10th QMUL S&E Programming Competition

Run by

School of Electronic Engineering and Computer Science

Sponsored by

Wednesday, 22nd February 2017

Note. *Your programs should read from standard input. The programs should process one input and print the result at the standard output. When judging your submission we will run your program several times on different inputs.*

Find the word

Given a matrix of letters (A-Z) and a word, check if the word can be found in the matrix. For instance, the word COMPUTE can be found in the following matrix as shown in boldface:

FA**OM**EWG
B**CAPTE**W
PAY**U**QJR
XWHAXIG
PAYQLHG

Words can start at any place in the matrix, but each of the following letters has to be one of the (up to) 8 neighbouring letters of the previous letter. Moreover, each letter from the matrix can only be used once.

Input Specification

Each input consists of:

- One line containing numbers n and m ($2 \le n,m \le 7$).
- Then n lines with m letters each (all uppercase).
- A blank line.
- One line containing a word w (all uppercase, at most 7 characters).

Output Specification

Output "Yes" or "No", depending on whether the word w can be found in the matrix or not.

Sample Inputs.

Input	Output
5 7 FAOMEWG BCAPTEW PAYUQJR XWHAXIG PAYQLHG COMPUTE	Yes
3 3 FAO BCA PEY CAP	No

Unit Fractions

A unit fraction is a fraction of the form *1/n* where *n* is greater or equal to 2. Given a bound *N*, find all the sums of the kind

$1/x = 1/y + 1/z$

involving only unit fractions, where $2 \leq x < y \leq z \leq N$. List them in lexicographical order on *(x,y,z)*.

Input

Each input consists of a single number $4 \leq N \leq 100$.

Output

For each input *N*, output the triples of positive integer numbers *(x,y,z)* which can be used to write a true equation involving unit fractions of the form described above.

Sample Inputs.

Input	Output
6	(2,3,6) (2,4,4) (3,6,6)
15	(2,3,6) (2,4,4) (3,4,12) (3,6,6) (4,6,12) (4,8,8) (5,10,10) (6,10,15) (6,12,12) (7,14,14)

Perfect Squares

Given two integers $1 \leq Min \leq Max$, return the number of integers that are perfect squares and are in the interval [Min,Max].

Input

Each input consists of two lines: The first line contains the lower bound *Min*, while the second line contains the upper bound *Max*. These are in the range $1 \leq Min \leq Max \leq 2^{50}$.

Output

For each input, output the number of integers that are perfect squares in the given interval.

Sample Inputs.

Input	Output
1 16	4
25 40	2

Set Difference

Given two sets of integers, *X* and *Y*, you are asked to return the set of all integers that appear exactly in one of the two sets.

For example, for the following sets X and Y:

[12,44,3,-1,34,11]
[5,0,12,3,55,34]

the result should be:

[-1,0,5,11,44,55]

Input

The input consists of two lines, each line containing a list (without repetitions) of integers in the inteval $[-2^{16},2^{16}]$.

Output

The output should be given as an ordered (increasing) list of integers.

Sample Inputs.

Input	Output
`[12,44,3,-1,34,11]` `[5,0,12,3,55,34]`	`[-1,0,5,11,44,55]`
`[4,8,12,31,43,42,0,-10,234,1123,443,221]` `[8,2,31,43,123,4211,0,-110,113,43,221]`	`[-110,-10,2,4,12,42,43,113,123,234,443,1123,4211]`

Interleaving

Given two lists of numbers from 1 to 9, of the same length, your program should compute the interleaving of the two lists that gives the largest number when read as a single decimal number.

For example, for the following lists of length 5:

[1, 4, 2, 5, 3]
[4, 5, 1, 3, 1]

their maximum interleaving is:

[4, 5, 1, 4, 2, 5, 3, 1, 3, 1]

corresponding to the decimal number 4514253131.

Input

The input consists of two lines, each line containing a list of digits in the range 1-9. The lists are guaranteed to have the same length $1 \leq L \leq 100$.

Output

Your program should output the interleaving as a single list of numbers.

Sample Inputs.

Input	Output
`[1,4,2,5,3]` `[4,5,1,3,1]`	`[4,5,1,4,2,5,3,1,3,1]`
`[1,2,1,2,1]` `[2,1,2,1,2]`	`[2,1,2,1,2,1,2,1,2,1]`

Pirates

A group of *N* pirates have to share *K* coins amongst themselves. The pirates have some constraints that need to be satisfied. The constraints are of the form

pirate *i* must receive *X* pirate *j*

where *i, j* are different numbers in the set *{1..N}*, and *X* is one of the following:

- more than
- one more than
- twice as much as

Moreover, each pirate must receive at least one coin; and all coins must be shared. Decide whether a sharing is possible that satisfies all the given constraints.

Input

The first line of the input will give the number of pirates $1 \leq N \leq 10$. The second line of the input gives the number of coins to be shared $0 \leq K \leq 20$. The third line describes the number of constraints *C* that need to be satisfied. The following *C* lines specify each of the constraints. These lines follow the pattern shown in the sample input below.

Output

Your program should output either "Yes" or "No", depending on whether a sharing that satisfies all the constraints is possible or not.

Sample Inputs.

Input	Output
6 pirates 10 coins 2 constraints: - pirate 1 must receive more than pirate 2 - pirate 2 must receive one more than pirate 3	Yes
6 pirates 10 coins 3 constraints: - pirate 1 must receive twice as much as pirate 2 - pirate 2 must receive one more than pirate 3 - pirate 3 must receive more than pirate 1	No