A. 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) $\longrightarrow$ (3, 2) $\longrightarrow$ (5, 2).

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

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

B. 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.

C. 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 ($1 \leq n \leq 300$, $1 \leq v \leq 10^9$, $0 \leq e \leq 50000$).

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

Next e lines describe one tube each in the format x y ($1 \leq x, y \leq n, x \neq y$) 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
```

D. Game with Powers

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

Vasya and Petya wrote down all integers from 1 to n to play the "powers" game (n can be quite large; however, Vasya and Petya are not confused by this fact).

Players choose numbers in turn (Vasya chooses first). If some number x is chosen at the current turn, it is forbidden to choose x or all of its other positive integer powers (that is, x^2, x^3, $\ldots$) at the next turns. For instance, if the number 9 is chosen at the first turn, one cannot choose 9 or 81 later, while it is still allowed to choose 3 or 27. The one who cannot make a move loses.

Who wins if both Vasya and Petya play optimally?

Input

Input contains single integer n ($1 \le n \le 10^9$).

Output

Print the name of the winner — "`Vasya`" or "`Petya`" (without quotes).

Examples

input
```
1
```
output
```
Vasya
```

input
```
2
```
output
```
Petya
```

input
```
8
```
output
```
Petya
```

Note

In the first sample Vasya will choose 1 and win immediately.

In the second sample no matter which number Vasya chooses during his first turn, Petya can choose the remaining number and win.

E. Princess and Her Shadow

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

Princess Vlada enjoys springing in the meadows and walking in the forest. One day — wonderful, sunny day — during her walk Princess found out with astonishment that her shadow was missing! "Blimey!", — she thought and started her search of the shadow in the forest.

Normally the Shadow is too lazy and simply sleeps under the Princess. But at this terrifically hot summer day she got bored of such a dull life, so she decided to play with Vlada.

The forest, where our characters entertain themselves, may be represented as a set of integer cells in the plane, where the Shadow and the Princess can move only up, down, left and right by 1. Some cells (as it happens in decent forests) are occupied by trees. The Shadow and the Princess are not allowed to enter a cell occupied by a tree. Unfortunately, these are the hard times for the forest, so there are very few trees growing here...

At first the Princess was walking within the cell (v_x, v_y), while the Shadow hid from the Princess in the cell (s_x, s_y). The Princess, The Shadow and the trees are located in the different cells.

The Shadow is playing with the Princess. As soon as the Princess moves by 1 in some direction, the Shadow simultaneously flies by 1 in the same direction, if it is possible (if the cell to fly to is not occupied by some tree); otherwise, the Shadow doesn't move. The Shadow is very shadowy, so our characters do not interfere with each other.

We say that the Shadow is caught by the Princess if after some move both of them are located in the same cell. Vlada managed to catch her Shadow! Can you?

Input

First line of the input contains the coordinates of the characters v_x, v_y, s_x, s_y and the number of trees m ($0 \le m \le 400$). The following m lines contain the coordinates of the trees.

All the coordinates are integers between -100 and 100, inclusive. The Princess, The Shadow and the trees are located in the different cells.

Output

If it is impossible for the Princess to catch the Shadow, print "`-1`" (without quotes).

Otherwise print a sequence of characters "`L`", "`R`", "`D`", "`U`", corresponding to the Princess's moves, following which she will be able to catch the Shadow at some turn (`L` — move to the left, `R` — to the right, `U` — up, `D` — down; axis x is directed to the right, y — up).

The number of characters (that is, the number of moves) must not exceed 10^6. All the Princess's moves should be correct, that is must not lead to the cell where a tree grows. It is allowed for the Princess and the Shadow to occupy the same cell before the last turn.

Examples

input

```
0 0 1 0 1
0 1
```

output

```
LLUR
```

input

```
5 0 3 0 8
2 -1
2 0
2 1
3 -1
4 1
4 0
3 1
4 -1
```

output

```
-1
```

input

```
3 2 1 1 3
0 1
1 0
0 0
```

output

```
DLL
```

Note

Below the pictures for the samples are given (Princess, Shadow and the trees are colored in pink, gray and black correspondingly; the blue dot marks the lattice center).

In the first case the Princess may make two left steps, one step upwards and one right step:

In the following case the Princess cannot catch the Shadow:

In the last sample the Princess may make two left steps and one down step (in any order):