TASK	**RIJEČI**	**OKVIR**	**REČENICE**	**KOLINJE**	**PAROVI**	**ODAŠILJAČI**

source code	`rijeci.pas`
	`rijeci.c`
	`rijeci.cpp`
		`okvir.pas`
		`okvir.c`
		`okvir.cpp`
			`recenice.pas`
			`recenice.c`
			`recenice.cpp`
				`kolinje.pas`
				`kolinje.c`
				`kolinje.cpp`
					`parovi.pas`
					`parovi.c`
					`parovi.cpp`
						`odasiljaci.pas`
						`odasiljaci.c`
						`odasiljaci.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	64 MB	32 MB
point value	**50**	**80**	**100**	**120**	**140**	**160**
	650					

Problems translated from Croatian by: **Paula Gombar**

One day, little Mirko came across a funny looking machine! It consisted of a **very very large** screen and a single button. When he found the machine, the screen displayed only the letter A. After he pressed the button, the letter changed to B. The next few times he pressed the button, the word transformed from B to BA, then to BAB, then to BABBA... When he saw this, Mirko realized that the machine alters the word in a way that all the letters B get transformed to BA and all the letters A get transformed to B.

Amused by the machine, Mirko asked you a very difficult question! After **K** times of pressing the button, how much letters A and how much letters B will be displayed on the screen?

INPUT

The first line of input contains the integer **K** ($1 \le \mathbf{K} \le 45$), the number of times Mirko pressed the button.

OUTPUT

The first and only line of output must contain two space-separated integers, the number of letters A and the number of letter B.

SCORING

In test data worth 20% of total points, **K** will be less or equal to 10.

SAMPLE TESTS

input 1 **output** 0 1	**input** 4 **output** 2 3	**input** 10 **output** 34 55

Mirko has assembled an excellent crossword puzzle and now he wants to frame it. Mirko's crossword puzzle consists of **M** x **N** letters, and the frame around it should be **U** characters wide on top, **L** characters on the left, **R** characters on the right and **D** characters on the bottom side.

The frame consists of characters # (hash) and . (dot) which alternate like fields on a chessboard. These characters should be arranged in a way that, if the frame is expanded to cover the entire crossword puzzle and we treat these characters as a chessboard, the # characters should be placed as the red fields on a chessboard (i.e. the top left field). See the examples below for a better understanding of the task.

INPUT

The first line of input contains two integers **M** and **N** ($1 \le$ **M**, **N** ≤ 10).

The second line of input contains integers **U**, **L**, **R**, **D** ($0 \le$ **U**, **L**, **R**, **D** ≤ 5).

The following **M** lines of input contains **N** characters – lowercase letters of the English alphabet. These lines represent Mirko's crossword puzzle.

OUTPUT

Output the framed crossword puzzle as stated in the text.

SAMPLE TESTS

<table>
<tr>
<td>

input

```
4 4
2 2 2 2
honi
oker
nera
irak
```

output

```
#.#.#.#.
.#.#.#.#
#.honi#.
.#oker.#
#.nera#.
.#irak.#
#.#.#.#.
.#.#.#.#
```

</td>
<td>

input

```
2 4
1 0 3 1
rima
mama
```

output

```
#.#.#.#
rima.#.
mama#.#
.#.#.#.
```

</td>
</tr>
</table>

Once upon a time in a land far far away, inhabited only by math students, Iva and Vedran were discussing self-explanatory sentences. A part of these sentences is exactly **one number** and it is **equal to the total number of letters in the sentence**. Some examples are: "This sentence has thirtyone letters.", "Blah blah seventeen".

Little Jurica overheard his friends' conversation and decided to impress them with the amount of self-explanatory sentences he knows by heart. He rushed back home and wrote a programme which will, given a sentence, tell him the minimum number he can put inside so that the sentence is valid. Unfortunately, his computer broke and now he needs your help. Write a programme to help Jurica!

The form of the sentence is: *word1 word2 word3* … $ *word_n-1 word_n*. The character $ represents the place where the number should be put in.
For example, the form of the sentence "this sentence has thirtyone letters" would be "this sentence has $ letters".

The rules that apply to writing numbers are the following:

- numbers from 1 to 10 are written "one", "two", "three", "four", "five", "six", "seven", "eight", "nine", "ten", respectively
- numbers from 11 to 19 are written "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"
- the remaining double digit numbers are written in a way that we name the tens' digit and add to it the name of the one digit remaining when we remove the tens' digit. Specially, if by removing the tens' digit we remain with zero, we add nothing to it
- the tens' digits (respectively from 2 to 9) are named the following: "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"
- three digit numbers are written in a way that we name the hundreds' digit number and add to it the number of the double digit number remaining. Specially, if by removing the hundreds' digit we remain with zero, we add nothing to it
- the hundreds' digits (respectively from 1 to 9) are named the following: "onehundred", "twohundred", "threehundred", "fourhundred", "fivehundred", "sixhundred", "sevenhundred", "eighthundred", "ninehundred"
- the rules that apply for numbers with more than three digits **are not relevant** because the input data will always be such that **the output is less than a thousand**

Examples of naming some numbers:

- 68 = "sixty" + "eight" = "sixtyeight"
- 319 = "threehundred" + "nineteen" = "threehundrednineteen"
- 530 = "fivehundred" + "thirty" = "fivehundredthirty"
- 971 = "ninehundred" + "seventy" + "one" = "ninehundredseventyone"

INPUT

The first line of input contains the integer **N** ($1 \le N \le 20$), the number of words in the sentence.

Each of the following **N** lines contains a word not longer than 50 lowercase letters of the English alphabet or the character $ (none of the words will be the name of a number).

The character $ will appear exactly once.

OUTPUT

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

The numbers are named as mentioned before, even if the sentence sounds gramatically incorrect.

The input data will be such that a solution will always exist and is less than 1000.

SCORING

In test cases worth 40% of total points, the required number for the sentence will be less or equal to 50.

SAMPLE TESTS

input	input	input
5 this sentence has $ letters	7 $ is the number of letters here	5 the letters are $ potato
output	**output**	**output**
this sentence has thirtyone letters	thirty is the number of letters here	the letters are twentynine potato

Clarification of the second example: Sentence is split in two lines because of the lack of space in the table. The total number of letters in the sentence is 6 + 2 + 3 + 6 + 2 + 7 + 4 = 30

Clarification of the third example: As you can see, this sentence is gramatically incorrect. Nevertheless, Jurica is not concerned by that, for he is a mathematician, not a linguist.

Melita has just returned from the annual pig slaughter. Don't worry, this is a regular thing in Croatia. The best part was the abundance of food! There was everything, starting from good spicy sausages, ham, black pudding, up to teewurst, top quality bacon and čvarci, all with warm white bread and butter. After these appetizers, it was the perfect time to whip up a deep pot full of sarma (Melita ate twentyish of them) as well as a large platter of fine roast pork, so soft that it almost melts in your mouth. They watered all of this down with copious gulps of the best dry white wine that made them even hungrier.

Butcher Bajs kept his award-winning ham for the very end. **N** people attented the annual pig slaughter, labeled with numbers from 1 to **N**. These people already ate a lot of meat: the **k**th person ate **A[k]** kilograms of meat so far. Bajs will distribute his ham to the people **in the ratio B**[1] : **B**[2] : … : **B**[N], exactly in that order, but he doesn't know the total amount (number of kilos) of ham which he will be distributing yet.

At the end of the slaughter, the Man of the Year will be declared. A ranking list is made according to the total kilos of meat eaten. Bajs impacts directly on this list by choosing the amount of ham to distribute. Although Bajs has been offered bribes many times, he refused each time, saying that he was an honest man who would not hurt a fly.

Bajs cares about order, because he's a nice gentleman, and wants to have the order of people in the exact form of 1, 2, 3, ..., **N**, respectively from the one who ate the most kilos of meat to those who ate less, allowing ties between participants. Help Bajs select the total amount of ham that he will distribute (in the ratio mentioned before) to achieve his intention.

INPUT

The first line of input contains an integer **N** (2 ≤ **N** ≤ 1000), the number of competitors for the Man of the Year award.

Each of the following **k** lines contains integers **A**[**k**] i **B**[**k**], as mentioned in the text (0 ≤ **A**[**k**], **B**[**k**] ≤ 10^6). At least one of the numbers **B**[**k**] will not be equal to 0.

OUTPUT

The first and only line of output must contain -1 if the required order cannot be achieved. Otherwise, output the required amount of ham in kilos, a real number (rounded up to 12 decimal places) between 0 and 10^7 (inclusive). If there are multiple possible solutions, output any.

SAMPLE TESTS

input	input	input
3 7 1 3 2 10 0	3 2 1 4 0 0 3	5 15 4 6 7 12 5 9 6 1 7
output	**output**	**output**
10.5	-1	87

Clarification of the first example: 10.5 kilos of ham is distributed in the ratio 1 : 2 : 0, which gives us 3.5, 7 and 0 kilos of ham, respectively. If we add this to the already eaten amount of meat, we conclude that the participants ate 10.5, 10 and 10 kilos in total, which is a valid order.

The distance between two integers is defined as **the sum of the absolute result of subtracting** their digits. For example, the distance between the numbers 4561 and 3278 is |4 – 3| + |5 - 2| + |6 - 7| + |1 - 8| = 12. If one of the numbers consists of fewer digits than the other, we fill it with leading zeroes. Therefore, the distance between the numbers 32 and 5678 is |0 - 5| + |0 - 6| + |3 - 7| + |2 - 8| = 21.

You are given two integers **A** and **B**. Calculate the sum of distances between each pair of numbers belonging in the interval [**A**, **B**]!

INPUT

The first and only line of input contains integers **A**, **B** ($1 \le \mathbf{A} \le \mathbf{B} \le 10^{50000}$).

OUTPUT

The first and only line of output must contain the required number from the text. Given that the number could be extremely large, output answer **modulo** 1 000 000 007.

SCORING

In test cases worth 20% of total points, **A** and **B** will not exceed 10000.

In test cases worth 40% of total points, **A** and **B** will not exceed 10^{100}.

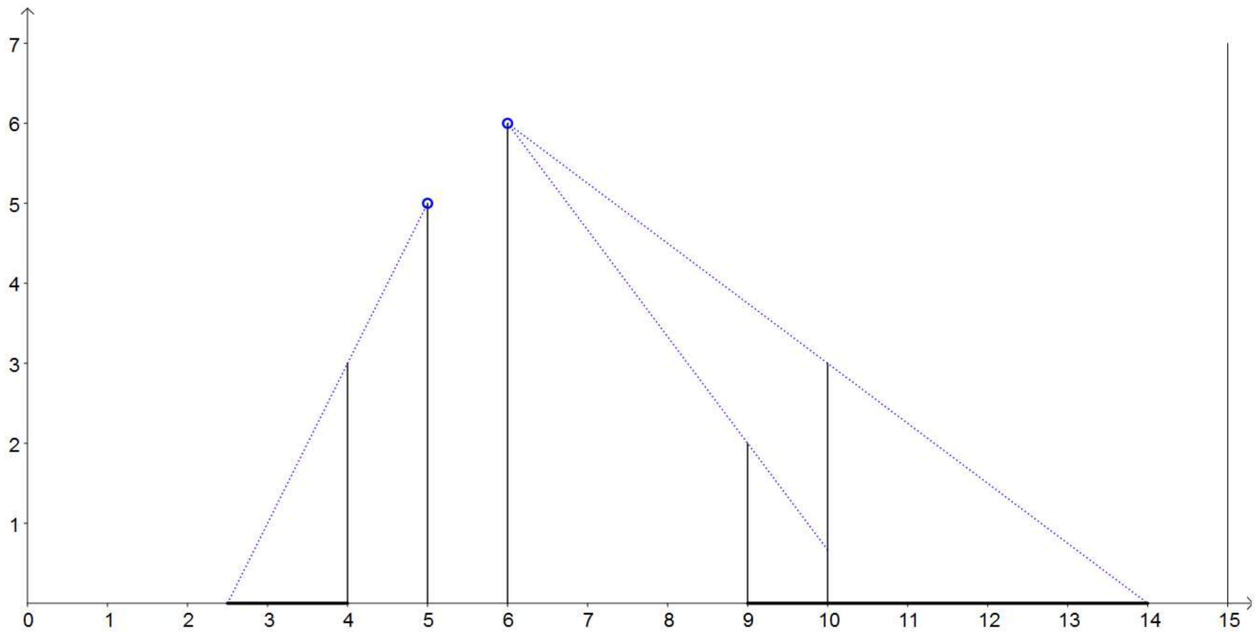
SAMPLE TESTS

input	input	input
1 5	288 291	1000000 10000000
output	**output**	**output**
40	76	581093400

Clarification of the second example: The distances are, respectively, (288, 289) = 1, (288, 290) = 9, (288, 291) = 8, (289, 290) = 10, (289, 291) = 9, (290, 291) = 1. Each of them counts twice, which is in total 2 * (1 + 9 + 8 + 10 + 9 +1) = 76.

The mayor has decided that it is high time to implement a new system of television transmitters. The city can be represented as **a segment of the length D** on which there are **buildings** of different heights. The width of a building is negligible. On top of some buildings, **television transmitters** are set, their dimensions are also negligible.

Television transmitters emit a television signal in all directions around them. The signal is spread through space **in straight lines** and **cannot pass through buildings**. A certain point in the city is considered **covered** if it is reached by a signal from an existing transmitter.



Find the segment of the city covered by television signal and output its length.

INPUT

The first line of input contains the integer **N** ($1 \leq N \leq 3 \cdot 10^5$), the number of buildings, and the integer **D** ($1 \leq D \leq 10^9$), the city length.

Each of the following **N** lines contains three numbers which describe the **i**th building:

1. a number which determines whether there is a transmitter on top of the building: 0 (no) or 1 (yes)
2. an integer X_i ($0 \leq X_i \leq D$), the distance between the building and the left end of the city
3. an integer H_i ($1 \leq H_i \leq 10^9$), the building height

The buildings are sorted in ascending order by the distance from the left end of the city. No two buildings will be located on the same distance from the left end of the city.

OUTPUT

The first and only line of output must contain the required length from the text.

Note: the maximum permissible deviation from the official solution is 10^{-3}.

SCORING

In test cases worth 30% of total points, **N** will be less or equal to 1000.

SAMPLE TESTS

input	input
3 10 1 2 6 0 4 3 0 8 2	5 15 0 4 3 1 5 5 1 6 6 0 9 2 0 10 3
output	**output**
6.000000	8.500000

Clarification of the second example: This example corresponds to the image from the text. The image depicts the city. The buildings are marked with vertical lines, and the transmitters with circles on the tops of the buildings. The bold lines on the x-axis represent the segment of the city not covered by television signal.