Question 1

You record all of the scoring activity at a basketball game. Points are scored by a 3-point shot, a 2-point field goal, or a 1-point free throw.

You know the number of each of these types of scoring for the two teams: the **Apples** and the **Bananas**. Your job is to determine which team won, or if the game ended in a tie.

**Input Specification:** The first line of input describe the scoring of the Apples, and the next line of input describe the scoring of the Bananas. For each team, the line contains the number of successful 3-point shots, successful 2-point field goals, and the number of successful 1-point free throws. Each number will be an integer between 0 and 100, inclusive.

**Output Specification:** The output will be a single character. If the Apples scored more points than the Bananas, output A. If the Bananas scored more points than the Apples, output B. Otherwise, output T, to indicate a tie.

**Sample Input 1**

10 3 7

8 9 6

**Output for Sample Input 1**

B

**Input for Sample Input 2**

7 3 0

6 4 1

**Output for Sample Input 2**

T

Question 2

**Problem Description:** You are playing a game with your friend to send messages back and forth. Your friend can encode a message to you by writing down a positive integer *N* and a symbol. You need to decode that message by writing out that symbol *N* times in a row on one line. Given a message that your friend has encoded, decode it.

**Input Specification**

The first line of input contains *L*, the number of lines in the message. The next *L* lines each contain one positive integer less than 80, followed by one space, followed by a (non-space) character.

**Output Specification**

The output should be *L* lines long. Each line should contain the decoding of the corresponding line of the input. Specifically, if line *i* + 1 of the input contained N x, then line *i* of the output should contain just the character x printed N times.

**Sample Input**

3

9 +

3 -

12 A

**Output for Sample Input**

+++++++++

---

AAAAAAAAAAAA

Question 3

**Problem Description:** Your new cell phone 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

Question 4

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



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:



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



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 100 characters. Each character is either H, representing a horizontal flip, or V, representing a vertical flip.

**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

**Sample Input 2**

VVHH

**Output for Sample Input 2**

1 2

3 4

Question 5

Given a string text, you want to use the characters of text to form as many instances of the word "balloon" as possible. You can use each character in text at most once. Return the maximum number of instances that can be formed.

Example 1:



Input: text = "nlaebolko"

Output: 1

Example 2:

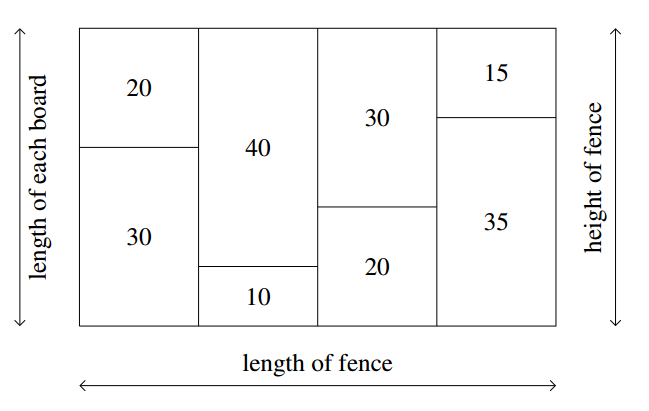


Input: text = "loonbalxballpoon"

Output: 2

Question 6

**Problem Description：** Tudor is a contestant in the Canadian Carpentry Challenge (CCC). To win the CCC, Tudor must demonstrate his skill at nailing wood together to make the longest fence possible using boards. To accomplish this goal, he has *N* pieces of wood. The *ith* piece of wood has integer length *Li*. A *board* is made up of **exactly two** pieces of wood. The length of a board made of wood with lengths *Li* and *Lj* is *Li* + *Lj*. A fence consists of boards that are the same length. The length of the fence is the number of boards used to make it, and the height of the fence is the length of each board in the fence. In the example fence below, the length of the fence is 4; the height of the fence is 50; and, the length of each piece of wood is shown:



Tudor would like to make the longest fence possible. Please help him determine the maximum length of any fence he could make, and the number of different heights a fence of that maximum length could have.

**Input Specification**

Please define a list contains all woods’ length data

**Output Specification**

Output two integers on a single line separated by a single space: the length of the longest fence and the number of different heights a longest fence could have.