Codeforces Beta Round #71

A. Bus Game

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

After Fox Ciel won an onsite round of a programming contest, she took a bus to return to her castle. The fee of the bus was 220 yen. She met Rabbit Hanako in the bus. They decided to play the following game because they got bored in the bus.

- Initially, there is a pile that contains x 100-yen coins and y 10-yen coins.
- They take turns alternatively. Ciel takes the first turn.
- In each turn, they must take exactly 220 yen from the pile. In Ciel's turn, if there are multiple ways to take 220 yen, she will choose the way that contains the maximal number of 100-yen coins. In Hanako's turn, if there are multiple ways to take 220 yen, she will choose the way that contains the maximal number of 10-yen coins.
- If Ciel or Hanako can't take exactly 220 yen from the pile, she loses.

Determine the winner of the game.

Input

The first line contains two integers x ($0 \le x \le 10^6$) and y ($0 \le y \le 10^6$), separated by a single space.

Output

If Ciel wins, print "`Ciel`". Otherwise, print "`Hanako`".

Sample test(s)

input
2 2
output
Ciel

input
3 22
output
Hanako

Note

In the first turn (Ciel's turn), she will choose 2 100-yen coins and 2 10-yen coins. In the second turn (Hanako's turn), she will choose 1 100-yen coin and 12 10-yen coins. In the third turn (Ciel's turn), she can't pay exactly 220 yen, so Ciel will lose.

B. Colorful Field

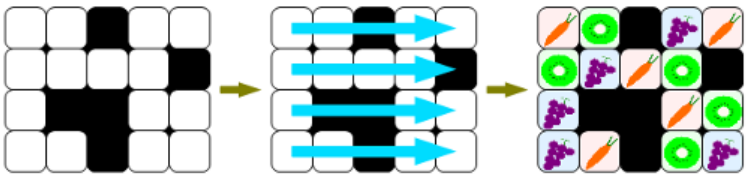
time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel saw a large field while she was on a bus. The field was a $n \times m$ rectangle divided into 1×1 cells. Some cells were wasteland, and other each cell contained crop plants: either carrots or kiwis or grapes.

After seeing the field carefully, Ciel found that the crop plants of each cell were planted in following procedure:

- Assume that the rows are numbered 1 to n from top to bottom and the columns are numbered 1 to m from left to right, and a cell in row i and column j is represented as (i, j).
- First, each field is either cultivated or waste. Crop plants will be planted in the cultivated cells in the order of $(1, 1) \rightarrow ... \rightarrow (1, m) \rightarrow (2, 1) \rightarrow ... \rightarrow (2, m) \rightarrow ... \rightarrow (n, 1) \rightarrow ... \rightarrow (n, m)$. Waste cells will be ignored.
- Crop plants (either carrots or kiwis or grapes) will be planted in each cell one after another cyclically. Carrots will be planted in the first cell, then kiwis in the second one, grapes in the third one, carrots in the forth one, kiwis in the fifth one, and so on.

The following figure will show you the example of this procedure. Here, a white square represents a cultivated cell, and a black square represents a waste cell.



Now she is wondering how to determine the crop plants in some certain cells.

Input

In the first line there are four positive integers n, m, k, t ($1 \le n \le 4 \cdot 10^4$, $1 \le m \le 4 \cdot 10^4$, $1 \le k \le 10^3$, $1 \le t \le 10^3$), each of which represents the height of the field, the width of the field, the number of waste cells and the number of queries that ask the kind of crop plants in a certain cell.

Following each k lines contains two integers a, b ($1 \le a \le n$, $1 \le b \le m$), which denotes a cell (a, b) is waste. It is guaranteed that the same cell will not appear twice in this section.

Following each t lines contains two integers i, j ($1 \le i \le n$, $1 \le j \le m$), which is a query that asks you the kind of crop plants of a cell (i, j).

Output

For each query, if the cell is waste, print `Waste`. Otherwise, print the name of crop plants in the cell: either `Carrots` or `Kiwis` or `Grapes`.

Sample test(s)

input

```
4 5 5 6
4 3
1 3
3 3
2 5
3 2
1 3
1 4
2 3
2 4
1 1
1 1
```

output

```
Waste
Grapes
Carrots
Kiwis
Carrots
Carrots
```

Note

The sample corresponds to the figure in the statement.

C. Beaver

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

After Fox Ciel got off a bus, she found that the bus she was on was a wrong bus and she lost her way in a strange town. However, she fortunately met her friend Beaver Taro and asked which way to go to her castle. Taro's response to her was a string s, and she tried to remember the string s correctly.

However, Ciel feels n strings $b_1, b_2, \ldots, b_n$ are really boring, and unfortunately she dislikes to remember a string that contains a boring substring. To make the thing worse, what she can remember is only the contiguous substring of s.

Determine the longest contiguous substring of s that does not contain any boring string, so that she can remember the longest part of Taro's response.

Input

In the first line there is a string s. The length of s will be between 1 and 10^5, inclusive.

In the second line there is a single integer n ($1 \le n \le 10$). Next n lines, there is a string b_i ($1 \le i \le n$). Each length of b_i will be between 1 and 10, inclusive.

Each character of the given strings will be either a English alphabet (both lowercase and uppercase) or a underscore (`'_'`) or a digit. Assume that these strings are case-sensitive.

Output

Output in the first line two space-separated integers len and pos: the length of the longest contiguous substring of s that does not contain any b_i, and the first position of the substring (0-indexed). The position pos must be between 0 and $|s|$ - len inclusive, where $|s|$ is the length of string s.

If there are several solutions, output any.

Sample test(s)

input

```
Go_straight_along_this_street
5
str
long
tree
biginteger
ellipse
```

output

```
12 4
```

input

```
IhaveNoIdea
9
I
h
a
v
e
N
o
I
d
```

output

```
0 0
```

input

```
unagioisii
2
ioi
unagi
```

output

```
5 5
```

Note

In the first sample, the solution is `traight_alon`.

In the second sample, the solution is an empty string, so the output can be «`0 0`», «`0 1`», «`0 2`», and so on.

In the third sample, the solution is either `nagio` or `oisii`.

D. Password

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

Finally Fox Ciel arrived in front of her castle!

She have to type a password to enter her castle. An input device attached to her castle is a bit unusual.

The input device is a $1 \times n$ rectangle divided into n square panels. They are numbered 1 to n from left to right. Each panel has a state either ON or OFF. Initially all panels are in the OFF state. She can enter her castle if and only if x_1-th, x_2-th, ..., x_k-th panels are in the ON state and other panels are in the OFF state.

She is given an array $a_1, ..., a_l$. In each move, she can perform the following operation: choose an index i ($1 \le i \le l$), choose consecutive a_i panels, and flip the states of those panels (i.e. ON $\longrightarrow$ OFF, OFF $\longrightarrow$ ON).

Unfortunately she forgets how to type the password with only above operations. Determine the minimal number of operations required to enter her castle.

Input

The first line contains three integers n, k and l ($1 \le n \le 10000$, $1 \le k \le 10$, $1 \le l \le 100$), separated by single spaces.

The second line contains k integers $x_1, ..., x_k$ ($1 \le x_1 < x_2 < ... < x_k \le n$), separated by single spaces.

The third line contains l integers $a_1, ..., a_l$ ($1 \le a_i \le n$), separated by single spaces. It is possible that some elements of the array a_i are equal value.

Output

Print the minimal number of moves required to type the password. If it's impossible, print `-1`.

Sample test(s)

input

```
10 8 2
1 2 3 5 6 7 8 9
3 5
```

output

```
2
```

input

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

output

```
-1
```

Note

One possible way to type the password in the first example is following: In the first move, choose 1st, 2nd, 3rd panels and flip those panels. In the second move, choose 5th, 6th, 7th, 8th, 9th panels and flip those panels.

E. Security System

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

Fox Ciel safely returned to her castle, but there was something wrong with the security system of the castle: sensors attached in the castle were covering her.

Ciel is at point $(1, 1)$ of the castle now, and wants to move to point (n, n), which is the position of her room. By one step, Ciel can move from point (x, y) to either $(x + 1, y)$ (rightward) or $(x, y + 1)$ (upward).

In her castle, c^2 sensors are set at points $(a + i, b + j)$ (for every integer i and j such that: $0 \le i < c, 0 \le j < c$).

Each sensor has a count value and decreases its count value every time Ciel moves. Initially, the count value of each sensor is t. Every time Ciel moves to point (x, y), the count value of a sensor at point (u, v) decreases by $(|u - x| + |v - y|)$. When the count value of some sensor becomes **strictly less than** 0, the sensor will catch Ciel as a suspicious individual!

Determine whether Ciel can move from $(1, 1)$ to (n, n) without being caught by a sensor, and if it is possible, output her steps. Assume that Ciel can move to every point even if there is a censor on the point.

Input

In the first line there are five integers n, t, a, b, c ($2 \le n \le 2 \cdot 10^5$, $0 \le t \le 10^{14}$, $1 \le a \le n - c + 1$, $1 \le b \le n - c + 1$, $1 \le c \le n$).

Please do not use the `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use the `cin` stream (also you may use the `%I64d` specificator).

Output

If Ciel's objective is possible, output in first line $2n - 2$ characters that represent her feasible steps, where i-th character is `R` if i-th step is moving rightward, or `U` if moving upward. If there are several solution, output **lexicographically first** one. Character `R` is lexicographically earlier than the character `U`.

If her objective is impossible, output `Impossible`.

Sample test(s)

input
5 25 2 4 1
output
RRUURURU

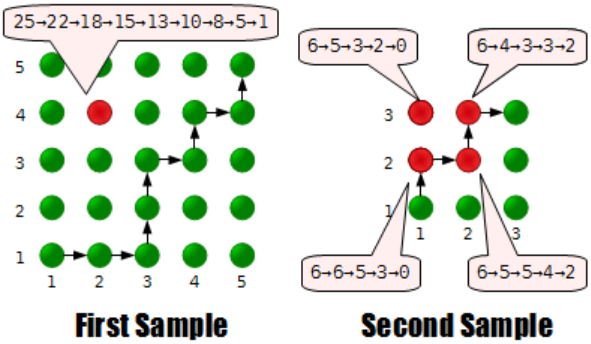
input
3 6 1 2 2
output
URUR

input
3 5 1 2 2
output
Impossible

input
20 492 11 4 8
output
RRRRRRRRRRRRRRRRUUUUURUUUUURUUUUUUUU

Note

The answers for the first sample and the second sample are shown on the picture:



Here, a red point represents a point that contains a sensor.
