This is the problem set for Day 5 (PM).

It includes topics covered during the week in preparation for the final contest session.

This problem set has **9 problems**. The items in the problem list below are links to each problem. You can click a problem name to jump to its page in the PDF file.

Problem A: Seriously, Is it Sorted?

Given a list of N integers, sort them in **descending order.**

It's hard to do it manually, but with the power of programming, sorting is just one line away from your reach!

Input
The first line of input contains an integer N ($1 \leq N \leq 100$).
The second line of input contains N integers, separated by single spaces. Each integer is within the range 0 to 100000000.

Output
Output the sorted list of integers, one integer on each line.

Sample Input #1	Sample Output #1
5	51
4 51 13 13 2	13
	13
	4
	2

Sample Input #2	Sample Output #2
3	5
4 5 3	4
	3

Sample Input #3	Sample Output #3
4	4
1 2 3 4	3
	2
	1

Problem B: Data Plan

Chris needs to add networking to his Tetris game.
He plans to send the data using strings. It's not the most efficient way, but it works.

In order to gauge how much data he has to send, he need to know how many characters are in the string he's sending.

Input

The first line of input contains an integer $T(1 \leq T \leq 10)$ which is the number of lines Chris wants to send.
The next lines of input are the lines Chris wants to send. Each line contains at most 100 characters and can only contain letters. The lines are given in order from line 1 to line N.

Output

For each line, output:
```
Line X is Y characters long.
```
where X is the line number and Y is the number of characters in that line.

Sample Input #1
```
2
GameStartSignal
PieceData10101end
```

Sample Output #1
```
Line 1 is 15 characters long.
Line 2 is 17 characters long.
```

Sample Input #2
```
3
A
Abc
Abcd
```

Sample Output #2
```
Line 1 is 1 characters long.
Line 2 is 3 characters long.
Line 3 is 4 characters long.
```

Problem C: Find the Median

Given a set of *N* doubles, print the median of the set.

What's the median?
For example, if you have the following set:

        5.0, 12.0, 3.0

First, you have to sort them in ascending order.

        3.0, 5.0, 12.0

The median is the middle number. In this case, it's the number at index 1: 5.0.

How about if there are an even number of elements?

        4.0, 3.2, 1.0, 0.5

Sort it:

        0.5, 1.0, 3.2, 4.0

Your two middle numbers are the elements at indices 1 and 2: 1.0 and 3.2. Get the average between them:

        1.0 + 3.2 = 4.2
        4.2/2 = 2.1

Input
The first line contains an integer $N(1 \leq N \leq 100)$, the number of numbers in the list.
This is followed by another line containing the N real numbers you have to find the median of, separated by spaces. The numbers will be expressed to a single decimal place.

Output
Output consists of only one line, which contains the median of the set, to four decimal places.

Sample Input #1	Sample Output #1
5	3.0000
9.1 83.0 2.9 0.0 3.0	

Sample Input #2	Sample Output #2
7	7.0000
1.0 2.0 3.5 8.2 7.0 22.0 32.0	

Sample Input #3	Sample Output #3
4	3.5000
5.0 33.4 2.0 1.2	

Problem D: ASCII GUI

You're given a task: to make a program that formats text neatly with line numbers and borders. Seems simple enough, but then there's another instruction: order the text in reverse. Can you do it?

Input
Input consists of exactly 5 lines of text. Each line contains at most 100 characters.

Output
Output consists of 7 lines.
The first and last lines contain 20 dashes, creating a horizontal border.
```
--------------------
```
The five lines in between contain the 5 lines of input, in reverse order, following this format:
```
<line number>:<line>
```
Line numbers start counting from 1. See the sample input and output.

Sample Input #1
```
apple
banana
I like cavendish bananas
dog
elephant
```

Sample Output #1
```
--------------------
5:elephant
4:dog
3:I like cavendish bananas
2:banana
1:apple
--------------------
```

Sample Input #2
```
Lorem ipsum dolor
sit amet, consectetur
adipiscing elit.
Pellentesque tempor
mauris felis,
```

Sample Output #2
```
--------------------
5:mauris felis,
4:Pellentesque tempor
3:adipiscing elit.
2:sit amet, consectetur
1:Lorem ipsum dolor
--------------------
```

Problem E: Mean Ranges

Sir Diy is teaching class today!

In statistics, two important values that can be obtained from a group of data are called the mean and range.

The mean is the average of all the given values, while the range is the difference between the highest value and the lowest one.

Given a set of numbers, your task is to print their respective ranges and means.

Input

The first line contains an integer N ($1 \leq N \leq 100$) which states the number of values in the data.
The second line contains the N values, which are all integers ranging from 1 to 100000.

Output

Print the range and the mean, in that order, separated by a single space, in one line.
Take note that the value of the mean may be a decimal number, so output the mean up to two decimal places.

Sample Input #1	Sample Output #1
4	3 2.50
1 2 3 4	

Sample Input #2	Sample Output #2
5	4 3.00
1 2 3 4 5	

Sample Input #3	Sample Output #3
6	0 1.00
1 1 1 1 1 1	

Problem F: Dictionary Time

I will give you a list of words. If I looked for those words in a dictionary, will I find the words in the same order?

Input

First line of input contains a single number $N(1 \le N \le 15)$ which indicates the number of words I'll search.

The next N lines will contain the words.

All words will be spelled with lowercase English letters.

Output

Simply output "yes" if the words are arranged as they would be in a dictionary; output "no" if they're not; without the quotation marks.

Sample Input #1
```
5
abecedarian
exsibilation
inaniloquent
onycophagy
usufructuary
```
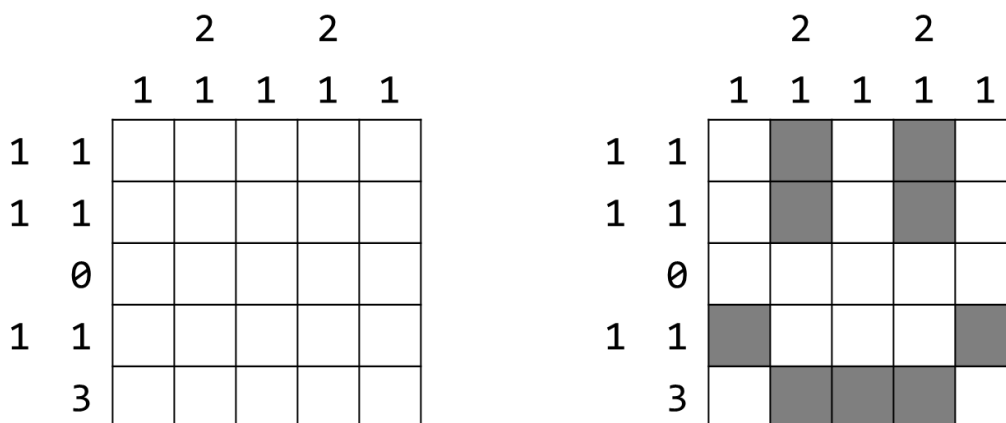
Sample Output #1
```
yes
```

Sample Input #2
```
5
i
dont
want
no
order
```

Sample Output #2
```
no
```

Problem G: Nonogram Hints

A **nonogram** is a type of logic puzzle that involves a grid and hints for each row and column.

Above is a nonogram puzzle. On the left is an unsolved puzzle, and on the right is a solved puzzle. Each cell in the grid can be either shaded or unshaded, and a puzzle is solved by shading the correct cells.

The hints for each row and column indicate how many groups of cells should be shaded, and how large each group should be. Two shaded cells are part of the same group if there are no unshaded cells between them.

For example, consider the following two rows:
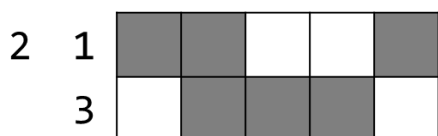
The first row has 2 hints: **2 1**
This means that there should be 2 groups of shaded cells in that row, one group of **2 cells** followed by another group containing **1 cell**.

The second row has 1 hint: **3**
This means that there should be a single group of **3 cells** in that row.

Here are examples of rows which are valid and invalid.
Row **a** and **b** are valid because the shaded squares follow the hints.
Row **c** is invalid because it only has a group containing 1 cell, while the hint indicates that the column should have a group of 2 cells followed by a group containing 1 cell.
Row **d** is invalid because the groups are in the wrong order.

Row **e** is invalid because it contains a single group of 3 cells, not following the hints.

Your task is to create a program that determines if a single given row is valid or invalid, given the hints for the row.

Input

The first line of input contains 2 integers, **N** and **M** ($1 \leq N, M \leq 100$).

N indicates the number of cells in the row.

M indicates the number of hints.

The second line of input contains **M** positive integers separated by spaces, which are the hints (h_i) in order from h_1 to h_M.

It is guaranteed that **M** minus one plus the sum of all the hints will be less than or equal to **N**.

This is followed **N** characters describing the row. A period (`.`) indicates the cell is unshaded, while an octothorpe (`#`) indicates the row is shaded.

Output

The output consists of a single line. Output `"Valid!"` (without the quotes) if the given row is valid, and output `"Invalid."` if the given row is invalid. All lines of output must end with a newline character.

Sample Input 1:
```
10 3
2 1 4 ##.#..####
```

Sample Output 1:
```
Valid!
```

Sample Input 2:
```
5 1
2 ..#.#
```

Sample Output 2:
```
Invalid.
```

The first sample input refers to this row (which is valid):

2 1 4

The second sample input refers to this row (which is invalid):

2

Problem H: Top Target

Commander Diy is assembling an army to attack the alien invaders that have decided to set foot on his home world. To guide him in assessing the enemy composition, he activates his trusty Scanner to list down all the surrounding enemy units.

The Commander wants to know what to focus his own army on. It would be a great advantage for him if he knew what kind of unit comprises most of the alien invaders. You have been chosen to design a program that would perform this task.

Make a program that will output the name of the top target, which is the enemy unit that appears most frequently in the list of enemy units. It is guaranteed that there is only one top target, and the inputs are case sensitive.

Input
The first line contains an integer N ($1 \leq N \leq 100$). N lines follow, each containing a name of an enemy unit.

Output
Output consists of two lines containing the name of the top target, then the number of times it appears. See the sample output for the format.

Sample Input	Sample Output
10	TOP TARGET: Soul Harvester
Soul Harvester	APPEARS 4 TIME(S)
Cyborg	
Mutant	
Soul Harvester	
Mutant	
Cyborg	
Soul Harvester	
Bug	
Bug	
Soul Harvester	

| Sample Input	Sample Output
4	TOP TARGET: Cyborg
Soul Harvester	APPEARS 2 TIME(S)
Cyborg	
Mutant	
Cyborg	

Problem I: Panel Puzzle

Toby is trying out ideas for a new game.

In order to make this game work, he has to make sure that his rotation function works correctly.

In this problem, you'll be given 2 panels of identical size. The first panel has blue lights, the second panel has red lights.

Two panels can be matched if all the red squares on the red panel can fill up any empty squares on the blue panel. **The red panel may be rotated 90, 180, or 270 degrees.**
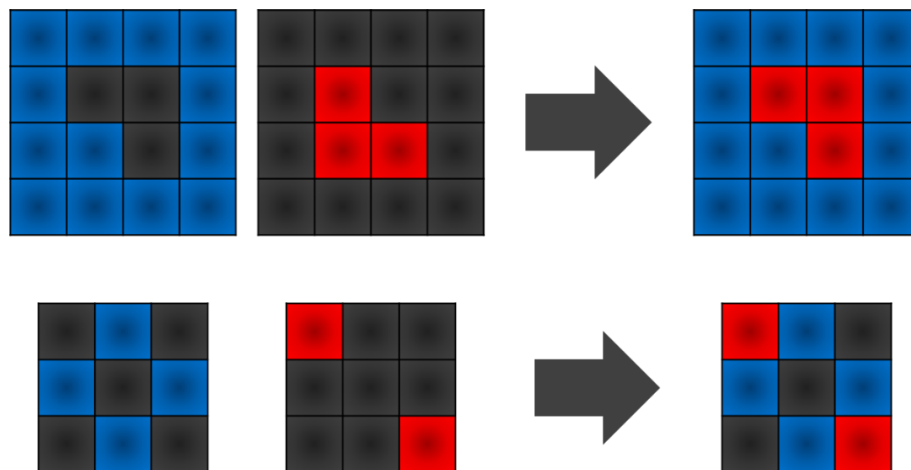
Note that not all empty squares have to be filled.

The panels have to line up exactly. See the examples below:

no match
(no matter how you rotate it)
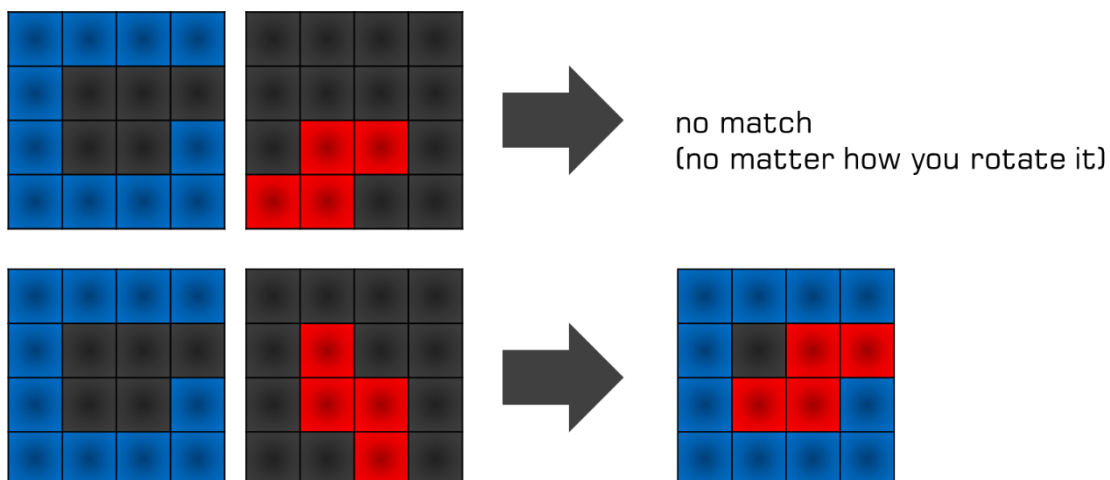
ASPC-A 2023 Day 5 PM Problem Set

Input

The first line of input contains an integer N ($1 \leq N \leq 10$), the size of the panels.
$2N$ lines follow, each containing N characters. `'#'` denotes a colored square, `'.'` denotes an empty square.
The first N lines that follow describe the blue panel.
The next N lines describe the red panel.

Output

Output `"Yes"` if the panels can be matched, and `"No"` otherwise.

Sample Input 1	Sample Output 1
4	Yes
####	
#..#	
##.#	
####	
....	
.##.	
.#..	
....	

Sample Input 2	Sample Output 2
3	Yes
.#.	
#.#	
.#.	
#..	
...	
..#	

Sample Input 3	Sample Input 3
4	No
####	
##..	
#..#	
####	
....	
#...	
##..	
.#..	

Sample Input 4	Sample Output 4
3	Yes
###	
#..	
..#	
.#.	
.##	

..#