Codeforces Round #188 (Div. 2)

A. Even Odds

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Being a nonconformist, Volodya is displeased with the current state of things, particularly with the order of natural numbers (natural number is positive integer number). He is determined to rearrange them. But there are too many natural numbers, so Volodya decided to start with the first n. He writes down the following sequence of numbers: firstly all odd integers from 1 to n (in ascending order), then all even integers from 1 to n (also in ascending order). Help our hero to find out which number will stand at the position number k.

Input

The only line of input contains integers n and k ($1 \le k \le n \le 10^{12}$).

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

Print the number that will stand at the position number k after Volodya's manipulations.

Examples

input
10 3
output
5

input
7 7
output
6

Note

In the first sample Volodya's sequence will look like this: {1, 3, 5, 7, 9, 2, 4, 6, 8, 10}. The third place in the sequence is therefore occupied by the number 5.

B. Strings of Power

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Volodya likes listening to heavy metal and (occasionally) reading. No wonder Volodya is especially interested in texts concerning his favourite music style.

Volodya calls a string powerful if it starts with "heavy" and ends with "metal". Finding all powerful substrings (by substring Volodya means a subsequence of consecutive characters in a string) in a given text makes our hero especially joyful. Recently he felt an enormous fit of energy while reading a certain text. So Volodya decided to count all powerful substrings in this text and brag about it all day long. Help him in this difficult task. Two substrings are considered different if they appear at the different positions in the text.

For simplicity, let us assume that Volodya's text can be represented as a single string.

Input
Input contains a single non-empty string consisting of the lowercase Latin alphabet letters. Length of this string will not be greater than 10^6 characters.

Output
Print exactly one number — the number of powerful substrings of the given string.

Please, do not use the %lld specifier to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the %I64d specifier.

Examples

input
heavymetalisheavymetal
output
3

input
heavymetalismetal
output
2

input
trueheavymetalissotruewellitisalsosoheavythatyoucanalmostfeeltheweightofmetalonyou
output
3

Note
In the first sample the string "heavymetalisheavymetal" contains powerful substring "heavymetal" twice, also the whole string "heavymetalisheavymetal" is certainly powerful.

In the second sample the string "heavymetalismetal" contains two powerful substrings: "heavymetal" and "heavymetalismetal".

C. Perfect Pair

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let us call a pair of integer numbers m-*perfect*, if at least one number in the pair is greater than or equal to m. Thus, the pairs (3, 3) and (0, 2) are 2-perfect while the pair (-1, 1) is not.

Two integers x, y are written on the blackboard. It is allowed to erase one of them and replace it with the sum of the numbers, $(x + y)$.

What is the minimum number of such operations one has to perform in order to make the given pair of integers m-perfect?

Input

Single line of the input contains three integers x, y and m ($-10^{18} \le x, y, m \le 10^{18}$).

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preffered to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

Print the minimum number of operations or "-1" (without quotes), if it is impossible to transform the given pair to the m-perfect one.

Examples

input
1 2 5
output
2

input
-1 4 15
output
4

input
0 -1 5
output
-1

Note

In the first sample the following sequence of operations is suitable: (1, 2) $\rightarrow$ (3, 2) $\rightarrow$ (5, 2).

In the second sample: (-1, 4) $\rightarrow$ (3, 4) $\rightarrow$ (7, 4) $\rightarrow$ (11, 4) $\rightarrow$ (15, 4).

Finally, in the third sample x, y cannot be made positive, hence there is no proper sequence of operations.

D. Ants

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

It has been noted that if some ants are put in the junctions of the graphene integer lattice then they will act in the following fashion: every minute at each junction (x, y) containing at least four ants a group of four ants will be formed, and these four ants will scatter to the neighbouring junctions $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$ — one ant in each direction. No other ant movements will happen. Ants never interfere with each other.

Scientists have put a colony of n ants into the junction (0, 0) and now they wish to know how many ants will there be at some given junctions, when the movement of the ants stops.

Input

First input line contains integers n ($0 \le n \le 30000$) and t ($1 \le t \le 50000$), where n is the number of ants in the colony and t is the number of queries. Each of the next t lines contains coordinates of a query junction: integers x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$). Queries may coincide.

It is guaranteed that there will be a certain moment of time when no possible movements can happen (in other words, the process will eventually end).

Output

Print t integers, one per line — the number of ants at the corresponding junctions when the movement of the ants stops.

Examples

input
1 3 0 1 0 0 0 -1
output
0 1 0

input
6 5 0 -2 0 -1 0 0 0 1 0 2
output
0 1 2 1 0

Note

In the first sample the colony consists of the one ant, so nothing happens at all.

In the second sample the colony consists of 6 ants. At the first minute 4 ants scatter from (0, 0) to the neighbouring junctions. After that the process stops.

E. Balance

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A system of n vessels with water is given. Several pairs of vessels are connected by tubes with transfusion mechanisms. One may transfer an integer amount of liters of water between two vessels connected by such tube (tube works in both directions). There might be multiple tubes between two vessels. Total number of tubes equals e. Volume of each vessel equals v liters. Of course, the amount of the water in any vessel cannot exceed v liters in the process of transfusions.

Given the initial amounts a_i of water in the vessels and the desired amounts b_i find a sequence of transfusions that deals with the task. Total number of transfusions must not exceed $2 \cdot n^2$.

Input

First line of the input contains integers n, v, e ($1 \le n \le 300$, $1 \le v \le 10^9$, $0 \le e \le 50000$).

Next two lines contain n integers each: initial a_i and the desired amounts b_i of water in corresponding vessels ($0 \le a_i, b_i \le v$).

Next e lines describe one tube each in the format x y ($1 \le x, y \le n, x \ne y$) for a tube between vessels number x and y. There might be multiple tubes between two vessels. You may assume that vessels are numbered from 1 to n in some way.

Output

Print "NO" (without quotes), if such sequence of transfusions does not exist.

Otherwise print any suitable sequence in the following format. On the first line print the total number of transfusions k (k should not exceed $2 \cdot n^2$). In the following k lines print transfusions in the format x y d (transfusion of d liters from the vessel number x to the vessel number y, x and y must be distinct). For all transfusions d must be a non-negative integer.

Examples

input
2 10 1 1 9 5 5 1 2
output
1 2 1 4

input
2 10 0 5 2 4 2
output
NO

input
2 10 0 4 2 4 2
output
0