

Задача А. Путешествие

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мебибайт |

В стране Байтландии существует n городов, занумерованных от 1 до n . Некоторые из городов соединены двунаправленными дорогами, при этом между двумя городами не может быть проложено более одной дороги.

Приехавший в страну путешественник хочет составить маршрут, который проходит по каждому из городов страны ровно один раз и возвращается в начальную точку. Кроме того, последовательность городов в порядке их посещения должна удовлетворять следующему неравенству: пусть города будут посещены в порядке m_1, m_2, \dots, m_n . Тогда должно существовать такое число i ($1 \leq i \leq n$), что $m_1 < m_2 < \dots < m_i > m_{i+1} > \dots > m_n$ (строго убывающая и строго возрастающая последовательности также являются частными случаями — при $i = 1$ и при $i = n$ соответственно).

Ваша задача — найти кратчайший маршрут с таким свойством, или же определить, что таких маршрутов не существует.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m ($2 \leq n \leq 200\,000$, $1 \leq m \leq 500\,000$).

Каждая из последующих m строк содержит описание одной дороги — три целых положительных числа a_i, b_i, w_i ($a_i \leq b_i \leq n$, $a_i \neq b_i$, $w_i \leq 10^9$), обозначающих, что дорога, соединяющая города a_i и b_i , существует и имеет длину w_i .

Формат выходного файла

В единственной строке выходного файла должно содержаться одно слово „NIE” (*нет* по-польски), если маршрутов, удовлетворяющих условию, не существует, или длина кратчайшего такого маршрута.

Примеры

| Standard input | Standard output |
|--|-----------------|
| 4 5 1 2 2 1 3 3 2 3 3 3 4 4 1 4 1 | 10 |
| 4 4 1 3 1 1 4 1 2 3 1 2 4 1 | NIE |

Задача В. Шахматная доска (Division 1 Only!)

Имя входного файла: Standard input
Имя выходного файла: Standard output
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На проводившемся журналом “Мат вокруг нас” конкурсе по решению шахматных задач жюри решило дать нестандартную задачу.

Задана шахматная доска размера $n \times n$, вертикали и горизонтали которой пронумерованы от 1 до n . Сколькими способами можно разместить на ней n ладей, пронумерованных от 1 до n так, чтобы никакие две ладьи не били друг друга и чтобы ладья с номером i не находилась ни в i -й вертикали, ни в i -й горизонтали? Так как число способов может быть слишком велико, ответ надо давать по модулю m .

Участников конкурса, привычных к задачам типа “Мат в 13 ходов”, эта задача поставила в тупик. А сможете ли решить её Вы?

Формат входного файла

В первой и единственной строке входного файла заданы два целых числа n и m ($1 \leq n \leq 10^{18}$, $1 \leq m \leq 10^6$).

Формат выходного файла

Выведите одно число — требуемое количество расстановок ладей, взятое по модулю m .

Пример

| Standard input | Standard output |
|----------------|-----------------|
| 3 120 | 4 |

Задача С. Парты

| | |
|-------------------------|---------------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 3 секунды |
| Ограничение по памяти: | 32 мебибайта |

В классе n учеников, причём n — чётное число. Некоторые ученики дружат между собой, при этом если ученик x дружит с учеником y , то и ученик y дружит с учеником x . Учитель хочет рассадить учеников по $n/2$ партам так, чтобы за каждой партой сидели два ученика, которые дружат между собой.

Вычислите количество способов, которым учитель может это сделать, при том, что все парты являются одинаковыми (и неразличимы между собой с точки зрения задачи).

Формат входного файла

В первой строке входного файла заданы два целых числа n ($2 \leq n \leq 26$) и m ($0 \leq m \leq n(n-1)/2$). Каждая из последующих m строк содержит два целых числа $1 \leq a, b \leq n$, обозначающих, что ученики a и b дружат между собой. Гарантируется, что n чётно и что каждая из пар $\{a, b\}$ встретится во входном файле не более одного раза.

Формат выходного файла

Выведите одно число — количество возможных способов рассадки учеников по партам в соответствии с пожеланиями учителя.

Пример

| Standard input | Standard output |
|--|-----------------|
| 6 5 1 4 1 5 2 4 2 5 3 6 | 2 |

Задача D. Разность

Имя входного файла: Standard input
Имя выходного файла: Standard output
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задано слово, состоящее из n строчных латинских букв ('a'-'z').

Для всех непустых подстрок этого слова вычислим разность d между количеством вхождений наиболее часто встречающейся в подстроке буквы и наименее часто встречающейся (но всё же встречающейся) в подстроке буквы (в случае подстроки, состоящей из одинаковых символов, эти буквы совпадают и разность равна 0).

Вычислите максимально возможное значение d .

Формат входного файла

В первой строке входного файла задано целое число n ($1 \leq n \leq 1\,000\,000$) — длина слова. Вторая строка содержит само слово — n строчных латинских букв.

Формат выходного файла

Выведите одно число — максимально возможное значение разности d для подстроки заданного слова.

Пример

| Standard input | Standard output |
|------------------|-----------------|
| 10 aabbaaabab | 3 |

Комментарий к примеру: Подстрокой, для которой достигается разность 3, будет **aaaba** — 4 вхождения **a** и одно вхождение **b**.

Задача Е. Мусор

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 9 секунд |
| Ограничение по памяти: | 256 мебибайт |

Город представляет собой n перекрёстков. Некоторые перекрёстки соединены улицами с двусторонним движением, причём каждую пару перекрёстков соединяет не более одной улицы.

Однажды мэр распорядился увеличить плату за сборку мусора. В результате некоторые жители города отказались оплачивать сборку мусора, предпочитая выкидывать мусор прямо на улицы, так что некоторые улицы города оказались завалены мусором.

Для того, чтобы убедить жителей всё же платить за уборку мусора, мэр издал секретное распоряжение: убрать мусор только с тех улиц, на которых большинство жителей оплатило уборку. Остальные улицы останутся неубранными, или — если вдруг эти улицы были чистыми — на них будет вывален мусор, собранный с других улиц. К распоряжению был приложен план города, на котором обозначены как те улицы, которые должны быть очищены от мусора, так и те, которые должны в итоге оказаться (или остаться) заваленными мусором.

Проблема состояла в том, что уборкой улиц города занимаются автоматизированные уборочные машины. Система предписаний (маршрут) данной машины задаётся следующим образом: машина начинает работу с заданного перекрёстка, проезжает по заданным улицам (при этом каждый перекрёсток, кроме исходного, может быть посещён не более одного раза) и возвращается на исходный перекрёсток (при этом исходный перекрёсток не может быть посещён ещё раз в середине пути). Если машина проезжает по грязной улице, она убирает мусор с этой улицы; если машина проезжает по чистой улице, она сбрасывает на ней мусор.

Мэр интересуется, возможно ли с помощью этих машин выполнить его распоряжение. Вам необходимо выяснить это, и, в случае, если возможно, построить пример соответствующего набора маршрутов.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^6$) — количество перекрёстков и улиц в городе соответственно. Перекрёстки занумерованы последовательными целыми числами от 1 до n .

Каждая из последующих m строк задаёт одну улицу. Улица задаётся четырьмя целыми числами: a , b , s и t ($1 \leq a < b \leq n$, $s, t \in \{0, 1\}$) — номера перекрёстков, соединённых дорогой, загрязнённость улицы на данный момент ($s = 0$ — улица чистая, $s = 1$ — улица грязная), загрязнённость улицы после реализации плана мэра ($t = 0$ — улица должна быть чистой, $t = 1$ — улица должна быть грязной) соответственно.

Гарантируется, что если существует набор маршрутов, который реализует распоряжение мэра, то существует и такой набор, для которого общее количество проездов автоматизированной уборочной машины по улицам не превосходит $5 \cdot m$.

Формат выходного файла

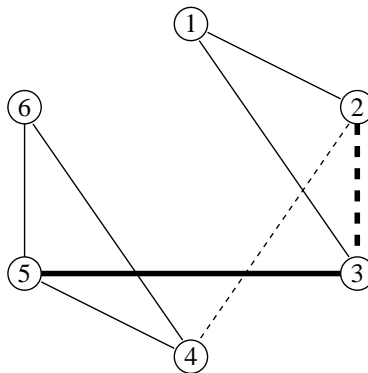
Если требуемого набора маршрутов не существует, в единственной строке выходного файла выведите слово “NIE” (*нет* по-польски).

Иначе выведите произвольный набор маршрутов, реализующий распоряжение мэра, для реализации которого машина должна проехать в общей сложности не более, чем по $5 \cdot m$ улицам. Первая строка выходного файла должна задавать количество маршрутов k . В последующих k строках должны быть описаны маршруты, по одному маршруту на строку. $i + 1$ -я строка выходного файла должна начинаться с целого числа k_i — количества улиц, из которых состоит i -й маршрут. Далее должны следовать $k_i + 1$ чисел — номера перекрёстков в порядке их объезда на данном маршруте.

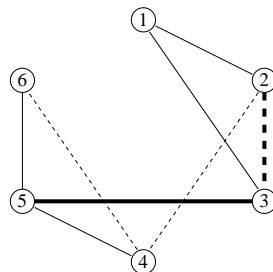
Примеры

На иллюстрациях к примерам чистые улицы обозначены тонкой линией, грязные — жирной. Улицы, которые должны быть очищены, обозначены разрывной линией, улицы, которые должны стать (или остаться) грязными — непрерывной линией.

| Standard input | Standard output |
|----------------|-----------------|
| 6 8 | 2 |
| 1 2 0 1 | 3 1 3 2 1 |
| 2 3 1 0 | 3 4 6 5 4 |
| 1 3 0 1 | |
| 2 4 0 0 | |
| 3 5 1 1 | |
| 4 5 0 1 | |
| 5 6 0 1 | |
| 4 6 0 1 | |



| Standard input | Standard output |
|----------------|-----------------|
| 6 8 | NIE |
| 1 2 0 1 | |
| 2 3 1 0 | |
| 1 3 0 1 | |
| 2 4 0 0 | |
| 3 5 1 1 | |
| 4 5 0 1 | |
| 5 6 0 1 | |
| 4 6 0 0 | |



Задача F. Автоострады (Division 1 Only!)

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 12 секунд |
| Ограничение по памяти: | 256 мегабайт |

Байтландия представляет собой небольшое государство, n крупнейших городов которого соединены $n - 1$ обычными дорогами с двусторонним движением.

По этим дорогам можно проехать между любыми двумя городами. Однако в последнее время обычные дороги оказались перегружены транспортом, так что было решено между некоторыми городами построить высокоскоростные автоострады.

Термин “путь” в данной задаче обозначает последовательность обычных дорог и/или автоострад, не проходящую дважды через один и тот же город. Для каждого двух городов x, y существует ровно один путь, который не содержит ни одной автоострады. Назовём этот путь “базовым”.

Те, кто едут из города x в город y , могут или использовать базовый путь, или ехать по некоторой автоостраде. В этом случае для предотвращения “пробок” на базовом пути должно выполняться следующее правило: путь не должен содержать городов, лежащих на базовом пути (естественно, за исключением городов x и y), а также должен содержать ровно одну автоостраду (и возможно, какое-то количество обычных дорог).

Ваша задача — написать программу, которая по заданной схеме дорог и автоострад отвечает на запросы о количестве допустимых путей между двумя указанными в запросе различными городами.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$) — количество городов в Байтландии. Города занумерованы целыми числами от 1 до n .

В каждой из последующих $n - 1$ строк заданы два целых числа a_i, b_i ($1 \leq a_i \neq b_i \leq n$), обозначающие, что города a_i и b_i соединены обычной дорогой.

Далее следует строка, содержащая одно целое число m ($1 \leq m \leq 10^5$) — количество автоострад. Каждая из последующих m строк содержит два целых числа a_i, b_i ($1 \leq a_i \neq b_i \leq n$), обозначающие, что города a_i и b_i соединены автоострадой.

Далее следует строка, содержащая одно целое число q ($1 \leq q \leq 5 \cdot 10^5$) — количество запросов. Каждая из последующих q строк содержит один запрос — два целых числа f_i, t_i ($1 \leq f_i \neq t_i \leq n$) — номера городов, количество путей между которыми необходимо вычислить.

Формат выходного файла

Выходной файл должен содержать ровно q строк. i -я строка должна содержать ответ на i -й запрос.

Пример

| Standard input | Standard output |
|----------------|-----------------|
| 9 | 1 |
| 1 2 | 4 |
| 2 3 | 2 |
| 4 2 | 2 |
| 1 5 | |
| 5 6 | |
| 7 5 | |
| 7 8 | |
| 9 7 | |
| 4 | |
| 2 5 | |
| 3 4 | |
| 6 4 | |
| 8 3 | |
| 4 | |
| 4 9 | |
| 2 5 | |
| 1 6 | |
| 1 7 | |

Задача G. Imprisoning Godzilla (Division 1 Only!)

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 6 секунд |
| Ограничение по памяти: | 256 мегабайт |

Жителям небольшого селения на окраине леса удалось поймать монстра, нападавшего на оказывавшихся в лесу людей. После поимки оказалось, что этот монстр представляет неизвестный науке вид, возможно даже, что это — последний представитель своего вида на планете. Так что было решено монстра не убивать, а содержать в специальном загоне до приезда учёных из столицы.

Было решено построить треугольный загон из прочных досок, а в качестве вершин треугольника использовать три дерева близлежащего леса. При этом, по мнению старейшин, площадь загона должна быть минимальна (но не равна нулю) — чтобы как можно большая часть леса была свободна от чудовища.

Ваша задача — по заданным координатам деревьев в лесу вычислить значение удвоенной площади загона.

Формат входного файла

Первая строка входного файла содержит целое число n ($3 \leq n \leq 2000$) — количество деревьев в близлежащем лесу. Каждая из последующих n строк содержит два целых числа x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты очередного дерева. Никакие два дерева не растут в одной и той же точке. При этом гарантируется, что всегда существуют три дерева, не лежащие на одной прямой.

Формат выходного файла

Выведите одно число — **удвоенную** наименьшую площадь невырожденного треугольника, образованного тремя точками, в которых растут деревья заданного леса.

Пример

| Standard input | Standard output |
|---|-----------------|
| 5 -7 4 -7 2 7 -2 -5 5 5 -4 | 4 |

Задача Н. Повороты дерева (Division 1 Only!)

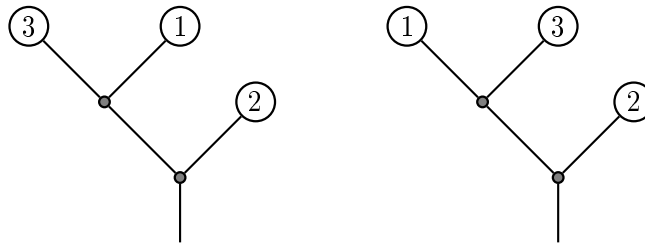
| | |
|-------------------------|---------------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 12 секунд |
| Ограничение по памяти: | 128 мегабайт |

Дерево *Rotatus Informatikus* обладает следующими свойствами:

- Дерево состоит из прямых веток, “развилок” и листьев. Ствол дерева, растущий из земли, также считается веткой.
- Каждая ветка заканчивается или развилкой, или листом.
- Каждая развилка состоит ровно из двух веток — левая ветка и правая ветка.
- Каждый лист на дереве помечен целым числом из диапазона $1..n$, при этом никакие два различных листа не помечены одинаковыми числами.
- В зависимости от освещения каждая развилка может сделать *поворот*, меняя местами левую и правую ветки.

Короной дерева называется последовательность целых чисел, которая получается при чтении чисел, написанных на листьях, слева направо.

Исследуя это дерево, один из учёных заинтересовался вопросом, насколько упорядоченной может оказаться корона дерева после некоторого количества поворотов. Упорядоченность дерева измеряется количеством *инверсий* в его короне a_1, a_2, \dots, a_n , то есть количеством пар (i, j) , $1 \leq i < j \leq n$ таких, что $a_i > a_j$.



Приведённое на примере дерево имеет 5 ветвей. Для изначального положения дерева (на рисунке слева) с короной 3, 1, 2 количество инверсий равно двум. Применяя поворот, получаем дерево с короной 1, 3, 2, в которой количество инверсий равно одному.

Напишите программу, которая определяет минимальное количество инверсий в дереве, которое может быть получено после применения некоторого количества поворотов к исходному дереву.

Формат входного файла

В первой строке входного файла задано целое число n ($2 \leq n \leq 10^6$), задающее количество листьев в дереве. В последующих строках рекурсивно задаётся само дерево:

- Если на конце условного ствола имеется лист, помеченный числом p ($1 \leq p \leq n$), то описанием дерева является единственная строка, содержащая единственное число p .
- Если на конце условного ствола имеется развилка, то описание дерева состоит из трёх частей:
 - Первая строка содержит число 0,
 - Далее следует описание левого поддерева (в качестве условного ствола берётся левая ветка).

- Далее следует описание правого поддерева (в качестве условного ствола берётся правая ветка).

Формат выходного файла

Выведите одно целое число: минимальное количество инверсий в короне дерева, которое может быть получено из заданного во входном файле с помощью поворотов.

Пример

| Standard input | Standard output |
|----------------|-----------------|
| 3 | 1 |
| 0 | |
| 0 | |
| 3 | |
| 1 | |
| 2 | |

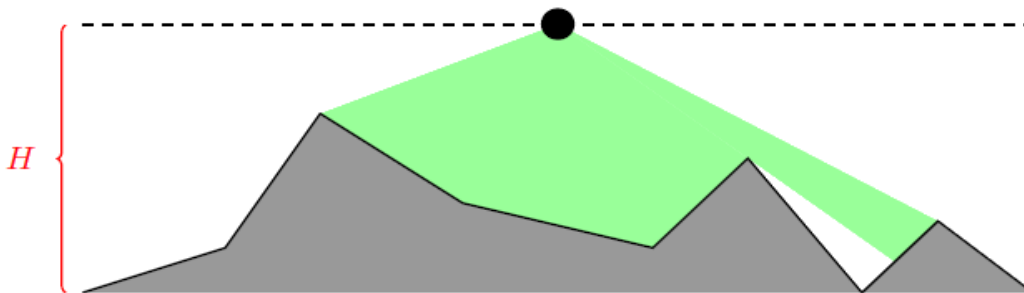
На иллюстрации изображено дерево, заданное в примере.

Задача I. Охота в горах

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мебибайт |

После поимки уже известного вам опасного монстра нападения на людей не прекратились. Исследования показали, что в горной местности активен ещё один экземпляр этого реликтового животного. Поиски были осложнены тем, что животное обладает феноменальной способностью к маскировке. Поэтому было принято решение привлечь для координации действий при отлове животного вертолёт.

Местность, в которой происходит действие, можно представить как двумерный профиль, полученный соединением нескольких точек (называемых “определяющими” точками рельефа) с различными x -координатами.



Про некоторые из “определяющих” точек известно, что в них может быть расположен вход в логово монстра.

Вертолёт может зависать неподвижно на высоте H (то есть на прямой $y = H$). Каждый такой вертолёт контролирует все точки, которые могут быть соединены отрезком с точкой, занимаемой вертолётom так, что отрезок не пересекает ломаную, образованную профилем местности, во внутренней точке этой ломаной (но может касаться вершины ломаной).

Ваша задача — по заданным точкам, в которых может быть логово монстра, вычислить, какое наименьшее количество вертолётов необходимо для того, чтобы одновременно контролировать все эти точки.

Формат входного файла

Первая строка входного файла содержит два целых числа n и H ($1 \leq n \leq 10^5$, $1 \leq H \leq 10^6$). Последующие n строк содержат описание профиля местности — последовательно соединяемые отрезками точки. Каждая из этих строк содержит три целых числа x_i , y_i и z_i ($0 \leq x_i \leq 10^6$, $0 \leq y_i < H$, $z_i \in \{0, 1\}$). (x_i, y_i) обозначают координаты точки, $z_i = 1$ обозначает, что в этой точке может быть логово монстра, в ином случае $z_i = 0$.

Гарантируется, что $y_1 = y_n = 0$ и что точки заданы в порядке возрастания x -координат.

Формат выходного файла

Выведите одно число — требуемое наименьшее количество вертолётов.

Пример

| Standard input | Standard output |
|---|-----------------|
| 9 30 0 0 1 15 5 1 25 20 1 40 10 1 60 5 1 70 15 1 82 0 1 90 8 1 100 0 1 | 2 |

Объяснение примера. Будет достаточно разместить два вертолѐта — в точках с $x = 31$.(6) и с $x = 112$.

Задача J. Сейф (Division 1 Only!)

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Компания «Strong Box» выпустила на рынок новый вид устройства: комбинаторный сейф. Комбинаторный сейф открывается с помощью одного колеса с цифрами. Колесо может находиться в n различных позициях, пронумерованных от 0 до $n - 1$. Некоторые из этих позиций открывают замок, некоторые — нет.

Известно, что если позиции колеса с номерами x и y открывают замок, то и позиция $(x + y) \bmod n$ тоже открывает замок (в частности, это верно и при $x = y$).

При тестировании были проверены k различных позиций колеса: m_1, m_2, \dots, m_k . Позиции m_1, m_2, \dots, m_{k-1} не открывают сейф, в то время как позиция m_k открывает.

По этой информации требуется вычислить, какое максимальное количество позиций колеса могут оказаться «открывающими».

Формат входного файла

В первой строке входного файла содержатся два целых числа n и k , разделённые пробелом ($1 \leq k \leq 250\,000$, $k \leq n \leq 10^{14}$).

Во второй строке заданы k различных целых чисел m_1, m_2, \dots, m_k , также разделённых пробелами ($0 \leq m_i < n$).

Гарантируется, что входные данные соответствуют некоторому комбинаторному сейфу, отвечающему условию задачи.

Формат выходного файла

Выведите одно целое число — максимальное количество позиций колеса, которые в соответствии с имеющейся информацией могут открывать замок сейфа.

Пример

| Standard input | Standard output |
|------------------------|-----------------|
| 42 5 28 31 10 38 24 | 14 |

Задача К. Круги (Division 2 Only!)

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

На плоскости задан круг радиуса R и n одинаковых кругов меньшего радиуса r , размещённых внутри большого круга так, чтобы каждый внутренний круг касался большого (изнутри) и двух соседних с ним внутренних (при $n = 2$ — одного оставшегося внутреннего). По заданным R и n вычислите r .

Формат входного файла

В первой строке входного файла задано количество тестовых примеров T ($1 \leq T \leq 4100$). В последующих T строках заданы тестовые примеры, по одному на строку. Каждый тестовый пример задан двумя числами: радиусом внешнего круга R (не более шести знаков после десятичной точки) и целым числом n — количеством внутренних кругов ($1 \leq R \leq 100$, $2 \leq n \leq 100$).

Формат выходного файла

Для каждого тестового примера в порядке их следования во входном файле выведите радиус внутреннего круга r с точностью до 10^{-3} .

Пример

| Standard input | Standard output |
|----------------|-----------------|
| 4 | 1.513 |
| 5.0 7 | 0.707 |
| 5.0 19 | 0.425 |
| 3.1415 20 | 28.000 |
| 56 2 | |

Задача L. Доставка ноутбуков (Division 2 Only!)

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Для церемонии награждения победителей крупного турнира по программированию водителю службы доставки крупной компьютерной фирмы нужно доставить X ноутбуков на автомобиле, который может донести максимум Y ноутбуков.

На складе ноутбуки лежат в ящиках, каждый из которых содержит (в зависимости от маркировки) от 0 до 25 ноутбуков. Пустой ящик маркирован буквой 'A', ящик с одним ноутбуком — буквой 'B', и так далее, то есть ящик с 25 ноутбуками будет маркирован буквой 'Z'. Ящики выставлены в ряд по одному, причём внешне они неразличимы.

Так как водитель потерял много времени в пробках, он решает не смотреть на маркировку ящиков. Сначала он выбирает случайным образом один из ящиков в ряду — «начальный» ящик. Затем он выбирает случайным образом «конечный» ящик между начальным и последним ящиком в ряду (включая как тот, так и другой). После чего все ящики, начиная с начального и заканчивая конечным, грузятся в машину — и водитель отправляется в путь.

Если количество ноутбуков не превосходит Y и не менее X , то водитель успешно выполнит заказ за одну поездку. Если в машине оказалось менее, чем X ноутбуков, то водителю придётся возвращаться на базу вторично. Если в машине оказалось более, чем Y ноутбуков, то машина сломается из-за перегрузки.

По заданным значениям X и Y , а также заданным расположению и маркировке ящиков на складе вычислите вероятность каждого из трёх вышеупомянутых исходов. Предполагается, что водитель всегда сможет загрузить в кузов выбранное количество ящиков и что оба раза водитель может выбрать любой из доступных для выбора ящиков с равной вероятностью.

Формат входного файла

Входной файл состоит из трёх строк. В первой строке задано целое число N ($1 \leq N \leq 2 \cdot 10^5$) — количество ящиков на складе. Вторая строка содержит последовательность заглавных латинских букв B_1, \dots, B_N — маркировки ящиков в порядке их следования в ряду от начала до конца. В последней строке заданы два целых числа X и Y . X — количество ноутбуков, которое требуется доставить, Y — максимальное количество ноутбуков, которое может перевезти автомобиль без поломки. ($0 \leq X \leq Y \leq 5 \cdot 10^4$).

Формат выходного файла

Выведите 3 числа — вероятности успешного выполнения задания, дополнительного возвращения на базу и поломки автомобиля соответственно с точностью не менее 10^{-6} .

Примеры

| | |
|-----------------|-----------|
| Standard input | |
| 4 | |
| ZDHA | |
| 2 10 | |
| Standard output | |
| 0.5 | 0.25 0.25 |

| Standard input | | |
|-----------------|--------------|--------------|
| 3 | | |
| DEF | | |
| 5 6 | | |
| Standard output | | |
| 0.3333333333 | 0.2777777778 | 0.3888888889 |

Задача М. Крюки с верёвками (Division 2 Only!)

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Однажды в Средиземье армия людей и армия эльфов совместно осаждали замок орков.

При штурме замка была использована следующая тактика: солдаты выстраивались в линию перед стеной замка, после чего каждый из них закидывал крюк с верёвкой на верх стены и лез наверх по этой верёвке. Если крюк не был брошен прямо, то возможен случай, при котором две верёвки пересекутся — тогда ни один из двух солдат, с которыми это произошло, не сможет забраться на стену. Во избежание таких инцидентов в каждой армии солдаты регулярно практиковались совместно забрасывать крюки.

Обе армии — как люди, так и эльфы — подготовлены отлично, так что верёвки от крюков, брошенные солдатами одной расы, заведомо не пересекутся. Однако возможен вариант, когда верёвка от крюка, брошенного эльфом, пересечётся с верёвкой от крюка, брошенного человеком.

Поэтому объединённая армия провела совместные учения. Ваша задача — по заданным позициям людей и эльфов, а также по точкам, в которых их крюки зацепились за верх стены определить, у какого количества различных пар людей и эльфов верёвки пересеклись.

Более формально, если в учениях принимали участие n человек и m эльфов, их позиции перед стеной занумерованы от 1 до $n + m$ слева направо. Точки попадания крюков на верхней части стены также занумерованы слева направо от 1 до $n + m$, при этом позиция i на стене находится точно напротив позиции i в строю. Верёвка от крюка, брошенного из позиции i в позицию j , пересекает верёвку от крюка, брошенного из позиции k в позицию l , тогда и только тогда, когда $(i < k$ и $j \geq l)$ или $(i > k$ и $j \leq l)$.

Верёвки, брошенные солдатами одной расы, гарантированно не пересекаются, никакие два солдата не занимают одно и то же место в строю. Если два крюка, брошенные представителями разных рас, попали в одно и то же место, то считается, что соответствующие верёвки пересекаются.

Формат входного файла

В первой строке входного файла содержатся два целых числа n и m — численность армии людей и армии эльфов соответственно ($1 \leq n, m \leq 3 \cdot 10^4$). В следующих n строках заданы позиции людей, в последующих m — позиции эльфов. Каждая из этих строк состоит из двух целых чисел i и j , разделённых пробелом — номера позиции, в которой стоит солдат, и номера места, в которое попал его крюк, соответственно ($1 \leq i, j \leq n + m$).

Формат выходного файла

Выведите количество различных пар солдат, у которых после броска верёвки пересеклись.

Примеры

| Standard input | Standard output |
|---|-----------------|
| 3 3 1 2 3 4 5 6 2 1 4 3 6 5 | 3 |
| 4 2 5 3 3 1 4 2 6 6 1 3 2 5 | 6 |

Задача N. 4 буквы (Division 2 Only!)

Имя входного файла: Standard input
Имя выходного файла: Standard output
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам задан список строк над алфавитом, состоящим из 4 букв 'X', 'Y', 'Z', 'W'. Найдите минимальную строку, содержащую все заданные строки в качестве подстрок. Если таких строк наименьшей длины несколько, выберите лексикографически минимальную.

Формат входного файла

В первой строке входного файла задано целое число n — количество строк в списке ($1 \leq n \leq 15$). Далее следуют сами строки. Каждая из заданных строк непуста, не превосходит по длине 100 символов и состоит только из букв 'X', 'Y', 'Z', 'W'.

Формат выходного файла

Выведите минимальную по длине (при наличии нескольких таковых — лексикографически минимальную из них) строку, содержащую все заданные строки в качестве подстрок.

Пример

| Standard input | Standard output |
|--------------------|-----------------|
| 2 ZXYWYW YWZ | ZXYWYWZ |

Задача О. Стрелялка (Division 2 Only!)

| | |
|-------------------------|-----------------|
| Имя входного файла: | Standard input |
| Имя выходного файла: | Standard output |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Одна из классических игр-стрелялок устроена следующим образом: у каждого игрока есть танк, задача — уничтожить выстрелом танк оппонента. Танки неподвижны, игроки стреляют по очереди и могут управлять углом наклона орудия и начальной скоростью снарядов. Начальная скорость не превышает $300m/s$, а также не может быть отрицательной. При этом оба танка представляют собой точки.

Кроме того, в игре реализован ветер. Считается, что ветер придаёт снаряду некоторое постоянное ускорение w , действующее вдоль оси x . Ускорение свободного падения в данной модели считается равным $9.8m/s^2$ и действует вдоль оси y в обратном направлении (то есть равно $-9.8m/s^2$).

На текущем этапе игры ваша задача усложнена — пушку танка заклинило под фиксированным углом, и вы можете управлять только скоростью выстрела.

Формат входного файла

В первой строке входного файла задано число T ($1 \leq T \leq 1000$) — количество тестовых примеров. Каждый тестовый пример задаётся в отдельной строке и представляет собой 6 чисел с плавающей точкой (гарантированно помещающихся в тип `double`): x_u, y_u, x_o, y_o, w, d — соответственно координаты в метрах вашего танка (x_u, y_u) и танка оппонента (x_o, y_o) ($0.0 \leq x_u < x_o \leq 1000.0$, $0.0 \leq y_u, y_o \leq 800.0$), ускорение w ($-2.0 \leq w \leq 2.0$) ветра в m/s^2 вдоль оси x , угол d ($0 < d < 78$) наклона пушки (при $d = 0$ выстрел идёт вдоль оси x по возрастанию координат, при $d = 90$ пушка стреляла бы вертикально вверх).

Формат выходного файла

Для каждого тестового примера выведите значение начальной скорости, при котором снаряд поражает цель, с точностью до 10^{-5} . Если цель при заданных условиях поразить невозможно, выведите -1 .

Пример

| Standard input |
|--|
| 2 |
| 24.980037311710223 288.98374755660933 382.68264850317206 369.12118991314696 0.0 21.444206874614686 |
| 390.2686697386107 201.33079962137526 440.4560006234233 692.7031898095327 0.0 51.0674695364569 |
| Standard output |
| 109.49391 |
| -1 |