Problem A. Практичные числа

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Практичное число — это натуральное число n, такое что все меньшие натуральные числа могут быть представлены в виде суммы различных делителей числа n. Например, 12 является практичным числом, поскольку все числа от 1 до 11 можно представить в виде суммы делителей 1, 2, 3, 4 и 6 этого числа (сами делители равны самим себе): $5 = 3 + 2; 7 = 6 + 1; 8 = 6+; 9 = 6 + 3; 10 = 6 + 3 + 1; 11 = 6 + 3 + 2$.

Вам необходимо написать программу, которая считает количество практичных чисел, меньших N.

Input

В первой и единственной строке находится число N ($1 \le N \le 200$).

Output

Требуется вывести количество практичных чисел, меньших N.

Examples

standard input	standard output
3	2
15	6

Problem B. Медиафасады

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Одна известная компания решила разместить на фасаде своего здания на улице Федора Достоевского рекламные баннеры. Фасад представляет собой прямоугольник размера MxN квадратных плиток. Рекламный баннер представляет собой прямоугольник HxW размера плитки и размещается на фасаде ровно по границам плиток.

Компания провела слепой аукцион, в котором всем желающим предлагалось указать положение их баннера на фасаде в виде координат левого нижнего угла и размера баннера в единицах длины плитки и цену, которую компания-рекламодатель готова заплатить, если баннер будет виден полностью. Если баннер будет виден не полностью, его цена уменьшается пропорционально видимой площади.

Ваша задача разместить баннеры на фасаде так, чтобы максимизировать рекламную выручку. Баннеры могут перекрывать друг друга, то есть если баннер компании «Добро» закроет собой часть баннера компании «Зло», то компания «Зло» заплатит за рекламу меньше пропорционально закрытой площади. Передвигать баннеры на другие позиции нельзя.

Input

Два целых числа M и N ($1 \le M, N \le 100$ - размер фасада, выраженный в единицах длины плитки. M - размер по вертикали (число рядов), N - размер по горизонтали (число столбцов). Целое число K ($0 \le K \le 1000$) - количество компаний, подавших заявки на участие в аукционе Далее для каждой компании подаются пять чисел: R, C, H, W, P, где R - номер ряда левого нижнего угла баннера, C - номер столбца левого нижнего угла баннера. Левый нижний угол фасада здания имеет координаты $(0, 0)$. H - высота баннера в единицах длины плитки, W - ширина баннера в единицах длины плитки. R, C, H, W - целые неотрицательные числа, допустимые для заданного размера фасада. P - целое число ($1 \le P \le 2^{31} - 1$), указывающее цену, которую компания готова заплатить за баннер.

Output

На стандартный поток вывода напечатайте максимальную возможную рекламную выручку с относительной точностью 1e-7.

Examples

standard input	standard output
10 10 1 0 0 10 10 1	1.0000000000
10 10 10 0 0 1 1 1 1 1 1 1 2 2 2 1 1 3 3 3 1 1 4 4 4 1 1 5 5 5 1 1 6 6 6 1 1 7 7 7 1 1 8 8 8 1 1 9 9 1 1 1 10	55.0000000000
1000 1000 2 0 0 500 1000 1000001 0 0 1000 500 1000000	1500001.0000000000

Problem C. Японский кроссворд

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Напишите программу, которая разгадывает японские кроссворды. Описание кроссвордов и некоторых возможных методик их отгадывания дано на веб-странице олимпиады.

Программе на вход подаются следующие входные данные: M, N - два целых числа ($1 \leq N, M < 100$), задающие количество строк и столбцов в кроссворде.

Далее следует описание N закодированных строк. Первое число в описании закодированной строки - количество серий закрашенных клеток в этой строке, а далее перечисляются число закрашенных клеток в каждой из серий. Далее следует описание M закодированных столбцов. Каждый столбец описывается аналогично строке, то есть сначала идет количество серий в данном столбце, а затем перечисляются количество закрашенных клеток в каждой серии.

Получив входные данные ваша программа должна разгадать кроссворд. Для каждой клетки кроссворда программа может либо указать, является ли клетка закрашенной или пустой, либо запросить подсказку.

Чтобы указать, что клетка является закрашенной ваша программа должна вывести три числа: R C 1, где R - номер строки, строки нумеруются от нуля и до M - 1; C - номер столбца, столбцы нумеруются от 0 до N - 1.

Чтобы указать, что клетка является свободной, ваша программа должна вывести три числа R C 0

Чтобы запросить подсказку, программа должна вывести R C -1

В ответ ваша программа получит либо число 0, если клетка свободна, либо число 1, если клетка закрашена.

В выводе вашей программы каждая из MxN клеток должна встретиться ровно один раз. Либо для нее указывается свободна она или закрашена, либо для нее запрашивается подсказка. Если ваша программа выводит координаты R, C, которые уже были ранее, ваша программа будет завершена с ошибкой выполнения (runtime error). Если ваша программа вывела, что клетка закрашена, а на самом деле она свободна, или наоборот, ваша программа будет завершена с ошибкой выполнения. Балл, которым будет оценена ваша программа, будет зависеть как от числа прошедших тестов, то есть тестов, на которых ваша программа дала правильный ответ, так и от числа подсказок, которые были запрошены программой (чем меньше подсказок, тем выше балл). Во время олимпиады будет проверяться только число прошедших тестов.

Время разгадывания одного кроссворда не должно превышать одну секунду.

Problem D. Трактора-1

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Однажды Петр разработал конструкцию красного беспилотного трактора, а затем выпустил для надежности N ($N \leq 100$) тракторов. Он хочет, чтобы каждый трактор попал в страну лугов, гор и коров. Петр живет на клетчатой плоскости,трактор представляет из себя одну клетку. К сожалению, страна Петра разделена со страной лугов, гор и коров бесконечным ферритовым забором во всех клетках с координатами (0, Y) кроме клетки (0,0). Страна Петра расположена справа от забора, страна лугов, гор и коров слева.

Помогите Петру написать программу для беспилотного трактора, такую, чтобы он из некоторого начального положения справа от забора попал в область слева от забора.

Управление трактором осуществляется следующим образом. Изначально в трактор загружается программа (во все трактора загружается одна и та же программа), которая в каждый момент времени решает, куда должен переместиться трактор – влево или вправо (ось X), вверх или вниз (ось Y) или остаться на месте (эти команды кодируются числами от 1 до 5, соответственно). После этого все трактора одномоментно совершают это перемещение. Если несколько тракторов пытаются попасть в одну и те же клетку , то вместо перемещения эти трактора остаются на месте. Так же трактор не может переместится если пытается переместиться в клетку, в которой остается другой трактор или в которой находится стена.

У каждого трактора есть 8 байт памяти, которую он может изменять каждый момент времени. У каждого трактора есть GPS, который позволяет определять его текущую координату.

Изначально координаты тракторов не превосходят по модулю 100.

Input

Каждый момент времени на вход программе подается пара целых чисел - текущая координата трактора. Следующие 8 чисел от 0 до 255, разделенные проблемами задают состояние памяти текущего трактора.

Output

Требуется вывести текущее действие трактора - число от 1 до 5, затем 8 байт - новое состояние памяти.

Note

Для каждого хода для каждого трактора запускается решение с данными, описанными во входном формате. Программа делает один ход и не должна предполагать, что можно сохранить дополнительные данные, кроме как в 8 байтах памяти трактора.

Пример ввода для одного хода:

```
1 0
0 0 0 0 0 0 0 0
```

Пример вывода:

```
1
1 0 0 0 0 0 0 0
```

Если за 5000 итераций трактора не переместились в левую полуплоскость, то решение на этом тесте считается неверным.

Задача тестируется оффлайн, баллы за тесты зависят от числа ходов для целевой конфигурации. Для тестов из примера выводится полный протокол проверки.

Problem E. Трактора-2

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

В предыдущей задаче возникает ситуация, когда несколько тракторов пытаются переместиться в одну и ту же клетку или трактор пытается переместиться в клетку, которая останется занятой другим трактором или стеной.

Вам необходимо написать программу, которая определяет итоговое расположение тракторов при выполнении одного хода.

Input

В первой строке находится число тракторов N ($1 \le N \le 100$). В следующих N строках задаются координаты тракторов x, y ($-1000 \le x, y \le 1000$) и число от 1 до 5, задающее действие трактора. Гарантируется, что трактора не находятся на стене.

Output

Требуется вывести N координат тракторов после выполнения хода.

Examples

standard input	standard output
3 1 2 3 3 3 2 4 4 4	1 3 3 3 4 4
2 1 1 2 2 1 1	2 1 1 1
2 1 1 2 3 1 1	1 1 3 1