Topic Stream Mashup: Constructives

A. 3-palindrome

1 second, 256 megabytes

In the beginning of the new year Keivan decided to reverse his name. He doesn't like palindromes, so he changed Naviek to Navick.

He is too selfish, so for a given n he wants to obtain a string of n characters, each of which is either 'a', 'b' or 'c', with no *palindromes* of length 3 appearing in the string as a substring. For example, the strings "abc" and "abca" suit him, while the string "aba" doesn't. He also want the number of letters 'c' in his string to be as little as possible.

Input
The first line contains single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the string.

Output
Print the string that satisfies all the constraints.

If there are multiple answers, print any of them.

input
2
output
aa

input
3
output
bba

A *palindrome* is a sequence of characters which reads the same backward and forward.

B. Unnatural Conditions

1 second, 256 megabytes

Let $s(x)$ be sum of digits in decimal representation of positive integer x. Given two integers n and m, find some positive integers a and b such that

- $s(a) \ge n$,
- $s(b) \ge n$,
- $s(a + b) \le m$.

Input
The only line of input contain two integers n and m ($1 \le n, m \le 1129$).

Output
Print two lines, one for decimal representation of a and one for decimal representation of b. Both numbers must not contain leading zeros and must have length no more than 2230.

input
6 5
output
6 7

input
8 16
output
35 53

In the first sample, we have $n = 6$ and $m = 5$. One valid solution is $a = 6, b = 7$. Indeed, we have $s(a) = 6 \ge n$ and $s(b) = 7 \ge n$, and also $s(a + b) = s(13) = 4 \le m$.

C. Grid game

1 second, 256 megabytes

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.

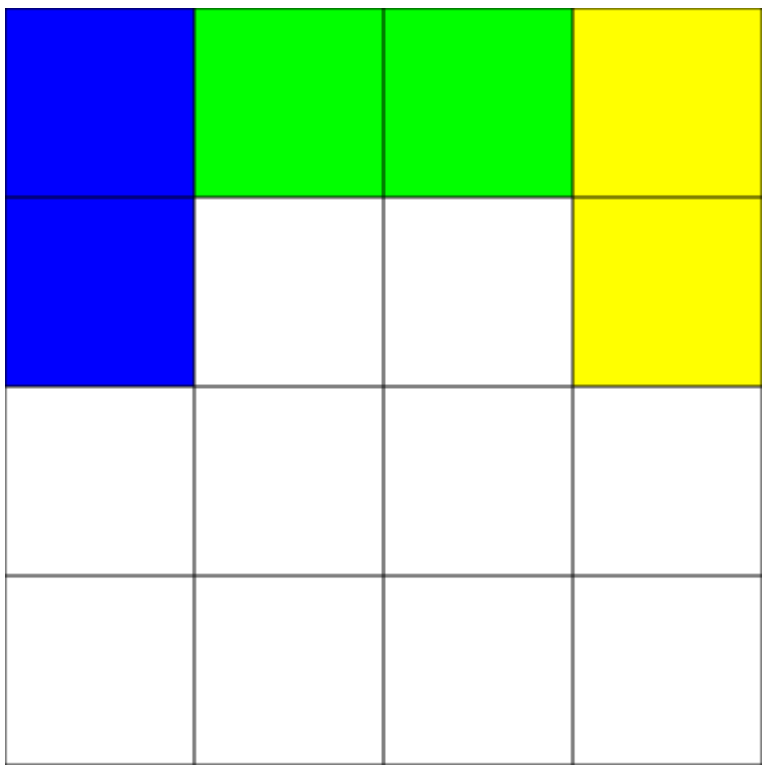
Input
The only line contains a string s consisting of zeroes and ones ($1 \le |s| \le 1000$). Zero describes vertical tile, one describes horizontal tile.

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.

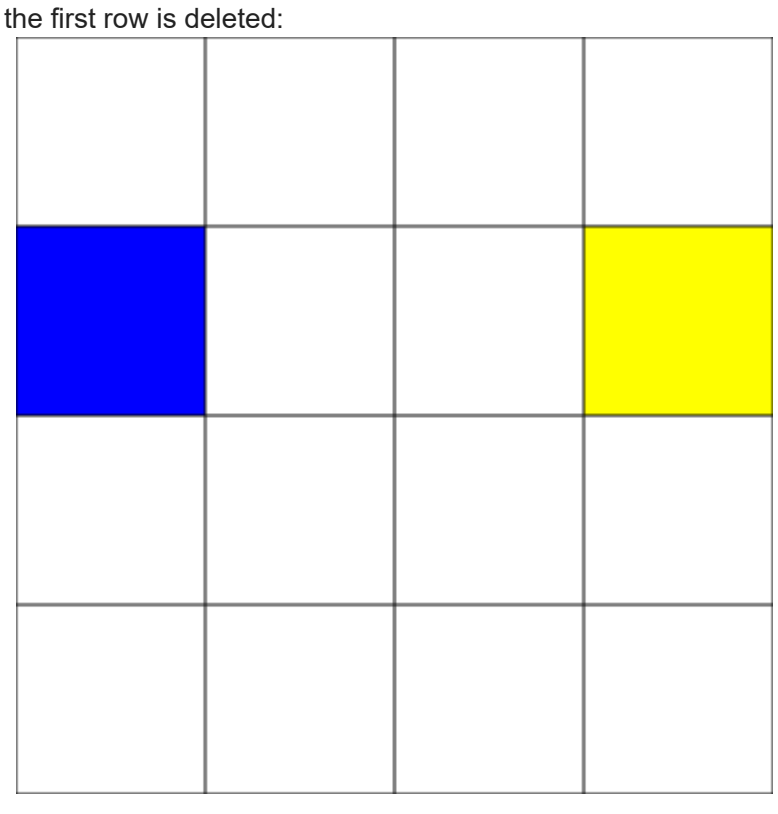
If there exist multiple solutions, print any of them.

input
010
output
1 1 1 2 1 4

Following image illustrates the example after placing all three tiles:



Then the first row is deleted:



D. Walking Between Houses

2 seconds, 256 megabytes

There are n houses in a row. They are numbered from 1 to n in order from left to right. Initially you are in the house 1.

You have to perform k moves to other house. In one move you go from your current house to some other house. You can't stay where you are (i.e., in each move the new house differs from the current house). If you go from the house x to the house y, the total distance you walked increases by $|x-y|$ units of distance, where $|a|$ is the absolute value of a. It is possible to visit the same house multiple times (but you can't visit the same house in sequence).

Your goal is to walk exactly s units of distance in total.

If it is impossible, print "NO". Otherwise print "YES" and any of the ways to do that. Remember that you should do exactly k moves.

Input

The first line of the input contains three integers n, k, s ($2 \le n \le 10^9$, $1 \le k \le 2 \cdot 10^5$, $1 \le s \le 10^{18}$) — the number of houses, the number of moves and the total distance you want to walk.

Output

If you cannot perform k moves with total walking distance equal to s, print "NO".

Otherwise print "YES" on the first line and then print exactly k integers h_i ($1 \le h_i \le n$) on the second line, where h_i is the house you visit on the i-th move.

For each j from 1 to $k-1$ the following condition should be satisfied: $h_j \ne h_{j+1}$. Also $h_1 \ne 1$ should be satisfied.

input
10 2 15
output
YES 10 4

input
10 9 45
output
YES 10 1 10 1 2 1 2 1 6

input
10 9 81
output
YES 10 1 10 1 10 1 10 1 10

input
10 9 82
output
NO

E. A Mist of Florescence

1 second, 256 megabytes

As the boat drifts down the river, a wood full of blossoms shows up on the riverfront.
"I've been here once," Mino exclaims with delight, "it's breathtakingly amazing."

"What is it like?"

"Look, Kanno, you've got your paintbrush, and I've got my words. Have a try, shall we?"

There are four kinds of flowers in the wood, Amaranths, Begonias, Centaureas and Dianthuses.

The wood can be represented by a rectangular grid of n rows and m columns. In each cell of the grid, there is exactly one type of flowers.

According to Mino, the numbers of connected components formed by each kind of flowers are a, b, c and d respectively. Two cells are considered in the same connected component if and only if a path exists between them that moves between cells sharing common edges and passes only through cells containing the same flowers.

You are to help Kanno depict such a grid of flowers, with n and m arbitrarily chosen under the constraints given below. It can be shown that at least one solution exists under the constraints of this problem.

Note that you can choose arbitrary n and m under the constraints below, they are not given in the input.

Input

The first and only line of input contains four space-separated integers a, b, c and d ($1 \le a, b, c, d \le 100$) — the required number of connected components of Amaranths, Begonias, Centaureas and Dianthuses, respectively.

Output

In the first line, output two space-separated integers n and m ($1 \le n, m \le 50$) — the number of rows and the number of columns in the grid respectively.

Then output n lines each consisting of m consecutive English letters, representing one row of the grid. Each letter should be among 'A', 'B', 'C' and 'D', representing Amaranths, Begonias, Centaureas and Dianthuses, respectively.

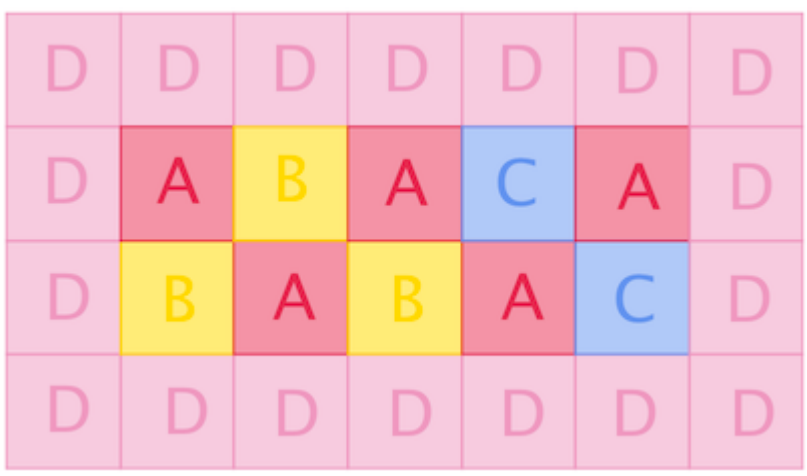
In case there are multiple solutions, print any. You can output each letter in either case (upper or lower).

input
5 3 2 1
output
4 7 DDDDDDD DABACAD DBABACD DDDDDDD

input
50 50 1 1
output
4 50 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC ABABABABABABABABABABABABABABABABABABABABABABABABAB BABABABABABABABABABABABABABABABABABABABABABABABABABABA DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD

input
1 6 4 5
output
7 7 DDDDDDD DDDBDBD DDCDCDD DBDADBD DDCDCDD DBDBDDD DDDDDDD

In the first example, each cell of Amaranths, Begonias and Centaureas forms a connected component, while all the Dianthuses form one.



F. Marco and GCD Sequence

1 second, 256 megabytes

In a dream Marco met an elderly man with a pair of black glasses. The man told him the key to immortality and then disappeared with the wind of time.

When he woke up, he only remembered that the key was a sequence of positive integers of some length n, but forgot the exact sequence. Let the elements of the sequence be $a_1, a_2, ..., a_n$. He remembered that he calculated $gcd(a_i, a_{i+1}, ..., a_j)$ for every $1 \le i \le j \le n$ and put it into a set S. gcd here means the greatest common divisor.

Note that even if a number is put into the set S twice or more, it only appears once in the set.

Now Marco gives you the set S and asks you to help him figure out the initial sequence. If there are many solutions, print any of them. It is also possible that there are no sequences that produce the set S, in this case print `-1`.

Input
The first line contains a single integer m $(1 \le m \le 1000)$ — the size of the set S.

The second line contains m integers $s_1, s_2, ..., s_m$ $(1 \le s_i \le 10^6)$ — the elements of the set S. It's guaranteed that the elements of the set are given in strictly increasing order, that means $s_1 < s_2 < ... < s_m$.

Output
If there is no solution, print a single line containing `-1`.

Otherwise, in the first line print a single integer n denoting the length of the sequence, n should not exceed 4000.

In the second line print n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^6)$ — the sequence.

We can show that if a solution exists, then there is a solution with n not exceeding 4000 and a_i not exceeding 10^6.

If there are multiple solutions, print any of them.

input
4 2 4 6 12
output
3 4 6 12

input
2 2 3
output
-1

In the first example $2 = gcd(4, 6)$, the other elements from the set appear in the sequence, and we can show that there are no values different from $2, 4, 6$ and 12 among $gcd(a_i, a_{i+1}, ..., a_j)$ for every $1 \le i \le j \le n$.

G. Gluttony

2 seconds, 256 megabytes

You are given an array a with n distinct integers. Construct an array b by permuting a such that for every non-empty subset of indices $S = \{x_1, x_2, ..., x_k\}$ $(1 \le x_i \le n, 0 < k < n)$ the sums of elements on that positions in a and b are different, i. e.

$$\sum_{i=1}^{k} a_{x_i} \neq \sum_{i=1}^{k} b_{x_i}.$$

Input
The first line contains one integer n $(1 \le n \le 22)$ — the size of the array.

The second line contains n space-separated distinct integers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^9)$ — the elements of the array.

Output
If there is no such array b, print `-1`.

Otherwise in the only line print n space-separated integers $b_1, b_2, ..., b_n$. Note that b must be a permutation of a.

If there are multiple answers, print any of them.

input
2 1 2
output
2 1

input
4 1000 100 10 1
output
100 1 1000 10

An array x is a permutation of y, if we can shuffle elements of y such that it will coincide with x.

Note that the empty subset and the subset containing all indices are not counted.

H. Down or Right

1 second, 256 megabytes

This is an interactive problem.

Bob lives in a square grid of size $n \times n$, with rows numbered 1 through n from top to bottom, and columns numbered 1 through n from left to right. Every cell is either allowed or blocked, but you don't know the exact description of the grid. You are given only an integer n.

Bob can move through allowed cells but only in some limited directions. When Bob is in an allowed cell in the grid, he can move **down or right** to an adjacent cell, if it is allowed.

You can ask at most $4 \cdot n$ queries of form "`?` r_1 c_1 r_2 c_2" ($1 \le r_1 \le r_2 \le n$, $1 \le c_1 \le c_2 \le n$). The answer will be "`YES`" if Bob can get from a cell (r_1, c_1) to a cell (r_2, c_2), and "`NO`" otherwise. In particular, if one of the two cells (or both) is a blocked cell then the answer is "`NO`" for sure. Since Bob doesn't like short trips, you can only ask queries with the manhattan distance between the two cells at least $n - 1$, i.e. the following condition must be satisfied: $(r_2 - r_1) + (c_2 - c_1) \ge n - 1$.

It's guaranteed that Bob can get from the top-left corner $(1, 1)$ to the bottom-right corner (n, n) and your task is to find a way to do it. You should print the answer in form "`!` `S`" where S is a string of length $2 \cdot n - 2$ consisting of characters '`D`' and '`R`', denoting moves down and right respectively. The down move increases the first coordinate by 1, the right move increases the second coordinate by 1. If there are multiple solutions, any of them will be accepted. You should terminate immediately after printing the solution.

Input
The only line of the input contains an integer n $(2 \le n \le 500)$ — the size of the grid.

Output
When you are ready to print the answer, print a single line containing "`!` `S`" where where S is a string of length $2 \cdot n - 2$ consisting of characters '`D`' and '`R`', denoting moves down and right respectively. The path should be a valid path going from the cell $(1, 1)$ to the cell (n, n) passing only through allowed cells.

Interaction
You can ask at most $4 \cdot n$ queries. To ask a query, print a line containing "`?` r_1 c_1 r_2 c_2" $(1 \le r_1 \le r_2 \le n, 1 \le c_1 \le c_2 \le n)$. After that you should read a single line containing "`YES`" or "`NO`" depending on the answer of the query. "`YES`" means Bob can go from the cell (r_1, c_1) to the cell (r_2, c_2), "`NO`" means the opposite.

Note that the grid is fixed before the start of your solution and does not depend on your queries.

After printing a query do not forget to output end of line and flush the output. Otherwise you will get `Idleness limit exceeded` or other negative verdict. To do this, use:

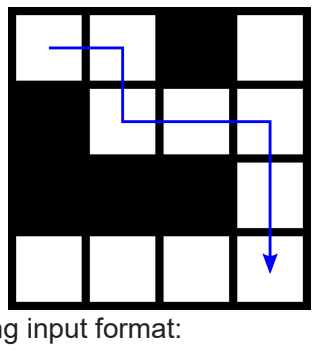
- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

Answer "BAD" instead of "YES" or "NO" means that you made an invalid query. Exit immediately after receiving "BAD" and you will see `Wrong answer` verdict. Otherwise you can get an arbitrary verdict because your solution will continue to read from a closed stream.

input
4 YES NO YES YES

output
? 1 1 4 4 ? 1 2 4 3 ? 4 1 4 4 ? 1 4 4 4 ! RDRRDD

The first example is shown on the picture below.



To hack, use the following input format:

The first line should contain a single integer n ($2 \le n \le 500$) — the size of the grid.

Each of the next n lines should contain a string of n characters '#' or '.', where '#' denotes a blocked cell, and '.' denotes an allowed cell.

For example, the following text encodes the example shown above:

```
4
..#.
#...
###.
....
```