CODEFORCES β
Sponsored by Telegram

Codeforces Round #467 (Div. 2)

A. Olympiad

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

The recent All-Berland Olympiad in Informatics featured n participants with each scoring a certain amount of points.

As the head of the programming committee, you are to determine the set of participants to be awarded with diplomas with respect to the following criteria:

- At least one participant should get a diploma.
- None of those with score equal to zero should get awarded.
- When someone is awarded, all participants with score **not less** than his score should also be awarded.

Determine the number of ways to choose a subset of participants that will receive the diplomas.

Input

The first line contains a single integer n $(1 \leq n \leq 100)$ — the number of participants.

The next line contains a sequence of n integers $a_1, a_2, \ldots, a_n$ $(0 \leq a_i \leq 600)$ — participants' scores.

It's guaranteed that at least one participant has non-zero score.

Output

Print a single integer — the desired number of ways.

Examples

input

```
4
1 3 3 2
```

output

```
3
```

input

```
3
1 1 1
```

output

```
1
```

input

```
4
42 0 0 42
```

output

```
1
```

Note

There are three ways to choose a subset in sample case one.

1. Only participants with 3 points will get diplomas.
2. Participants with 2 or 3 points will get diplomas.
3. Everyone will get a diploma!

The only option in sample case two is to award everyone.

Note that in sample case three participants with zero scores cannot get anything.

B. Vile Grasshoppers

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

The weather is fine today and hence it's high time to climb the nearby pine and enjoy the landscape.

The pine's trunk includes several branches, located one above another and numbered from 2 to y. Some of them (more precise, from 2 to p) are occupied by tiny vile grasshoppers which you're at war with. These grasshoppers are known for their awesome jumping skills: the grasshopper at branch x can jump to branches $2 \cdot x, 3 \cdot x, \ldots, \lfloor \frac{y}{x} \rfloor \cdot x$.

Keeping this in mind, you wisely decided to choose such a branch that none of the grasshoppers could interrupt you. At the same time you wanna settle as high as possible since the view from up there is simply breathtaking.

In other words, your goal is to find the highest branch that cannot be reached by any of the grasshoppers or report that it's impossible.

Input

The only line contains two integers p and y ($2 \leq p \leq y \leq 10^9$).

Output

Output the number of the highest suitable branch. If there are none, print `-1` instead.

Examples

input
3 6
output
5

input
3 4
output
-1

Note

In the first sample case grasshopper from branch 2 reaches branches 2, 4 and 6 while branch 3 is initially settled by another grasshopper. Therefore the answer is 5.

It immediately follows that there are no valid branches in second sample case.

C. Save Energy!

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

Julia is going to cook a chicken in the kitchen of her dormitory. To save energy, the stove in the kitchen automatically turns off after k minutes after turning on.

During cooking, Julia goes to the kitchen every d minutes and turns on the stove if it is turned off. While the cooker is turned off, it stays warm. The stove switches on and off instantly.

It is known that the chicken needs t minutes to be cooked on the stove, if it is turned on, and $2t$ minutes, if it is turned off. You need to find out, how much time will Julia have to cook the chicken, if it is considered that the chicken is cooked evenly, with constant speed when the stove is turned on and at a constant speed when it is turned off.

Input

The single line contains three integers k, d and t ($1 \leq k, d, t \leq 10^{18}$).

Output

Print a single number, the total time of cooking in minutes. The relative or absolute error must not exceed 10^{-9}.

Namely, let's assume that your answer is x and the answer of the jury is y. The checker program will consider your answer correct if $\frac{|x-y|}{max(1,y)} \leq 10^{-9}$.

Examples

input
3 2 6
output
6.5

input
4 2 20
output

```
20.0
```

Note

In the first example, the chicken will be cooked for 3 minutes on the turned on stove, after this it will be cooked for $\frac{3}{6}$. Then the chicken will be cooked for one minute on a turned off stove, it will be cooked for $\frac{1}{12}$. Thus, after four minutes the chicken will be cooked for $\frac{3}{6} + \frac{1}{12} = \frac{7}{12}$. Before the fifth minute Julia will turn on the stove and after 2.5 minutes the chicken will be ready $\frac{7}{12} + \frac{2.5}{6} = 1$.

In the second example, when the stove is turned off, Julia will immediately turn it on, so the stove will always be turned on and the chicken will be cooked in 20 minutes.

D. Sleepy Game

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

Petya and Vasya arranged a game. The game runs by the following rules. Players have a directed graph consisting of n vertices and m edges. One of the vertices contains a chip. Initially the chip is located at vertex s. Players take turns moving the chip along some edge of the graph. Petya goes first. Player who can't move the chip loses. If the game lasts for 10^6 turns the draw is announced.

Vasya was performing big laboratory work in "Spelling and parts of speech" at night before the game, so he fell asleep at the very beginning of the game. Petya decided to take the advantage of this situation and make both Petya's and Vasya's moves.

Your task is to help Petya find out if he can win the game or at least draw a tie.

Input

The first line of input contain two integers n and m — the number of vertices and the number of edges in the graph ($2 \le n \le 10^5$, $0 \le m \le 2 \cdot 10^5$).

The next n lines contain the information about edges of the graph. i-th line ($1 \le i \le n$) contains nonnegative integer c_i — number of vertices such that there is an edge from i to these vertices and c_i distinct integers $a_{i,j}$ — indices of these vertices ($1 \le a_{i,j} \le n$, $a_{i,j} \ne i$).

It is guaranteed that the total sum of c_i equals to m.

The next line contains index of vertex s — the initial position of the chip ($1 \le s \le n$).

Output

If Petya can win print «`Win`» in the first line. In the next line print numbers $v_1, v_2, ..., v_k$ ($1 \le k \le 10^6$) — the sequence of vertices Petya should visit for the winning. Vertex v_1 should coincide with s. For $i = 1... k - 1$ there should be an edge from v_i to v_{i+1} in the graph. There must be no possible move from vertex v_k. The sequence should be such that Petya wins the game.

If Petya can't win but can draw a tie, print «`Draw`» in the only line. Otherwise print «`Lose`».

Examples

input

```
5 6
2 2 3
2 4 5
1 4
1 5
0
1
```

output

```
Win
1 2 4 5
```

input

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

output

```
Lose
```

input

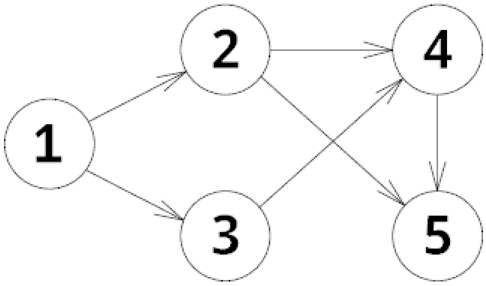
```
2 2
1 2
1 1
1
```

output

```
Draw
```

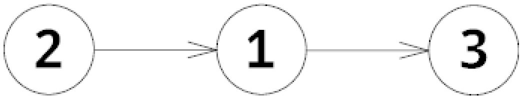
Note

In the first example the graph is the following:



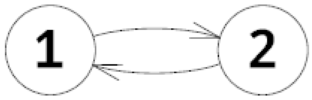
Initially the chip is located at vertex 1. In the first move Petya moves the chip to vertex 2, after that he moves it to vertex 4 for Vasya. After that he moves to vertex 5. Now it is Vasya's turn and there is no possible move, so Petya wins.

In the second example the graph is the following:



Initially the chip is located at vertex 2. The only possible Petya's move is to go to vertex 1. After that he has to go to 3 for Vasya. Now it's Petya's turn but he has no possible move, so Petya loses.

In the third example the graph is the following:



Petya can't win, but he can move along the cycle, so the players will draw a tie.

E. Lock Puzzle

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Welcome to another task about breaking the code lock! Explorers Whitfield and Martin came across an unusual safe, inside of which, according to rumors, there are untold riches, among which one can find the solution of the problem of discrete logarithm!

Of course, there is a code lock is installed on the safe. The lock has a screen that displays a string of n lowercase Latin letters. Initially, the screen displays string s. Whitfield and Martin found out that the safe will open when string t will be displayed on the screen.

The string on the screen can be changed using the operation «$\texttt{shift}\ x$». In order to apply this operation, explorers choose an integer x from 0 to n inclusive. After that, the current string $p = \alpha\beta$ changes to $\beta^R\alpha$, where the length of β is x, and the length of α is $n - x$. In other words, the suffix of the length x of string p is reversed and moved to the beginning of the string. For example, after the operation «$\texttt{shift}\ 4$» the string «$\texttt{abcacb}$» will be changed with string «$\texttt{bcacab}$ », since $\alpha = \texttt{ab}$, $\beta = \texttt{cacb}$, $\beta^R = \texttt{bcac}$.

Explorers are afraid that if they apply too many operations «$\texttt{shift}$», the lock will be locked forever. They ask you to find a way to get the string t on the screen, using no more than 6100 operations.

Input

The first line contains an integer n, the length of the strings s and t ($1 \le n \le 2\,000$).

After that, there are two strings s and t, consisting of n lowercase Latin letters each.

Output

If it is impossible to get string t from string s using no more than 6100 operations «$\texttt{shift}$», print a single number -1.

Otherwise, in the first line output the number of operations k ($0 \le k \le 6100$). In the next line output k numbers x_i corresponding to the operations «$\texttt{shift}\ x_i$» ($0 \le x_i \le n$) in the order in which they should be applied.

Examples

input

```
6
abacbb
babcba
```

output

```
4
6 3 2 3
```

input

```
3
aba
bba
```

output

```
-1
```

In the first example, after applying the operations, the string on the screen will change as follows:

1. abacbb $\longrightarrow$ bbcaba
2. bbcaba $\longrightarrow$ ababbc
3. ababbc $\longrightarrow$ cbabab
4. cbabab $\longrightarrow$ babcba