Codeforces Round #534 (Div. 1)

A. Grid game

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

You are given a 4x4 grid. You play a game — there is a sequence of tiles, each of them is either 2x1 or 1x2. Your task is to consequently place all tiles from the given sequence in the grid. When tile is placed, each cell which is located in fully occupied row or column is deleted (cells are deleted at the same time independently). You can place tile in the grid at any position, the only condition is that tiles (and tile parts) should not overlap. Your goal is to proceed all given figures and avoid crossing at any time.

Input

The only line contains a string s consisting of zeroes and ones ($1 \le |s| \le 1000$). Zero describes vertical tile, one describes horizontal tile.

Output

Output $|s|$ lines — for each tile you should output two positive integers r, c, not exceeding 4, representing numbers of smallest row and column intersecting with it.

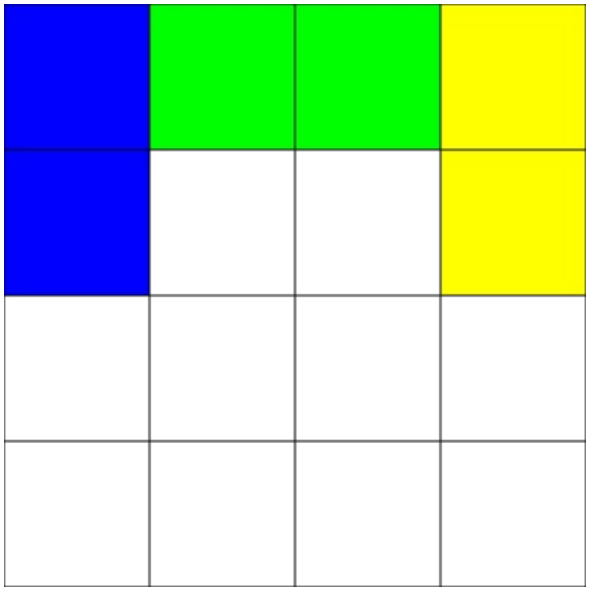
If there exist multiple solutions, print any of them.

Example

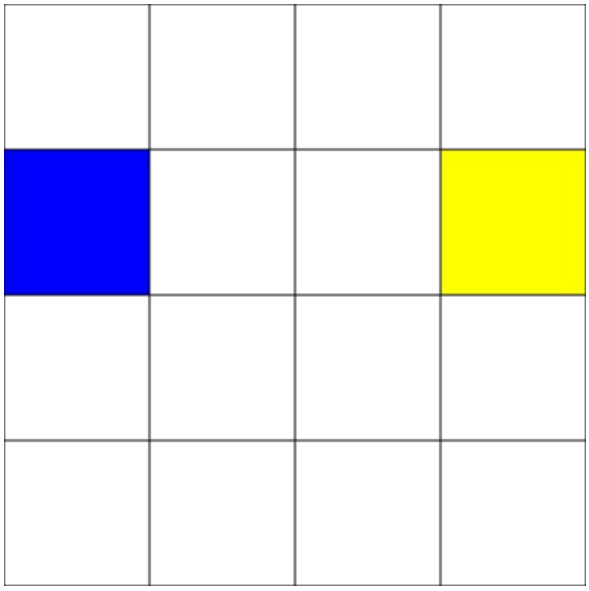
input
010
output
1 1 1 2 1 4

Note

Following image illustrates the example after placing all three tiles:



Then the first row is deleted:



B. Game with modulo

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

This is an interactive problem.

Vasya and Petya are going to play the following game: Petya has some positive integer number a. After that Vasya should guess this number using the following questions. He can say a pair of non-negative integer numbers (x, y). Petya will answer him:

- "x", if $(x \bmod a) \geq (y \bmod a)$.
- "y", if $(x \bmod a) < (y \bmod a)$.

We define $(x \bmod a)$ as a remainder of division x by a.

Vasya should guess the number a using **no more, than 60 questions**.

It's guaranteed that Petya has a number, that satisfies the inequality $1 \leq a \leq 10^9$.

Help Vasya playing this game and write a program, that will guess the number a.

Interaction

Your program should play several games.

Before the start of any game your program should read the string:

- "`start`" (without quotes) — the start of the new game.
- "`mistake`" (without quotes) — in the previous game, you found the wrong answer. Your program should terminate after reading this string and it will get verdict "Wrong answer".
- "`end`" (without quotes) — all games finished. Your program should terminate after reading this string.

After reading the string "`start`" (without quotes) the new game starts.

At the beginning, your program should ask several questions about pairs of non-negative integer numbers (x, y). You can only ask the numbers, that satisfy the inequalities $0 \leq x, y \leq 2 \cdot 10^9$. To ask a question print "? x y" (without quotes). As the answer, you should read one symbol:

- "x" (without quotes), if $(x \bmod a) \geq (y \bmod a)$.
- "y" (without quotes), if $(x \bmod a) < (y \bmod a)$.
- "e" (without quotes) — you asked more than 60 questions. Your program should terminate after reading this string and it will get verdict "Wrong answer".

After your program asked several questions your program should print the answer in form "! a" (without quotes). You should print the number a satisfying the inequalities $1 \leq a \leq 10^9$. It's guaranteed that Petya's number a satisfied this condition. After that, the current game will finish.

We recall that your program can't ask more than 60 questions during one game.

If your program doesn't terminate after reading "`mistake`" (without quotes), "`end`" (without quotes) or "`e`" (without quotes), it can get any verdict, because it will continue reading from closed input. Also, if your program prints answer or question in the incorrect format it can get any verdict, too. **Be careful.**

Don't forget to flush the output after printing questions and answers.

To flush the output, you can use:

- fflush(stdout) in C++.
- System.out.flush() in Java.
- stdout.flush() in Python.
- flush(output) in Pascal.
- See the documentation for other languages.

It's guaranteed that you should play at least 1 and no more than 100 games.

Hacks:

In hacks, you can use only one game. To hack a solution with Petya's number a ($1 \le a \le 10^9$) in the first line you should write a single number 1 and in the second line you should write a single number a.

Example

input
start x x start x x y start x x y y end
output
? 0 0 ? 10 1 ! 1 ? 0 0 ? 3 4 ? 2 5 ! 2 ? 2 4 ? 2 5 ? 3 10 ? 9 1 ! 3

Note

In the first test, you should play 3 games with Petya's numbers 1, 2 and 3.

In the first game, Petya will answer "x" (without quotes) to any question, because $(x \bmod 1) = 0$ for any integer x.

In the second game, if you will ask pair $(0, 0)$, the answer will be "x" (without quotes), because $(0 \bmod 2) \ge (0 \bmod 2)$. But if you will ask pair $(2, 5)$, the answer will be "y" (without quotes), because $(2 \bmod 2) < (5 \bmod 2)$, because $(2 \bmod 2) = 0$ and $(5 \bmod 2) = 1$.

C. Johnny Solving

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

Today is tuesday, that means there is a dispute in JOHNNY SOLVING team again: they try to understand who is Johnny and who is Solving. That's why guys asked Umnik to help them. Umnik gave guys a connected graph with n vertices without loops and multiedges, such that a degree of any vertex is at least 3, and also he gave a number $1 \le k \le n$. Because Johnny is not too smart, he promised to find a simple path with length at least $\frac{n}{k}$ in the graph. In reply, Solving promised to find k simple by vertices cycles with representatives, such that:

- Length of each cycle is at least 3.
- Length of each cycle is not divisible by 3.
- In each cycle must be a representative - vertex, which belongs only to this cycle among all **printed** cycles.

You need to help guys resolve the dispute, for that you need to find a solution for Johnny: a simple path with length at least $\frac{n}{k}$ (n is not necessarily divided by k), or solution for Solving: k cycles that satisfy all the conditions above. If there is no any solution - print -1.

Input

The first line contains three integers n, m and k ($1 \le k \le n \le 2.5 \cdot 10^5, 1 \le m \le 5 \cdot 10^5$)

Next m lines describe edges of the graph in format v, u ($1 \le v, u \le n$). It's guaranteed that $v \ne u$ and all m pairs are distinct.

It's guaranteed that a degree of each vertex is at least 3.

Output

Print PATH in the first line, if you solve problem for Johnny. In the second line print the number of vertices in the path c ($c \geq \frac{n}{k}$). And in the third line print vertices describing the path in route order.

Print CYCLES in the first line, if you solve problem for Solving. In the following lines describe **exactly** k cycles in the following format: in the first line print the size of the cycle c ($c \geq 3$). In the second line print the cycle in route order. Also, the first vertex in the cycle must be a **representative**.

Print -1 if there is no any solution. The total amount of printed numbers in the output must be at most 10^6. It's guaranteed, that if exists any solution then there is a correct output satisfies this restriction.

Examples

input
4 6 2 1 2 1 3 1 4 2 3 2 4 3 4
output
PATH 4 1 2 3 4

input
10 18 2 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 2 3 3 4 2 4 5 6 6 7 5 7 8 9 9 10 8 10
output
CYCLES 4 4 1 2 3 4 7 1 5 6

D. Professional layer

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Cardbluff is popular sport game in Telegram. Each Cardbluff player has ever dreamed about entrance in the professional layer. There are n judges now in the layer and you are trying to pass the entrance exam. You have a number k — your skill in Cardbluff.

Each judge has a number a_i — an indicator of uncertainty about your entrance to the professional layer and a number e_i — an experience playing Cardbluff. To pass the exam you need to convince all judges by playing with them. You can play only **one** game with each judge. As a result of a particular game, you can divide the uncertainty of i-th judge by any natural divisor of a_i which is at most k. If GCD of all indicators is equal to 1, you will enter to the professional layer and become a judge.

Also, you want to minimize the total amount of spent time. So, if you play with x judges with total experience y you will spend $x \cdot y$ seconds.

Print minimal time to enter to the professional layer or -1 if it's impossible.

Input

There are two numbers in the first line n and k ($1 \leq n \leq 10^6$, $1 \leq k \leq 10^{12}$) — the number of judges and your skill in Cardbluff.

The second line contains n integers, where i-th number a_i ($1 \leq a_i \leq 10^{12}$) — the uncertainty of i-th judge

The third line contains n integers in the same format ($1 \le e_i \le 10^9$), e_i — the experience of i-th judge.

Output
Print the single integer — minimal number of seconds to pass exam, or -1 if it's impossible

Examples

input
3 6 30 30 30 100 4 5
output
18

input
1 1000000 1 100
output
0

input
3 5 7 7 7 1 1 1
output
-1

E. Radix sum

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let's define **radix sum** of number a consisting of digits $a_1, \ldots, a_k$ and number b consisting of digits $b_1, \ldots, b_k$(we add leading zeroes to the shorter number to match longer length) as number $s(a, b)$ consisting of digits $(a_1 + b_1) \mod 10, \ldots, (a_k + b_k) \mod 10$. The **radix sum** of several integers is defined as follows: $s(t_1, \ldots, t_n) = s(t_1, s(t_2, \ldots, t_n))$

You are given an array $x_1, \ldots, x_n$. The task is to compute for each integer $i (0 \le i < n)$ number of ways to consequently choose one of the integers from the array n times, so that the **radix sum** of these integers is equal to i. Calculate these values modulo 2^{58}.

Input
The first line contains integer n — the length of the array($1 \le n \le 100000$).

The second line contains n integers $x_1, \ldots x_n$ — array elements($0 \le x_i < 100000$).

Output
Output n integers $y_0, \ldots, y_{n-1}$ — y_i should be equal to corresponding number of ways modulo 2^{58}.

Examples

input
2 5 6
output
1 2

input
4 5 7 5 7
output
16 0 64 0

Note
In the first example there exist sequences: sequence $(5, 5)$ with radix sum 0, sequence $(5, 6)$ with radix sum 1, sequence $(6, 5)$ with radix sum 1, sequence $(6, 6)$ with radix sum 2.