

## Problem A. Donut

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 seconds  
Memory limit: 1024 mebibytes

В данной задаче мы будем использовать расстояние Чебышёва на декартовой плоскости. Расстояние Чебышёва между двумя точками  $(p_x, p_y)$  и  $(q_x, q_y)$  на плоскости определяется как  $\max(|p_x - q_x|, |p_y - q_y|)$ .

Определим **пончик** с центром  $(x_c, y_c)$ , внутренним радиусом  $l$  и внешним радиусом  $r$  как множество таких точек  $(x, y)$ , что  $l \leq \max(|x - x_c|, |y - y_c|) \leq r$ .

На плоскости выбраны  $n$  точек, пронумерованных последовательными целыми числами от 1 до  $n$ . Каждая точка имеет некоторую стоимость.

Требуется разместить на плоскости пончик внутреннего радиуса  $l$  и внешнего радиуса  $r$  так, чтобы центр находился в точке с целыми координатами, а сумма стоимостей всех выбранных точек, принадлежащих пончику, была наибольшей возможной.

Ваша задача — вывести эту стоимость.

### Input

Первая строка входа содержит три целых числа — количество выбранных точек  $n$ , внутренний радиус пончика  $l$  и внешний радиус пончика  $r$  ( $1 \leq n \leq 10^5$ ,  $1 \leq l \leq r \leq 10^9$ ).

Последующие  $n$  строк задают выбранные точки.  $i$ -я из этих строк содержит три целых числа  $x$ ,  $y$  и  $s$  — координаты точки и её стоимость ( $-10^9 \leq x, y \leq 10^9$ ,  $-10^4 \leq s \leq 10^4$ ). При этом некоторые выбранные точки могут совпадать.

### Output

Выведите одно целое число — максимальную сумму стоимостей точек, принадлежащих пончику при его оптимальном размещении.

### Examples

standard input	standard output
4 1 1 0 1 1 0 -1 1 1 0 -100 -1 0 -100	2
4 1 2 0 1 1 0 -1 1 1 0 -100 -1 0 -100	1

## Problem C. Earthquake

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 512 mebibytes

Остров Sodor соединён с Англией мостами.

Так как остров находится далеко от английского побережья, существует много (ровно  $n$ ) путей, которые соединяют остров и Англию через несколько мостов. В частности, путь  $i$  соединяет Англию и Sodor через  $k_i$  ( $k_i \geq 1$ ) мостов. Обозначим  $j$ -й мост (считая от острова Sodor) пути  $i$  как  $B[i, j]$ . На иллюстрации ниже указаны два пути ( $n = 2$ ) и пять мостов, то есть  $k_1 = 2$  и  $k_2 = 3$ .

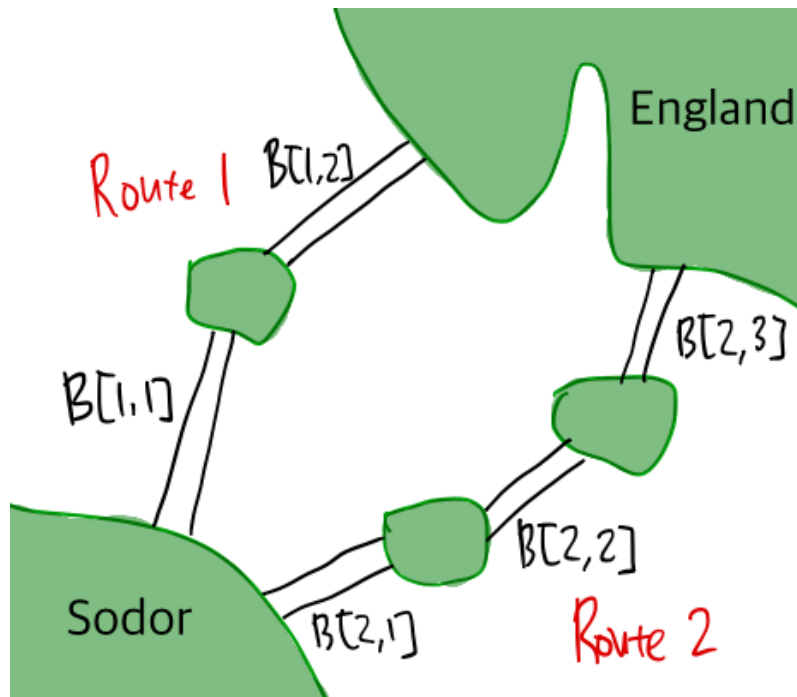


Figure 1: Two routes and five bridges

Для любых двух соседних мостов маршрута  $B[i, j]$  и  $B[i, j + 1]$ , где  $j + 1 \leq k_i$ , их объединение — это небольшой островок, который является точкой соединения двух мостов. Как показано на рисунке, мосты не пересекаются и все островки тоже попарно различны, так что общее количество путей действительно равно  $n$ . В частности, если какой-то из мостов пути  $i$  повреждён, то путь  $i$  не может быть использован для передвижения.

В связи с недавними землетрясениями некоторые мосты повреждены и не могут использоваться. На данный момент неизвестно, какие мосты выдержали землетрясение, а какие оказались разрушенными. Для каждого моста, впрочем, известна вероятность того, что этот мост повреждён. Пусть  $p[i, j]$  ( $0 < p[i, j] < 1$ ) — вероятность того, что мост  $B[i, j]$  выдержал землетрясение. Считаем, что все события типа “мост выдержал землетрясение” являются независимыми.

Спасатели хотят узнать, существует ли какой-либо путь, соединяющий Sodor и Англию. Для проверки, повреждён ли мост, необходимо выслать разведгруппу к соответствующему мосту. Отправка разведгруппы к одному мосту стоит 1 условную единицу. Требуется выяснить математическое ожидание суммы, затраченной на то, чтобы достоверно выяснить, существует ли хотя бы один путь, соединяющий Sodor и Англию при том, что последовательность проверок выбирается так, чтобы минимизировать это математическое ожидание.

## Input

Первая строка входа содержит одно целое число  $n$  — количество путей ( $2 \leq n \leq 1000$ ).

Каждая из последующих  $n$  строк задаёт информацию об одном маршруте.  $i$ -я из этих строк начинается целым числом  $k_i$ , задающим количество мостов в  $i$ -м маршруте ( $1 \leq k_i \leq 1000$ ). Далее следует  $k_i$  целых чисел  $q[i, j]$ , ( $1 \leq j \leq k_i$ ).  $q[i, j]$  — целое число между 1 и 999 включительно, задающее вероятность  $p[i, j] = q[i, j]/1000$ .

## Output

Выведите одно число — матожидание суммы, затраченной на выяснение того, существует ли путь между Англией и островом, с абсолютной или относительной точностью не хуже  $10^{-9}$ .

## Examples

standard input	standard output
2 3 900 900 900 2 100 100	3.0081
3 1 240 1 310 1 50	2.2144

## Note

Первый пример описывает изображённый на рисунке мост. Интуитивно кажется, что маршрут 1 с хорошей вероятностью будет целым, в то время как маршрут 2, скорее всего, разрушен. Оптимальная последовательность — выслать разведгруппу к трём мостам первого маршрута (в произвольном порядке до тех пор, пока статус первого маршрута не будет прояснён), а затем, если потребность ещё осталась, проверить два моста второго маршрута.

Во втором примере оптимальным является проверить второй маршрут, затем первый, а затем, если понадобится, третий.

## Problem D. Dynamic Input Tool

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 512 mebibytes

В новом текстовом редакторе CodingSlave 1.0 существует две операции редактирования строки:

1. Добавить один символ в конце текущей строки.
2. Выбрать непустую подпоследовательность текущей строки и дописать её в конец текущей строки.

Изначально текущая строка пуста.

Вам задано слово, составленное из строчных латинских букв. Вычислите минимальное количество операций, требуемое для того, чтобы получить это слово.

### Input

Первая строка входа содержит слово  $S$ , состоящее из строчных латинских букв ( $1 \leq |S| \leq 10^6$ ).

### Output

Выведите одно целое число: минимальное количество операций, требуемых для того, чтобы построить строку  $S$ , используя CodingSlave 1.0.

### Examples

standard input	standard output
aaa	3
aabaaaabaa	5

## Problem E. Central Lake

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 seconds  
Memory limit: 512 mebibytes

Jaehyun живёт в городе, представляющем собой круг радиуса  $R$ . На окружности есть 360 000 точек, занумерованных против часовой стрелки: переход против часовой стрелки из точки 359 999 приводит в точку 0. Все расстояния между парами соседних точек одинаковы.

Первоначально в некоторых из этих 360 000 точек построены дома (всего  $N$  домов), жители которых могли ходить друг к другу в гости по прямому пути. Однако в центре города решили устроить озеро радиуса  $r$ , центр которого совпадает с центром города. Более того, время от времени поступают распоряжения о сносе существующих домов и постройке новых.

Jaehyun хочет после каждого распоряжения знать максимум из кратчайших расстояний по суше между двумя существующими домами.

### Input

Первая строка входа содержит два целых числа  $R$  и  $r$ : радиус города и радиус озера, соответственно ( $10 \leq R \leq 10^5$ ,  $1 \leq r < R$ ).

Вторая строка содержит целое число  $N$  ( $2 \leq N \leq 100\,000$ ) — первоначальное количество домов.

Третья строка содержит  $N$  целых чисел  $a_1, a_2, \dots, a_N$  — номера точек, в которых расположены  $N$  домов ( $0 \leq a_i < 360\,000$ , все  $a_i$  попарно различны).

Следующая строка содержит целое число  $Q$  — количество распоряжений ( $1 \leq Q \leq 100\,000$ ).

Последующие  $Q$  строк задают распоряжения. Каждая из этих строк содержит по два целых числа  $q$  и  $x$  ( $1 \leq q \leq 2$ ,  $0 \leq x < 360\,000$ ). Распоряжение имеет следующий формат в соответствии с типом  $q$ :

- “1  $x$ ”: построить дом в точке с номером  $x$ .
- “2  $x$ ”: снести дом в точке с номером  $x$ .

Гарантируется, что в момент выполнения распоряжения типа 1 в точке с номером  $x$  нет дома, а в момент выполнения распоряжения типа 2 в точке с номером  $x$  есть дом. Также гарантируется, что в любой момент времени существует не менее двух домов.

### Output

Выведите  $Q + 1$  строк. В первой строке выведите наибольшее расстояние по суше между домами перед выполнением запроса (но с уже существующим озером). В следующих  $Q$  строках выведите требуемое расстояние после соответствующего распоряжения. Ответ принимается с абсолютной или относительной погрешностью не хуже  $10^{-6}$ .

## Example

standard input	standard output
10 5	14.14213562
2	22.55649583
0 90000	22.55649583
4	19.93850195
1 180000	10
1 240000	
2 0	
2 90000	

## Problem F. Computing MDSST

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 512 mebibytes

В этой задаче рассматриваются взвешенные неориентированные графы, в которых все рёбра имеют положительный вес.

Пусть  $D(G, i, j)$  — длина кратчайшего пути в графе  $G$  между вершинами  $i$  и  $j$ .

Задан полный взвешенный неориентированный граф  $G$ , который состоит из  $n$  вершин, пронумерованных последовательными целыми числами от 1 до  $n$ . Среди остовных деревьев графа  $G$ , остовное дерево с минимальной суммой расстояний — это такое остовное дерево  $T$ , для которого  $S(T) = \sum_{1 \leq i < j \leq n} D(T, i, j)$  минимально. Ваша задача найти остовное дерево с минимальной суммой расстояний в  $G$  и вывести  $S(T)$ .

### Input

Первая строка содержит целое число  $n$  — количество вершин в графе ( $2 \leq n \leq 15$ ).  $i$ -я из последующих  $n - 1$  строк содержит  $n - i$  целых чисел, разделённых пробелами.  $j$ -е целое число — длина ребра между вершинами  $i$  и  $i + j$ .

Все длины расположены между 1 и  $10^9$  включительно.

### Output

Выведите одно целое число — значение  $S(T)$  для остовного дерева с минимальной суммой расстояний.

### Example

standard input	standard output
4 3 2 1 5 6 7	18

## Problem L. XOR Transformation

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 1024 mebibytes

Задан целочисленный массив длины  $N$ :  $X = [x_0, x_1, \dots, x_{N-1}]$ .

Определим преобразование  $F_k(X)$  массива  $X$   $F_k(x) = [f_{k,0}(X), f_{k,1}(X), \dots, f_{k,N-1}(X)]$ , следующим образом:

- $k$  — целое число между 1 и  $N$ , включительно.
- $f_{k,i}(X) = \bigoplus_{j=0}^{k-1} x_{(i+j) \bmod N}$ , где  $i$  — целое число между 0 и  $N-1$ , включительно и  $\oplus$  — побитовый XOR.

Вам также даны два целых числа  $T$  и  $K$ . Вычислите значение  $F_K^T(X)$  и выведите его. Обратите внимание, что  $F_K^1(X) = F_K(X)$  и  $F_K^t(X) = F_K(F_K^{t-1}(X))$  для  $t > 1$ .

### Input

Первая строка содержит три целых числа  $N$ ,  $K$  и  $T$  ( $1 \leq K \leq N \leq 10^5$ ,  $1 \leq T \leq 10^{18}$ ).

Вторая строка содержит  $N$  целых неотрицательных чисел  $x_0, x_1, \dots, x_{N-1}$  ( $0 \leq x_i \leq 10^9$ ), где  $x_i$  — это  $i$ -й элемент массива  $X$  при нумерации с нуля.

### Output

Пусть  $F_K^T(X) = [a_0, a_1, \dots, a_{N-1}]$ . Выведите  $N$  целых чисел  $a_0, a_1, \dots, a_{N-1}$ .

### Examples

standard input	standard output
5 3 1 3 0 2 1 2	1 3 1 0 1
5 3 2 3 0 2 1 2	3 2 0 0 3
5 3 3 3 0 2 1 2	1 2 3 0 2
5 3 15 3 0 2 1 2	3 0 2 1 2
11 5 10000000000000000000 2 2 4 5 9 1 5 7 7 1 8	13 4 5 8 1 0 5 10 3 4 8



## Problem M. Ants

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 Mebibytes

При движении муравьи образуют колонны так, что каждый муравей, кроме первого находится строго позади другого муравья. В случае если две колонны встречаются, муравьи начинают перепрыгивать друг через друга.

С момента встречи колонн, каждую секунду происходит следующее. Если какие-то два муравья, двигающиеся друг навстречу другу, оказываются рядом, то они меняются позициями (один перепрыгивает через другого). Восстановите порядок муравьёв по прошествии  $T$  секунд.

### Input

В первой строке входного файла содержатся два целых числа  $N_1$  и  $N_2$  — количество муравьёв в первой и во второй колоннах соответственно. Следующие две строки задают порядок муравьёв в первой и второй колоннах (начиная с лидера колонны). Каждый муравей обозначен заглавной латинской буквой, причём никакие два муравья не обозначены одной и той же буквой вне зависимости от того, в какой колонне они идут. Последняя строка входного файла содержит целое число  $T$  ( $0 \leq T \leq 50$ ).

### Output

В выходной файл выведите в одной строке порядок муравьёв после  $T$  секунд. Считать, что первая колонна муравьёв движется слева направо, а вторая — справа налево.

### Examples

standard input	standard output
3 3 ABC XYZ 0	CBAXYZ
3 3 ABC XYZ 2	CXBYAZ
3 4 XLQ CRUZ 3	CQRLUXZ

## Problem N. Manhattan Circle

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 Mebibytes

В Манхэттенской метрике расстояние между двумя точками  $A(x_1, y_1)$  и  $B(x_2, y_2)$  определяется следующим образом:

$$D(A, B) = |x_1 - x_2| + |y_1 - y_2|$$

Остальные определения такие же, как и в евклидовой геометрии, включая и определение окружности:

Окружностью называется геометрическое место точек на плоскости, расположенных на фиксированном расстоянии (называемом радиусом) от фиксированной точки (называемой центром окружности).

По заданному радиусу  $R$  выведите площадь круга в манхэттенской метрике.

### Input

Во входном файле задано одно целое число — радиус  $R$  ( $1 \leq R \leq 10^4$ )

### Output

Выведите площадь круга заданного радиуса в манхэттенской метрике с точностью не хуже  $10^{-9}$ .

### Examples

standard input	standard output
1	2.0

## Problem O. Polygon

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 Mebibytes

Рассмотрим выпуклый  $N$ -угольник, такой, что никакие три его диагонали не пересекаются в одной точке внутри  $N$ -угольника. Найдите количество точек пересечения диагоналей для такого  $N$ -угольника.

### Input

Входной файл содержит одно число  $N$ ,  $3 \leq N \leq 100$ .

### Output

Выведите одно число — количество точек пересечения диагоналей.

### Examples

standard input	standard output
3	0
4	1
6	15

## Problem P. Sequence

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 mebibytes

Задана бесконечная целочисленная последовательность

$$A_0 = 0 \tag{1}$$

$$A_i = (A_{i-1} + q) \pmod p, \text{ for all } i > 0 \tag{2}$$

Число  $p$  — простое.

Посчитайте, сколько чисел из интервала  $[0..p-1]$  не встретятся в этой последовательности.

### Input

Первая строка входа содержит одно целое число  $p$ . Вторая строка содержит одно целое число  $q$ . ( $0 \leq q \leq p \leq 10^{100}$ ).

### Output

Выведите одно целое число — ответ к задаче.

### Example

standard input	standard output
5 1	0

## Problem Q. Songs

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 512 Mebibytes

Каждый вечер жители небольшой деревни собираются у костра и поют песни. Среди жителей есть певец, который, если он присутствует у костра, исполняет новую песню, которую ещё никто из жителей не слышал, после чего жители расходятся по домам. Если певец отсутствует, то жители поют друг другу все песни певца, которые они уже слышали.

По заданному списку жителей, присутствовавших у костра в течение  $E$  последовательных вечеров, выведите список тех жителей, которые слышали знают все песни, исполнявшиеся за это время.

### Input

Первая строка входного файла содержит целое число  $N$ ,  $1 \leq N \leq 100$  — количество жителей. Жители занумерованы от 1 до  $N$ , причём первый номер присвоен певцу. Вторая строка содержит целое число  $E$ ,  $1 \leq E \leq 50$  — количество вечеров. Каждая из последующих  $E$  строк содержит список жителей, которые были у костра в соответствующий вечер. Каждая строка начинается с целого положительного числа  $K$ ,  $2 \leq K \leq N$  — количества присутствующих в этот вечер жителей, далее идут  $K$  чисел — номера жителей. Гарантируется, что певец появится у костра как минимум один раз.

### Output

Выведите по одному на каждой строке в порядке возрастания, начиная с певца, номера тех жителей, которые знают все песни, исполненные певцом за  $E$  вечеров.

### Example

standard input	standard output
4	1
3	2
2 1 2	4
3 2 3 4	
3 4 2 1	
8	1
5	2
4 1 3 5 4	6
2 5 6	8
3 6 7 8	
2 6 2	
4 2 6 8 1	