора, показав свою якобы любовь к миру и прогрессу. Для этого он хочет развернуть фермерское дело прямо посреди столицы провинции.

Столица провинции была построена в основном из соображений удобства обороны от врага. В городе есть n оборонительных башен, отмеченных на плане точками. Некоторые пары башен соединены стенами, которые на плане нарисованы как прямолинейные отрезки, соединяющие соответствующие точки. Никакие две стены не пересекаются, поэтому соответствующие им отрезки на плане не имеют общих точек, за исключением, возможно, одной точки, которая является концевой для обоих отрезков. Башня не может быть расположена в стене, поэтому на каждом отрезке на плане отмечено ровно две точки, которые являются концевыми точками отрезка.

Для ведения сельского хозяйства нужно выбрать некоторую защищённую область в городе. Чтобы область была защищённой, она должна быть полностью окружена стенами. На плане область должна быть невырожденным многоугольником, стороны которого являются стенами, а вершины — башнями. Граница многоугольника не может иметь самопересечений и самокасаний. При обходе границы каждая вершина-башня должна встречаться не более одного раза.

К сожалению, противники Форвина не собираются прекращать войну в провинции. Поэтому выбранную область, оазис фермерства, нужно будет оборонять от внешних врагов. На все башни и стены, находящиеся на границе области, нужно будет разместить охрану. Форвин прикинул для каждой башни и каждой стены, сколько солдат нужно для её обороны.

Для того чтобы фермерская территория выглядела как можно более мирной, нужно сделать среднюю плотность военного населения на ней как можно меньше. Помогите лорду Форвину выбрать область так, чтобы отношение общего количества солдат, обороняющих границу, к площади области было минимально возможным.

Формат входных данных

В первой строке входного файла записано два целых числа n и m , где n — количество башен в городе, а m — количество стен ($2 \leq n \leq 500$).

В следующих n строках описаны башни. Каждая из них содержит три целых числа x_i , y_i и c_i , где x_i , y_i — координаты башни, а c_i — количество солдат, которое требуется для её обороны ($|x_i|, |y_i| \leq 500, 1 \leq c_i \leq 500$).

В последних m строках описаны стены города. В каждой из них записано по три целых числа a_j , b_j и d_j , где a_j , b_j — номера башен, расположенных на концах стены, а d_j — количество солдат, необходимое для обороны этой стены ($1 \leq a_j < b_j \leq n, 0 \leq d_j \leq 500$). Башни нумеруются с единицы в том порядке, в котором они описаны в файле.

Гарантируется, что никакие две башни не расположены в одной точке, между двумя башнями может быть не более одной стены, никакая башня не расположена в стене, стены не пересекаются (точное определение написано выше в условии).

Формат выходных данных

В первую строку выходного файла необходимо вывести целое число — количество башен на границе найденной области. Во второй строке должны быть записаны через пробел номера башен, которые расположены на границе области, в порядке обхода границы против часовой стрелки. Если в столице невозможно отгородить защищённую область, то нужно вывести единственную строку “Barragar in danger”.

Средняя плотность военного населения для найденной вами области может быть неточной, но относительная погрешность не должна превышать 10^{-9} .

Примеры

input.txt	output.txt
10 13 2 2 5 5 1 3 7 3 6 4 3 7 1 5 2 5 5 5 7 8 50 3 7 3 8 1 1 3 3 50 1 2 3 2 3 7 2 4 20 3 9 3 3 4 5 1 4 27 4 6 0 4 8 3 6 8 0 3 7 7 1 5 2 5 8 3 7 8 1	7 1 2 3 4 6 8 5
5 3 2 4 1 1 1 1 4 1 1 3 3 1 4 3 1 1 2 0 2 3 0 4 5 0	Barrayar in danger

Замечание

В первом примере оптимальная область имеет площадь 17, для обороны границы требуется 51 солдат. В результате получается плотность военных $51/17 = 3$.

Самая большая по площади область (обход "1 2 3 7 8 5") даёт отношение $92/30 = 3\frac{1}{15}$. Область с обходом "4 3 7 8 6" даёт отношение $84/13 = 6\frac{6}{13}$.