Review of Previous contents:

- Pass into functions with objects (pass address instead of values). Example 2019Q4
- Recursive application. Example 2019Q3
- String Manipulation Example 2020Q4

Homework:

- Geometric analysis Example 2020 Q3

2019Q4

Problem J4/S1: Flipper

Problem Description
You are trying to pass the time while at the optometrist. You notice there is a grid of four numbers:

1	2
3	4

You see lots of mirrors and lenses at the optometrist, and wonder how flipping the grid horizontally or vertically would change the grid.

Specifically, a "horizontal" flip (across the horizontal centre line) would take the original grid of four numbers and result in:

3	4
1	2

A "vertical" flip (across the vertical centre line) would take the original grid of four numbers and result in:

2	1
4	3

Your task is to determine the final orientation of the numbers in the grid after a sequence of horizontal and vertical flips.

Input Specification
The input consists of one line, composed of a sequence of at least one and at most 1 000 000 characters. Each character is either `H`, representing a horizontal flip, or `V`, representing a vertical flip.

For 8 of the 15 available marks, there will be at most 1 000 characters in the input.

Output Specification
Output the final orientation of the four numbers. Specifically, each of the two lines of output will contain two integers, separated by one space.

Sample Input 1
```
HV
```

Output for Sample Input 1
```
4 3
2 1
```

La version française figure à la suite de la version anglaise.

Sample Input 2
```
VVHH
```

Output for Sample Input 2
```
1 2
3 4
```

Problem J3: Cold Compress

Problem Description

Your new cellphone plan charges you for every character you send from your phone. Since you tend to send sequences of symbols in your messages, you have come up with the following compression technique: for each symbol, write down the number of times it appears consecutively, followed by the symbol itself. This compression technique is called *run-length encoding*.

More formally, a block is a substring of identical symbols that is as long as possible. A block will be represented in compressed form as the length of the block followed by the symbol in that block. The encoding of a string is the representation of each block in the string in the order in which they appear in the string.

Given a sequence of characters, write a program to encode them in this format.

Input Specification

The first line of input contains the number N, which is the number of lines that follow. The next N lines will contain at least one and at most 80 characters, none of which are spaces.

Output Specification

Output will be N lines. Line i of the output will be the encoding of the line $i + 1$ of the input. The encoding of a line will be a sequence of pairs, separated by a space, where each pair is an integer (representing the number of times the character appears consecutively) followed by a space, followed by the character.

Sample Input

```
4
+++===!!!!
777777......TTTTTTTTTTTT
(AABBC)
3.1415555
```

Output for Sample Input

```
3 + 3 = 4 !
6 7 6 . 12 T
1 ( 2 A 2 B 1 C 1 )
1 3 1 . 1 1 1 4 1 1 4 5
```

Explanation of Output for Sample Input

To see how the first message (on the second line of input) is encoded, notice that there are 3 + symbols, followed by 3 = symbols, followed by 4 `!` symbols.

Problem J4: Cyclic Shifts

Problem Description

Thuc likes finding cyclic shifts of strings. A *cyclic shift* of a string is obtained by moving characters from the beginning of the string to the end of the string. We also consider a string to be a cyclic shift of itself. For example, the cyclic shifts of `ABCDE` are:

`ABCDE`, `BCDEA`, `CDEAB`, `DEABC`, and `EABCD`.

Given some text, T, and a string, S, determine if T contains a cyclic shift of S.

Input Specification

The input will consist of exactly two lines containing only uppercase letters. The first line will be the text T, and the second line will be the string S. Each line will contain at most 1000 characters.

For 6 of the 15 available marks, S will be exactly 3 characters in length.

Output Specification

Output `yes` if the text, T, contains a cyclic shift of the string, S. Otherwise, output `no`.

Sample Input 1

```
ABCCDEABAA
ABCDE
```

Output for Sample Input 1

```
yes
```

Explanation of Output for Sample Input 1

`CDEAB` is a cyclic shift of `ABCDE` and it is contained in the text `ABC`**`CDEAB`**`AA`.

Sample Input 2

```
ABCDDEBCAB
ABA
```

Output for Sample Input 2

```
no
```

Explanation of Output for Sample Input 2

The cyclic shifts of `ABA` are `ABA`, `BAA`, and `AAB`. None of these shifts are contained in the text `ABCDDEBCAB`.

Problem J3: Art

Problem Description

Mahima has been experimenting with a new style of art. She stands in front of a canvas and, using her brush, flicks drops of paint onto the canvas. When she thinks she has created a masterpiece, she uses her 3D printer to print a frame to surround the canvas.

Your job is to help Mahima by determining the coordinates of the smallest possible rectangular frame such that each drop of paint lies inside the frame. Points on the frame are not considered inside the frame.

Input Specification

The first line of input contains the number of drops of paint, N, where $2 \leq N \leq 100$ and N is an integer. Each of the next N lines contain exactly two positive integers X and Y separated by one comma (no spaces). Each of these pairs of integers represents the coordinates of a drop of paint on the canvas. Assume that $X < 100$ and $Y < 100$, and that there will be at least two distinct points. The coordinates $(0, 0)$ represent the bottom-left corner of the canvas.

For 12 of the 15 available marks, X and Y will both be two-digit integers.

Output Specification

Output two lines. Each line must contain exactly two non-negative integers separated by a single comma (no spaces). The first line represents the coordinates of the bottom-left corner of the rectangular frame. The second line represents the coordinates of the top-right corner of the rectangular frame.

Sample Input

```
5
44,62
34,69
24,78
42,44
64,10
```

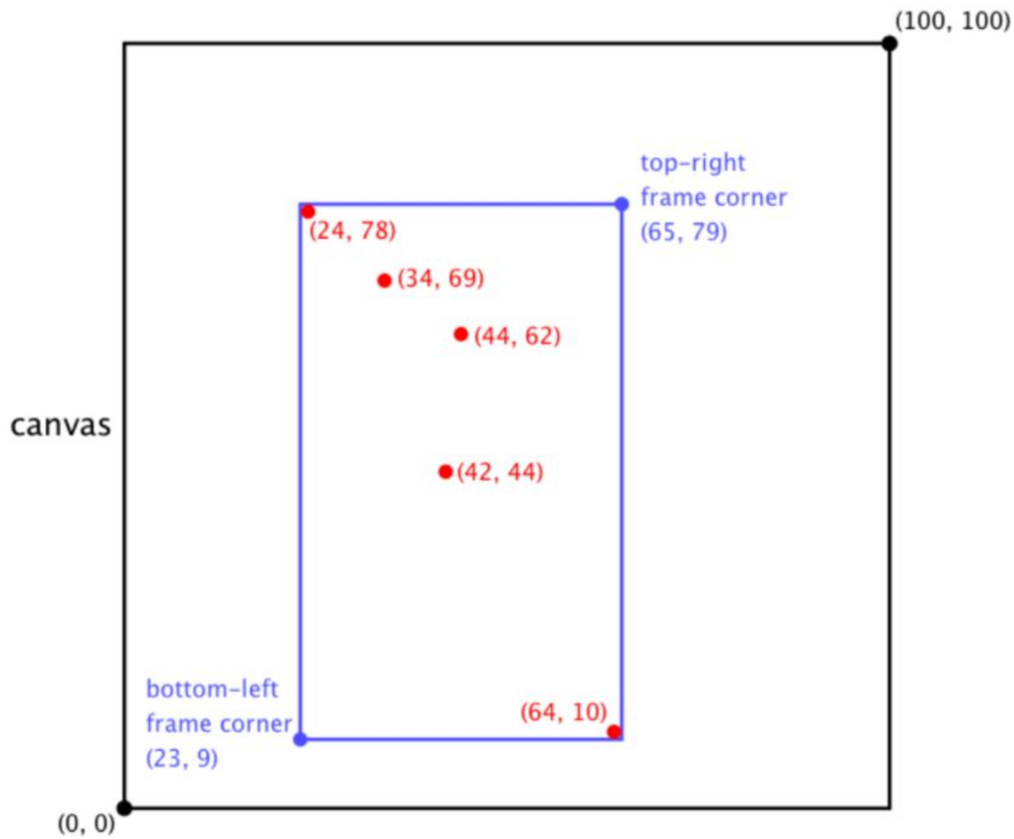
Output for Sample Input

```
23,9
65,79
```

Explanation of Output for Sample Input

The bottom-left corner of the frame is $(23, 9)$. Notice that if the bottom-left corner is moved up, the paint drop at $(64, 10)$ will not be inside the frame. (See the diagram on the next page.) If the corner is moved right, the paint drop at $(24, 78)$ will not be inside the frame. If the corner is moved down or left, then the frame will be larger and no longer the smallest rectangle containing all the drops of paint. A similar argument can be made regarding the top-right corner of the frame.

La version française figure à la suite de la version anglaise.


(100, 100)
top-right
frame corner
(65, 79)
(24, 78)
(34, 69)
(44, 62)
canvas
(42, 44)
bottom-left
frame corner
(23, 9)
(64, 10)
(0, 0)