TASK	BAKA	SUME	DOBRI	BUREK	JEDAN	BAKTERIJE

source code	`baka.pas`
	`baka.c`
	`baka.cpp`
		`sume.pas`
		`sume.c`
		`sume.cpp`
			`dobri.pas`
			`dobri.c`
			`dobri.cpp`
				`burek.pas`
				`burek.c`
				`burek.cpp`
					`jedan.pas`
					`jedan.c`
					`jedan.cpp`
						`bakterije.pas`
						`bakterije.c`
						`bakterije.cpp`

input	standard input (*stdin*)					
output	standard output (*stdout*)					
time limit	1 second	1 second	1 second	1 second	1 second	1 second
memory limit	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
point value	**50**	**80**	**100**	**120**	**140**	**160**
	650					

Problems translated from Croatian by: **Ivan Pilat**

Mirko's grandma still uses an ancient pulse dial telephone with a rotary dial as shown in the following picture:

For each digit that we want to dial, we need to turn the rotary dial clockwise until the chosen digit reaches the finger stop (metal fin). Then we let go of the dial and wait for it to return to its original position before we can dial another digit. In our modern, instant gratification world, the dial return often lasts much longer than our patience. More precisely, dialling the digit 1 takes a total of two seconds, while dialling any larger digit takes an additional second for each additional finger circle counting from 1 to the dialled digit (as shown in the picture).

Mirko's grandma remembers phone numbers by memorizing a corresponding word which, when dialled, results in the correct number being dialled. When dialling a word, for each letter, we dial the digit which has that letter written next to it on the dial (for example, the digit 7 for the letter S). For example, the word UNUCIC[1] corresponds to the number 868242. Your task is determining, for a given word, the **total time** required to dial that word.

INPUT

The first and only line of input contains a single word consisting of between 2 and 15 (inclusive) uppercase English letters.

OUTPUT

The first and only line of output must contain the required dialling time.

SCORING

In test data worth at least 30% of total points, the input word will contain only vowels.

In test data worth an *additional* 30% of total points, the input word will contain only letters smaller than P.

1 'Little grandson' in Croatian.

SAMPLE TESTS

input	**input**
WA	UNUCIC
output	**output**
13	36

Clarification of the first example: The corresponding phone number is 92, which needs 10 + 3 seconds to dial.

Once upon a time, there existed a sequence **A** consisting of **N** positive integers. You don't know the sequence itself, but you do know the sum of every two elements of the sequence. Find the sequence **A**!

INPUT

The first line of input contains the positive integer **N** ($2 \leq N \leq 1000$).

Each of the following **N** lines contains **N** positive integers smaller than or equal to 100 000, forming the table **S**. The following relations hold: **S**(**i**, **j**) = **A**[**i**] + **A**[**j**] for **i** ≠ **j**, and **S**(**i**, **j**) = 0 for **i** = **j**. Here **S**(**i**, **j**) denotes the number in the **i**th row and **j**th column of the table, and **A**[**i**] denotes the **i**th element of the sequence **A**.

It is guaranteed that for any input data set there exists a **unique sequence of positive integers A** with the given properties.

OUTPUT

The first and only line of output must contain the required sequence **A** (in the form of **N** space-separated positive integers).

SAMPLE TESTS

input	**input**
2 0 2 2 0	4 0 3 6 7 3 0 5 6 6 5 0 9 7 6 9 0
output	**output**
1 1	2 1 4 5

You are given a sequence **A** consisting of **N** integers (not to be confused with the sequence from the previous task). We will call the **i**th sequence element **good** if it equals the sum of some **three** elements in positions strictly smaller than **i** (an element can be used more than once in the sum).

How many good elements does the sequence contain?

INPUT

The first line of input contains the positive integer **N** ($1 \leq \mathbf{N} \leq 5000$), the length of the sequence **A**.

The second line of input contains **N** space-separated integers representing the sequence **A** ($-100\,000 \leq \mathbf{A}_i \leq 100\,000$).

OUTPUT

The first and only line of output must contain the number of good elements in the sequence.

SCORING

In test data worth at least 40% of total points, $\mathbf{N} \leq 50$.

In test data worth at least 70% of total points, $\mathbf{N} \leq 500$.

SAMPLE TESTS

input	**input**	**input**
2 1 3	6 1 2 3 5 7 10	3 -1 2 0
output	**output**	**output**
1	4	1

Baker Crumble has just baked **N** triangular burek[2] pastries. Each pastry can be represented in the Cartesian coordinate system as a triangle with vertices in integer coordinate points.

The baker's mischievous son Joey has just taken a large knife and started to cut the pastries. Each cut that Joey makes corresponds to a horizontal (y = **c**) or vertical (x = **c**) line in the coordinate system. Help the baker assess the damage caused by Joey's pastry cutting. Your task is to determine, for each Joey's cut, how many pastries are affected (such that both the left and right parts of the cut pastry have areas greater than zero).

INPUT

The first line of input contains the positive integer **N** ($2 \leq \mathbf{N} \leq 100\,000$), the number of burek pastries.

Each of the following **N** lines contains six nonnegative integers smaller than 10^6. These numbers are, in order, the coordinates ($\mathbf{x_1}$, $\mathbf{y_1}$), ($\mathbf{x_2}$, $\mathbf{y_2}$), ($\mathbf{x_3}$, $\mathbf{y_3}$) of the three pastry-triangle vertices. The three vertices will not all be on the same line. The pastries can touch as well as overlap.

The following line contains the positive integer **M** ($2 \leq \mathbf{M} \leq 100\,000$), the number of cuts.

Each of the following **M** lines contains a single cut line equation: "x = **c**" or "y = **c**" (note the spaces around the equals sign), where **c** is a nonnegative integer smaller than 10^6.

OUTPUT

For each cut, output a line containing the required number of cut pastries.

SCORING

In test data worth at least 40 points, $\mathbf{M} \leq 300$.

In test data worth an *additional* 40 points, the vertex coordinates of all triangles will be smaller than 1000.

2 Turkish/Balkanian flaky dough pastry, often filled with cheese or minced meat.

SAMPLE TESTS

input

```
3
1 0 0 2 2 2
1 3 3 5 4 0
5 4 4 5 4 4
4
x = 4
x = 1
y = 3
y = 1
```

output

```
0
1
1
2
```

input

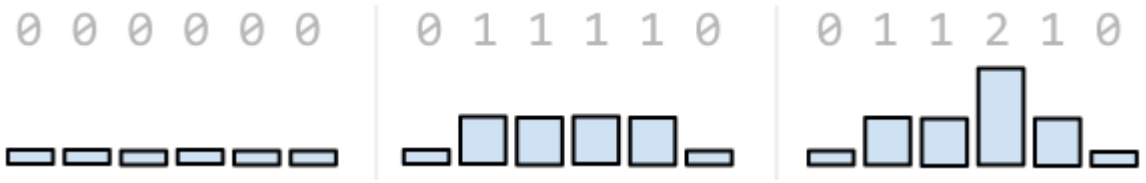
```
4
2 7 6 0 0 5
7 1 7 10 11 11
5 10 2 9 6 8
1 9 10 10 4 1
4
y = 6
x = 2
x = 4
x = 9
```

output

```
3
2
3
2
```

Some people like to pretend that they are a pharaoh. Or a dolphin. Luka is one such person.

He has built a relief consisting of a long line of **N** columns with nonnegative integer heights. The heights of all columns were initially zero. The relief was build in steps, where in each step Luka would select a **contiguous subsequence of columns with equal heights** and raise all columns in the subsequence, **except** the first and last column, by **one**.



Hundreds of years have passed, and **some of the columns** have been **stolen**. Luka's great-great-...-great-grandson is trying to determine the number of possible reliefs that could have been built by Luka such that the remaining columns' heights match the original relief.

INPUT

The first line of input contains the positive integer **N** ($1 \leq \mathbf{N} \leq 10\,000$), the number of columns in Luka's relief.

The second line of input contains **N** space-separated integers $\mathbf{h}_i$ ($-1 \leq \mathbf{h}_i \leq 10\,000$), the column heights. A height of -1 represents a stolen column.

OUTPUT

The first and only line of output must contain the required number of possible reliefs modulo 1 000 000 007.

SAMPLE TESTS

input	**input**	**input**
3 -1 2 -1	3 -1 -1 -1	6 -1 -1 -1 2 -1 -1
output	**output**	**output**
0	2	3

Luka is bored in chemistry class yet again. This time, he's playing with intelligent bacteria. He has arranged his **K** bacteria on a rectangular board divided in **N** rows, labelled with numbers from 1 to **N** starting from the top, and **M** columns, labelled with numbers from 1 to **M** starting from the left.

Each bacterium begins its adventure in a certain cell, facing one of the four neighbouring cells, and carries out the following actions every second:

1. Reads the number **X** dedicated to that bacterium in the current cell.
2. Turns 90 degrees clockwise, **X** times.
3. If it is facing a cell outside the board, it turns 180 degrees.
4. Finally, it moves to the cell that it is facing.

Luka has placed a trap in one cell. The trap will activate and kill the bacteria as soon as they **all** step on that cell in the **same second**.

Since Luka only has two hours of chemistry class today, help him determine how long the game will last, in seconds.

INPUT

The first line of input contains the positive integers **N** ($3 \leq \mathbf{N} \leq 50$), **M** ($3 \leq \mathbf{M} \leq 50$), and **K** ($1 \leq \mathbf{K} \leq 5$).

The second line of input contains the positive integers **X** and **Y**, the row and column where Luka has placed the trap.

The remainder of the input consists of bacteria descriptions, for each bacterium **i** from 1 to **K**:

- two positive integers $\mathbf{X}_i$, $\mathbf{Y}_i$ – the row and column of the starting cell of bacterium **i**, and the character $\mathbf{C}_i$ representing the starting direction that the bacterium is facing (U – up, R – right, D – down, L – left).

- **N** by **M** matrix of digits between 0 and 9, inclusive; the digit in row **x** and column **y** represents the number in cell (**x**, **y**) dedicated to bacterium **i**.

OUTPUT

The first and only line of output must contain the total duration of Luka's game, in seconds. If the game will never end, output -1.

SAMPLE TESTS

input	**input**	**input**
3 3 1 2 2 1 1 R 010 000 000	3 4 2 2 2 3 4 R 2327 6009 2112 3 2 R 1310 2101 1301	4 4 3 4 3 1 1 U 1001 0240 3322 2327 1 3 L 9521 2390 3020 2421 2 2 D 3397 2013 1102 7302
output	**output**	**output**
3	8	296