Problem A. Спортивное домино

Input file:	`game.in`
Output file:	`game.out`
Time limit:	3 секунды
Memory limit:	64 мебибайта

На прошедших в Лондоне олимпийских играх делегация Байтландии не участвовала — дело в том, что в Байтландии популярны несколько менее классические виды спорта, и соревнования даже по вполне обычным виды спорта проводятся по необычным правилам.

Пристрастие байтландцев к необычным видам спорта коснулось и спортивного покера: игра в покер идёт не картами, а фишками байтландского домино. Соответственно игра называется «спортивное домино», хотя к привычному нам домино правила игры отношения не имеют.

Каждая фишка байтландского домино, подобно фишке обычного домино, состоит из двух квадратных половинок. Каждая из половинок окрашена в некоторый цвет, кроме того, на каждой половинке написано некоторое число (возможно, отрицательное). При этом набор является полным в следующем смысле: если на одной из фишек есть число A цвета C_1, а на какой-то другой — B цвета C_2, то в наборе обязательно присутствует ровно одна, соответствующая этим числам, фишка (составленная из клетки цвета C_1 с числом A и клетки цвета C_2 с числом B).

Например, если в наборе имеются фишки $(2_{red} : 2_{green})$ и $(3_{red} : -4_{blue})$, то в нем должны быть ещё и фишки $(2_{red} : 3_{red})$, $(2_{green} : 3_{red})$, $(2_{red} : -4_{blue})$, $(2_{green} : -4_{blue})$.

Перед партией в «спортивное домино» судья подсчитал суммы чисел на всех фишках, при этом вышло ровно N отрицательных, L равных нулю и K положительных. Впоследствии участниками матча была подана апелляция на то, что комплект был неполным.

Разбирающую этот случай коллегию судей заинтересовал вопрос, существует ли какой-либо набор целых чисел, участвующих в игре (с некоторым количеством цветов), для которого это возможно. Каждое из чисел по модулю не должно превосходить 10^6.

Input

Выходной файл состоит из одной строки, в которой заданы три целых положительных числа N, L и K, причем $N + L + K \leq 90$.

Output

Если искомые наборы существуют, то в выходной файл нужно вывести любой из них в одну строчку, все числа через пробел. Если же набора не существует, то вывести строку "`Impossible`".

Examples

game.in	game.out
1 1 1	1 2 -2
1 1 2	Impossible

Problem B. Вперёд к победе!

Input file:	goforwin.in
Output file:	goforwin.out
Time limit:	2 секунды
Memory limit:	64 мебибайта

Соревнования по стендовой стрельбе в Байтландии проводятся так: перед каждой попыткой случайным образом запускается одна из N мишеней различной формы и размера, по которой и стреляет стрелок.

Во время финала чемпионата Байтландии для победы действующему чемпиону надо было обязательно реализовать последнюю попытку. При этом известно, что в X из N разновидностей мишеней чемпион в последней попытке попадает с вероятностью 1, а в оставшиеся $N-X$ — с вероятностью 0.5 (линейные размеры этих мишеней поменьше). Вычислите вероятность того, что чемпион сохранит своё звание.

Input

В первой строке входного файла заданы два целых числа N и X ($1 \le N \le 50,\ 0 \le X < N$).

Output

В единственной строке выведите вероятность того, что чемпион в последней попытке попадёт в мишень, в виде несократимой дроби. Максимально точно следуйте формату вывода, представленному в примере.

Example

goforwin.in	goforwin.out
20 10	3/4

Problem C. Графомарафон

Input file:	graphon.in
Output file:	graphon.out
Time limit:	2 секунды
Memory limit:	64 мебибайта

Правила марафона в Байтландии также модернизированы. Соответствующий неолимпийский вид спорта называется графомарафоном или графоном.

Забег проходит по улицам города, стартует на одной из площадей, финиширует на другой. При этом минимальный путь между площадями по улицам города равен длине дистанции забега. Таких путей может быть несколько (существование как минимум одного такого пути гарантируется).

Город состоит из N площадей, занумерованных трёхзначными числами от 100 до $N+99$. Некоторые площади соединены между собой улицами, при этом никакая площадь не может быть соединена сама с собой и любые две площади соединены не более, чем одной улицей. Для каждой улицы известна её длина. Бежать по улице участники могут как в одну, так и в другую сторону.

Марафон стартует на площади 100, финиширует на площади $N + 99$. Требуется найти лексикографически минимальную последовательность площадей от старта до финиша, задающую кратчайший путь.

Input

Входной файл состоит из не более, чем 100 тестовых примеров. В первой строке каждого тестового примера заданы два целых числа N и M — количество площадей и количество улиц ($2 \le N \le 100,\ 1 \le M \le 500$) соответственно. В последующих M строках заданы по три целых числа a_i, b_i, l_i — номера площадей, соединённых улицей и длина соответствующей улицы ($100 \le a_i, b_i \le 99 + N,\ a_i \ne b_i,\ 1 \le l_i \le 100$). Входной файл завершается тестовым примером с $M = N = 0$, обрабатывать который не нужно.

Output

Для каждого тестового примера в отдельной строке выведите последовательность номеров площадей, задающих лексикографически наименьший кратчайший путь от старта до финиша.

Example

graphon.in	graphon.out
3 3 101 100 2 102 100 5 102 101 3 0 0	100 101 102

Problem D. Наибольшая сумма

Input file:	greatestsum.in
Output file:	greatestsum.out
Time limit:	2 секунды
Memory limit:	64 мебибайта

Один из игроков в «спортивное домино», регулярно проигрывавший в результате невезения, решил построить формулу, определяющую его возможные результаты. После анализа всех факторов появилось следующее правило.

Пусть $f(x) = (1 - sign^2(x) - sign(x)) \cdot ((a|x| + b) \mod p)$.

Определим бесконечную последовательность A следующим образом: $A_0 = k$, $A_i = f(A[i-1])$ для всех $i > 0$. A_i и есть ожидаемая сумма выигрыша в i-й после начала наблюдений день.

Игрок хочет выяснить, в течение какого периода времени ему стоит активно играть. Более формально, рассмотрим все конечные подпоследовательности A (конечная подпоследовательность последовательности A — это последовательность, составленная из одного или нескольких идущих подряд элементов последовательности A)., которые образуются одним или несколькими подряд идущими элементами последовательности A.

Среди этих подпоследовательностей он просит Вас найти такую, сумма элементов которой максимальна и вывести соответствующую сумму.

Input

В первой строке входного файла заданы четыре целых числа a, b, p и k ($1 \le p \le 1000,\ 0 \le a, b, k \le 1000$).

Output

Выведите максимальную возможную сумму элементов подпоследовательности. Если эта сумма бесконечно велика, то выведите "Infinite".

Example

greatestsum.in	greatestsum.out
0 0 1 1	1
1 1 2 2	Infinite

Problem E. Зелёный биатлон

Input file:	greenbiathlon.in
Output file:	greenbiathlon.out
Time limit:	2 секунды
Memory limit:	64 мебибайта

По аналогии с хоккеем на траве в Байтландии изобрели «зелёный биатлон». Задача участника — как можно быстрее преодолеть дистанцию, при этом точно отстрелявшись из арбалета на каждом огневом рубеже.

Огневые рубежи расположены вдоль одной прямой на различном расстоянии от точки старта, при этом рубежи могут располагаться по разные стороны от старта. Порядок прохождения огневых рубежей определяет сам участник. Гонка заканчивается в момент завершения последней стрельбы.

В соревнованиях по «зелёному биатлону» решил принять участие чемпион Байтландии по обычному биатлону. Проведённые тренерами измерения показали следующее: время, которое он затрачивает на каждом огневом рубеже для того, чтобы поразить все мишени, равно 1. Время, которое чемпион затрачивает на преодоление одного метра дистанции, равно $1 + S_{left}$, где S_{left} — количество оставшихся огневых рубежей (после каждой стрельбы нагрузка на участника уменьшается, и бежать становится проще).

По заданным расстояниям от точки старта и огневых рубежей до начала стадиона вычислите, за какое минимальное время чемпион сможет пройти дистанцию

Input

В первой строке входного файла задано целое число T ($1 \le T \le 60$) — количество тестовых примеров. Далее следуют тестовые примеры. Каждый пример задан в одной строке, в которой сначала идёт целое число N ($1 \le N \le 50$) — количество огневых рубежей. Далее заданы $N + 1$ целых положительных чисел, не превосходящих 10^5, первые N из которых задают расстояние от начала стадиона до каждого из огневых рубежей, а последнее задаёт расстояние от начала стадиона до точки старта.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — минимальное время, за которое чемпион пройдёт дистанцию.

Example

greenbiathlon.in	greenbiathlon.out
2 3 100 200 300 100 4 15 25 45 85 40	503 279

Problem F. Атлетические упражнения

Input file:	gym.in
Output file:	gym.out
Time limit:	2 секунды
Memory limit:	64 мебибайта

Соревнования по толканию ядра в Байтландии проводятся на площадке в форме выпуклого многоугольника. Точка, на которой размещается атлет, выбирается таким образом, что при её соединении со всеми вершинами площадки получаются N треугольных секторов равной площади. При каждой новой попытке участник обязан попасть ядром в новый сектор; таким образом, вводится естественное ограничение на число подходов.

Вам заданы вершины площадки. Требуется построить точку броска, то есть выбрать такую точку внутри площадки, при соединении которой с её вершинами площади любых двух из N полученных треугольников отличаются не более, чем на 10^{-3}.

Input

В первой строке входного файла записано целое число N — количество вершин выпуклого многоугольника, являющегося контуром площадки ($3 \le N \le 10^5$). Далее — каждая пара с новой строки — координаты всех вершин многоугольника, заданных в порядке обхода по часовой стрелке. Координаты каждой вершины задаются парой вещественных чисел x_i и y_i, где $-10^4 < x_i, y_i < 10^4, 3 \le i \le N$.

Output

В выходной файл нужно выдать через пробел два вещественных числа — координаты точки броска с максимально возможной точностью. Если для данного стадиона точки с соответствующими свойствами не существует, выведите "`Impossible`" (без кавычек).

Examples

gym.in	gym.out
4 1 1 1 -1 -1 -1 -1 1	0.0000 0.0000
4 0 0 0.25 0.4330127 0.75 0.9330127 1 0	Impossible