

Problem A. Маршрут жизни

Input file: **stdin**
Output file: **stdout**
Time limit: 1 секунда
Memory limit: 256 мегабайт

Поверхность открытой совсем недавно планеты `ilc5l` представляет собой абсолютно гладкую сферу радиуса R . Для исследования планеты объединенными аэрокосмическими силами Земли был направлен спускаемый аппарат Вальдемар-63, который должен был опуститься на поверхность в точке A . Но в расчетах были допущены ошибки при разложении простых чисел на множители, и в результате Вальдемар-63 попал в точку B . К счастью, аппараты серии Вальдемар снабжены ходовым механизмом, который позволит добраться до изначальной точки назначения. При этом энергетические запасы аппарата весьма ограничены, поэтому необходимо проложить минимально возможный путь до точки A .

Ситуация осложняется тем, что некая точка O на поверхности планеты излучает волны неизвестного характера, распространяющиеся в радиусе r от неё (расстояние измеряется по поверхности). Как показали предыдущие 62 миссии, любой аппарат, попадающий в зону действия излучения, мгновенно выходит из строя. Таким образом, маршрут Вальдемара-63 следует строить, избегая зоны излучения.

Найдите минимальное расстояние, которое придется преодолеть спускаемому аппарату для перемещения из точки B в точку A .

Input

В первой строке целые числа X_A, Y_A ($-90 \leq X_A \leq 90, -180 \leq Y_A \leq 180$) — координаты (широта и долгота в градусах) точки A . Во второй строке — координаты точки B . В третьей — три числа: координаты O и радиус действия излучения r . В четвёртой строке — положительное целое число, не превышающее 1000 — радиус планеты R . Гарантируется, что точки A и B находятся вне зоны действия излучения.

Output

Вывести искомое расстояние. Ответ считается правильным, если абсолютная или относительная погрешность не превышает 10^{-4} .

Examples

stdin	stdout
55 49 55 129 90 0 1 2	1.510687

Problem B. Разреженные скобочки

Input file: stdin
Output file: stdout
Time limit: 1 секунда
Memory limit: 256 мегабайт

— Да закройте же скобки, мужчина!..
Вишневский В.

Порядок вычислений в арифметических выражениях задается расстановкой скобок, например, $(3 \cdot (2 + 1)) \cdot (4 - 5)$. Если удалить все элементы выражения за исключением скобок, то оставшиеся символы образуют *скобочную последовательность* $((()))()$. Предположим теперь, что кроме открывающих и закрывающих скобок может присутствовать ещё один символ, не нарушающий правильность скобочных последовательностей. Например, символ «0». Такую последовательность будем называть *разреженной скобочной последовательностью*. Понятие разреженной скобочной последовательности можно определить и так:

- Пустая строка считается разреженной скобочной последовательностью.
- Если S и T — разреженные скобочные последовательности, то строки $0S$, $S0$, (S) и ST также являются разреженными скобочными последовательностями.

Глубиной разреженной скобочной последовательности называется максимальная разность между количеством открывающих и закрывающих скобок в префиксе последовательности. (Префиксом строки S называется строка, которую можно получить из S удалением некоторого количества символов с конца строки. Например, префиксами строки « $ABCAB$ » являются строки «», « A », « AB », « ABC » и « $ABCA$ » и « $ABCAB$ »). Так, глубина последовательности « $(0)(0())0$ » равна двум (префикс « $(0)(0($ » содержит три открывающие и одну закрывающую скобки).

Необходимо по заданным значениям n и k вычислить количество разреженных скобочных последовательностей из n символов, которые имеют глубину вложения скобок, равную k .

Input

Единственная строка содержит разделенные пробелом целые числа n и k ($1 \leq n \leq 300$, $0 \leq k \leq n$).

Output

Выведите количество разреженных скобочных последовательностей из n символов, которые имеют глубину k . Ответ запишите по модулю $(10^9 + 9)$.

Examples

stdin	stdout
3 0	1
3 1	3
3 2	0

Note

В первом примере возможна единственная разреженная скобочная последовательность с глубиной вложения скобок 0: 000.

Во втором примере возможны три разреженные скобочные последовательности с глубиной вложения скобок 1: 0(), (0) и ()0.

Problem C. Шоколадные Треугольники

Input file: **stdin**
Output file: **stdout**
Time limit: 4 секунды
Memory limit: 256 мегабайт

*Пирамиды были большими квадратными
треугольниками, построенными в пустыне.
Из сочинения американского студента.*

Компания «Тортофф» производит вашу любимую шоколадную продукцию. Для вашего удобства мы выпускаем шоколад исключительно в форме выпуклого n -угольника. Вы можете разламывать его на куски вдоль *непересекающихся* внутри многоугольника диагоналей. По результатам опросов наших экспертов, самые вкусные куски шоколада имеют треугольную форму.

Любой покупатель может помочь нам улучшить качество нашей продукции. Для компании важно узнать количество способов разломать наш любимый шоколад ровно на k треугольных частей.

Input

В единственной строке записаны два целых числа n и k ($3 \leq n \leq 300$, $0 \leq k \leq n - 2$).

Output

Выведите одно число — количество способов разрезания n -угольника ровно на k треугольных частей с помощью непересекающихся внутри многоугольника диагоналей. Ответ запишите по модулю $(10^9 + 9)$.

Examples

stdin	stdout
4 0	1
4 1	0
4 2	2

Problem D. БДБД

Input file: `stdin`
Output file: `stdout`
Time limit: 9 секунд
Memory limit: 256 мегабайт

Большая Древесная База Данных создана для того, чтобы в ней можно было надежно сохранить и раскрасить любое дерево. В новой версии БДБД запланирован новый функционал, для реализации которого потребуется вновь переосмыслить теорию графов.

В БДБД хранится взвешенное дерево. В языке запросов Системы Управления Большой Древесной Базы Данных (СУБДБД) предусмотрены два вида запросов:

1. «1 v d c » — покрасить все вершины, находящиеся на расстоянии не более d от вершины v , в цвет c . Все вершины изначально окрашены в цвет с номером 0.
2. «2 v » — вывести цвет вершины v .

Необходимо запрограммировать работу СУБДБД и ответить на все запросы пользователя.

Input

В первой строке число N ($1 \leq N \leq 10^5$) — количество вершин дерева.

Следующие $N - 1$ строк содержат описание ребер, по три числа в строке a_i, b_i, w_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$, $1 \leq w_i \leq 10^4$), где i -ое ребро имеет вес w_i и соединяет вершины a_i и b_i .

В следующей строке число Q ($1 \leq Q \leq 10^5$) — число запросов. В каждой из Q следующих строк запросы одного из двух видов:

1. Числа 1, v, d, c ($1 \leq v \leq N$, $0 \leq d \leq 10^9$, $0 \leq c \leq 10^9$).
2. Числа 2, v ($1 \leq v \leq N$).

Все числа во входных данных целые.

Output

Для каждого запроса второго типа необходимо вывести в отдельной строке цвет запрошенной вершины.

Examples

stdin	stdout
5	6
1 2 30	6
1 3 50	0
3 4 70	5
3 5 60	7
8	
1 3 72 6	
2 5	
1 4 60 5	
2 3	
2 2	
1 2 144 7	
2 4	
2 5	

Problem E. Горох-Сити

Input file: **stdin**
Output file: **stdout**
Time limit: 2 секунды
Memory limit: 256 мегабайт

Царь Горох был мудрым и дальновидным правителем. Всё время проводил в государственных делах и заботах. Государство развивалось, народ размножался и строился. Новые дома росли, как грибы после дождя. Соседние цари завидовали и подумывали о войне. Чтобы обезопасить столицу от врагов, задумал царь Горох построить новую каменную стену, охватывающую всю столицу с её домами.

Для того, чтобы карта города красиво смотрелась на стене в кабинете царя, было принято решение о том, что столица должна иметь прямоугольные границы. При этом площадь города должна оставаться минимально возможной, а все N домов, зарегистрированные в муниципалитете, должны оказаться в границах прямоугольного города.

Input

В первой строке входного файла целое число N — количество домов в городе ($3 \leq N \leq 80\,000$). В следующих N строках пары целых чисел X, Y — декартовы координаты домов ($-25\,000 \leq X, Y \leq 25\,000$). Гарантируется, что в городе есть как минимум три дома, не лежащие на одной прямой.

Output

В четырех строках выведите координаты вершин прямоугольника в порядке обхода против часовой стрелки. Координаты следует выводить с точностью не менее 5 знаков после запятой. Ответ считается верным, если абсолютная или относительная погрешность значения площади прямоугольника не превосходит 10^{-5} и все дома лежат внутри него или находятся на расстоянии менее 10^{-5} от его сторон.

Examples

stdin	stdout
3	0.000000 2.000000
0 0	0.000000 0.000000
2 2	2.000000 0.000000
2 0	2.000000 2.000000

Problem F. Красивые суммы

Input file: stdin
Output file: stdout
Time limit: 1 секунда
Memory limit: 256 мегабайт

Красивыми суммами называют суммы нескольких подряд идущих положительных целых чисел. Например, суммы $7 + 8$ и $4 + 5 + 6$ — красивые, а сумма $3 + 5 + 7$ — некрасивая, хотя результат суммирования во всех случаях равен 15. (Сумма из одного слагаемого 15 тоже считается красивой.) Исходя из этого, *красотой* целого положительного числа будем называть количество представлений этого числа в виде красивых сумм. Например, красота числа 15 равна 4, поскольку 15 представляется в виде красивых сумм четырьмя способами: $15 = 7 + 8 = 4 + 5 + 6 = 1 + 2 + 3 + 4 + 5$.

Из двух целых чисел более красивым считается то, у которого больше представлений в виде красивых сумм. При равенстве количеств таких представлений предпочтение в красоте отдаётся меньшему из них. Например, 15 — наименьшее целое число красоты 4.

Вам необходимо найти наименьшее целое положительное число красоты n .

Input

В единственной строке записано целое число n ($1 \leq n \leq 10^5$).

Output

Выведите искомое число по модулю $(10^9 + 9)$.

Examples

stdin	stdout
3	9
4	15

Problem G. Нанобудильники

Input file: **stdin**
Output file: **stdout**
Time limit: 1 секунда
Memory limit: 256 мегабайт

Будильник с боем — в заданное время бьёт вас до полного пробуждения.

На столе старого часовщика лежат n остановившихся нанобудильников, пронумерованных натуральными числами от 1 до n . Нанобудильники измеряют время в часах, причём в одном часе миллион минут, а каждая минута длится миллион секунд. Для отладки механизмов часовщик должен синхронизировать время на всех нанобудильниках. Для этого он передвигает стрелки *вперед* на некоторое время (возможно, нулевое). Величину такого передвижения назовем временем перевода.

Ваша задача — подсчитать *наименьшее* суммарное время перевода, необходимое для того, чтобы все нанобудильники показывали одинаковое время.

Input

В первой строке записано единственное целое число n — количество нанобудильников ($2 \leq n \leq 10^5$). В каждой i -ой из n следующих строк указано время h, m, s , которое показывают i -ые часы. Целые числа h, m и s указывают количество часов, минут и секунд соответственно ($0 \leq h < 12, 0 \leq m < 10^6, 0 \leq s < 10^6$).

Output

В единственной строке запишите через пробел три целых числа h, m и s — наименьшее суммарное время перевода, где h, m и s — целые числа, указывающие количество часов, минут и секунд соответственно ($0 \leq m < 10^6, 0 \leq s < 10^6$).

Examples

stdin	stdout
2 10 0 0 3 0 0	5 0 0
3 11 999999 999999 0 0 0 11 999999 999999	0 0 2

Problem H. Обед

Input file: `stdin`
Output file: `stdout`
Time limit: 1 секунда
Memory limit: 256 мегабайт

Болото имеет вид узкой полосы длины n . Поверхность болота сплошь покрыта листьями водяной лилии размера 1, пронумерованными от 1 до n , на каждом из которых расположилась муха. На одном из листьев вместо мухи сидит лягушонок Квайт, и у него начинается обед. Он может прыгнуть на соседний лист или перепрыгнуть через один лист в любую сторону. Приземлившись, он съест сидящую там муху. Квайт уже большой лягушонок, а листья не очень надежные, поэтому, когда он прыгает, лист, на котором он находился, начинает тонуть.

Для того чтобы насытиться, Квайту необходимо съесть всех мух, начав свой путь с листа под номером s и закончив на листе под номером f . При прыжке на соседний лист Квайт тратит больше энергии, чем при прыжке через лист.

Необходимо спланировать прыжки лягушонка так, чтобы он смог насытиться, затратив минимальное количество прыжков на соседние листья.

Input

В единственной строке записаны три целых числа n, s, f ($2 \leq n \leq 10\,000, 1 \leq s, f \leq n$) — количество листьев на болоте, номер стартового и номер финишного листьев.

Output

Выведите минимальное количество прыжков на соседние листья, которое потребуется для насыщения лягушонка. Если насытиться не получится, выведите единственное число -1 .

Examples

<code>stdin</code>	<code>stdout</code>
4 1 2	1

Problem I. Бухгалтерская система счисления

Input file: stdin
Output file: stdout
Time limit: 5 секунд
Memory limit: 256 мегабайт

Новейшая Бухгалтерская Система Счисления — самая бухгалтерская во всём мире. Её автор Цейзенпок — лучший специалист соответствующего министерства. Любое целое положительное число n в этой системе по основанию m представляется в виде суммы m слагаемых:

$$n = C_{x_m}^m + C_{x_{m-1}}^{m-1} + C_{x_{m-2}}^{m-2} + \dots + C_{x_1}^1,$$

причем x_1, x_2, \dots, x_m — целые числа такие, что $0 \leq x_1 < x_2 < \dots < x_m$. Числа $C_k^m = \frac{k!}{m!(k-m)!}$ наши специалисты называют бухгалтерскими коэффициентами. Каждое число n в этой системе записывается в виде $n = \overline{(x_m) \dots (x_2)(x_1)}$, причём считается, что $0! = 1$ и $C_k^m = 0$, если $m > k$. Например, число 9 в бухгалтерской системе по основанию 3 записывается в виде **(4)(3)(2)**, так как $9 = C_4^3 + C_3^2 + C_2^1$, а число 1 в этой системе по основанию 2 выглядит так: **(2)(0)**, поскольку $1 = C_2^2 + C_0^1$.

Вам необходимо составить программу, которая находит представление целого положительного числа n в бухгалтерской системе счисления по основанию m .

Input

В единственной строке записаны два целых числа n и m ($1 \leq n \leq 10^{16}$, $2 \leq m \leq 1000$).

Output

Единственная строка должна содержать последовательность из m разделенных пробелом целых чисел x_m, \dots, x_2, x_1 , образующих запись числа n в бухгалтерской системе счисления. Число x_m является первой (слева) цифрой в записи числа n , а x_1 — его последней цифрой.

Examples

stdin	stdout
9 3	4 3 2
5 2	3 2

Problem J. Формула Цейзенпока

Input file: **stdin**
Output file: **stdout**
Time limit: 1 секунд
Memory limit: 256 мегабайт

Ученый Цейзенпок с планеты `i1c5l` стал известен на всю Вселенную благодаря своему недавнему открытию — формуле Цейзенпока. У этой формулы всего три аргумента: n , k и m , а значением этой формулы является число сочетаний из n по k по модулю m .

Пока вся Вселенная думает, для каких целей применима данная формула, нам требуется автоматизировать её вычисление.

Input

Входной файл содержит три целых числа n, k, m , разделенных пробелами ($1 \leq n \leq 10^{18}$, $0 \leq k \leq n$ и $2 \leq m \leq 1\,000\,000$).

Output

Выведите значение формулы для данных n, k, m .

Examples

<code>stdin</code>	<code>stdout</code>
2 1 3	2
4 2 5	1

Problem K. Мы делили апельсин

Input file: **stdin**
Output file: **stdout**
Time limit: 1 секунда
Memory limit: 256 мегабайт

На далекой планете `ilc51` обществом из n человек собран урожай в k апельсинов. Теперь стоит задача дележа урожая.

В этом обществе принята цветовая дифференциация штанов с элементами демократии, поэтому дележ урожая происходит следующим образом. В зависимости от цвета штанов каждый человек получает ранг от 1 до n . У всех людей штаны разного цвета. Далее человек с рангом 1 объявляет свое решение: кому и сколько апельсинов достанется. После этого все n человек голосуют «за» или «против». Если хотя бы половина людей проголосует «за», то решение объявляется принятым, в противном случае, человека, внесшего предложение, изгоняют из общества, а право объявить предложение переходит к человеку с рангом 2, и процедура повторяется.

При принятии решений и голосовании каждый человек действует оптимально для себя: пытается получить как можно больше апельсинов и при вариантах с равным количеством апельсинов стремится, чтобы в обществе осталось как можно меньше людей. Если человека изгоняют из общества, то считается, что он получил отрицательное число апельсинов. Из всех оптимальных решений человек может выбрать любое. Каждый человек знает, что остальные тоже действуют оптимально, руководствуясь теми же принципами.

Но один из этих людей обладает уникальными штанами-хамелеонами. Эти штаны могут по желанию хозяина принимать один из m цветов. Никакой из этих цветов не встречается у остальных людей.

Обладатель хамелеоновых штанов точно знает, какой ранг он получит для каждого цвета своих штанов. Ему осталось выяснить минимальное и максимальное количество апельсинов, которое он получит при каждом из цветов штанов.

Input

В первой строке целые числа n , k и m ($1 \leq n, k \leq 10^9$, $1 \leq m \leq 10^5$) — количество людей, апельсинов и возможных цветов штанов-хамелеонов соответственно.

Во второй строке m целых чисел a_1, a_2, \dots, a_m ($1 \leq a_i \leq n$), где a_i — ранг обладателя хамелеоновых штанов при цвете i .

Output

Для каждого a_i на отдельной строке вывести минимальное и максимальное количество апельсинов, которые получит обладатель штанов цвета i или «-1 -1» (без кавычек), если его изгонят из общества.

Examples

stdin	stdout
2 10 2 1 2	10 10 0 0
3 1 2 1 3	0 0 1 1

Note

В первом примере человек первого ранга забирает все апельсины себе и голосует «за».

Во втором примере человек с рангом 1 должен отдать апельсин человеку третьего ранга. Если первый заберет апельсин себе, то остальные проголосуют "против". Если первый отдаст апельсин второму, то второй все равно будет «против», т.к. знает, что, изгнав первого, он тоже получит апельсин, но людей в обществе останется меньше. Так как третий тоже будет «против», то этот вариант не подходит для первого.

Problem L. Бассейн счастья

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 256 мегабайт

В Казани к предстоящему Чемпионату мира по водным видам спорта построен бассейн на N дорожек. По некоторым из этих дорожек уже плавают люди.

Наблюдая за людьми, татарские учёные разделили дорожки на счастливые и несчастливые. Несчастливой дорожка считается в том случае, если по ней плавает максимальное количество людей. То есть, нет другой дорожки, по которой плавало бы больше людей, чем по этой. По несчастливым дорожкам плавают несчастные люди. Все остальные дорожки считаются счастливыми. По счастливым дорожкам плавают счастливые люди.

Учёные решили сделать как можно больше людей счастливыми. Для этого они согласовали с администрацией бассейна возможность переместить одного человека на одну из *соседних* дорожек, если это потребуется. При этом с первой дорожки пловца можно переместить только на вторую, а с последней только на предпоследнюю.

Input

В первой строке записано единственное целое число N — количество дорожек в бассейне ($3 \leq N \leq 10^5$). Во второй строке N целых чисел p_i , разделённых пробелом — распределение людей по дорожкам, где число p_i задаёт количество пловцов на i -ой дорожке ($0 \leq p_i \leq 10^5$).

Output

Выведите единственное число — минимально возможное число несчастных пловцов.

Examples

стандартный ввод	стандартный вывод
3 1 3 5	5
4 1 0 1 0	2