Unknown Language Round #4

A. Hexagonal numbers

time limit per test: 2 seconds
memory limit per test: 64 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
2
output
6

input
5
output
45

B. Gnikool Ssalg

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

You are given a string. Reverse its characters.

Input

The only line of input contains a string between 1 and 100 characters long. Each character of the string has ASCII-code between 33 (exclamation mark) and 126 (tilde), inclusive.

Output

Output the characters of this string in reverse order.

Sample test(s)

input
secrofedoc
output
codeforces

input
!ssalg-gnikool5
output
5looking-glass!

C. Decimal 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 integer numbers. Calculate the sum of its elements.

Input

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

Output

Output the sum of the elements of the array.

Sample test(s)

input
5 1 2 3 4 5

output
15

input
7 100 5 45 86 14 20 30

output
300

D. Exponentiation

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

You are given integers a, b and c. Calculate a^b modulo c.

Input

Input data contains numbers a, b and c, one number per line. Each number is an integer between 1 and 100, inclusive.

Output

Output a^b mod c.

Sample test(s)

input
2 5 40

output
32

input
2 5 26

output
6

E. Tribonacci numbers

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

Tribonacci numbers are a sequence of numbers, defined as follows:

- $t_0 = t_1 = 0$,
- $t_2 = 1$,
- $t_i = t_{i-1} + t_{i-2} + t_{i-3}$.

You are given n; calculate n-th tribonacci number modulo 26.

Input

The only line of input contains an integer n ($1 \le n \le 1000$).

Output

Output n-th tribonacci number modulo 26.

Sample test(s)

input
4
output
2

input
9
output
18

F. Prime factorization

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

You are given an integer n. Output its prime factorization.

If $n = a_1^{b_1} a_2^{b_2} \ldots a_k^{b_k}$, where a_k are prime numbers, the output of your program should look as follows: $a_1\ a_1\ \ldots\ a_1\ a_2\ a_2\ \ldots\ a_2\ \ldots\ a_k\ a_k\ \ldots\ a_k$, where factors are ordered in non-decreasing order, and each factor a_i is printed b_i times.

Input

The only line of input contains an integer n ($2 \le n \le 250$).

Output

Output the prime factorization of n, as described above.

Sample test(s)

input
245
output
5 7 7

input
13
output
13

G. CAPS LOCK ON

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

You are given a string which consists of letters and other characters. Convert it to uppercase, i.e., replace all lowercase letters with corresponding uppercase ones. Keep the rest of characters unchanged.

Input

The only line of input contains a string between 1 and 100 characters long. Each character of the string has ASCII-code between 33 (exclamation mark) and 126 (tilde), inclusive.

Output

Output the given string, converted to uppercase.

Sample test(s)

input
cOdEfOrCeS
output
CODEFORCES

input
ulr#4:befunge-RULES!
output
ULR#4:BEFUNGE-RULES!

H. Balanced brackets

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

A sequence of brackets is called balanced if one can turn it into a valid math expression by adding characters «+» and «1». For example, sequences «(())()», «()» and «(()(()))» are balanced, while «)(», «(()» and «(())(» are not.

You are given a string which consists of opening and closing round brackets. Check whether it is a balanced bracket sequence.

Input

The only line of input contains a string between 1 and 100 characters long, inclusive. Each character in the string will be «(» or «)».

Output

Output «YES» if the bracket sequence is balanced, and «NO» otherwise (quotes for clarity only).

Sample test(s)

input
(()(()))()
output
YES

input
())()
output
NO

I. Array sorting

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

Sorting arrays is traditionally associated with high-level languages. How hard can it be in Befunge? 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 60, 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 60 1 60 1 60 1 60 1 60 1

output
1 1 1 1 1 60 60 60 60 60

J. Date calculation

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

In Gregorian calendar a typical year consists of 365 days and 12 months. The numbers of days in the months are: $31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31$. If year index is divisible by 400, or divisible by 4 but not by 100, the year becomes leap year, with one extra day in the second month (the one which typically has 28 days).

You are given the index of the year and the index of the day in the year. Find out the date of this day (day and month it will fall upon).

Input

The first line of input contains the year index, between 1600 and 2400, inclusive. The second line contains the day index, between 1 and 366, inclusive. It is guaranteed that the day index will be valid for this year, i.e., day 366 will occur only in a leap year.

Output

Output the index of the day and the index of the month, separated with a space.

Sample test(s)

input
2011 324
output
20 11

input
2012 274
output
30 9

Note

All indexes are 1-based.