Surprise Language Round #6

A. Hexagonal Numbers

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

Hexagonal numbers are figurate numbers which can be calculated using the formula $h_n = 2n^2 - n$. You are given n; calculate n-th hexagonal number.

Input

The only line of input contains an integer n ($1 \le n \le 100$).

Output

Output the n-th hexagonal number.

Sample test(s)

input
3
output
15

input
6
output
66

B. A + Reverse B

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

You are given numbers a and b. Calculate the sum of a and reverse of b. A reverse of a number is a number which contains the same digits in reverse order. For example, reverse of 230 is 32, and reverse of 0 is 0.

Input

The input contains two integers a and b ($0 \le a, b \le 10^9$), separated by a single space. The numbers are given without leading zeros.

Output

Output the sum of a and reverse of b.

Sample test(s)

input
5 15
output
56

input
73 9180
output
892

C. LCM

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

Least common multiple (LCM) of two numbers is the smallest positive integer which is divisible by both of them. You are given integers a and b. Calculate their LCM.

Input

The input contains two integers a and b ($1 \leq a, b \leq 10^3$), separated by a single space.

Output

Output LCM(a, b).

Sample test(s)

input
10 42
output
210

input
123 41
output
123

D. Asterisks

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

You are given a number n. Print n lines, i-th line should consist of i characters "*". Lines' indices are 1-based.

Input

The only line of input contains an integer n ($1 \le n \le 50$).

Output

Output the described pattern.

Sample test(s)

input
3
output
* ** ***

input
6
output
* ** *** **** ***** ******

E. HQ9+

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

HQ9+ is a joke programming language which has only four one-character instructions:

- "H" prints "Hello, World!",
- "Q" prints the whole source code of the program itself (at each call),
- "9" prints the lyrics of "99 Bottles of Beer" song,
- "+" increments the value stored in the internal accumulator.

Instructions "H" and "Q" are case-sensitive and must be uppercase. The characters of the program which are not instructions are ignored.

You are given a program written in HQ9+. You have to figure out whether executing this program will produce any output.

Input

The input will consist of a single line p which will give a program in HQ9+. String p will contain between 1 and 100 characters, inclusive. ASCII-code of each character of p will be between 33 (exclamation mark) and 126 (tilde), inclusive.

Output

Output "YES", if executing the program will produce any output, and "NO" otherwise (quotes for clarity only).

Sample test(s)

input
Hello!
output
YES

input
Progopedia
output
NO

Note

In the first case the program contains only one instruction — "H", which prints "Hello, World!".

In the second case none of the program characters are language instructions.

F. Binary Notation

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

You are given a positive integer n. Output its binary notation.

Input

The only line of input data contains an integer n ($1 \le n \le 10^6$).

Output

Output the binary notation of n (without any leading zeros).

Sample test(s)

input
5
output
101

input
101
output
1100101

Note

In the first example $5 = 1 * 2^2 + 0 * 2^1 + 1 * 2^0$.

G. Array Sorting

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

Sorting arrays is traditionally associated with high-level languages. How hard can it be in Roco? Sort the given array in non-descending order.

Input

The first line of input contains an integer n ($1 \le n \le 100$) — the size of the array. The following n lines contain the elements of the array, one per line. Each element of the array is an integer between 1 and 100, inclusive. The array might contain duplicate elements.

Output

Output space-separated elements of the sorted array.

Sample test(s)

input
5 7 1 9 7 3

output
1 3 7 7 9

input
10 100 1 100 1 100 1 100 1 100 1

output
1 1 1 1 1 100 100 100 100 100

H. Stack

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

In this problem we'll use a stack which supports two types of operations:

- Push a given number on the stack.
- Pop two numbers from the stack, perform a given operation (addition or multiplication) on them and push the result on the stack.

You are given a string which describes the sequence of operations to be performed on the stack. i-th character corresponds to i-th operation:

- If i-th character is a digit, push the corresponding number on the stack.
- If i-th character is «+» or «*», perform the corresponding operation.

Initially the stack is empty. Output the topmost number on the stack after executing all given operations.

Input

The only line of input contains a string of operations, consisting of characters «+», «*» and digits (`0..9`). The length of the string will be between 1 and 20 characters, inclusive.

The given sequence of operations is guaranteed to be correct, i.e. the stack will have at least two elements before every math operation. The numbers on the stack will never exceed 10^6.

Output

Output a single number — the topmost element of the stack after performing all given operations.

Sample test(s)

input
12+3*66*+
output
45

input
149
output
9

Note

In the first case the stack will end up containing a single number — the result of calculating (1+2)*3+6*6.

In the second case there are no math operations, so the answer will be the last number pushed on the stack.