

## Surprise Language Round #7

### A. Crystal Ball Sequence

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

Crystal ball sequence on hexagonal lattice is defined as follows:  $n$ -th element is the number of lattice points inside a hexagon with  $(n + 1)$  points on each side. The formula is  $H_n = 3 \cdot n \cdot (n + 1) + 1$ . You are given  $n$ ; calculate  $n$ -th element of the sequence.

#### Input

The only line of input contains an integer  $n$  ( $0 \leq n \leq 9$ ).

#### Output

Output the  $n$ -th element of crystal ball sequence.

#### Examples

<b>input</b>
1
<b>output</b>
7

  

<b>input</b>
3
<b>output</b>
37

## B. Hexakosioihexekontahexaphobia

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

This (unpronounceable) word means simply fear of number 666.

You are given a string of digits. Check whether it is scary for a person suffering from this phobia, i.e., whether it contains number 666 as a substring.

### Input

The input will consist of a single string  $p$ . The string contains between 1 and 100 digits ('0'-'9'), inclusive. The string doesn't contain any other characters except digits.

### Output

Output "YES" if given string contains number 666, and "NO" otherwise (quotes for clarity only).

### Examples

<b>input</b>
123098
<b>output</b>
NO
<b>input</b>
16660
<b>output</b>
YES
<b>input</b>
1606061
<b>output</b>
NO

### Note

Note that 666 must be a contiguous substring of  $p$ , not a subsequence (see sample 3).

## C. Eval

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

You are given a simple arithmetic expression of the form  $a?b$ , where  $a$  and  $b$  are integer constants, and  $?$  can be one of the following operations: '+' (addition), '-' (subtraction), '\*' (multiplication), '/' (integer division) or '%' (modulo operation).

Output the result of evaluation of this expression.

### Input

The input is a single line containing an expression  $a?b$ . Here  $a$  and  $b$  are integers between 1 and 999, inclusive;  $?$  is an operation character: '+', '-' (ASCII code 45), '\*', '/' or '%'.

### Output

Output a single integer — the result of evaluation of this expression.

### Examples

<b>input</b>
123+456
<b>output</b>
579

<b>input</b>
192/5
<b>output</b>
38

<b>input</b>
945%19
<b>output</b>
14

## D. Caesar Cipher

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

Caesar cipher is one of the simplest encryption techniques. To transform the original message into encrypted one using key  $k$ , one has to replace each letter with a letter which is  $k$  positions later in the alphabet (if this takes the position beyond Z, the rest of it is counted from the start of the alphabet). In a more formal way, if letters of the alphabet are enumerated starting with 0, the result of encryption for character  $X$  will be  $(x + k) \bmod 26$  (26 is the number of letters in the Latin alphabet).

You are given the original message and the encryption key  $k$ . Output the result of encryption.

### Input

The first line of input contains an integer  $k$  ( $0 \leq k \leq 25$ ) — the encryption key.

The second line contains the original message — a sequence of uppercase Latin letters ('A'-'Z'). The length of the message is from 1 to 10, inclusive.

### Output

Output the result of encryption.

### Examples

<b>input</b>
5 CODEFORCES
<b>output</b>
HTIJKTWHJX
<b>input</b>
13 FALSERULEZ
<b>output</b>
SNYFREHYRM

## E. Chessboard

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

Chessboard is a board of  $n \times n$  squares arranged in two alternating colors (black and white). Top left square is white. You are given board size  $n$ . Output an image of a chessboard, with black and white squares marked with '#' and '.' characters, respectively.

### Input

The only line of input contains an integer  $n$  ( $1 \leq n \leq 9$ ).

### Output

Output an image of  $n \times n$  chessboard.

### Examples

input
4
output
<pre>.#.# #.#. .#.# #.#.</pre>

## F. Pairwise Sums

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

You are given an array of  $n$  integers. For each element output the sum of itself and the previous element. For the first element, output the sum of the first and the last elements of the array.

### Input

The input consists of a single line of space-separated integers. The first number is  $n$  ( $2 \leq n \leq 50$ ) — the size of the array. The following  $n$  numbers are the elements of the array ( $1 \leq a_i \leq 1000$ ).

### Output

Output the sums  $a_1 + a_n, a_2 + a_1, \dots, a_n + a_{n-1}$ , separated with spaces.

### Examples

<b>input</b>
4 1 2 3 4
<b>output</b>
5 3 5 7

<b>input</b>
5 5 46 372 81 9
<b>output</b>
14 51 418 453 90

## G. Hamming Distance

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

Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. You are given two strings; calculate the distance between them.

### Input

The input consists of two lines. Each line contains a string of characters 'A'-'Z' between 1 and 100 characters, inclusive. The strings have equal length.

### Output

Output Hamming distance between the strings.

### Examples

<b>input</b>
CODECHEF TOPCODER
<b>output</b>
6

  

<b>input</b>
HAMMING DISTANC
<b>output</b>
6

## H. 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 FALSE? Sort the given array in non-descending order.

### Input

The input consists of a single line of space-separated integers. The first number is  $n$  ( $1 \leq n \leq 10$ ) — the size of the array. The following  $n$  numbers are the elements of the array ( $1 \leq a_i \leq 100$ ).

### Output

Output space-separated elements of the sorted array.

### Examples

input
3 3 1 2
output
1 2 3

input
7 12 2 3 44 5 60 2
output
2 2 3 5 12 44 60