Problem A. Array access

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 mebibytes

Во время курса по написанию компиляторов студенты получили следующее задание: реализовать парсер для выражений языка Паскали, содержащих обращения к элементам массива.

Студент Вася реализовал только небольшое подмножество языка: целую константу 0 и один двумерный массив a. Иначе говоря, написанный им «язык» имел следующий словарь:

```
expr ::= 0 | a[expr,expr]
```

Преподаватель вынужден был дать Васе дополнительное задание: пусть массив a задаётся следующим образом:

```
a: array [0..N - 1, 0..N - 1] of 0..N
```

По заданным N и значениям элементов массива построить состоящее из наименьшего количества байт выражение на реализованном Васей подмножестве Паскаля, значение которого равно N.

Input

Первая строка входного файла содержит целое число N. В последующих N строках задано по N целых чисел — значения массива a_{ij}, перечисленного сначала по строкам, а затем слева направо ($1 \leqslant N \leqslant 22$, $0 \leqslant a_{ij} \leqslant N$).

Output

Выведите одну строку — требуемое выражение. Выражение должно соответствовать определённой в условии задачи грамматике. Если построить N невозможно, выведите "IMPOSSIBLE". Если выражений минимальной длины несколько, выведите любое из них.

Examples

input.txt	output.txt
2 1 2 0 0	a[0,a[0,0]]
3 2 0 0 0 3 0 0 0 0	IMPOSSIBLE

Problem B. Ball of fir

Input file:	input.txt.
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	256 mebibytes

Математики из Института Новогодних Исследований построили математическую модель новогодней ёлки, в которой ёлка представлена в виде шара радиусом 1 метр. Однако моделирование гирлянды является более сложной задачей...

Гирлянда моделируется как кривая, расположенная на поверхности сферы. Кривая начинается в самой верхней точке сферы и заканчивается в её самой нижней точке. При этом кривая делает ровно N оборотов вокруг сферы.

Более формально, рассмотрим сферические координаты φ (долгота) и ϑ (широта). Начальное значение долготы равно 0, при этом долгота делает ровно N оборотов и её финальное значение также равно 0.

Начальное значение широты равно 90 градусов, а финальное значение равно -90 градусов. Изменение широты идёт пропорционально изменению долготы.

Для заданного N требуется вычислить длину кривой, которая моделирует гирлянду.

Input

Входной файл содержит одно целое число N ($1 \leqslant N \leqslant 100$).

Output

Выведите одно вещественное число — длину кривой в метрах не менее, чем с тремя точными знаками после десятичной точки.

Examples

input.txt.	output.txt
1	5.270
3	12.612

Problem C. Compression Research

Input file:	input.txt
Output file:	output.txt
Time limit:	5 seconds
Memory limit:	256 mebibytes

Многие алгоритмы сжатия основаны на поиске часто повторяющихся подстрок во входных данных. Так как часто искать повторения по всему тексту оказывается непрактично, на каждом шаге работы алгоритма рассматривается *окно сжатия* ограниченной длины.

Юный программист Вася хочет разработать свой алгоритм сжатия; однако в работе над алгоритмом он встретил следующую задачу:

Рассмотрим входную строку, состоящую из N битов. Пусть окном сжатия является любая подстрока, состоящая из M последовательных битов. Для каждого окна сжатия требуется определить наиболее часто встречающуюся подстроку длины L ($L \leqslant M$) и найти её количество вхождений.

Input

Первая строка входного файла содержит целые числа L и M ($1 \leqslant M \leqslant N \leqslant 2 \cdot 10^5$, $1 \leqslant L \leqslant 100$). Вторая строка имеет длину N и составлена из символов '0' или '1'.

Output

Выведите $N - M + 1$ целых чисел — частоты наиболее часто встречающейся подстроки заданной длины для каждого окна сжатия.

Examples

input.txt	output.txt
2 10 0101010101	5
1 3 1110000	3 2 2 3 3

Note

В первом примере из условия задачи длина окна сжатия равна длине строки, так что рассматривается всего одно окно сжатия. Наиболее часто встречающейся подстрокой длины 2 является подстрока "01", которая встречается 5 раз.

Problem D. Door and wallpaper

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

Во время ремонта юному строителю Васе поручили оклеить стену обоями. Стена представляет собой прямоугольник шириной W метров и высотой H метров с прямоугольной дверью w метров в ширину и h метров в высоту, расположенной с левой стороны стены.

Обои упакованы в рулоны. Полоса обоев в каждом рулоне имеет ширину 1 метр и длину D метров. Вася может разрезать полосу на полосы меньшей длины, сохраняя при этом ширину.

При этом Вася должен соблюдать следующие правила:

- Оклеить обоями всю стену, кроме дверного проёма, при этом соседние полосы обоев не могут пересекаться по фигуре ненулевой площади.
- Обои наклеиваются вертикально; над дверью наклеиваются полосы длины $H - h$, в остальных местах — полосы длины H. Использовать «составные» полосы запрещается.
- Все задействованные рулоны обоев разрезаются на один и тот же набор полос.

Требуется вычислить наименьшее количество рулонов, которые потребуются Васе, чтобы выполнить поручение.

Input

Входной файл содержит пять целых чисел W, H, w, h, D ($1 \leqslant W, H, w, h \leqslant 10^9$; $1 \leqslant D \leqslant 2 \times 10^9$; $h \leqslant H$; $w \leqslant W$). Гарантируется, что входные данные таковы, что задача всегда имеет решение.

Output

Выведите одно целое число — минимальное количество рулонов обоев, которое потребуется Васе.

Examples

input.txt	output.txt
5 3 3 1 6	3

Note

В примере из условия задачи Вася разрезает три рулона на полосы как $3 + 2 + 1$. Если бы можно было резать рулоны на различные наборы, то хватило бы двух: один режется как $3 + 3$, второй — как $2 + 2 + 2$.

Problem E. Elite number

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

Назовём целое число *элитным*, если оно делится на каждую цифру своего десятичного представления.

По заданному целому числу x требуется найти наименьшее элитное число, большее или равное x.

Input

Входной файл содержит одно целое число x ($1 \leqslant x \leqslant 10^{10}$.

Output

Выведите одно целое число — наименьшее элитное число, большее или равное x.

Examples

input.txt	output.txt
10	11
7777777777	7777777777
579916	593595

Problem F. Far Eastern Federal Clock

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

В одной очень большой стране было много провинций. Однажды правительство заметило, что жители наиболее удалённой провинции чем-то недовольны, и решило выяснить, в чём дело.

Социологические исследования обнаружили интересный феномен: проблема была в том, что каждый житель провинции должен был покупать часы для того, чтобы отслеживать время. После этого правительство распорядилось построить огромную башню с гиганскими механическими часами, чтобы жители могли смотреть на часы на башне и экономить потраченные на покупку часов деньги.

В конце концов башня была сооружена и настал момент открытия. После торжественного открытия обнаружилась небольшая проблема — часы показывали время неправильно.

Отремонтировать часы уже было невозможно — суммы, выделенные на строительство сооружения, уже давно освоены. Тогда правительство распорядилось назначить Старшего Часовщика, который бы подводил часы, двигая их стрелки руками.

При этом:

- Ровно в полночь часы должны быть подведены (и должны показывать полночь).
- В течение остальных суток часы должны подводиться с периодичностью один раз в p минут.
- Различие между временем, показанным на часах, и реальным временем не должно превосходить m минут. При этом выбирается наименьшая возможная разница, то есть если в 00:05 часы показывают 23:50, то разница равна 15 минутам.

Так как стрелки часов очень тяжёлые, Старший Часовщик попросил Вас минимизировать усилие, которое он должен затратить на движение стрелок часов в течение суток.

У часов две стрелки — минутная и часовая. Каждую минуту минутная стрелка прыгает на t градусов по часовой стрелке, а часовая — на $t/12$ (для правильно идущих часов $t = 6$).

Подстройка производится сразу же после прыжка стрелки, и усилие, требуемое для подстройки, равно сумме углов в градусах между текущим и правильным положением для каждой стрелки.

Input

Входной файл содержит вещественное число t, заданное не более, чем с тремя знаками после десятичной точки и целое число m ($0 \leqslant t \leqslant 10^4$, $1 \leqslant m \leqslant 10^4$).

Output

Выведите два числа. Первое из них — минимальное суммарное усилие, которое будет затрачено Старшим Часовщиком на подведение стрелок за сутки, второе — соответствующая частота подведения стрелок p ($1 \leqslant p \leqslant 1440$). Если решений с минимальным суммарным усилием несколько, выведите то из них, для которого p максимально.

Examples

input.txt	output.txt
12 1	9360.0000 2
18.0 90	1440.0000 30
6 1	0.0000 1440

Note

В первом примере часы идут вдвое быстрее, чем следует, так что каждую минуту ошибка увеличивается на одну минуту. В этом случае требуется подводить часы каждые две минуты.

Во втором примере часы идут втрое быстрее, чем требуется, однако приемлемая ошибка куда больше. Получается так, что каждые 30 минут позиция минутной стрелки совпадает с правильной, и, если выбрать интервал, равный 30 минутам, то двигать надо только часовую стрелку.

Problem G. Ganking

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

В популярной онлайновой игре “Attack Of The Moderns 3” играют две команды по 5 игроков в каждой. Во время матча игроки перемещаются по карте и пытаются убить игроков другой команды, применяя оружие и заклинания. Убитые игроки восстанавливаются через некоторый интервал времени.

Все попытки атаки на игроков сохраняются в статистике игры. Каждая атака описывается числами t, a, v, k, где t — время в секундах, прошедшее после начала игры, a — номер атакующего игрока, v – номер игрока, который был атакован, $k = 1$, если в результате данной атаки противник был убит, и 0 в противном случае.

Игроки из первой команды пронумерованы последовательными натуральными числами от 1 до 5, игроки из второй команды — последовательными натуральными числами от 6 до 10.

Gank — ситуация, когда один или несколько игроков атакуют и убивают игрока, сокомандники которого находятся где-то ещё и не могут ему помочь.

Более формально, пусть G — множество игроков, которые атаковали противника за последние T секунд до того, как он был убит. *Gank* считается, если в этот период времени:

- убитый противник атаковал только игроков из множества G;
- участники из множества G атаковали только убитого противника и были атакованы только им.

Все участники из множества G, которые атаковали убитого противника, являются участниками gank-а.

По заданному T и по последовательности из N описаний попыток атаки вычислите для каждого игрока количество gank-ов, в которых тот участвовал.

Input

Первая строка входного содержит целые числа N и T. Каждая из N последующих строк содержит четвёрку целых чисел t_i, a_i, v_i, k ($1 \leqslant N \leqslant 10000$, $1 \leqslant t_i \leqslant t_{i+1} \leqslant 10^5$, $1 \leqslant T \leqslant 10^5$. Верно, что или $1 \leqslant a_i \leqslant 5 < v_i \leqslant 10$, или $1 \leqslant v_i \leqslant 5 < a_i \leqslant 10$). Гарантируется, что время между последовательными убийствами одного и того же игрока больше T.

Output

Выведите 10 целых чисел. i-е из этих чисел задаёт количество gank-ов, в которых участвовал i-й игрок.

Examples

input.txt	output.txt
7 4 1 1 6 0 2 2 6 0 5 3 6 1 10 7 1 1 10 7 2 1 20 8 1 1 20 3 7 1	0 1 2 0 0 0 0 1 0 0

Problem H. Hexes in viewport

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

Компания, занимающаяся разработкой компьютерных игр, начала разработку новой игры, которая проходит на поле, представляющем собой шестиугольную сетку. Первая подзадача, которую надо решить — отображение видимой участнику части решётки. При этом для ускорения принято решение использовать псевдографическое представление.

Шестиугольная решётка состоит из «псевдоправильных» шестиугольников с длиной стороны N символов. В каждом шестиугольнике верхняя и нижняя сторона составлены из N символов '_' characters (ASCII 95), правая верхняя и левая нижняя стороны — из N символов '
' (ASCII 92), левая верхняя и правая нижняя — из N символов '/' (ASCII 47). Остальное поле занято символами '.' (ASCII 46).

Предпорагается, что решётка бесконечна, при этом в точке $(0; 0)$ находится самый левый символ верхней стороны шестиугольника. Видимая участнику часть решётки представляет собой прямоугольник.

Напишите программу, которая по заданным координатам верхнего левого угла области видимости x и y и w и h (ширине и высоте области видимости) выводит содержимое области видимости.

Input

Входной файл содержит пять целых чисел N, x, y, w, h ($1 \leqslant N \leqslant 100, 0 \leqslant x, y \leqslant 10^9, 1 \leqslant w, h \leqslant 100$).

Output

Выведите h строк, каждая из которых содержит по w символов — содержимое области видимости.

Example

input.txt	output.txt
3 0 2 33 10	....\...../.....\...../.....\.... .....___/.......___/.......___ ...../...\...../...\...../... ..../.....\...../.....\...../.... ___/.......___/.......___/..... ...\...../...\...../...\..... ..../.....\...../.....\...../.... .....___/.......___/.......___ ...../...\...../...\...../... ..../.....\...../.....\...../....

Problem I. IT over the bridge

Input file:	input.txt
Output file:	output.txt
Time limit:	2 seconds
Memory limit:	64 mebibytes

У некоторой крупной организации есть офисное здание, в котором проведена локальная сеть. Недавно организация получина новый офис, и локальная сеть была расширена на новое здание.

Так как новое здание находится на удалённом острове, то время от времени происходят разные неприятные события: обрывы соединения, отключения питания, потери пакетов и так далее.

Инженеры из IT-отдела для мониторинга сети установили два сервера, по одному в каждом здании. Каждый сервер создаёт log-файл, в котором он сохраняет типы ошибок для каждой обнаруженной им ошибки.

Теоретически оба лога должны быть идентичны. Однако, получилось так, что у каждой системы мониторинга также бывают сбои: некоторые ошибки сети определяются только одним сервером, время от времени серверы сообщают о несуществующих ошибках, даже информация о временах событий не соответствует, так как часы на серверах не синхронизированы.

В результате для хранения логов был разработан формат, в котором каждый лог представляется как L символов, по одному символу для каждой ошибки сети. Чтобы отфильтровать ошибки мониторинга, *реальная история ошибок* определяется как максимальная последовательность ошибок одного типа, которая встречается в обоих логах в одном и том же порядке.

При изучении проблем сети инженерам часто требуется сравнивать не только полные логи, но и отдельные части логов.

По заданным двум строкам a и b, задающим два лога, и последовательности из N запросов, каждый из которых имеет форму a_1, a_2, b_1, b_2, выведите для каждого зпроса длину реальной истории ошибок, которая получается при сравнении участка между символами с номерами a_1 и a_2 включительно в первом логе и символами с номерами от b_1 до b_2 включительно во втором логе.

Input

Первые две строки входного файла содержат две строки a и b, имеющие одинаковую длину L.

Третья строка содержит целое число N. Каждая из последующих N строк содержит по 4 целых числа — значения $a_{i,1}a_{i,2}b_{i,1}b_{i,2}$ ($1 \leqslant L \leqslant 100$, $1 \leqslant N \leqslant 10^6$, $1 \leqslant a_{i,1} \leqslant a_{i,2} \leqslant L$, $1 \leqslant b_{i,1} \leqslant b_{i,2} \leqslant L$, строки a и b состоят из строчных латинских букв).

Output

Выведите N целых чисел — ответы на соответствующие запросы

Examples

input.txt	output.txt
abcd abcd 2 1 2 3 4 1 3 2 4	0 2
aaazaaz zzzazzz 1 4 7 1 7	3

Problem J. Jokers

Input file:	input.txt
Output file:	output.txt
Time limit:	1 second
Memory limit:	256 mebibytes

Однажды Марфа Геннадьевна купила колоду из 54 карт, содержащую два джокера. Она выбрала случайным образом N из этих карт (вероятность вытащить каждую карту при этом одинакова).

Вычислите вероятность того, что среди выбранных карт будет как минимум один джокер.

Input

Входной файл содержит одно целое число N ($2 \leqslant N \leqslant 54$).

Output

Выведите одно число — требуемую вероятность с точностью не хуже 10^{-6}.

Examples

input.txt	output.txt
2	0.073375
10	0.338924