Codeforces Round #268 (Div. 2)

A. I Wanna Be the Guy

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

There is a game called "I Wanna Be the Guy", consisting of n levels. Little X and his friend Little Y are addicted to the game. Each of them wants to pass the whole game.

Little X can pass only p levels of the game. And Little Y can pass only q levels of the game. You are given the indices of levels Little X can pass and the indices of levels Little Y can pass. Will Little X and Little Y pass the whole game, if they cooperate each other?

Input

The first line contains a single integer n ($1 \leq n \leq 100$).

The next line contains an integer p ($0 \leq p \leq n$) at first, then follows p distinct integers $a_1, a_2, ..., a_p$ ($1 \leq a_i \leq n$). These integers denote the indices of levels Little X can pass. The next line contains the levels Little Y can pass in the same format. It's assumed that levels are numbered from 1 to n.

Output

If they can pass all the levels, print "`I become the guy.`". If it's impossible, print "`Oh, my keyboard!`" (without the quotes).

Sample test(s)

input
4 3 1 2 3 2 2 4
output
I become the guy.

input
4 3 1 2 3 2 2 3
output
Oh, my keyboard!

Note

In the first sample, Little X can pass levels [1 2 3], and Little Y can pass level [2 4], so they can pass all the levels both.

In the second sample, no one can pass level 4.

B. Chat Online

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

Little X and Little Z are good friends. They always chat online. But both of them have schedules.

Little Z has fixed schedule. He always online at any moment of time between a_1 and b_1, between a_2 and b_2, ..., between a_p and b_p (all borders inclusive). But the schedule of Little X is quite strange, it depends on the time when he gets up. If he gets up at time 0, he will be online at any moment of time between c_1 and d_1, between c_2 and d_2, ..., between c_q and d_q (all borders inclusive). But if he gets up at time t, these segments will be shifted by t. They become $[c_i + t, d_i + t]$ (for all i).

If at a moment of time, both Little X and Little Z are online simultaneosly, they can chat online happily. You know that Little X can get up at an integer moment of time between l and r (both borders inclusive). Also you know that Little X wants to get up at the moment of time, that is suitable for chatting with Little Z (they must have at least one common moment of time in schedules). How many integer moments of time from the segment $[l, r]$ suit for that?

Input

The first line contains four space-separated integers p, q, l, r ($1 \le p, q \le 50$; $0 \le l \le r \le 1000$).

Each of the next p lines contains two space-separated integers a_i, b_i ($0 \le a_i < b_i \le 1000$). Each of the next q lines contains two space-separated integers c_j, d_j ($0 \le c_j < d_j \le 1000$).

It's guaranteed that $b_i < a_{i+1}$ and $d_j < c_{j+1}$ for all valid i and j.

Output

Output a single integer — the number of moments of time from the segment $[l, r]$ which suit for online conversation.

Sample test(s)

input
1 1 0 4 2 3 0 1

output
3

input
2 3 0 20 15 17 23 26 1 4 7 11 15 17

output
20

C. 24 Game

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

Little X used to play a card game called "24 Game", but recently he has found it too easy. So he invented a new game.

Initially you have a sequence of n integers: $1, 2, ..., n$. In a single step, you can pick two of them, let's denote them a and b, erase them from the sequence, and append to the sequence either $a + b$, or $a - b$, or $a \times b$.

After $n - 1$ steps there is only one number left. Can you make this number equal to 24?

Input

The first line contains a single integer n $(1 \leq n \leq 10^5)$.

Output

If it's possible, print "YES" in the first line. Otherwise, print "NO" (without the quotes).

If there is a way to obtain 24 as the result number, in the following $n - 1$ lines print the required operations an operation per line. Each operation should be in form: "a op b = c". Where a and b are the numbers you've picked at this operation; op is either "+", or "−", or "*"; c is the result of corresponding operation. Note, that the absolute value of c mustn't be greater than 10^{18}. The result of the last operation must be equal to 24. Separate operator sign and equality sign from numbers with spaces.

If there are multiple valid answers, you may print any of them.

Sample test(s)

input
1

output
NO

input
8

output
YES 8 * 7 = 56 6 * 5 = 30 3 - 4 = -1 1 - 2 = -1 30 - -1 = 31 56 - 31 = 25 25 + -1 = 24

D. Two Sets

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

Little X has n distinct integers: $p_1, p_2, ..., p_n$. He wants to divide all of them into two sets A and B. The following two conditions must be satisfied:

- If number x belongs to set A, then number $a - x$ must also belong to set A.
- If number x belongs to set B, then number $b - x$ must also belong to set B.

Help Little X divide the numbers into two sets or determine that it's impossible.

Input

The first line contains three space-separated integers n, a, b ($1 \le n \le 10^5$; $1 \le a, b \le 10^9$). The next line contains n space-separated distinct integers $p_1, p_2, ..., p_n$ ($1 \le p_i \le 10^9$).

Output

If there is a way to divide the numbers into two sets, then print "YES" in the first line. Then print n integers: $b_1, b_2, ..., b_n$ (b_i equals either 0, or 1), describing the division. If b_i equals to 0, then p_i belongs to set A, otherwise it belongs to set B.

If it's impossible, print "NO" (without the quotes).

Sample test(s)

input
4 5 9 2 3 4 5
output
YES 0 0 1 1

input
3 3 4 1 2 4
output
NO

Note

It's OK if all the numbers are in the same set, and the other one is empty.

E. Hack it!

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

Little X has met the following problem recently.

Let's define $f(x)$ as the sum of digits in decimal representation of number x (for example, $f(1234) = 1 + 2 + 3 + 4$). You are to calculate $\sum_{i=l}^{r} f(i) \bmod a$.

Of course Little X has solved this problem quickly, has locked it, and then has tried to hack others. He has seen the following C++ code:

```
ans = solve(l, r) % a;
if (ans <= 0)
  ans += a;
```

This code will fail only on the test with $\sum_{i=l}^{r} f(i) \equiv 0 \pmod{a}$. You are given number a, help Little X to find a proper test for hack.

Input

The first line contains a single integer a ($1 \le a \le 10^{18}$).

Output

Print two integers: l, r ($1 \le l \le r < 10^{200}$) — the required test data. Leading zeros aren't allowed. It's guaranteed that the solution exists.

Sample test(s)

input
46
output
1 10

input
126444381000032
output
2333333 2333333333333