CodeWars 2018 Barcelona

Problems

#	Problem	Points
1	Savings for the Fallas	1
2	Printing books	1
3	To be able or not to be able. That is the question.	2
4	How old are you?	2
5	Storytelling	2
6	It’s the final countdown	3
7	Which graphics card should I buy?	3
8	Run, Forrest, run!	3
9	3D Coffee	4
10	Triathlon timing	4
11	Character counting	4
12	Tracking student’s progress	5
13	Quadratic equation solver	5
14	Power of 2	6
15	Not-allowed-entry-charset	6
16	Emirp numbers	7
17	Magic sum	10
18	Space battleship	10
19	Cross-stitch	12
20	Car race	12
21	Game of life	15
22	Nested triangles	15
23	Matrix Code	15
24	Ancient formulas	15
25	Secret door	15
26	Skiing	20
27	Digital castellers	20
28	Meowy’s Island	20
29	Hexagons	23
30	Maze	23
31	Keyboard	30

Introduction

Savings for the Fallas

1 point

Ana wants to know how much money she will spend to travel from Barcelona to Valencia to enjoy the "Fallas". She only has a few days to continue saving money to go. Ana will send you a list with the prices of the train tickets. The first line will be the one-way ticket and the second will be the return ticket. Please, can you tell her how many euros she should save?

Come on, help Ana!

Input

The input consists of two integers in two lines:
<One-way ticket cost>
<Return ticket cost>

Output

Print out the total of euros that should save following this output format: Ana should save a total of <total euros> euros.

Example

Input

```
50
35
```

Output

```
Ana should save a total of 85 euros.
```

Printing books

1 point

To earn some extra money, you decide to start working in a printing house. In your first day, you realize the boss looks worried. He should print lots of copies of a book (it's a best-seller!) and he doesn't know how many ink cartridges will remain after printing all the copies. The boss knows that he needs 3 ink cartridges for each book and he has written in a paper the number of books to print and the number of cartridges stored in the warehouse. Using your programming skills, try to help the boss ensure that there are always enough ink cartridges in the warehouse to print all the books.

HINT: Consider that there will be always enough ink cartridges in the warehouse.

Input

The input consists of two integers in two lines:
<Number of books to print>
<Number of ink cartridges available in the warehouse>

Output

Print the number of remaining cartridges after printing all the books.

Example

Input

```
10
80
```

Output

```
50
```

Introduction

To be able or not to be able. That is the question.

2 points

Ana wants to buy a new house but she doesn’t know if she has enough money. She will tell you how many euros she has saved, and how many euros the house costs. Can you help her to decide if she has enough money to buy the new house?

Come on, help again Ana!

Input

The input consists of two integers in two lines:
<Ana’s savings>
<Cost of the new house>

Output

Print out the one of the following outputs once you confirm Ana is able or not to buy a new house: Ana can buy the house !!
Ana can NOT buy the house :(

Example 1

Input

```
500000
300000
```

Output

```
Ana can buy the house !!
```

Example 2

Input

```
50000
300000
```

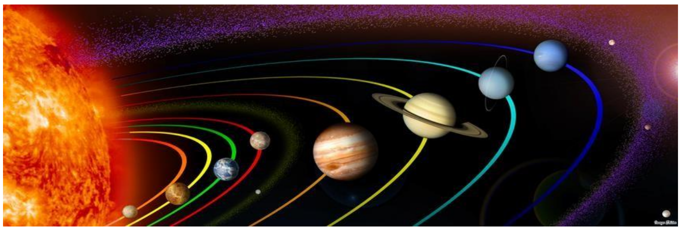
Output

```
Ana can NOT buy the house :(
```

How old are you?

2 points

The actual definition of a year is the time it takes to a planet to complete a single orbit around the Sun. That is, here on Earth, we consider a year to be 365 days.



But if you were to live in another planet of our Solar System – a year would work out to something else. Fortunately, we have a simple table to get the equivalence in Earth days of any planet's year.

Planet	Revolution period in Earth days
Mercury	88
Venus	225
Earth	365
Mars	687
Jupiter	4333
Saturn	10759
Uranus	30689
Neptune	60182

Since we plan to travel around the Solar System it is important to have conversion software to know the Earth age of the interplanetary travelers. Can you provide an easy way to convert the age expressed in years in a given planet to its value in Earth years?

Input

The input consists of two lines. The first line is an integer indicating your current age in years in the planet where you live. The second line is a string with the name of the planet where you live.
<Current age in years in the planet where you live>
<Name of the planet where you live>

Output

Print out your age in Earth years as an integer value.

Example

Input

```
10
Mars
```

Output

```
18
```

Storytelling

2 points

A storyteller has been told that nowadays the stories must be dynamic, so he wants to create a program that everybody could use. This program would have a static storyline and a dynamic part, where the reader would introduce his personal information: name, age, gender, city, favourite sport, favourite team and its ideal job.

HINT: Beware of the gender (boy/girl) and the consequent changes derived.

This is the static storyline:

Name **is a** ***age*** **year-old** ***gender*****.** ***Pronoun*** **is living with** ***possesive pronoun*** **parents in an apartment in the centre of** ***city*****, where** ***pronoun*** **hangs out with** ***possesive pronoun*** **friends. Moreover, in** ***possesive pronoun*** **free time** ***pronoun*** **plays** ***favourite sport*** **in a team called** ***favourite team*****.** ***name*** **would like to pursue a career in** ***ideal job*** **when** ***pronoun*** **is older, that's why** ***pronoun*** **is studying hard.**

Input

The input will be the dynamic data provided in seven lines.

Output

Print out the whole story including the static storyline part filled with the provided dynamic data.

Example

Input

```
Ainhoa 22
girl
Donostia
basketball
Mundarro
social working
```

Output

```
Ainhoa is a 22 year-old girl. She is living with her parents in an apartment in the
centre of Donostia, where she hangs out with her friends. Moreover, in her free time
she plays basketball in a team called Mundarro. Ainhoa would like to pursue a career
in social working when she is older, that's why she is studying hard.
```

It’s the final countdown

3 points

Different space agencies around the world plan to send manned missions to Mars during 21st century.

Many challenges are involved in these projects, like economic funding and astronaut health risks. Some scientists are currently developing and testing the different technologies to provide the first step towards Mars colonization.

And here is a critical part -- we need you to develop the software to generate the final countdown for the spaceship launch. Would you be able to do it?

Input

The input will be a single positive integer number indicating the first value of the countdown.

Output

Print out the countdown, that is the sequence backward counting starting with the input value.

Example

Input

```
3
```

Output

```
3 2 1 0
```

Which graphics card should I buy?

3 points

Your old GPU (Graphics Processing Unit) can no longer support newer video games and you want to study several options to check which is the one that will give you those extra fps (frames per second) needed. To evaluate the GPU performance the program will test its frequency versus the minimum required frequency of several video games.

Input

The input will be a sequence of lines with integer numbers representing frequencies (all in MHz). The first number will be frequency of the GPU to test. The rest of the numbers will be the minimum frequencies required for a specific game title to perform properly. If the game frequency is 0, the process should stop, and not consider that value.

Output

The output will be the number of video games that will run perfect for the selected GPU.

Example

Input

```
1809
1700
1900
1200
0
```

Output

```
2
```

Introduction

Run, Forrest, run!

3 points

Forrest is passionate about running. To have an accurate tracking of all his activity while running he bought a U.S. sport watch. It reports the distance run every day only in miles. Forrest knows that it is recommended that running shoes should be replaced every 622 kilometers and needs a program to sum up all the year running activity to decide whether to replace or not his running shoes. Keep in mind that 1 mile is approximately 1.6 kilometers.

Input

The input will be a sequence of 365 integer values in a single line representing the miles run daily during the last year.

Output

The output of the program reports a simple string stating "Yes" or "No" to know if the running trainers must be replaced.

Example

Input

```
16 4 1 15 12 20 14 2 7 10 4 14 5 15 16 21 13 3 16 11 18 17 10 20 2 18 7 12 11 5 10 8
12 1 6 1 6 12 2 10 19 8 14 13 5 6 8 12 17 1 10 4 18 6 3 7 3 1 14 11 3 14 11 13 6 13 10
14 4 11 3 10 17 18 13 11 17 7 11 3 12 4 9 2 5 15 20 20 16 19 20 18 14 8 9 15 18 21 8 3
13 15 20 17 2 12 8 15 8 4 8 10 11 20 15 1 10 5 16 11 19 11 20 6 18 6 13 21 6 8 6 11 14
14 2 14 7 11 9 6 1 7 1 4 16 20 12 15 4 5 2 20 5 17 15 13 18 18 10 17 7 14 21 19 13 17
2 1 10 11 1 5 19 6 2 12 6 14 6 16 16 15 15 11 10 21 10 11 1 21 12 3 18 20 2 9 20 20 18
5 12 13 17 9 12 1 1 18 7 15 5 21 13 20 16 2 9 10 9 1 3 8 15 16 6 14 15 1 2 9 18 18 2 4
16 14 16 2 1 21 4 3 15 16 16 3 20 6 21 5 1 20 14 4 14 14 7 2 14 9 17 14 20 21 21 19 16
20 11 18 11 5 11 21 6 16 7 18 11 9 20 2 9 12 7 5 14 14 12 15 1 3 8 13 11 20 7 5 9 4 3
15 1 1 19 11 15 12 15 21 10 11 19 19 18 15 3 6 4 20 19 6 21 17 7 2 18 4 19 8 14 16 6
13 3 15 2 12 3 8 4 17 6 7 8 11 14 16 21 20 4 15 5 14 13 3 1 21 2 13 8 4 6 8 10
```

Output

```
Yes
```

3D Coffee

4 points

You have just bought a 3D printer and you want to run a successful 3D coffee bar business. To calculate the price of every printed part you sell, you just apply the following formula:

price = volume printed [cm^3] x p [€/cm^3]

Where p is the price per cm^3. If the customer doesn't have a drink in the coffee bar the value of p is 2 €/cm^3. But you promote a special offer that if customer buys one or more drinks then the value of p is reduced to 1.8 €/cm^3.

So to compute the final price for each printer part your cash register will receive two values: one specifying the volume of the printed part and a second one detailing if a customer has had at least one drink.

Input

The input consists of several lines of two values following this format:
<3D volume printed with two decimals> <Just Y or N to report if the customer has had a drink> The input ends with a line with -1 integer value.

Output

Print out the invoice price for each 3D volume printed with two decimals rounding.

Example

Input 3
Output 90 3
100.. 13

```
50.12 Y
50.12 N
100 Y

180.00
```

10 Triathlon timing

4 points

Introduction

A triathlon is a multidisciplinary race that combines three different sports: swimming, cycling, and running. Although the three disciplines are practiced one after the other, the classificatory system tracks the time of each of them individually. At the end of the race, the three registered times are added up to determine the final time.

Given the three registered times for swimming, cycling and running, compute the final race time in the appropriate format (XXhYYmZZs). The triathlon will not last longer than 72 hours.

Input

The three registered times for swimming, cycling and running of triathlon are provided following the time format XXhYYmZZs.

Output

The final race time of triathlon in format XXhYYmZZs.

Example

Input

```
00h28m43s
01h02m31s
00h37m17s
```

Output

```
02h08m31s
```

11 Character counting

4 points

Introduction

Your Computer Science teacher needs a program to analyze text. He is interested in counting how many uppercase and lowercase letters the text contains. The rest of the characters in the text (numbers, punctuation, math symbols, etc.) must be also counted except spacing characters that must be ignored. Would you be so kind to write this program?

Input

The input will be a line with the text to be analyzed.

Output

Print out the number of uppercase, lowercase and other characters found in the text.

Example

Input

```
Hi there! How are you today? The answer is 4*8 + 34 - 2 = (2^6)
```

Output

```
Uppercase 3
Lowercase 29
Other 16
```

12 Tracking student's progress

5 points

Introduction

Every quarter your teacher collects the student's notes of the every class where he teachs. He reviews the results and compare the different classes he is teaching. This way he can improve his lessons. To avoid doing this task manually with a pocket calculator you can help him by creating a program. This program should find out the number of students that is below the average grade. Beware that the student's notes can be written with decimal number, as in real life.

Input

The input consists of several lines containing the student's notes of the whole class expressed in decimal format.

Output

Print out the number of students below the average.

Example

Input

```
8
0
4.3 6.5
4.7
```

Output

```
2
```

13 Quadratic equation solver

5 points

Introduction

Jeremy is nervous. He has his first Quadratic Equation exam, and he is about to begin. But Jeremy is the best young programmer of its classroom, and he wonders if he could programmatically solve any Quadratic Equation in the world with a simple code.

Jeremy knows very well that a generic Quadratic Equation has the form

$$ax^2 + bx + c = 0$$

Where ***a***, ***b***, and ***c*** are the coefficients of the equation. Could you help Jeremy and write a program that receives as input the coefficients of the equations and tells you the solution?

Remember that the roots of a Quadratic Equation can be computed as

$$x_{+,-} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Input

The input of the program is the value of the coefficients of the equation, separated by spacing characters.

Output

The output of the program is the solution (the roots of the quadratic equation).

Remember that a quadratic equation can have complex roots when b^2-$4ac$ is lesser than 0. Your program must answer “It has complex Roots!” if that is the case (Example 1).

When the roots are not complex, the solution must be shown featuring first x+ and second x- with two decimals rounding (Example 2).

Example 1

Input

```
1 1 5
```

Output

```
It has complex Roots!
```

Example 2

Input `1`

```
2.5 -4.5
```

Output `x_+ = 1.21; x_-`

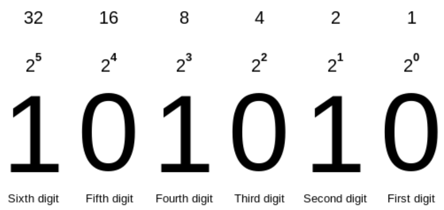
```
= -3.71
```

14 Power of 2

6 points

Introduction

In Computer Science we deal daily with binary numbers. As you may know a binary number is represented in a row of bits that could have a value of 0 or 1. It is assigned to each bit a position number, ranging from zero to N-1, where N is the number of bits in the binary representation used. Usually, this is simply the exponent for the corresponding bit weight in base-2 (such as in $2^{31}..2^{0}$).



Consider the decimal number 42, as an example, its representation in binary is 101010. Meaning that the position of 2^5 is 1, 2^4 is 0,2^3 is 1, 2^2 is 0, 2^1 is 1 and 2^0 is 0. So, the exponent of the highest power of 2 is 5 (2^5).

Given a decimal number find out the exponent of the highest power of 2.

Input

The input will be a single positive integer number higher than 0.

Output

Print out the position of first one from left in binary.

Example 1

Input

5

Output

2

15 Not-allowed-entry-charset

6 points

Introduction

An online Web form includes a section for entering freeform text. However, there are some characters or numbers that cannot be inserted into our back-end database. To solve this problem, we need you to create a program that, given a string and a not-allowed character set, will remove all occurrences of the not-allowed literal characters from the entry of the user string.

Input

The input will consist of two lines: <the given string>
<the not-allowed-entry-char-set>

Output

Print out the original given string after removing the not-allowed-entry-charset.

Example 1

Input

```
12345678
3456
```

Output 1278

Example 2

Input

```
87654321
346
```

Output

```
87521
```

16 Emirp numbers

7 points

Introduction

A prime is a number that is only divisible by one and itself, which is essentially saying that it has no divisor. Nowadays primes are essential for secure communications. Most modern computer cryptography works by using the prime factors of large numbers.

When you reverse the digits of most primes you get a composite number (for example, 43 becomes 34). That is not the case for palindromic primes that read the same forward and backward (for example, 727), so reversing a palindromic prime gives you the same prime.

Then there is an special category, the emirp numbers. An emirp (the word "prime" written backwards) is a prime whose reversal is also prime, but which is not a palindromic prime (for example, 13 becomes 31). The first emirp numbers are 13, 17, 31, 37, 71, 73, 79, 97, 107, 113, 149, 157, ...

Let's write a program to find out whether an integer number is emirp.

Input

The input will be an integer number.

Output

Print out whether the given numer is emirp or not.

Example 1

Input `13`

Output

```
13 is an emirp number
```

Example 2

Input

```
11
```

Output

```
11 is not an emirp number
```

17 Magic sum

10 points

Introduction

One day, while you are sitting in front of your computer, something incredible happens. The computer starts making some noise and in a blast of light everything goes blank. When you opened your eyes, you found yourself in a weird and magical world. The computer must have sent you to another planet in a different parallel universe.

Walking around, in a hurry to find a way to come back home, you found out a message in a poster saying: "Solve the sum of all digits of a given number to go home!". You stopped and told yourself - wait, to come back home, the only thing is to solve a simple sum? is all that takes! –

Unfortunately, in this world, the basic mathematical operations are quite different than the ones in your home world. Thus, you need to understand them in order to give the correct answer and come back home.

The magic sum operation requires you to review a sequence of digits and find the sum of all digits with the following properties:

- In order to consider a digit, it has to match the next digit in the sequence. The sequence is circular, so the digit after the last digit is the first digit in the sequence.
- If a digit is even, you must multiply it by its position in the list before sum it. In this case, the first digit is considered to have the position 1.
- If a digit is odd, just sum it.

For instance:

- 1122 : outputs 7 = 1+6 ⟶ because the first digit (1) matches the second digit and its odd, we sum its value. The third digit (2) matches the fourth one and because its even, we multiply it by its position (that is 3).
- 123 : outputs 0 ⟶ because none of the elements matches the next in the sequence. - 565 : outputs 5 ⟶ because the the third digit (5) matches the first one and its odd.

Will you be able to solve the sum and come back home?

Input

The input consists of several lines with given numbers
Each number consists in a sequence of digits. All digits are between 1 and 9.

Output

The output must consist in several lines with the integers representing the magic sum of the input numbers.

Example

Input

```
1122
123
565
1
```

Output

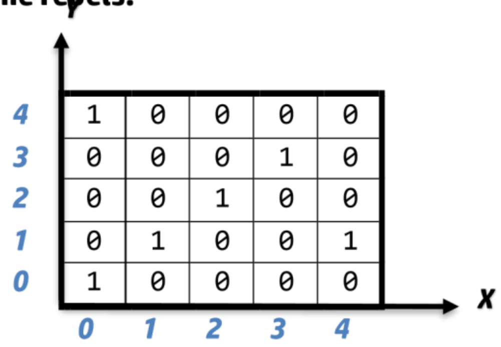
```
7
0
5
1
```

18 Space battleship

10 points

Introduction

A long time ago in a galaxy far, far away.... the Rebel and Imperial fleets are fighting against each other in the final battle near Coruscant. Princess Leia has ordered you to provide the total amount of hits against the Imperial forces. To ease your job you are given an square matrix representing the battle scenario with the X/Y coordinates where the imperial ships are present. You are also provided with the
X/Y coordinates of the shots fired by the rebels.



Y						
4	1	0	0	0	0	
3	0	0	0	1	0	
2	0	0	1	0	0	
1	0	1	0	0	1	
0	1	0	0	0	0	
	0	1	2	3	4	X

Sometimes the rebels fire more than once to the same objective. Beware of not double counting these shots. Can you develop a program to provide this data as soon as possible? Remember you are her last hope.

Input

The first value of the input will be an integer number representing the size of the square matrix. It will be followed by the matrix where 1 means an imperial spaceship is present and 0 is just empty space. Then the number of shots is reported followed by the coordinates of every shot.

Output

Print out the number of successful hits on the Imperial fleet.

Example

Input

```
5
1 0 0 0 0
0 0 0 1 0
0 0 1 0 0 0 1 0 0 1
1 0 0 0 0
5
1 1
4 4
```

```
1 1
4 1
0 4
```

Output

```
3
```

19 Cross-stitch

12 points

Introduction

As a hobby you decided to do a cross-stitch picture. You have an schema with different colors and you want to know how many skeins of yarn you need for every color. Every stitch consumes 10mm of yarn and every skein has 1m of yarn.

Every character represents an unique color except white spaces and new lines.

Input

The input will be a line with an integer that represents the total number of colors, followed by a series of lines representing the picture where each different character represents a color.

Output

A report detailing the number of skeins needed by color. The output order is not relevant.

Example

Input

```
3
------------------
--------AA--------
------AAAAAA------
----AAAAAAAAAA----
--------AA--------
------AAAAAA------
----AAAAAAAAAA----
--------AA--------
------AAAAAA------
----AAAAAAAAAA----
--------OO--------
--------OO--------
--------OO--------
```

Output

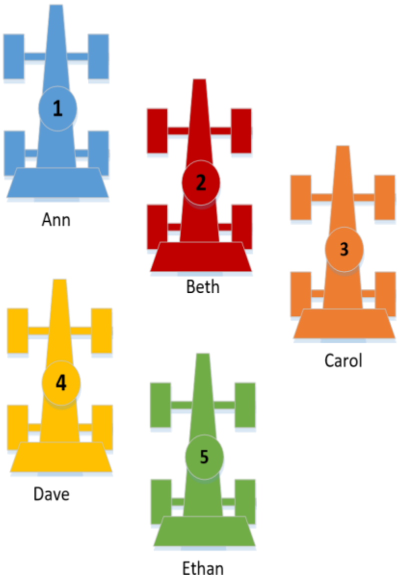
```
- 2 skeins of yarn -
- 1 skeins of yarn A
- 1 skeins of yarn O
```

20 Car race

12 points

Introduction

A famous website portal has asked you to help them publish the race results of car races.



You have managed to subscribe to a news feed service that reports the order of the racers in the grid at the beginning of the race and any overtakes that occur during the race. This will suffice to let you keep track of each racer's position until the end of the race. Can you automate this task writing a piece of software?

Input

The input format is as follows:

- **The first line consists of the list of racers following the order in the grid. The name of the racers consists of a first name without any white spaces. In the example; Ann is first followed by Beth, which is followed by Carol, etc.**
- **The following lines contain the overtakes that occur over the course of the race. All the overtake announcements will consist one car taking the position of the car that was immediately ahead of it. Do not consider the case of the racers that are lapped twice by the leading racers.**

Output

The output consists of the list of racers, one per line, in the order in which they have finished the race.

Example

Input

```
Ann Beth Carol Dave Ethan
Ethan overtakes Dave
Carol overtakes Beth
Carol overtakes Ann
Beth overtakes Ann
Dave overtakes Ethan
```

Output

```
Carol
Beth
Ann
Dave
Ethan
```

21 Game of life

15 points

Introduction

The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970.

The game is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input. One interacts with the Game of Life by creating an initial configuration and observing how it evolves.

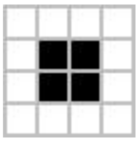
We want to simulate such experiment, by creating a simulator of this Game of Life.

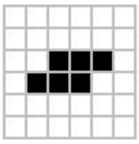
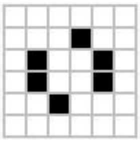
The rules of the game are simple.

We have a board of N x N size, that will be filled with life or dead cells (# or .). After each turn, every state of the cell will be determined by the previous state according to the following rules:

1. Any live cell with fewer than two live neighbors dies, as if caused by under-population.
2. Any live cell with two or three live neighbors' lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by over-population.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

The neighbors of a cell A are determined by all the surroundings cells of the cell A (vertically, horizontally and diagonally). Some patterns will survive forever, other may oscillate among different status and other may even move or create other patterns.

This will be a stable pattern, . As you can see all the dead cells will remain dead as they only have 1 or 2 neighbors, and the already alive cells will always live because they have 3 neighbors.

An oscillator is a pattern that returns to its original state, in the same orientation and position, after a finite number of turns. The toad is a repeating pattern that oscillates between  and . Those cells that appear, do so because they have exactly 3 neighbors, and those cells that die do so because they have 4 or more neighbors or they have 1 or less neighbors.

Following these rules, we can create complex and bigger patterns.

Your goal will be create an application that can load a board of size N x N from the input, and will produce as output the final state of the board after m turns.

You can see some examples of input and output in the following sections.

HINT: Don’t forget that the total size of the board can be up to 100 x 100, so be careful with sizes, and don’t forget to be careful when updating your board (wink, wink). By the way, doing it by hand could take LOTS of time (specially if we will test it with high numbers of iterations…).

Input

< m that represents the number of turns >
< N that represents the board size >
< board data matrix displayed as N rows x N columns >

Output

Print out the resulting board data.

Example 1

Input

```
2
5
. . . . .
. . # # .
. . # # .
. . . . .
. . . . .
```

Output

```
. . . . .
. . # # .
. . # # .
. . . . .
. . . . .
```

Example 2

Input

```
2
5
# . # # .
# . . # .
. . # # .
. . . . .
. . . . .
```

Output

```
. . # # .
. # . . # .
 . . # .
. . . . .
. . . . .
```

22 Nested triangles

15 points

Introduction

Given an input nesting level number, within the range from 0 to 9, draw a set of triangles which sit inside each other. Please note in the examples that a nesting level of 0 means a 0 triangle.

Input

The input will be a single integer number indicating the level of nesting between 0 and 9.

Output

Print out the nested triangles.

Example 1

Input 1

Output

```
 1
111
```

Example 2

Input

```
3
```

Output

```
     3
    323
   32123
  3211123
 322222223
33333333333
```

C++

23 Matrix Code

15 points

Introduction

You work as a technical consultant for the new upcoming sci-fi film “Matrix Code”.

In the movie, a white hat hacker named “Robt” is studying an unknown new malware (malicious software) that is obfuscating all the internet communications. During the investigation Robt gets absorbed into the machine computing world by the malware and he will have to fight it from the inside.

This movie will use a lot of special effects related to how the computers interact with the real world and the director and art team want to include subliminal messages in the representation of the obfuscated transmissions.

You are asked to develop a prototype to allow the movie team to obfuscate text communications. Since the movie will be set in the 1990’s, the output will have to be represented in 4:3 screen format, to simulate the old CRT monitors.

At the moment, only one obfuscating algorithm is requested. This one, should change all the letters from the text by the next one in the alphabet. If the obfuscated text message does not fill the whole screen, the character “ should be shown, except in those pixels that are located in the screen’s diagonal, where # should be shown instead.

Important notes

- Format 4:3 means, only character displays of 4x3, 8x6, 12x9, 16x12, etc, are allowed
- Only letters should change, any other symbol should remain the same. • The diagonal is formed by the positions (x, y) where x=y
- Each element should be separated by a space.

Input

A text of any length in a single line (only ASCII characters)

Output

The output will be the minimum matrix of format 4:3 showing the obfuscated text with the requested restrictions

Example 1

Input

```
Hi Robt, the matrix has you and you will not escape!
```

Output

```
I j   S p c u ,   u i f
n b u s j y   i b t z p
v   b o e   z p v x j m
m   o p u   f t d b q f
! # " " " " " " " " " "
" " # " " " " " "
" " " " " " # " " " " "
" " " " " " " # " " " "
" " " " " " " " # " " "
```

Example 2

Input

```
Hi Robt, to be able to exit from the matrix you must analyze the data prompted in the
screen. Then you will fully understand the code. Zzz.
```

Output

```
I j   S p c u ,   u p   c f   b c
m f   u p   f y j u   g s p n   u
i f   n b u s j y   z p v   n v t
u   b o b m z A f   u i f   e b u
b   q s p n q u f e   j o   u i f
t d s f f o .   U i f o   z p v
x j m m   g v m m z   v o e f s t
u b o e   u i f   d p e f .   a A
A . " " " " " " " " " " " " " " #
" " " " " "
" " " " " " " " " " # " " " " "
" " " " " " " " " " " # " " " "
```

```
//...
   // Takes the input message and shifts letters
   // by their next ones in the alphabet string translate
   (string inputMessage){   // After 'z', it comes 'a' again
   string alphabetLowerCase = "abcdefghijklmnopqrstuvwxyza";
   string alphabetUpperCase = "ABCDEFGHIJKLMNOPQRSTUVWXYZA";
   string alphabet = alphabetLowerCase + alphabetUpperCase;
   string outputMessage = "";   // Iterate over the input
   message   for (int i = 0; i < inputMessage.size(); ++i){
   char newChar;
       //search the alphabet for the char at position i
   size_t found = alphabet.find(inputMessage[i]);     if
   (found != string::npos){
         // Find the letter and take the next one in the list.
   newChar = alphabet[found+1];
       }
       // Copy the character. No conversion here
   else newChar = inputMessage[i];     // Append
   it into the output message     outputMessage
   += newChar;
     }
     return outputMessage;
   }

   int main(){
     string inputMessage;
   getline(cin, inputMessage);
   paint(translate(inputMessage)); }
```

24 Ancient Formulas

15 points

Introduction

Archaeologists found ancient mathematical formulas that use parenthesis "()", brackets "[]" and braces "{}" to group sub-expressions in order to provide better clarity than if they only used parenthesis "()".

Here is an example: [1 + 1] + (2 + { 1 + [4 * (2 + 1) + 3]})

In any case, the meaning of these grouping symbols is equivalent and they simply indicate the opening and closing of a sub-expression group.

Interestingly, we discovered that some of the closing symbols do not always match with the opening symbol type. For instance, in the formula "2+({3-2]-1)" the opening symbol '{' should be closed using a matching '}' instead of a ']' for consistency.

You should write a program that reads one of these ancient formulas and adjust any closing symbols to match the corresponding opening symbol type.

Input

The input is one line with a formula that may contain several grouping symbols. You can assume that the number of opening and closing symbols match and the only problem is with the consistency between the opining type and the corresponding closing type.

Output

The output corresponds of a first line with the input formula after adjusting any closing symbols that do not match the opening symbol types. And a second line that informs of the number of closing symbols that had to be modified in the original formula to fix it using the format "# fixes made to the formula."; where # is the number of changes.

Example 1

Input

```
2+({3-2]-1)
```

Output

```
2+({3-2}-1)
1 fixes made to the formula.
```

Example 2

Input

```
[ sin(1) + rho) + phi (a + { 1 + [ 4 * (2 pi + 1) + b]]]
```

Output

```
[ sin(1) + rho] + phi (a + { 1 + [ 4 * (2 pi + 1) + b]}) 3
fixes made to the formula.
```

25 Secret door

15 points

Introduction

As every year, our grandpa Santa Claus is in his journey to deliver presents to all good girls and boys around the world in his sleigh led by magical reindeer. But this time, and out of his imagination, the magical sleigh crashed and Santa landed in a maze of rooms created by the Grinch. By the time Santa recovered from this problem, there are only a few minutes until midnight and Santa is seeking help from a good girl or boy to help him out.

The maze is as follows: every room is connected, by a secret hall, to another room and there is only one room with an exit to the exterior, marked with a 0 (zero).

When you are in one room, you take the secret hall to another room. From there you walk to other room, and so on until you find the exit. The path is said to be unidirectional, i.e. one room is only connected to another room. For instance, room 1 is connected to room 3, but room 3 is not connected to room 1. In the other hand, there are rooms that are connected to themselves, so they don't have an exit.

For instance:
0 ⟶ 0 : has an exit by definition
1 ⟶ 3 : has an exit via 3
2 ⟶ 1 : has an exit via 1, then 3
3 ⟶ 0 : has an exit directly connected to 0
4 ⟶ 2 : has an exit via 2, then 1, then 3
5 ⟶ 5 : do not has an exit

Santa would like to know how many rooms, that are connected to the exit, are so he can continue his journey. In the previous example, we would give Santa the answer of 5 (5 rooms with an exit). All of them but room 5.

Would you help Santa to find the way out so he can deliver all the presents?

Input

<n> positive integer number greater than 0 indicating how many rooms there are.
<room ID> <room ID connection> for every room we give the room id and its connection.The room ID will be between 0 and n-1. The given sequence can be in any order.

Output

Print out the positive integer indicating how many rooms have an exit.

Example

Input

```
6
0 0
1 3
2 1
3 0
4 2
5 5
```

Output

```
5
```

26 Skiing

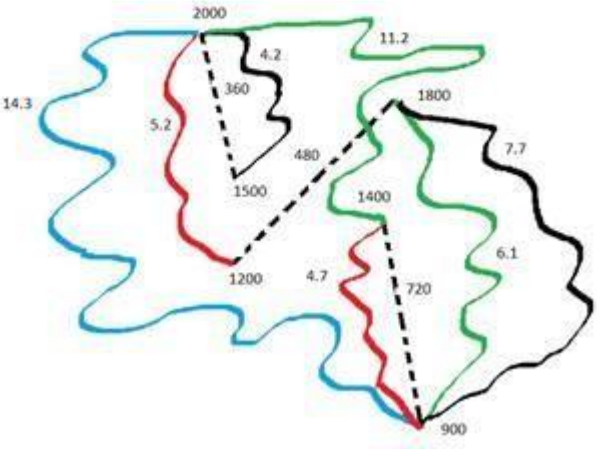
20 points

Introduction

A ski station consists of lifters and slopes.

Lifters bring you uphill from a lower starting point to a higher ending point, while slopes are the other way around. Lifters should have at least one slope connected to its starting and ending point.

Slopes have a difficulty level, which limits the maximum speed a skier can reach on them: 20 km/h on green slopes, 30 km/h on blue slopes, 40 km/h on red slopes, and 50 km/h on black slopes. Slopes can start or end in another slope or in a lifter. Multiple slopes and lifters can be connected.



Given a ski domain map, which contains lifters and slopes, determine the minimum time (in seconds) needed to go from the top of the ski station to the bottom.

Note that, for the sake of simplicity, we cannot have two or more starting and ending points with the same height. Moreover, in case of trouble, we can always call the relief helicopter, which will bring us to the bottom of the ski resort in two hours.

Input

The map should be read as follows: The first number in the input is the number of lifters and slopes the ski station has. Then, for each element, the input provides a letter that specifies if it is a lifter (L) or a slope (S). For lifters, following we have the time a lifter requires to reach its end in seconds, followed by its starting point and ending point height, both given in meters. For slopes, we have the difficulty level, the length in kilometers, and the starting point and the ending point heights, both given in meters as well.

Output

Print out the minimum time in seconds with a resolution of two decimals needed to go from the top of the ski station to the bottom.

Example

Input

```
10
L 360 1500 2000
L 480 1200 1800
L 720 900 1400
S black 4.2 2000 1500
S red 5.2 2000 1200
S blue 14.3 2000 900
S green 11.2 2000 1400
S red 4.7 1400 900
S black 7.7 1800 900
S green 6.1 1800 900
```

Output

```
1502.40
```

Note: in this example the fastest way down is 2000 – red slope – 1200 – lifter – 1800 – black slope – 900

27 Digital Castellers

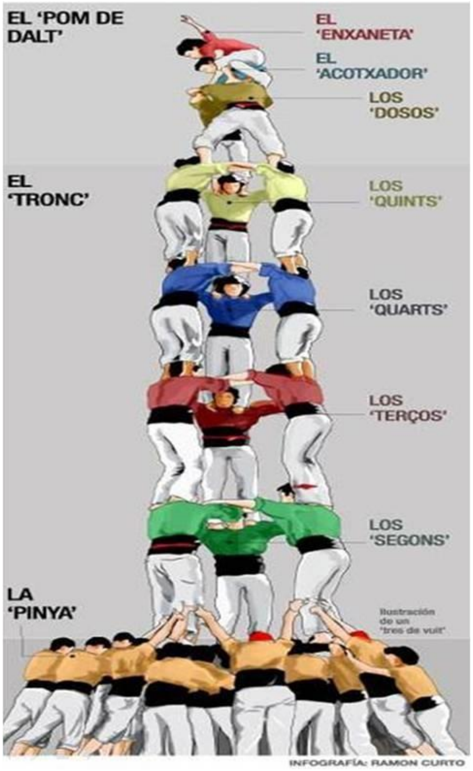
20 points

Introduction

A *Castell* is a human tower built traditionally in festivals at many locations within Catalonia. People that forms part of a *Castell* are known as *castellers*.

So, in our a computer programming contest you are requested to build digital castellers.

There are three definite parts of a digital castle; the *pinya* , the *tronc* , and the *pom* de *dalt* or the crown of the castle.



The *pinya* is the base of the *Castell*, and it's composed by:

- The *soca*, in the first floor
- The *folre*, in the second floor. It's optional.
- The *manilles*, in the third floor. It's optional, and only allowed in top of a *folre*.

The *tronc* is the main visible structure of the *Castell*, and it's between the *pinya* and the *pom de dalt*. The number of *castellers* in each floor of the *tronc* is defined in the name of the *Castell*.

The *pom de dalt* is the top of the *Castell* and it's composed by:

- The *dosos*, two *castellers* just on top of the *tronc*.
- The *acotxador*, one *casteller* on top of the *dossos*.
- The *enxaneta*, who crowns the *Castell*, on top of the *acotxador*.

In addition, a *Castell* may have an *agulla*. This is a central tower in the centre of the *Castell*,with the same height as the *tronc*.

The most common nomenclature to describe a *Castell* is, in Catalan

N de M [amb X] [i Y] [i Z]

Where N, M, X, Y and Z are the variables that define the structure of the *Castell*.

- N is the number of *castellers* in each floor of the *tronc*, with 1 <=N <= 10
- M is the height of the *Castell* in *castellers*, with 3 <= M <= 10
- X, Y and Z are optional, and they may be, in that order ○ *folre* ○ *manilles* ○ *l'agulla*

N and M are expressed in catalan with the following nomenclature

- **Only for N ○ 1 = *pilar***
    - **2 = *dos* or *torre***

- **For N and M ○ 3 = tres ○ 4 = quatre ○ 5 = cinc ○ 6 = sis ○ 7 = set ○ 8 = vuit ○ 9 = nou**
    - **10 = deu**

The Castells constructed by a Pilar (N = 1) does not have the *dosos* nor the *acotxador*.

The problem consists on, given a Castell description, represent it graphically

- **Each *casteller* is represetend with a # character**
- **The *Castell* has to be as symmetrical as possible**
- **If the total width of the *tronc* is odd, there will be a space between the *dosos*.**
- **If the total width of the *tronc* is even, the *acotxador* and *l'enxaneta* will be in the left.**

Input

A string with the definition of the castell to draw.

Output

The digital castell

Example 1

Input `quatre de set amb`
`folre`

Output

```
   #
   #
   ##
  ####
  ####
 ######
########
```

Example 2

Input `tres de sis`

Output

```
  #
  #
 # #
 ###
 ###
#####
```

Example 3

Input `cinc de nou amb folre i manilles`

Output

```
     #
     #
    # #
   #####
   #####
   #####
  #######
 #########
###########
```

28 Meowy's Island

20 points

Introduction

Meowy, the adventurer cat, has found a new island to live during his journeys. However, this island has a very irregular terrain and it is quite singular: the sea level grows rapidly during the day. The problem is that Meowy hates water, although he thinks that the island is really beautiful. So he needs to know the level of the sea at every hour in order to avoid flooded areas.

Meowy has studied the island carefully, thus he knows the following:

- There is a map of the island represented as a grid, where the sea cells are marked with '.' and the island cells contain a height value in the interval [1,9].

- When the sea level increases, the water from a cell floods the neighboring cells (consider the 8 directions: north, north-east, east, south-east, south, south-west, west, north-west) only if the height of the neighboring cell is less or equal than the sea level.

- There might be holes in the island (cells with lesser height than the surrounding cells, see top right part of the island in the Example 2). Holes are flooded only when at least one of the surrounding higher cells are covered by the sea.

- There might be sea cells inside the island (see bottom left part of the island in Example 2).

Input

The input of the program is:
< The rows of the map. >
< The columns of the map. >
< The values of the height map. >
< The initial sea level. >
< The final sea level. >

Output

The state of the map at every height level of the sea. Cells with water are represented by a 'W' and cells without water are represented with a space ' '.

Example 1

Input

```
10
29
```

```
.............................
.12...........55555.........
..222223344....41115.........
...79972347774479974241155...
...7..85575335473373443......
...7788547473557972..........
..493285.743754721222........
.558.275..77...9..212........
.553443...........222........
.............................
0
5
```

Output

```
Sea level: 0
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
W  WWWWWWWWWWWW     WWWWWWWWW
WW         WWWW     WWWWWWWWW
WWW                       WWW
WWW WW                 WWWWWW
WWW                WWWWWWWWWW
WW      W            WWWWWWWW
W   W   WW  WWW WW   WWWWWWWW
W      WWWWWWWWWWW   WWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
Sea level: 1
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WW WWWWWWWWWWWW     WWWWWWWWW
WW         WWWW     WWWWWWWWW
WWW                   WW  WWW
WWW WW                 WWWWWW
WWW                WWWWWWWWWW
WW      W        W   WWWWWWWW
W   W   WW  WWW WW   WWWWWWWW
W      WWWWWWWWWWW   WWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW Sea
level: 2
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWW     WWWWWWWWW
WWWWWWW    WWWW     WWWWWWWWW
WWW    W            W WW  WWW
WWW WW                 WWWWWW
WWW               WWWWWWWWWWW
WW   W  W       WWWWWWWWWWWWW
W   WW  WW  WWW WWWWWWWWWWWWW
```

```
W      WWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW Sea level: 3
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWW     WWWWWWWWW
WWWWWWWWW  WWWW     WWWWWWWWW
WWW    WW           W WW  WWW
WWW WW     WW   WW W  WWWWWWW
WWW         W     WWWWWWWWWWW
WW  WW  W  W    WWWWWWWWWWWWW
W   WW  WW  WWW WWWWWWWWWWWWW
W  W  WWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW Sea
level: 4
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWW     WWWWWWWWW
WWWWWWWWWWWWWWWWWWW WWWWWWWWW
WWW    WWW   WW    WWWWW  WWW
WWW WW     WW W WW WWWWWWWWWW
WWW     W W W     WWWWWWWWWWW
WWW WW  W WW  W WWWWWWWWWWWWW
W   WW  WW  WWW WWWWWWWWWWWWW
W  WWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
Sea level: 5
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWW    WWW   WW    WWWWWWWWWW
WWW WW WW WWWWW WW WWWWWWWWWW
WWW    WW W WWW   WWWWWWWWWWW
WWW WW WW WW WW WWWWWWWWWWWWW
WWW WW WWW  WWW WWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWW
```

Example 2

Input

```
15
31
...............................
.333.............4444444.......
.323...........6644444411......
.333.........4666644443233.....
..33333333..666666644..33......
.3322233344444444444...21......
.332.2.3355444445544533333333..
.3322233544544544445...233223..
..322334555544544445...233..3..
..33334454454445544555..33333..
...321...332234...2222333......
...24446666666666666666444.....
..22147777777777777777774......
..211..444444444444444444......
...............................
0
4
```

Output

```
Sea level: 0
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
W   WWWWWWWWWWWWW       WWWWWWW
W   WWWWWWWWWWW          WWWWWW
W   WWWWWWWWW             WWWWW
WW        WW         WW  WWWWWW
W                   WWW  WWWWWW
W   W W                      WW
W                   WWW      WW
WW                  WWW   WW WW
WW                    WW     WW
WWW   WWW      WWW       WWWWWW
WWW                       WWWWW
WW                       WWWWWW
WW   WW                  WWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
Sea level: 1
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
W   WWWWWWWWWWWWW       WWWWWWW
W   WWWWWWWWWWW        WWWWWWWW
```

```
W   WWWWWWWWW             WWWWW
WW        WW         WW  WWWWWW
W                   WWW WWWWWWW
W   W W                      WW
W                   WWW      WW
WW                  WWW   WW WW
WW                    WW     WW
WWW  WWWW      WWW       WWWWWW
WWW                       WWWWW
WW  W                    WWWWWW
WW WWWWW                 WWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
Sea level: 2
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
W   WWWWWWWWWWWWW       WWWWWWW
W   WWWWWWWWWWW        WWWWWWWW
W   WWWWWWWWW          W  WWWWW
WW        WW         WW  WWWWWW
W  WWW              WWWWWWWWWWW
W  WWWW                      WW
W  WWW              WWWW  WW WW
WW WW               WWWW  WW WW
WW                    WW     WW
WWW WWWWW      WWWWWWW   WWWWWW
WWWW                      WWWWW
WWWWW                    WWWWWW
WWWWWWW                  WWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
Sea level: 3
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWW       WWWWWWW
WWWWWWWWWWWWWWW        WWWWWWWW
WWWWWWWWWWWWW         WWWWWWWWW
WWWWWWWWWWWW         WWWWWWWWWW
WWWWWWWWWW          WWWWWWWWWWW
WWWWWWWWW           WWWWWWWWWWW
WWWWWWWW            WWWWWWWWWWW
WWWWWWW             WWWWWWWWWWW
WWWWWW                WWWWWWWWW
WWWWWWWWWWWWWW WWWWWWWWWWWWWWWW
WWWW                      WWWWW
WWWWW                    WWWWWW
WWWWWWW                  WWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
Sea level: 4
```

```
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWW  WWWWWWWWWWWWWW
WWWWWWWWWWWWWW    WWWWWWWWWWWWW
WWWWWWWWWWWW       WWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWW  WWWW  WW WWWWWWWWWWW
WWWWWWWW WW WW WWWW WWWWWWWWWWW
WWWWWWWW    WW WWWW WWWWWWWWWWW
WWWWWWWW WW WWW  WW   WWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWW                WWWWWWWW
WWWWWW                  WWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
```

29 Hexagons

23 points

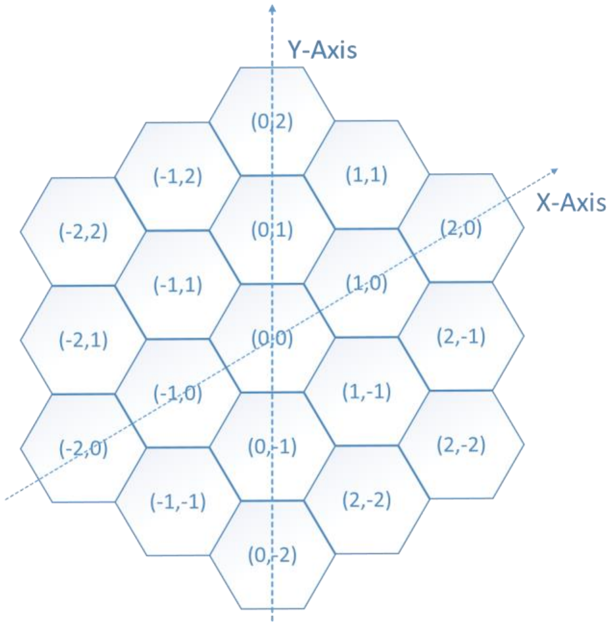
Introduction

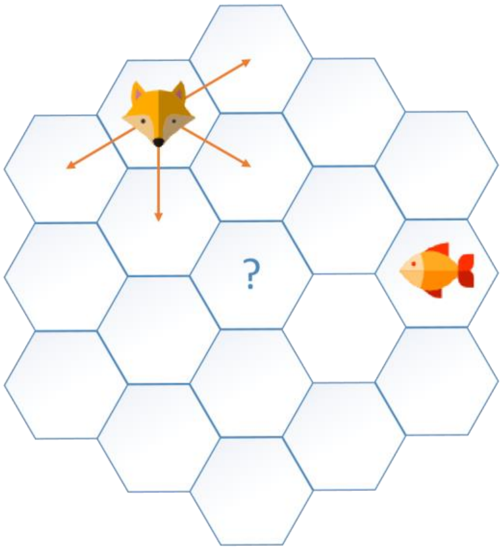
Clusters of hexagonal columns known as the Giant's Causeway can be found along the coast of Northern Ireland. Legends claim the causeway was built by Irish giant Finn MacCool, who had been challenged to a fight by Scottish giant Benandonner.

The columns vary in height, and some have a small hole in the top. When the waves break with the columns, sometimes fishes get trapped in the holes.

Foxes live in the area, and they love to eat fish. To get the trapped fishes, a fox has to move above the columns. Each fox has two characteristics: the maximum jump, and the maximum fall it can survive. A fox only can jump to a near hexagon if it differential height is less or equal to its maximum jump, and only can fall to a near hexagon if the differential height is less or equal to its maximum fall it can survive.

We can identify each hexagonal column using a (x,y) notation using the following convention:





We want to know if a fox positioned in a certain hexagon is able to get a certain fish.

Input

- Multiple lines, each one with a hexagon x,y coordinates and its height
- Line with end character #
- Multiple lines, each one with the maximum jump, the maximum fall and the fox and the fish coordinates

Output

For each line with a fox and fish description:

- If exists a path: “The fox says: what a delicious fish!”
- If not: “The fish says: not today, little fox!”

Example

Input

```
0 0 0
0 1 1
0 2 2
0 3 5
0 4 0
#
0 0 0 0 0 0
1 1 0 0 0 2
1 1 0 0 0 3
3 1 0 0 0 3
3 1 0 0 0 4
3 5 0 0 0 4
```

Output

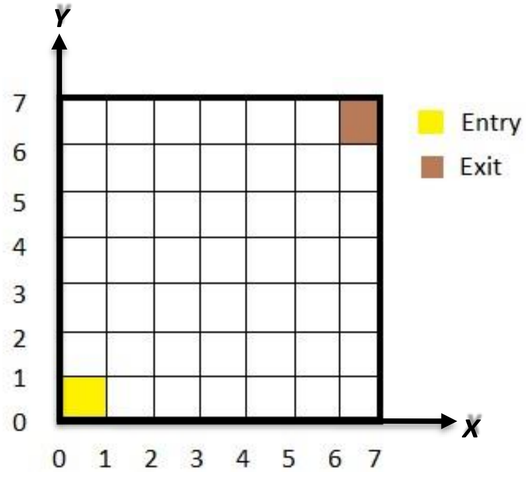
```
The fox says: what a delicious fish!
The fox says: what a delicious fish!
The fish says: not today, little fox! The
fox says: what a delicious fish!
The fish says: not today, little fox!
The fox says: what a delicious fish!
```

30 Maze

23 points

Introduction

Calculate the shortest path from the entry square to the exit square in a given maze of size DimX x DimY. The entry will be in the bottom left of the maze and the exit in the upper right of maze. The movements along the maze can be one square up, down, right or left.



In this example it is shown a maze of 7 x 7

RESTRICTIONS:
DimX > 1
DimY > 1
N ≥ 0
$0 \le OX_n, DX_n \le DimX$
$0 \le OY_n, DY_n \le DimY$

Input

For each test, the first line will be DimX, number of columns of the maze.
The second line will be DimY, number of rows of the maze.
The third line will be N, number of walls in the maze.
The next N lines will be “OXn, OYn, DXn, DYn”, the coordinates of the origin and the destiny of the wall n.

Output

The output must one line. It will contain the distance from the entry to the exit. In case it is impossible to reach the exit the output will be -1.

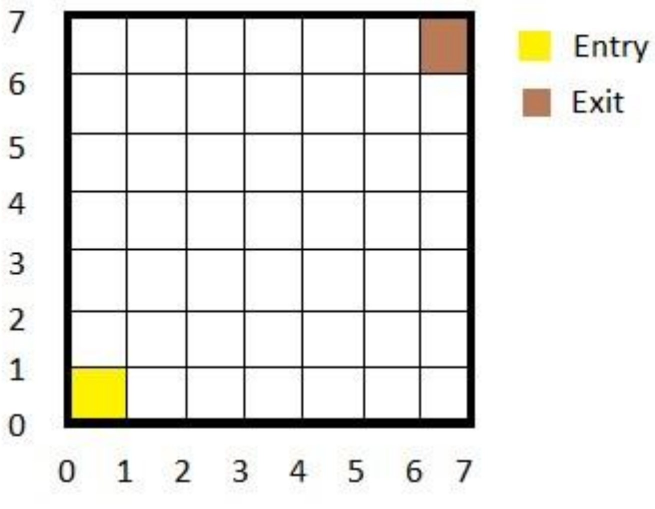
Example 1

Input

```
7
7
0
```

Output

```
12
```



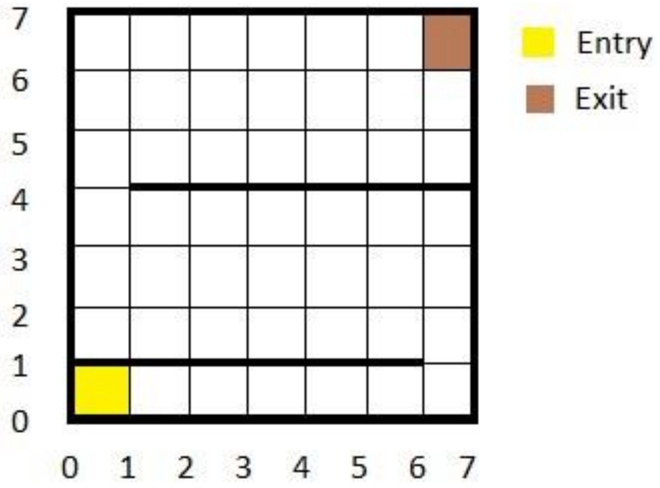
Example 2

Input

```
7
7
2
0,1,6,1
1,4,7,4
```

Output

```
24
```



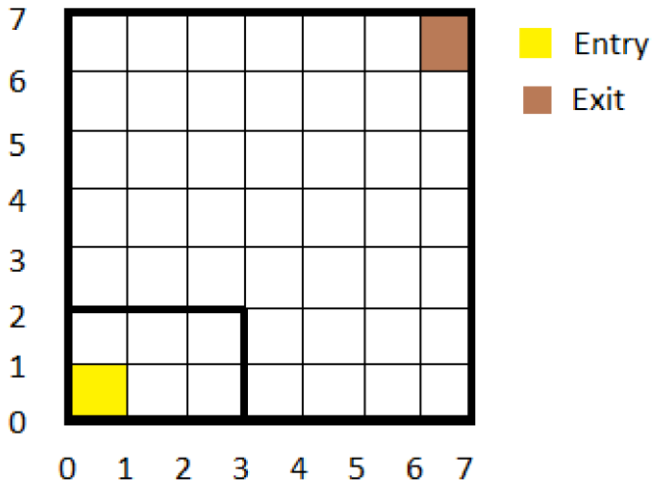
Example 3

Input

```
7
7
2
0,2,3,2
3,2,3,0
```

Output

```
-1
```



31 Keyboard

30 points

Introduction

Nowadays, being proficient in English is mandatory. You know that, and that's why you've spent the last months studying hard for an official English exam. Since you're a computer programmer, you've chosen to do the computer based exam. The first part of the exam is the writing test. You arrive at the room, you sit in front of the computer and… OMG! The keyboard is not a standard one! It has several letters in each key.

You need to do the writing test with that strange keyboard. You have a text you have to write using that keyboard and you need to minimize the number of mistakes.

The length of the text you want to write and the text you will actually write must be the same. A mistake occurs when the letter of the ith position in the ideal text is not the same as the letter in the ith position in the real text.

For example, imagine you want to write HELLO_WORLD and you have 5 different keys, with the following letters each key:

```
HEL
O_WOKLD
WOR
HE
LL
```

The best way to write HELLO WORLD is HE|LL|O_WOKLD. In that case we've used the 2nd, 4th and 5th keys and we've done 1 mistake. We've written a K instead of an R.

Input

Each case consist of the word W you want to write, then the number N of keys in the keyboard followed by N different sets of letters.

Output

Print the minimum number of mistakes you'll do when you want to write the word W using this strange keyboard. Remember that W and the word you'll actually write must have the same length. If no combination of keys gives a word of the same length as W, print -1.

Example 1

Input

```
HELLO_WORLD
5
HEL
O_WOKLD
WOR
HE
LL
```

Output

```
1
```

Example 2

Input

```
WE_ARE_THE_CHAMPIONS
9
ADKE_
__
ARE
CHAM_PIO
DDKK
WEEEE
W
KLLLE
IIWW
```

Output

```
9
```