G. Graphic Settings

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

Recently Ivan bought a new computer. Excited, he unpacked it and installed his favourite game. With his old computer Ivan had to choose the worst possible graphic settings (because otherwise the framerate would be really low), but now he wants to check, maybe his new computer can perform well even with the best possible graphics?

There are m graphics parameters in the game. i-th parameter can be set to any positive integer from 1 to a_i, and initially is set to b_i ($b_i \le a_i$). So there are different combinations of parameters. Ivan can increase or decrease any of these parameters by 1; after that the game will be restarted with new parameters (and Ivan will have the opportunity to check chosen combination of parameters).

Ivan wants to try all p possible combinations. Also he wants to return to the initial settings after trying all combinations, because he thinks that initial settings can be somehow best suited for his hardware. But Ivan doesn't really want to make a lot of restarts.

So he wants you to tell the following:

- If there exists a way to make exactly p changes (each change either decreases or increases some parameter by 1) to try all possible combinations and return to initial combination, then Ivan wants to know this way.
- Otherwise, if there exists a way to make exactly p - 1 changes to try all possible combinations (including the initial one), then Ivan wants to know this way.

Help Ivan by showing him the way to change parameters!

Input

The first line of input contains one integer number m ($1 \le m \le 6$).

The second line contains m integer numbers $a_1, a_2, ..., a_m$ ($2 \le a_i \le 1000$). It is guaranteed that .

The third line contains m integer numbers $b_1, b_2, ..., b_m$ ($1 \le b_i \le a_i$).

Output

If there is a way to make exactly p changes (each change either decreases or increases some parameter by 1) to try all possible combinations and return to initial combination, then output `Cycle` in the first line. Then p lines must follow, each desribing a change. The line must be either `inc x` (increase parameter x by 1) or `dec x` (decrease it).

Otherwise, if there is a way to make exactly p - 1 changes to try all possible combinations (including the initial one), then output `Path` in the first line. Then p - 1 lines must follow, each describing the change the same way as mentioned above.

Otherwise, output `No`.

Examples

input

```
1
3
1
```

output

```
Path
inc 1
inc 1
```

input

```
1
3
2
```

output

```
No
```

input

```
2
3 2
1 1
```

output

```
Cycle
inc 1
inc 1
inc 2
dec 1
dec 1
dec 2
```