

Problem A. Irreducible Polynomials

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

Имеется некоторое простое число p . Возьмем $Z_p = \{0, 1, \dots, p - 1\}$ — поле вычетов по модулю p . В этом поле операции умножения и сложения выполняются по модулю p . Теперь рассмотрим нормированные многочлены над этим полем вида

$$f(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0,$$

где n — степень многочлена, x — переменная, $a_i \in Z_p$ — коэффициенты.

Так, например, многочлен $x^2 + x + 1$ в поле Z_2 неприводим. Нормированный многочлен называется неприводимым, если нельзя представить его в виде $f(x) = p(x) \cdot q(x)$, где $p(x), q(x)$ — многочлены степени меньшей, чем $f(x)$. И это единственный неприводимый многочлен второй степени в поле Z_2 .

Ваша задача — вычислить количество нормированных неприводимых многочленов степени n в поле Z_p . Поскольку это количество может быть большим, то требуется получить лишь остаток от деления этого числа на m .

Limits

Числа p, n, m — целые, p — простое.

$$1 \leq p, n, m \leq 10^9.$$

Input

Единственная строка входного файла содержит три числа p, n, m .

Output

Выведите в выходной файл количество неприводимых многочленов степени n в поле Z_p по модулю m .

Example

standard input	standard output
2 2 10	1
3 4 100	18

Problem B. Game

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

Двое игроков играют в следующую игру. Каждый из игроков называет некоторую последовательность, состоящую из 0 и 1. После этого начинают подбрасывать симметричную монету до тех пор, пока результаты последних бросков не совпадут с последовательностью одного из игроков (значение 0 соответствует решке, 1 – гербу). Выигрывает, естественно, тот игрок, чья последовательность выпадет раньше. Требуется по заданным последовательностям, названным игроками, определить вероятность победы первого игрока.

Limits

Последовательности, названные игроками, непусты и имеют длину не более 50. Ни одна из последовательностей не является суффиксом другой.

Input

В первой строке содержится последовательность первого игрока, во второй строке — второго игрока (без пробелов).

Output

Выведите вероятность того, что последовательность первого игрока выпадет раньше, чем последовательность второго игрока, с точностью не менее 10^{-8} .

Example

standard input	standard output
001 110	0.50000000
00 10	0.25000000

Problem C. Paint the Fence

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

Имеется забор, состоящий из N досок, следующих одна за другой. Требуется покрасить его таким образом, чтобы каждая доска была окрашена целиком в один из имеющихся C цветов. При этом некоторые пары цветов являются несовместимыми, то есть не допускается, чтобы после покраски две соседние доски были окрашены в соответствующие цвета. Всего имеется M таких пар. Ваша задача — определить количество допустимых раскрасок (то есть таких, при которых нет двух соседних досок, покрашенных в несовместимые цвета).

Limits

N, C, M — целые числа.

$1 \leq N \leq 10^{18}$, $1 \leq C \leq 100$, $0 \leq M \leq C(C + 1)/2$.

Input

В первой строке содержатся три числа N, C, M . В каждой из последующих M строк содержится по два числа, определяющих пару несовместимых цветов.

Output

Выведите остаток от деления количества допустимых раскрасок забора на $10^9 + 7$.

Example

standard input	standard output
2 3 2 1 2 2 3	5
3 2 1 1 2	2

Problem D. Cities

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

На координатной прямой расположены N городов. Было решено выбрать K различных пар этих городов и назвать города каждой пары городами-побратимами по отношению друг к другу. При этом у каждого города может быть не более одного города-побратима. Требуется определить максимальное и минимальное значение суммы расстояний между всеми парами городов-побратимов.

Limits

N, K – целые числа.

$1 \leq N \leq 100000, 0 \leq K \leq N/2$. Координаты городов не превышают 10^9 по абсолютной величине.

Input

В первой строке содержатся числа N и K . Во второй – N чисел, определяющих координаты городов.

Output

В единственной строке нужно вывести два числа – максимальное и минимальное суммарное расстояния, которые могут получиться при выборе K пар городов-побратимов.

Example

standard input	standard output
5 2 0 3 4 7 9	13 3
3 1 2 7 5	5 2

Problem E. Vacation on a River

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

Семейная пара решила провести отдых на берегу реки. Джордж любит высокие места и хочет доехать туда, где берег будет как можно выше над уровнем реки. Его жена Мэри, наоборот, боится высоты и хочет провести отдых там, где берег будет как можно ниже. Сейчас же они едут на машине по односторонней главной дороге, вдоль которой имеется N поворотов к реке. Дорога за каждым из этих поворотов ведет к реке и каждый из супругов знает высоту, на которой расположено место, к которому ведет соответствующая дорога. За рулём сидит Джордж, но Мэри может отвлекать Джорджа так, что тот может не заметить очередного поворота *за исключением последнего поворота* (и Джордж знает об этом). Все повороты настолько похожи, что, подъезжая к очередному повороту, Джордж не может знать наверняка, к какому месту тот выводит.

Главная дорога продолжается и за последним поворотом и тоже ведет к месту у реки с известной высотой берега. Очевидно, что Джордж может применить одну из следующих стратегий: повернуть на первом замеченном им повороте, повернуть на втором из таких поворотов, повернуть на третьем и т.д., либо вообще не сворачивать (разумеется, в случае, если окажется, что в действительности он заметит меньше поворотов, чем требуется в намеченной им стратегии, пара прибудет в место, расположенное в конце главной дороги).

Требуется определить оптимальную стратегию для Джорджа в предположении, что Мэри также будет действовать оптимально.

Limits

N, h_i — целые числа.

$$1 \leq N \leq 10^5, 0 \leq h_i \leq 1000.$$

Input

В первой строке содержится число N . Во второй строке записано $N + 1$ чисел h_i , определяющих высоту места, к которому ведет дорога от i -го поворота (h_{N+1} — высота места, к которому выводит главная дорога, если никуда с нее не сворачивать).

Output

Выведите в первой строке максимальную среднюю высоту места, к которому может добраться Джордж, при оптимальном противодействии Мэри. Во второй строке выведите $N + 1$ число — вероятности, с которыми Джордж должен применять каждую из своих чистых стратегий для достижения этой высоты. Все величины должны выводиться с точностью не менее 10^{-6} . В случае, если существует несколько оптимальных стратегий, следует выбирать ту из них, у которой вероятность выбора стратегии “Нигде не сворачивать” максимальна. Если и таких несколько — ту, которая имеет наибольшую вероятность выбора N -го поворота, и т.д.

Example

standard input	standard output
2 0 6 3	4.000000 0.333333 0.666667 0.000000
3 2 3 4 2	2.800000 0.400000 0.400000 0.200000 0.000000

Problem F. Representable numbers

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

Заданы два целых числа a и b . Число x назовем *представимым*, если его можно представить в виде суммы $x = x_1 + x_2 + \dots + x_n$ конечного (возможно нулевого) числа слагаемых x_i , каждое из которых равно a или b . Требуется определить, сколько различных чисел на отрезке $[A, B]$ являются представимыми.

Limits

a, b — целые числа, не превосходящие по модулю 10000.

A, B — целые числа, не превосходящие по модулю 10^{18} .

$A \leq B$.

Input

В единственной строке содержатся числа a, b, A, B .

Output

Выведите количество представимых для заданных a и b чисел из отрезка $[A, B]$.

Example

standard input	standard output
4 5 7 12	4
6 10 20 30	6

Problem G. Quadratic permutation

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

Рассмотрим перестановку чисел от a до b . Будем называть ее *квадратичной*, если для каждого элемента его сумма с элементом, который попадает на его место после перестановки, является точным квадратом. Более точно, квадратичной перестановкой называется такая биекция p множества целых чисел от a до b самого на себя, что для любого i выполняется $i + p(i) = j^2$ для некоторого целого числа j . Требуется для заданных a и b найти квадратичную перестановку.

Limits

a, b – целые числа.

$0 \leq a \leq 100, 0 \leq b \leq 100000, a \leq b$.

Input

В единственной строке содержатся числа a и b .

Output

Выведите $b - a + 1$ чисел, определяющих значения $p(i)$ для всех i от a до b , где p – некоторая квадратичная перестановка. Если такой перестановки при заданных a и b не существует, выведите одно число -1 .

Example

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

Problem H. De Bruijn's Cycle

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

Циклом де Брёйна порядка n для множества $D = \{0, 1, \dots, b-1\}$ называется циклическая последовательность a_0, a_1, \dots, a_{l-1} такая, что каждый вектор длины n над множеством D встречается в этой последовательности ровно один раз (т.е. для любых $b_0, b_1, \dots, b_{n-1} \in D$ существует единственное k в пределах от 0 до $l-1$ такое, что $b_j = a_{(k+j) \bmod l}$ для всех $j = \overline{0, n-1}$). Требуется построить такую последовательность.

Limits

n, b – целые числа.

$1 \leq n \leq 1000, 1 \leq b \leq 10, b^n \leq 10^7$.

Input

В единственной строке содержатся числа n и b .

Output

В единственной строке выведите цикл де Брёйна порядка n для множества b -ичных цифр (без пробелов).

Example

standard input	standard output
2 3	001102122
3 2	00010111

Problem I. Card Duel

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

Двое игроков играют в игру. Каждый из игроков изначально получает некоторое количество карточек (пусть будет n_1 и n_2 соответственно). Каждый ход игроки выбирают по одной из имеющихся у них на руках карточек и затем открывают их. Более слабая карта сбрасывается в отбой, а более сильную игрок, показавший ее, забирает обратно.

В случае, если игроки показали одинаковые карты, они обе сбрасываются в отбой. Игра продолжается до тех пор, пока хотя бы у одного из игроков не закончатся карты. Если при этом у одного из игроков осталась еще хотя бы одна карта, он получает 1 очко, а соперник — 0. Если же у обоих игроков закончились карты, каждый получает по 0.5 очков. Всего есть N типов карт. Отношение силы карт может быть нетранзитивным и задается матрицей A . A_{ij} равно 1, если карта i бьет карту j , и 0 в противном случае. Требуется определить цену этой игры для первого игрока в предположении, что второй игрок играет оптимально.

Limits

$1 \leq n_1, n_2, N \leq 8$.

$A_{ij} + A_{ji} = 1$ при $i \neq j$, $A_{ii} = 0$.

Input

В первой строке задается число N . Последующие N строк содержат по N чисел, определяющих матрицу A . Следующая строка содержит число n_1 и еще n_1 чисел, каждое из которых определяет тип соответствующей карточки первого игрока. В последней строке в аналогичном формате задаются карточки второго игрока.

Output

Выведите цену игры для первого игрока с точностью не менее 10^{-8} .

Example

standard input	standard output
3 0 1 1 0 0 1 0 0 0 2 3 2 1 1	0.00000000
3 0 1 0 0 0 1 1 0 0 3 1 2 3 3 1 2 3	0.50000000
3 0 1 0 0 0 1 1 0 0 3 1 2 3 3 2 2 3	0.66666667

Problem J. Plane Boarding

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

N человек садятся в самолет, в котором есть в точности N мест. У каждого из пассажиров есть билет на некоторое место в этом самолете и нет двух одинаковых билетов. Однако некоторые из пассажиров являются сумасшедшими. Люди заходят по одному в салон самолета. Сумасшедшие люди при входе в самолет не смотрят на билет, а садятся на некоторое место, выбирая его равновероятно из свободных, нормальные же люди занимают место, указанное в билете. Но если место нормального пассажира занято, то, чтобы не начинать скандал, он садится на любое свободное место с равной вероятностью. Требуется определить для каждого пассажира вероятность того, что он займет при входе в самолет место, указанное в его билете.

Limits

N – целое число, $1 \leq N \leq 100000$.

Input

В первой строке задается число N . Во второй строке записаны N чисел, каждое из которых определяет соответствующего пассажира в порядке входа в салон самолета (0 обозначает нормального человека, 1 – сумасшедшего).

Output

Выведите N чисел, каждое из которых определяет вероятность того, что соответствующий пассажир займет свое место. Все значения должны быть выведены с точностью не меньше 10^{-8} .

Example

standard input	standard output
3 0 1 0	1.00000000 0.50000000 0.50000000
4 1 0 1 0	0.25000000 0.75000000 0.33333333 0.33333333