

Problem A

Pizza delivery

Your Irish pizza and kebab restaurant is doing very well. Not only is the restaurant full almost every night, but there is also an ever increasing number of deliveries to be made, all over town. To meet this demand, you realize that it will be necessary to separate the delivery service from the restaurant. A new large kitchen, only for baking pizzas and being a base for deliveries, has to be established somewhere in town.

The main cost in the delivery service is not the making of the pizza itself, but the time it takes to deliver it. To minimize this, you need to carefully plan the location of the new kitchen. To your help you have a database of all last year's deliveries. For each block in the city, you know how many deliveries were made there last year. The kitchen location will be chosen based on the assumption that the pattern of demand will be the same in the future.

Your city has a typical suburban layout – an orthogonal grid of equal-size square blocks. All places of interest (delivery points and the kitchen) are considered to be located at street crossings. The distance between two street crossings is the Manhattan distance, i.e., the number of blocks you have to drive vertically, plus the number of blocks you have to drive horizontally. The total cost for a delivery point is its Manhattan distance from the kitchen, times the number of deliveries to the point. Note that we are only counting the distance *from* the kitchen *to* the delivery point. Even though we always drive directly back to the kitchen after a delivery is made, this (equal) distance is not included in the cost measure.

Input specifications

On the first line, there is a number, $1 \leq n \leq 20$, indicating the number of test cases. Each test case begins with a line with two integers, $1 \leq x \leq 100$, $1 \leq y \leq 100$, indicating the size of the two-dimensional street grid. Then follow y lines, each with x integers, $0 \leq d \leq 1000$, indicating the number of deliveries made to each street crossing last year.

Output specifications

For each test case, output the least possible total delivery cost (the sum of all delivery costs last year), assuming that the kitchen was located optimally. There should be one line for each test case, with an integer indicating the cost, followed by a single space and the word 'blocks'.

Sample input

```
2
4 4
0 8 2 0
1 4 5 0
0 1 0 1
3 9 2 0
6 7
0 0 0 0 0 0
0 1 0 3 0 1
2 9 1 2 1 2
8 7 1 3 4 3
1 0 2 2 7 7
0 1 0 0 1 0
0 0 0 0 0 0
```

Output for sample input

```
55 blocks
162 blocks
```

Задача В. Стритрейсинг

Ты стоишь на светофоре на своей машине, а рядом с тобой или через машину стоит такой же, как ты. Ты с ним не знаком, ты даже понятия не имеешь, кто он такой, но ты знаешь: сейчас начнется оно... Вы не сигналили друг другу, не газуете, но оба понимаете: да, сейчас будет оно, то самое. И по сигналу светофора с визгом резины и ревом выхлопной системы вы срываетесь вперед, пытаетесь выяснить, чья машина быстрее. Из всех машин, стоящих на светофоре, только вы вдвоем сорвались. Если одним из них был ты, то ты настоящий стритрейсер.

Чаще всего сибирские гонщики собираются на недостроенной взлетно-посадочной полосе за городом. Ориентир — развилка перед аэропортом Толмачёво, после которой поворачиваете налево и едете минут пять. Потом поворот направо на запасную полосу — и вы на месте. Соревнуются и днем и ночью.

Однажды ночью сотрудники ГИБДД расставили вдоль трассы знаки ограничения скорости и уселись в засаде с радаром. Очередные соревнования пришлось проводить, соблюдая скоростной режим. Напоминаем, что знак ограничения скорости предписывает двигаться со скоростью, не превышающей указанную на нем. Действие знака начинается в месте установки и прерывается следующим знаком. С начала трассы до первого знака действует обычное ограничение 90 км/час.

За какое минимально возможное время проедет трассу ваша машина, если максимальное ускорение, развиваемое двигателем a_1 м/сек², а максимальное замедление торможения a_2 м/сек²? В начале трассы ваш автомобиль неподвижен.

Входные данные

В первой строке файла записано вещественное число S — длина трассы ($0 < S \leq 10000$ м). Вторая строке входного файла содержит два вещественных числа a_1 и a_2 ($0 < a_1, a_2 \leq 10$ м/сек²). В третьей строке находится целое число N — количество установленных знаков ($0 \leq N \leq 100$). В последующих N строках файла даны через пробел пары вещественных чисел S_i, V_i — расстояние от начала трассы, на котором установлен i -ый знак ($1 \leq i \leq N$) и ограничение скорости в км/час, указанное на нём ($0 \leq S_i < S, 0 < V_i \leq 500$), соответственно. Знаки записаны по порядку, по мере удаления от старта ($S_i < S_{i+1}$ при $1 \leq i < N$).

Выходные данные

Выходной файл должен содержать одно вещественное число с точностью до двух десятичных знаков после запятой — минимальное время в секундах.

Примеры

Входной файл: <i>input.txt</i>	Выходной файл: <i>output.txt</i>
1000 5 10 0	42.50
1000 5 10 1 100 45	78.81

Problem C

S-Nim

Arthur and his sister Caroll have been playing a game called Nim for some time now. Nim is played as follows:

- The starting position has a number of heaps, all containing some, not necessarily equal, number of beads.
- The players take turns choosing a heap and removing a positive number of beads from it.
- The first player not able to make a move, loses.

Arthur and Caroll really enjoyed playing this simple game until they recently learned an easy way to always be able to find the best move:

- Xor the number of beads in the heaps in the current position (i.e. if we have 2, 4 and 7 the *xor-sum* will be 1 as $2 \text{ xor } 4 \text{ xor } 7 = 1$).
- If the xor-sum is 0, too bad, you will lose.
- Otherwise, move such that the xor-sum becomes 0. This is always possible.

It is quite easy to convince oneself that this works. Consider these facts:

- The player that takes the last bead wins.
- After the winning player's last move the xor-sum will be 0.
- The xor-sum will change after every move.

Which means that if you make sure that the xor-sum always is 0 when you have made your move, your opponent will never be able to win, and, thus, you will win.

Understandibly it is no fun to play a game when both players know how to play perfectly (ignorance is bliss). Fortunately, Arthur and Caroll soon came up with a similar game, *S-Nim*, that seemed to solve this problem. Each player is now only allowed to remove a number of beads in some predefined set S , e.g. if we have $S = \{2, 5\}$ each player is only

allowed to remove 2 or 5 beads. Now it is not always possible to make the xor-sum 0 and, thus, the strategy above is useless. Or is it?

Your job is to write a program that determines if a position of S -Nim is a losing or a winning position. A position is a winning position if there is at least one move to a losing position. A position is a losing position if there are no moves to a losing position. This means, as expected, that a position with no legal moves is a losing position.

Input specifications

Input consists of a number of test cases.

For each test case: The first line contains a number k ($0 < k \leq 100$) describing the size of S , followed by k numbers s_i ($0 < s_i \leq 10000$) describing S . The second line contains a number m ($0 < m \leq 100$) describing the number of positions to evaluate. The next m lines each contain a number l ($0 < l \leq 100$) describing the number of heaps and l numbers h_i ($0 \leq h_i \leq 10000$) describing the number of beads in the heaps.

The last test case is followed by a 0 on a line of its own.

Output specifications

For each position:

- If the described position is a winning position print a 'W'.
- If the described position is a losing position print an 'L'.

Print a newline after each test case.

Sample input

```
2 2 5
3
2 5 12
3 2 4 7
4 2 3 7 12
5 1 2 3 4 5
3
2 5 12
3 2 4 7
4 2 3 7 12
0
```

Output for sample input

```
LWW
WWL
```

Задача D. Прогрессии

Жители страны Прогрессляндии слишком буквально поняли лозунг своего великого правителя "Больше прогрессий — хороших и разных" и решили сосчитать, сколько всего прогрессий они могут придумать. К нашему великому счастью, они знают только целочисленные строго возрастающие арифметические прогрессии в диапазоне от 0 до N , причем прогрессия обязательно должна начинаться со священного числа 0 и иметь хотя бы два элемента.

К сожалению, они недостаточно прогрессивны, чтобы решить эту проблему. Помогите им.

Входные данные

В первой строке входного файла записано одно число N ($0 \leq N \leq 10^{12}$).

Выходные данные

В выходной файл нужно вывести одно число — количество различных целочисленных строго возрастающих конечных арифметических прогрессий, начинающихся с нуля и лежащих в диапазоне от 0 до N , включительно. В прогрессии должно быть не менее двух различных целых чисел. При этом прогрессии, содержащие разное число членов считаются различными.

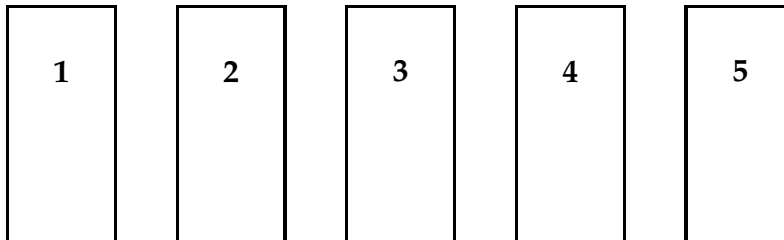
Пример

Входной файл: <i>input.txt</i>	Выходной файл: <i>output.txt</i>
3	5

Problem E

Who owns the Amiga?

In a corridor in a student dormitory, there are five rooms numbered **1, 2, 3, 4** and **5**; room number **1** is the left-most room. The rooms have doors in different colours: **blue, green, red, white** and **yellow**, but not necessarily in that order.



In these rooms live five students **Anna, Bernhard, Chris, David** and **Ellen** of five different nationalities **Danish, Finnish, Icelandic, Norwegian** and **Swedish**. (Both the names and the nationalities are given in alphabetical order, so it does *not* follow automatically that Anna is Danish.)

These students have one computer each, and these computers are of different kinds: **Amiga, Atari, Linux, Mac** and **Windows** (given here in alphabetical order). They each have their own favourite programming language: **C, C++, Java, Pascal** and **Perl** (also listed in alphabetical order).

You want to find out who owns the Amiga computer based on some facts about the students.

Input specifications

The input consists of several scenarios. The first input line contains a number 1–1000 indicating how many scenarios there are.

Each scenario starts with a line with a number 1–2000 telling how many fact lines there are for that scenario. Then follow the fact lines which each contains three words separated by one or more spaces:

- The first and third word is one of these names:

1 2 3 4 5
blue green red white yellow
anna bernhard chris david ellen

danish finnish icelandic norwegian swedish
amiga atari linux mac windows
c c++ java pascal perl

(Note that no uppercase letters are used.)

- The second word specifies a relationship; it is one of

same-as left-of right-of next-to

same-as tells that the first and third fact words apply to the same room; for instance

blue same-as bernhard

tells that Bernhard lives in the room with a blue door.

left-of tells that the first fact word applies to the room immediately to the left of the one to which the third fact word applies. For example,

chris left-of perl

means that Chris lives in the room immediately to the left of the Perl programmer.

right-of tells that the first fact word applies to the room immediately to the right of the one to which the third fact word applies.

next-to tells that the two fact words apply to rooms next to each other. For example,

swedish next-to linux

means that the Swedish student lives in the next room (either to the left or the right) of the owner of the Linux computer.

You may assume that there are no inconsistencies in the input data. In other words, there will in every scenario be at least one person who may own the Amiga without violating the constraints.

Output specifications

For each scenario, you should print a line starting with

scenario #*n*:

where *n* is the scenario number. If you can determine who (i.e., Anna, Bernhard, Chris, David or Ellen) owns the Amiga, you continue the line with

xxxx owns the amiga.

or, if you cannot name the Amiga owner, you print

cannot identify the amiga owner.

Sample input

```
2
8
red same-as 1
danish same-as 1
perl same-as 5
atari same-as 2
linux same-as 3
mac same-as 4
windows same-as 5
anna same-as 1
8
chris left-of amiga
chris left-of 4
c same-as 1
danish same-as 1
red same-as 1
linux same-as red
windows next-to linux
mac left-of swedish
```

Output for sample input

```
scenario #1: anna owns the amiga.
scenario #2: cannot identify the amiga owner.
```

Задача F. Памятник

В одном городе к юбилею решили поставить на главной площади памятник. Его привезли поздно вечером, установили на специально подготовленную площадку и накрыли тентом. Во время церемонии открытия под звуки духового оркестра тент сняли. И тут все присутствующие увидели, что памятник стоит вверх ногами. Его нужно срочно поставить правильно на то же самое место.

Памятник имеет вид параллелепипеда. Он очень тяжелый, его нельзя передвигать, можно только переворачивать через ребра. На всех боковых сторонах памятника находится одно и то же изображение.

Главная площадь города имеет форму прямоугольника со сторонами N и M ($1 \leq M \leq N \leq 100$). С одним из углов площади совмещено начало координат. Ось OX проходит по большей стороне, а OY — по меньшей. Площадка, на которую нужно поставить памятник, точно совпадает по размерам с основанием памятника. Это прямоугольник со сторонами A и B ($1 \leq A \leq N$, $1 \leq B \leq M$, $B \leq A$), которые параллельны сторонам площади (большая сторона площадки параллельна большей стороне площади). Расположение площадки задается координатами ее угла, ближайшего к началу координат.

Определим длину элементарного переворота памятника как длину ребра, перпендикулярного ребрам основания, на котором памятник находился до этого переворота. Длина траектории перемещения памятника определяется как сумма длин элементарных переворотов. Требуется найти наименьшую длину траектории, по которой можно переместить памятник и поставить правильно.

Входные данные

В первой строке входного файла записаны два числа N и M — размеры площади.

В следующей строке заданы размеры памятника и координаты площадки. Это пять чисел A , B , C , X , Y , где A и B — длины ребер основания памятника, C — его высота ($1 \leq C \leq 100$), X и Y — координаты ($0 \leq X \leq N - A$, $0 \leq Y \leq M - B$). Все числа целые и записаны через пробел.

Выходные данные

В выходной файл необходимо вывести одно целое число — длину минимальной траектории перемещения памятника. Если памятник переместить невозможно, то нужно вывести 0.

Пример

Входной файл: <i>input.txt</i>	Выходной файл: <i>output.txt</i>
10 10 1 1 1 2 2	6

Задача G. Молекулы

Органические вещества отличаются потрясающим разнообразием. Они различаются по химическому составу (по пропорциям составляющих вещество элементов), по химической формуле молекул (по порядку расположения и химическим связям атомов) и по пространственной форме молекул. Проблема восстановления по химическому составу органического вещества его химической формулы и проблема пространственного конфигурирования молекул по их химическим формулам — чрезвычайно актуальные задачи биоинформатики. К сожалению, они отличаются высокой вычислительной сложностью.

Рассмотрим проблему химической формулы и пространственной конфигурации органических молекул при следующих упрощающих предположениях:

1. вещество состоит только из атомов водорода H (валентность 1), кислорода O (валентность 2), азота N (валентность 3) и углерода C (валентность 4);
2. молекула или группа молекул вещества представляет собой плоскую прямоугольную решётку из m строк и n столбцов, в узлах которой могут находиться или атомы перечисленных элементов, или «дырки» — пустоты, не занятые никаким атомом;
3. каждый атом в решётке может быть связан с атомами, своими соседями по решётке, но с каждым соседом он связан не более чем одной связью, а общее число его связей совпадает с его валентностью.

Ваша задача — написать программу, которая для данной прямоугольной решётки, в узлах которой могут находиться элементы H, O, N, C или дырки «.», определит, может ли она описывать структуру одной или нескольких молекул.

Входные данные

Входной файл для программы описывает одну решетку. В его первую строку записаны через пробел два целых числа M и N ($1 \leq M, N \leq 50$), задающих число строк и число столбцов этой решётки. Следующие M строк содержат ровно по N символов «H», «O», «N», «C», «.», представляющих элементы или «дырки» в порядке слева направо в соответствующей строке решётки.

Выходные данные

В выходной файл программы нужно вывести слово **Valid**, если данная решетка может описывать структуру одной или нескольких молекул, или **Invalid**, в противном случае.

Примеры

Входной файл: <i>input.txt</i>	Выходной файл: <i>output.txt</i>
3 4 HOH. NCOH OO..	Valid
3 4 HOH. NCOH OONH	Invalid
2 3 HOH HOH	Valid
4 10 OOOOOOOOOO OOOOOOOOOO OOOONOOOOO OOOOOOOOOO	Invalid

Problem H

Tourist

A lazy tourist wants to visit as many interesting locations in a city as possible without going one step further than necessary. Starting from his hotel, located in the north-west corner of city, he intends to take a walk to the south-east corner of the city and then walk back. When walking to the south-east corner, he will only walk east or south, and when walking back to the north-west corner, he will only walk north or west. After studying the city map he realizes that the task is not so simple because some areas are blocked. Therefore he has kindly asked you to write a program to solve his problem.

Given the city map (a 2D grid) where the interesting locations and blocked areas are marked, determine the maximum number of interesting locations he can visit. Locations visited twice are only counted once.

Input specifications

The first line in the input contains the number of test cases (at most 20). Then follow the cases. Each case starts with a line containing two integers, W and H ($2 \leq W, H \leq 100$), the width and the height of the city map. Then follow H lines, each containing a string with W characters with the following meaning:

- '.' Walkable area
- '*' Interesting location (also walkable area)
- '#' Blocked area

You may assume that the upper-left corner (start and end point) and lower-right corner (turning point) are walkable, and that a walkable path of length $H + W - 2$ exists between them.

Output specifications

For each test case, output a line containing a single integer: the maximum number of interesting locations the lazy tourist can visit.

Sample input

```
2
9 7
*.....
.....**#.
..**...#*
..#####*#
.*.#*.*#
...#**...
*.....
5 5
.*.*.
*###.
*.*.*
.###*
.*.*.
```

Output for sample input

```
7
8
```

Задача I. Перекачка данных

Имеется два компьютера, с одного из которых нужно передать информацию размером V мегабайт на другой. В качестве носителя информации имеется два USB-диска с размерами V_1 и V_2 мегабайт, скоростями чтения R_1 и R_2 и записи W_1 , W_2 мегабайт в секунду, соответственно.

Конструктивные особенности дисков таковы, что чтение или запись происходят целое число секунд, в том числе, если на источнике информации или на приемнике места меньше, чем может быть записано за секунду, копирование будет все равно занимать секунду. На каждом из компьютеров имеется только один USB-разъем. Поэтому в один и тот же момент времени можно либо переписывать информацию с компьютера на один носитель, либо читать данные с диска и переписывать их на компьютер. Требуется определить, за какое минимальное время можно перенести данные с одного компьютера на другой, если переключение дисков происходит за нулевое время.

Входные данные

В первой строке входного файла записано одно целое число V — объем информации, который нужно перекачать.

В следующих двух строках записано по три числа — параметры каждого из дисков. Сначала V_1 , R_1 , W_1 , а затем, соответственно, V_2 , R_2 и W_2 .

Все числа даны в диапазоне от 1 до 300, включительно.

Выходные данные

В выходной файл нужно вывести одно число — минимальное время перекачки заданного объема данных.

Пример

Входной файл: <i>input.txt</i>	Выходной файл: <i>output.txt</i>
80 10 10 10 20 20 20	6

Задача J. Города – конкуренты

В стране Флатландии имеется N городов, каждый город выпускает один определенный товар из M товаров. Города, выпускающие один и тот же товар, являются городами-конкурентами. Города Флатландии хотят объединиться в торговую сеть. Как показали исследования, для того, чтобы какие-то города могли объединиться в торговую сеть, необходимо и достаточно, чтобы из любого города A можно было добраться по дорогам сети в любой другой город B . Причем, в такой сети не должно быть дорог между городами-конкурентами, т.е. города-конкуренты не являются соседями в этой сети.

Ваша задача — по заданным городам Флатландии построить сеть дорог таким образом, чтобы длина сети была минимальной. Длина сети есть сумма длин построенных дорог. Дороги между городами строятся таким образом, чтобы попасть на них можно было только из городов, которые они соединяют. Для того чтобы избежать пересечений, дороги можно располагать на разных уровнях.

Входные данные

В первой строке входного файла записаны через пробел два целых числа N и M ($1 \leq N \leq 200$, $1 \leq M \leq 200$). В последующих N строках описаны города Флатландии, описание каждого города занимает одну строку. В соответствующей строке записаны через пробел три целых числа X_i , Y_i , Z_i , где X_i и Y_i — координаты i -го города, а Z_i — номер продукции, которую выпускает этот город ($-10000 \leq X_i, Y_i \leq 10000$, $1 \leq Z_i \leq M$, $1 \leq i \leq N$).

Выходные данные

В первую строку выходного файла необходимо вывести одно вещественное число с точностью до 0.001 — минимальную длину сети дорог.

Если города объединить в торговую сеть невозможно, то вывести -1 .

Примеры

Входной файл: <i>input.txt</i>	Выходной файл: <i>output.txt</i>
8 3 3 3 1 3 10 1 5 6 2 10 6 3 10 10 2 15 3 1 15 6 3 15 10 2	29.909
4 1 0 0 1 0 5 1 5 0 1 5 5 1	-1

Problem K

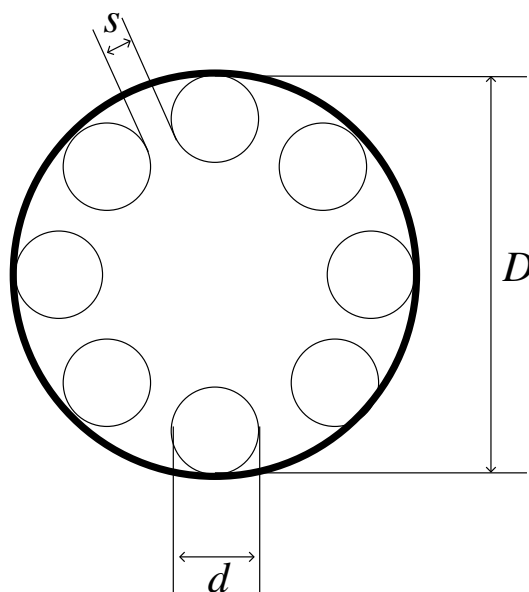
Ball bearings

The Swedish company SKF makes ball bearings. As explained by Britannica Online, a ball bearing is

“one of the two types of rolling, or anti friction, bearings (the other is the roller bearing).

Its function is to connect two machine members that move relative to one another so that the frictional resistance to motion is minimal. In many applications, one of the members is a rotating shaft and the other a fixed housing. Each ball bearing has three main parts: two grooved, ring like races and a number of balls. The balls fill the space between the two races and roll with negligible friction in the grooves. The balls may be loosely restrained and separated by means of a retainer or cage.”

Presumably, the more balls you have inside the outer ring, the smoother the ride will be, but how many can you fit within the outer ring? You will be given the inner diameter of the outer ring, the diameter of the balls, and the minimum distance between neighboring balls. Your task is to compute the maximum number of balls that will fit on the inside of the outer ring (all balls must touch the outer ring).



Input specifications

The first line of input contains a positive integer n that indicates the number of test cases. Then follow n lines, each describing a test case. Each test case consists of three positive floating point numbers, D , d , s , where D is the inner diameter of the outer ring, d is the diameter of a ball, and s is the minimum distance between balls. All parameters are in the range $[0.0001, 500.0]$.

Output specifications

For each test case output a single integer m on a line by itself, where m is the maximum number of balls that can fit in the ball bearing, given the above constraints. There will always be room for at least three balls.

Sample input

```
2
20 1 0.1
100.0 13.0 0.2
```

Output for sample input

```
54
20
```

Задача L. Светофоры

Все знают, что на ночных улицах опасно. Но в данном случае речь идет не о преступниках и маньяках. Когда наступает ночь, и силы зла властвуют безраздельно, там действуют те, с кем не встретишься днем — темные маги, вампиры и прочая нечисть. Их сила велика, и с ними нельзя справиться обычным оружием. Но по следу "ночных охотников" идут те, кто веками сражается с порождениями сумрака и побеждает их, неукоснительно соблюдая при этом Договор, заключенный тысячелетия тому назад между Светлыми и Темными... Имя им — Ночной Дозор. Их предназначение — сохранение равновесия между Добром и Злом, нарушение которого вызывает разрушения, войны, революции, вселенские катастрофы. Каждый плохой человеческий поступок — измена, предательство, убийство, равно, как и хороший, ложится на чашу весов, перевешивая их то в одну, то в другую сторону. Именно поэтому и силы Света, и силы Тьмы вынуждены существовать в двух мирах: реальном и потустороннем, пытаясь либо подтолкнуть человека к греху, либо отвести от него...

В городе Н-ске, на одном из перекрестков силы Зла нарушили вековой договор. С другого перекрестка машина "Горсвет" направляется к злополучному месту. За какое время доедут силы Света, если у них есть карта города со схемой работы светофоров, и они поедут по оптимальному маршруту с максимально разрешенной скоростью 60 км/час?

Карта города представляет собой прямоугольник размером $N \times M$ км ($1 \leq N, M \leq 25$). Движение в Н-ске организовано по $M+1$ улицам, идущим параллельно с севера на юг, и $N+1$ авеню, идущим параллельно с запада на восток. Расстояние между двумя соседними улицами (авеню) составляет 250 метров.

По традиции, улицы нумеруются (с запада на восток) подряд идущими натуральными числами, начиная с единицы. Авеню обозначаются (с севера на юг) подряд идущими буквами латинского алфавита, начиная с А. Таким образом, каждый перекресток города можно однозначно обозначить парой из буквы и числа, например С17.

На каждом перекрестке может находиться светофор. Для i -го светофора известно число K_i (целое $1 \leq K_i \leq 180$), определяющее интервалы цикла смены его состояний: для потоков, едущих с запада и с востока, сначала $(K_i - 1)$ секунд горит зеленый свет, затем 1 секунду горит желтый, затем K_i секунд горит красный; а для потоков, едущих с севера и с юга, сначала K_i секунд горит красный, затем $(K_i - 1)$ секунд — зеленый, затем 1 секунду — желтый. Через перекресток разрешается проезжать напрямую или поворачивать на зеленый и желтый свет. В момент поступления сигнала о нарушении договора, каждый светофор находился в D_i секундах от начала цикла (D_i — целое, $0 \leq D_i < 2 * K_i$).

Входные данные

В первой строке входного файла через пробел записаны числа N и M . Во второй строке через пробел записаны обозначения начального и конечного перекрестка. В третьей строке записано количество светофоров K , где $0 \leq K \leq (N+1) * (M+1)$. В последующих K строках через пробел записаны обозначение перекрестка и числа K_i и D_i .

Входные данные

Выходной файл должен содержать одно целое число — минимальное время проезда в секундах.

Примеры

Входной файл: <i>input.txt</i>	Выходной файл: <i>output.txt</i>
5 1 F2 A2 3 A2 60 0 C1 100 10 C2 180 180	75
5 5 A1 F6 7 A3 120 0 B3 180 0 C3 180 60 D3 100 60 E3 100 0 F1 5 0 F2 3 0	150