VK Cup 2016 - Wild Card Round 1

A. Lazy Caterer Sequence

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Lazy caterer sequence is defined as the maximum number of pieces formed when slicing a convex pancake with n cuts (each cut is a straight line). The formula is $C_n = n \cdot (n + 1) / 2 + 1$. You are given n; calculate n-th element of the sequence.

Input

The only line of the input contains an integer n ($0 \le n \le 100$).

Output

Output the n-th element of lazy caterer sequence.

Examples

input
2
output
4

input
5
output
16

B. Seasons

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given a name of a month. Output the season of the year to which it belongs (based on Northern hemisphere).

Input

The input consists of a single string containing the name of one of the twelve months (January, February, March, April, May, June, July, August, September, October, November or December). The string is capitalized as given here.

Output

Output a single string — the season of the year to which the given month belongs (winter, spring, summer or autumn). The name of the season should be in lowercase.

Examples

input
April
output
spring

input
November
output
autumn

Note

Assume that winter is December through February, spring is March through May, summer is June through August and autumn is September through November.

C. Array Sum

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given an array of integers. Calculate the sum of its elements.

Input

The i-th line of the input contains an integer a_i ($0 \le a \le 1000$) — the i-th element of the array. The size of the array is between 1 and 10, inclusive. Note that the size of the array is not given explicitly!

Output

Output a single integer — the sum of the elements of the array.

Examples

input
2 15 110 3
output
130

input
90 0 21 331 45
output
487

D. Maximal Difference

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given an array of integers a_i. Find the largest absolute value of difference between adjacent elements of the array $max(abs(a_i - a_{i+1}))$.

Input

The only line of the input contains a list of space-separated integers a_i ($1 \le a_i \le 100$) — elements of the array. The size of the array is between 2 and 10, inclusive. Note that the size of the array is not given explicitly!

Output

Output a single integer — the largest absolute value of difference between adjacent elements of the array.

Examples

input
2 10 4 8 6 12
output
8

input
3 3
output
0

E. Divisibility Check

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given an array of integers. Check whether there exists a number in this array which is divisible by all other numbers in this array. Output 1, if such a number exists, and 0 otherwise.

Input

The only line of the input contains a list of space-separated integers a_i ($1 \le a_i \le 100$) — elements of the array. The size of the array is between 2 and 10, inclusive. Note that the size of the array is not given explicitly!

Output

Output `1` if there exists element of this array which is divisible by all other elements of the array, and `0` otherwise.

Examples

input
6 12 4
output
1

input
3 13
output
0

input
26 13 12
output
0

F. Primes in Interval

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given two integers a and b ($a \le b$). How many prime numbers are there on the interval from a to b, inclusive?

Input

The input contains two integers a and b ($2 \le a \le b \le 1\,000\,000$), separated by a single space.

Output

Output a single integer — the number of primes between a and b, inclusive.

Examples

input
10 20
output
4

input
23 23
output
1

input
271 566
output
46

G. Hungarian Notation

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

In Hungarian notation, a variable name is prefixed with a letter or a group of letters which are mnemonics for the type of that variable. For the purposes of this problem we will consider only two data types: integer and real.

You are given the meaningful part of variable name in lowercase and a sample value that it will store. Integer values will be written as a sequence of digits. Real values will be written using fixed-point notation: the value is represented with a mandatory decimal point, one or more digits in the decimal part and without exponent part.

Your task is to construct a name of this variable in Hungarian notation in the following way. Convert the first letter of meaningful part of the name to uppercase, and prepend a prefix: 'i' for integer variables and 'f' for real ones.

Input

The first line of the input contains a string of lowercase letters of English alphabet. The length of the string will be between 1 and 10, inclusive.

The second line of the input contains a string of digits and zero or one decimal point '.'. The length of the string will be between 1 and 11, inclusive. It's guaranteed that the decimal point '.' will not be the last character of the string.

Output

Output a single string — the name of the variable in Hungarian notation.

Examples

input
count 18
output
iCount

input
weight 3.95
output
fWeight

H. Rotate Matrix

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

You are given a square matrix of integer numbers. Rotate it 90 degrees clockwise (see examples for clarification of rotation).

Input

The input consists of n lines ($1 \le n \le 10$, n is not given explicitly). Each of the lines contains n space-separated integers; j-th integer in i-th line corresponds to matrix element m_{ij} ($1 \le m_{ij} \le 100$).

Output

Output the rotated matrix in the same format as the input.

Examples

input
1 2 3 4 5 6 7 8 9

output
7 4 1 8 5 2 9 6 3

input
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

output
13 9 5 1 14 10 6 2 15 11 7 3 16 12 8 4