



Important Contest Instructions!!

Please read the following instructions carefully. They contain important information on how to run your programs and submit your solutions to the judges. If you have any questions regarding these instructions, please ask a volunteer before the start of the competition.

Program Input

Most programs will require input. You have two options:

- 1) Your program may read the input from a file. The input data will be in the local directory in the file **probXX.txt**, where 'XX' is the problem number.
- 2) Your program may read the input from the keyboard (standard in). You may type everything on the keyboard, or you may copy the data from **probXX.txt** into the standard in. **Tip:** Type 'Ctrl-Z <return>' to signal the end of keyboard input.

Note: An easy way to enter keyboard data is by redirecting the contents of a file to your program. For example, if you are executing prob01, the input file **prob01.txt** can be redirected to the standard in of your program using syntax like this (examples are shown for each of the allowed languages):

```
%> java prob01 < prob01.txt
%> java -jar js.jar prob01.js < prob01.txt
%> python prob01.py3 < prob01.txt
%> prob01.exe < prob01.txt
```

Your program will behave exactly as if you were typing the input at the keyboard.

Program Output

All programs must send their output to the screen (standard out, the default for any print statement).

Submitting your Programs

Interpreted Programs (Java, JavaScript, Python) Your program must be named probXX.java / probXX.js / probXX.py2 / probXX.py3, where 'XX' corresponds to the problem number. For Python, use the extension that matches the Python version you are using. Please submit only the source (.java, .js, .py2 or .py3). For java, the main class must be named probXX. Note there is no capitalization. All main and supporting classes should be in the default (or anonymous) package.

Native Programs (C, C++, etc.) Your program should be named probXX.exe, where 'XX' corresponds to the problem number.

You are strongly encouraged to submit solutions for Problems #0 and #1 (see next pages) prior to the start of the competition to ensure that your build environment is compatible with the judges' and that you understand the Input and Output methods

Problem.0 Hello, Teach

+ 1 point

Summary

The sole purpose of this problem is to allow each team to submit a test program to ensure the programs generated by their computer can be judged by our judging system. Your task for this program is a variation on the classic "Hello World!" program.

On this day in 1867, the United States Congress created the Department of Education. In tribute, all you need to do is print out "Hello, Teachers!" to the screen.

Output

| Hello, Teachers!



Problem.1 Social Graces

+ 1 point

Summary

Without good social interaction skills, life will be more challenging for you. And if you don't know how to do Input and Output properly, the CodeWars competition will not go well for you either. So, here's your chance for a little practice on both before the actual CodeWars contest begins.

Write a program to respond kindly to someone on social media, using their first name.

Input

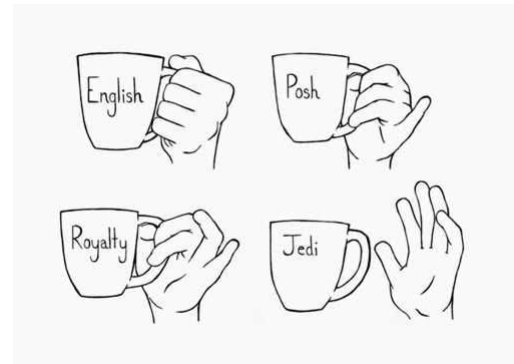
Input will consist of a single line with a single word that is your new friend's first name.

| Dolores

Output

Respond to your social media friend with the following text, including the input name as shown in the example.

| While we seem to disagree on this issue, Dolores, I respect your opinion and look forward to further discussion!



Problem.2 Will You Make it?

+ 2 points

What a nightmare! You woke up late and still have to travel to CodeWars. Do you have time?

Input

Read 3 integers on one line separated by spaces "H M S", representing the Hours until CodeWars starts, the miles you need to travel, and your speed in miles per hour.

```
| 25 100 4
```

Output

Repeat the input, then add a period and space and either "I will make it!" or "I will be late!"

```
| 25 100 4. I will make it!
```

Other Examples

```
| Input: 2 60 29
| Output: 2 60 29. I will be late!
|
| Input: 3 2 1
| Output: 3 2 1. I will make it!
```

Problem.3 At the end of your rope

+ 2 points

The latest consideration for a CodeWars mascot has been put out to pasture, literally. The dejected goat has been tied to the corner of a square barn in the middle of a grass field. At least he gets to eat!

Input

Read 1 positive integer, the length in feet of the rope. Maximum of 100 feet.

| 20

Output

Print the area (in square feet of grass) the goat can reach. It normally would be a full circle, but two walls of the barn are in the way. The rope is shorter than the walls of the barn. The area of a circle with radius r is $A = \pi * r * r$. Use $\pi = 3.14159$.

| 942.48



Problem.4 A Taxing Question

+ 3 points

In most countries (unlike the U.S.) sales taxes are included in the listed price of items. So, if the menu lists a price of \$2.50 for a Jiffy Burger, then that's what you pay -- the sales tax is included already. Much simpler, right?

Johnny, however, is a tax freak. So, while traveling in Outer Tasmania, Johnny wants to determine how much of each purchase is made up of tax. So, your job is to write a program that will compute the tax portion of a purchase, given the local tax rate and the purchase amount.

```
pretax_price = total / (1.0 + tax_rate)
tax_amount = pretax_price * tax_rate
```

The first parameter of your input will be an integer that represents the number of tax rate & purchase amount pairs to analyze. The remaining input parameters consist of sequential tax rate/purchase amounts pairs as shown in the example below. All rates and purchase amounts will include no more than two decimal places, and the computed tax amount should NOT be rounded up. All output amounts should be displayed out to two decimal places as shown in the example.



Sample Input

```
5
8.25
500
5.00
17.35
2.37
412.13
6
99.99
7.5
77.77
```

Sample Output

```
On your $500.00 purchase, the tax amount was $38.11
On your $17.35 purchase, the tax amount was $0.83
On your $412.13 purchase, the tax amount was $9.54
On your $99.99 purchase, the tax amount was $5.66
On your $77.77 purchase, the tax amount was $5.43
```

All purchase amounts will be less than \$10,000.

Problem.5 Building Pyramids

+ 3 points

The weather has kept everyone inside. The math teacher has given everyone an interesting assignment. Take cubes of sugar and build pyramids. Then she asks how many cubes you used. The problem is each team built a pyramid of a different height. Can you help everyone count their cubes quickly?

Input

Read 1 positive integer, the height of the pyramid. Each layer is a full square and has one less cube along its length than the layer beneath it. The top layer is a single cube. The maximum value will be 1000 (it has been a long winter.)

| 6

Output

Print the number of cubes in the entire pyramid.

| 91

Explanation

The bottom layer has 6x6 cubes (36). The next higher layers have 25, 16, 9, 4, and 1 cubes each, for a total of 91.



Problem.6 Glad to Meet You

+ 3 points

Several couples have arrived at a party and everyone introduces themselves and shakes hands.

Being the ever diligent student of human behavior, you wonder just how many times introductions were made.

Input

Read one integer N, the number of couples who arrived. Maximum of 100 (that's a big party!)

| 4

Output

If everyone shakes hands once with every other person there (not shaking hands with themselves or with their date), print the number of handshakes.

| 24

Explanation

For 4 couples, call them AA, BB, CC, DD. Both of couple AA each shake hands with BB, CC, DD, for a total of 12 handshakes. Couple BB has already shaken hands with AA, so now shakes hands with CC and DD (8 handshakes). Finally, Couple CC shakes hands with DD (4 handshakes.) Total of 24 handshakes.

Example 2 Input

| 10

Example 2 Output

| 180



Problem.7 It's CodeWars Math time!

+ 4 points

You gotta know how to do math to get through CodeWars. So, show your style!

Input

Each line of input has 2 positive integers N and M, separated by one space. The first integer (N) is between 0 and 1 million. The second (M) will be a 0, 1, 2, or 3. Your program must find the next integer that is Even (M=0), a Prime (M=1), a Square (M=2) or a Cube (M=3).

```
12 3
20 2
999999 3
3 3
12 0
0 0
```

Output

For each line, your program must output the next positive integer greater than N that is Even (0), Prime (1), a Square (2), or a Cube (3). If both N and M equal 0, your program should end (and not print).

```
27
25
1000000
8
14
```

Problem.8 Number Scrabble

+ 4 points

For some reason, your school's football team has chosen to spell out the numbers on their jerseys instead of using the usual digits. Being great fans, you're going to be ready to cheer for your favorite players by bringing letter cards so you can spell out their number. Each fan has different favorites, so they each need to bring different sets of letters.

The English spellings for the numbers 0 to 12 are:

```
| ZERO ONE TWO THREE FOUR FIVE SIX  
| SEVEN EIGHT NINE TEN ELEVEN TWELVE
```

Input

Read a set of integers from 0 to 12, separated by spaces, representing one fan's favorite players. The last integer will be 999, marking the end of the line.

```
| 0 1 4 7 9 11 999
```

Output

Print the same numbers, then a period and a space. Then, in alphabetical order, print all the letters the fan needs to be able to spell any one of the jersey numbers provided.

```
| 0 1 4 7 9 11. E E E F I L N N O R S U V Z
```

Example Input 2

```
| 2 3 999
```

Example Output 2

```
| 2 3. E E H O R T W
```

Problem.9

Rock Paper Scissors Lizard Spock

+ 5 points

Implement the game: Rock Paper Scissors Lizard Spock [1]

The Rules Are:

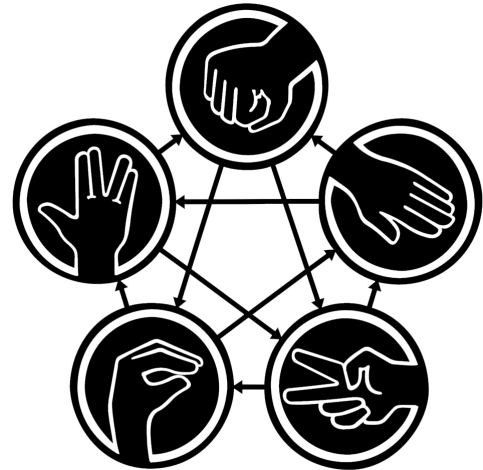
```
Scissors cuts Paper
Paper covers Rock
Rock crushes Lizard
Lizard poisons Spock
Spock smashes Scissors
Scissors decapitates Lizard
Lizard eats Paper
Paper disproves Spock
Spock vaporizes Rock
Rock crushes Scissors
```

You will receive the choices from both players at the same time in the form of:

```
| Choice1 Choice2
```

You are guaranteed to only receive one game choice at a time. You are guaranteed that the game choice will be formatted properly, and will exist.

If the same choice is made for both sides, the result is a tie. State the winner in terms of the first choice made (the choice to the left of the space). Else, state the outcome and the winner.



Sample In 1:

Scissors Lizard

Sample Out 1:

SCISSORS WINS, Scissors decapitates Lizard

Sample In 2:

Scissors Rock

Sample Out 2:

SCISSORS LOSES, Rock crushes Scissors

Sample In 3:

Rock Rock

Sample Out 3:

TIE, ROCK does not affect ROCK

[1]

Game source: <http://samkass.com/theories/RPSSL.html>

Game license: <https://creativecommons.org/licenses/by-nc/3.0/>

Problem.10 Age Calculator

+ 5 points

You want to know the day you will turn 18, and decide to write a program to tell you that. Your friend was born on a leap day, and wants to use your program too, and she also wants to know how old she'd be on the date the last Avengers movie will come out.

To do all that you will need to read in data containing an unknown number of lines. Each line will either be two dates or a date and an integer. Given a line with two dates separated by a space, calculate the age of a person taking the first date as their birth date, and the second date as the date to check their age.

Example:

```
| 1967-01-31 1984-01-31
```

Given a date and an integer separated by a colon, calculate what date a person will be the age given by the integer, if the integer is greater than zero and less than one hundred.

Example:

```
| 1968-08-06:39
```

Note:

```
| Dates will range from Jan. 1st 1583 to Dec. 31 3000.  
| Keep in mind leap years! If the year is evenly divisible by 4, it is a leap year. However, if the year is  
| also divisible by 100 evenly, it is NOT a leap year (unless it is also divisible by 400).  
| Most modern programming languages have Date libraries which should make your life easier.
```

If the age given is zero, instead calculate the date the person will STILL be zero years old. If the age is greater than 99, calculate the age as usual, but add "(good job!)" to the end of the output.

Output as follows:

Example In:

```
| 1968-08-06:39  
| 1967-01-31 1984-01-31  
| 2001-08-13:99  
| 2019-07-04:110  
| 2005-12-24:0
```

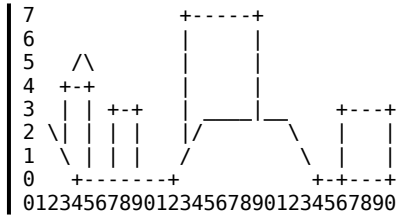
Example Out:

```
| Will be 39 on 2007-08-06 if born on 1968-08-06  
| If born on 1967-01-31, will be 17 years old on 1984-01-31  
| Will be 99 on 2100-08-13 if born on 2001-08-13  
| Will be 110 on 2129-07-04 if born on 2019-07-04 (good job!)  
| Will still be 0 up to 2006-12-23 if born on 2005-12-24
```

Problem.11

+ 5 points

Draw the outline of the San Francisco skyline. I've been there, but this is totally what I see in my imagination, Not all lines will have the input size max characters. Also, my panoramic camera has some ordering issues, so be careful.



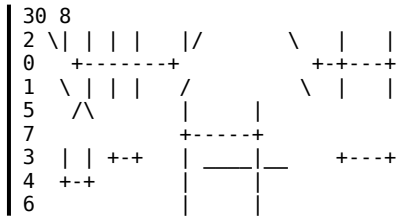
Input:

Two integers, The width of your skyline and the height of your skyline, The Y coordinate and the string that represents the skyline at that elevation.

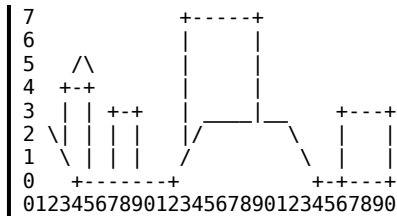
Output

Draw the resulting skyline from the input provided.

Sample Input:



Sample Output:



Problem.12 There's Always Room for Pi

+ 5 points

The captain of your starship needs you to check the access tubes for the garbage chute. Since you have no intention of climbing around down there (lest you run into an angry Wookie climbing out), you have decided to write a program to check the tubes using drones. Before you send your drone down there though, you need to check if it will fit. The access tubes are circular, so you'll need to have your program calculate Pi while it flies. In order to save time, you will set your algorithm precision so it will run faster in larger tubes. You'll check the output of your program by looking at a random digit after it is done calculating, and comparing it against the expected digit.

You plan to implement Nilakantha's Series to calculate Pi. The series is:

$$Pi = 3 + (4/2*3*4) - (4/4*5*6) + (4/6*7*8) - (4/8*9*10) + \dots$$

Generic formula:

```
n = 1000 //the number of times to iterate over the series
sign = 1 //determines whether or not to add or subtract
Pi = 3 //base starting value of Pi

Loop from i=2 until (n*2)
  If sign is 1
    Add (4/i*(i+1)*(i+2)) to Pi
  else
    Subtract (4/i*(i+1)*(i+2)) from Pi
  Multiply sign by negative 1
  Add 2 to i
End Loop
```

Input

```
Line 1 will contain the value for n. Your program will need to run for that number of
iterations.
Line 2 will contain the decimal digit to output. Your program may calculate something
like: 3.141592653589787, but you should only output the decimal digit asked for in the input

200000
14
```

Output

```
If you only calculate Pi with 200000 iterations, it will only be accurate to the 13th
decimal place. All of the other digits will be wrong. But that's fine, because you
don't need high accuracy for the larger tunnels you're sure your drone will fit in.

8
```

Problem.13 More Things In Heaven & Earth, Than Poor Ratios

+ 5 points

Your last algebra test tells you that you need to work on your ability to solve ratio comparison problems. You kept getting the math wrong. So, you've decided to create a function which will take 2 ratios with an unknown variable "X" and have it solve for "X" so you can check your homework and learn how to do it properly, hopefully before the next test!

The formula for determining "X" when comparing two ratios is to simply take the two ratios setup as fractions, cross multiply and solve for "X".
E.G.:

```
X/10 = 20/200
200X = 200
200X/200 = 200/200
X = 1
```

You will receive 4 lines in your input. Line 1 will be the numerator of the ratio on the left side of the equal sign. Line 2 will be its denominator. Line 3 will be the numerator on the right, and line 4 its denominator. 3 of the inputs will be numbers and 1 of the inputs will simply be an "X". Your program will need to be able to solve for X regardless of its location in the equation input.

Input

```
20
100
X
590000
```

Input Explanation

```
| That input should map to an equation as follows:
| 20/100 = X/590000
```

Output

```
| 118000.0
```

Note

```
| Answers correct to the first decimal place will be accepted as correct
```

Problem.14 Phi Won't Fly for Pi

+ 5 points

The Da Vinci Code claims that the Fibonacci sequence, also known as the golden ratio, unlocks the meaning of life, the universe, and everything. You don't believe that, but your painter does. The artist you've hired to paint your wall plans to do so as a series of expanding blocks in alternating black and white paint. Your painter has the vision, but you get to do the math!

The standard Fibonacci sequence is:

```
| 0,1,1,2,3,5,8,13,21...
```

The sequence is created by simply adding the sum of the prior 2 entries in the sequence together. The formula is:

```
| F(n) = F(n-1) + F(n-2), for n > 1; assuming F(0)=0,
| and F(1)=1 unless otherwise stated.
```

However, your painter hasn't decided whether to start painting at the first few blocks, or until later, so they want you to run calculations with various starting values for the sequence.

For example, if the sequence is started with $F(0)=1$ and $F(1)=2$, then the first few numbers in the sequence would be:

```
| 1,2,3,5,8,13...
| Which is just shifted a bit to the right.
```

Your calculations must be able to accept any starting value for $F(0)$ and $F(1)$ for the Fibonacci sequence. You will receive 3 numbers on 3 lines. The first number on the first line will be $F(0)$. The 2nd will be $F(1)$, and the 3rd will be $F(n)$ (e.g. how many numbers in the sequence to generate). You will need to output all numbers generated in the sequence from $F(0)$ up to $F(n)$, separated by a comma.

Example Input

```
| 2
| 4
| 10
```

Example Output

```
| 2,4,6,10,16,26,42,68,110,178
```



Problem.15 A Crafty Display

+ 5 points

You are responsible for the window shelf display at the new craft store. You have several unique books and candles you can arrange in the window. The owner would like a different display every day.

The shelf can fit up to 4 items and each display must include at least one book and one candle. (Left-to-right order doesn't matter; just consider the unique items in the display.)

Knowing how many books and candles you have, can you determine how many days you can decorate before needing more?

For example, if you have 3 books (abc) and 2 candles(xy), there 6 ways to place 2 items (ax ay bx by cx cy), 9 ways to place 3 items (abx aby acx acy bcx bcy axy bxy cxy), and 5 ways to place 4 items (abcx abcy abxy acxy bcxy), for a total of 20 different sets.

Input

Read two integers B and C on one line, representing the number of books and candles you have.

| 3 2

Output

Print the number of different displays you can create.

| 20

Problem.16 Too Too Many Words

+ 6 points

You are working on on your first novel, and while the spell checking and grammar checking tools are helping a lot, one of of the things which keeps getting missed is is accidental duplicate words. You have decided to to write your own program which will remove duplicate words from your text.

Given an arbitrarily long string of of words, remove all duplicate words.

But you must keep two special cases: "is is" and "had had". For these special exceptions, only 2 in a row are allowed. For example, if you find "is is is", change it to "is is".

Ignore case of letters when comparing for duplicates. You will receive one line of input which will be a long string containing 0-256KB of letters and spaces only. When deleting words, keep the first ones.

Input

| I am Am someone who had haD HAD a dog and and knowing what that is is Is iS a a a a happy thing

Output

| I am someone who had haD a dog and knowing what that is is a happy thing

Problem.17 Character Counter

+ 6 points

You are part of a new team tasked with coming up with a shortcut keyboard to be implemented on a wrist-panel for astronauts on the first extra-galactic space flight mission. Space Command has tasked your team with researching a new keyboard layout to replace QWERTY. Dvorak has already been eliminated.

Space is limited on the wrist-panels, so only a few letters will be visible at a time. Your team wants to sample real messages the astronauts are sending each other now and determine which characters are used the most.

Your program will need to read a string of letters and spaces and count how often each character is used. Then print the characters in descending ASCII order. (ASCII 'A' is 65. 'a' is 97. A space is 20.) If any character repeats 10 or more times, move that character to a new list instead, and sort that list in ascending ASCII order. You will display both lists ("regular" and "10+ count"), separated by a semi-colon.

Input

Input is a single line of input between 1 and 256 characters.

Output

Output on a single line all characters in the sorted lists with the character followed immediately by the count of that character in square brackets ([]). Separate the two lists ("regular" and "10+ count") with a semi-colon. For the space character, use the underscore character (_). For example:

```
ccooo DEEEEEEEEEE
o[3]c[2]D[1]_[1];E[10]
```

If no character repeats 10 times, you should still print the semi-colon at the end of the output. Conversely, if you only have characters that repeat 10+ times, you should start your output with the semi-colon.

Example Input

```
I am the very model of a modern Major General Ive information vegetable animal and mineral Im very well
acquainted too with matters mathematical I understand equations both the simple and quadratical
```

Example Output

```
y[2]w[2]v[4]u[4]s[4]q[3]p[1]j[1]h[5]g[1]f[2]d[8]c[3]b[2]M[1]I[4]G[1];_[31]a[22]e[20]i[10]l[10]m[11]n[12]o
[10]r[10]t[15]
```

Problem.18 Fortune Teller

+ 7 points

Code a simulation of a "MASH" (Mansion Apartment Shack House) type game. MASH is a multi-player paper-and-pencil game, commonly played by kids and intended to predict one's future with a story.

Setup for the game is simple. First, one creates some categories that will comprise your story, and under each category lists some possible options. In the paper-and-pencil game, the player would determine a "magic number" by closing their eyes and drawing on the paper while their friend counted until they said "stop" to determine their "magic number." In the simulation you will be coding, you will simply read in this "magic number" from the input.

To play the game, step through each category's options until the iterator ("magic number") has been reached, eliminate the option you are on. Repeat the process until only a single option remains, then move on to the next category, and repeat.

Example with a Magic Number of 4:

```
Colors Category:
Red
Blue
Green
```

Starting with Red as "1", move to Blue as "2", then Green as "3", then back up to Red as "4". Red is eliminated. Start over with Blue as "1", then Green as "2", then back up to Blue as "3" then to Green as "4". Green is eliminated. Blue is the final option left under the category, so stop iterating.

Input

Each input dataset will contain only four categories, but the number of options per category will vary.

The first line of input specifies the number of options for each category. This is followed for four sets of category data. Category names will be on their own line followed by the the word "Category" and a colon. Available options for each category will follow, one per line. Following the category data, the next two lines of the input will be a line that says "Magic Number:" followed by a line with an integer "magic number." It will always be 1 or greater (never zero or negative).

Output

Print the player's "MASH Story" with the final option in each category.

Example In:

```
4
Degree Category:
Associates
Bachelors
Masters
Doctorate
State Category:
California
Texas
New York
Ohio
Car Category:
Tesla
Ford
Honda
Jeep
Years In College Category:
3
4
5
6
Magic Number:
3
```

Example Out:

```
Your MASH Story:
Degree - Associates
State - California
Car - Tesla
Years in College - 3
```

Problem.19 Simple String Encryption

+ 8 points

We are a global multi-mega media warehouse and we need our bits encrypted! We are looking for skilled encryptors and we've a good test to weed out the unskilled. We look forward to your attempts!

We will give you a Password and the text to encode. If your encoding matches our expectations, we will gladly reward you by further considering your efforts.

First, create an integer "key" from the password by parsing each character in the password: take the ASCII value of each character and alternately add and subtract each, starting with addition. (In case you don't know the ASCII values, upper-case A-Z are 65-90, and lower-case a-z are 97-122.) If the calculated key is less than 32, add 32 to the value until it is greater than or equal to 32. If the calculated key is greater than 126, subtract 16 from the value until it is less than or equal to 126.

Example key from password: ABC (A=65, B=66, C=67)

```
key = 0 + 65 - 66 + 67
key = 66
```

Next, to encrypt the string, take each character in the string and multiply the ASCII value of the character by the key, then convert the product to base-16.

Input

The first line of the input will be the password to use. All remaining lines will be strings to encrypt.

```
PASSWORD
Hello
World
```

Output

Output the key, and each base-16 encrypted string in a comma delimited list with no spaces.

```
Key = 37
a68,e99,f9c,f9c,100b
c93,100b,107a,f9c,e74
```

Problem.20 Phone Passcode Guessing

+ 8 points

Your friend has bet you \$20 that you cannot figure out their 4-digit phone passcode given 3 guesses. Using the smudges left on their screen over the numbers, and your knowledge of your friend's passcode rules, try to figure out their passcode.

Your friend always chooses a passcode where the first 2 digits (as a 2-digit number) are the square of the second 2 digits (as a 2-digit number).

Input

The dataset represents a phone passcode input screen:

```
| 123
| 456
| 789
| X0X
```

The dataset will be marked with C or U.

C = clearly a number in the passcode.

U = unclear, possibly used in the passcode.

If a location is a digit or an X, then it isn't in the passcode.

EXAMPLE INPUT:

```
| 1C3
| 4U6
| 789
| XCX
```

For this input, 2 and 0 are definitely used. 5 might be. No other digits are in the passcode.

Output

If you only find one possible combination of digits which could be the passcode, output:
CODE IS: #####

If two or three possible passcodes are found, output each on a new line as:

POSSIBLE MATCH: #####

POSSIBLE MATCH: #####

POSSIBLE MATCH: ##### (if 3 codes are found, output 3 lines)

If four or more possible passcodes are found, output:

WILL NOT WIN BET

If no possible passcodes exist, output:

COULD NOT DETERMINE CODE

EXAMPLE OUTPUT:

```
| CODE IS: 2505
```

Problem.21 Chinese Zodiac Receipt

+ 8 points

You've just started a new job at a large food conglomerate, Nom Nom Foods. Your first assignment is to make an enhancement to the customer receipts printed at restaurants in their Chinese food division. In order to make things a little more fun, Marketing would like you, given the receipt date (passed as a string), to print out 1) the Zodiac sign corresponding to that date, and 2) the Chinese Zodiac animal corresponding to that year. You should use the format shown in the example below.

Because the restaurant cash registers are sometimes placed into testing mode by the Federal Department of Weights and Measures, your code will need to gracefully handle all valid ISO-8601 Gregorian/Joda dates between the range of Jan. 1 1583 and Dec. 31 3001.

Remember to account for leap years! A year is a leap year if it is evenly divisible by 4 (unless it is evenly divisible by 100, then it is not -- unless it is also divisible by 400).

To determine the Zodiac output, the rules are simple. The Chinese Zodiac animal for the year rotates every 12 years. The 12 most recent animals were:

```
2008 - Rat
2009 - Ox
2010 - Tiger
2011 - Rabbit
2012 - Dragon
2013 - Snake
2014 - Horse
2015 - Sheep
2016 - Monkey
2017 - Rooster
2018 - Dog
2019 - Pig
```

The date ranges for the Zodiac sign of a person's birth have been determined by the company's HR department to be:

```
Aries      March 21 - April 19
Taurus     April 20 - May 20
Gemini     May 21 - June 20
Cancer     June 21 - July 22
Leo        July 23 - August 22
Virgo      August 23 - September 22
Libra      September 23 - October 22
Scorpio    October 23 - November 21
Sagittarius November 22 - December 21
Capricorn  December 22 - January 19
Aquarius   January 20 - February 18
Pisces     February 19 - March 20
```

Example In:

```
| 2019-03-02
```

Example Out:

```
| If you were born on March 2, your sign is Pisces.
| 2019 is the year of the Pig.
```

Problem.22 Insta-where?

+ 9 points

Good news, you got a job at Instagram right out of college! Bad news, your very first assignment requires that you work with a data file with less-than-perfect data sources.

You need to create a process to figure out a list of user's countries based on their recorded GPS coordinates.

The data file you will be using is in the format of:

```
Users List
-----
Country List
```

The "-----" marks a data region change.

You'll need to cross-reference each user's location against the official Instagram country list which will be in the form of:

```
Name| (x.xx,y.yy)
Example:
Mara Jade-Skywalker| (3.33,6.91)
```

The country list will be in the form of:

```
Country Name: (x1,y1);(x2,y2);(x3,y3);(x4,y4)
Example:
Japan: (45,8);(46,8);(45,11);(46,11)
```

If the user's location was unknown, their coordinates will be (-100,-100). If a user's country can't be found, simply list those users under "Unknown".

A user's location will be considered to be "inside" a country if the rectangle defined by the coordinates of the country contains the point, or the point is on the border of the rectangle (which includes the corners).

Once you have determined the country each user should belong to (or that it is unknown) output an alphabetized list of the countries, with the users listed under their country (do not list countries with no users). Put any "unknowns" at the end/bottom of the list.

You DO NOT need to alphabetize the list of users listed under each country.

Sample (Abbreviated) Input

```
Mara Jade-Skywalker| (45.01,9.01)
-----
Japan: (45,8);(46,8);(45,11);(46,11)
USA: (1,6);(9,6);(1,12);(9,12)
```

Sample Output

```
[Japan]
-Mara Jade-Skywalker
```


Problem.23 Scoring Social Media Posts

+ 10 points

Your favorite social media network has opened up an API for you to control your daily 'feed' of posts. You want to take the opportunity to clean up your feed and implement your own scoring method to sort posts from your friends and family as well as celebrities which you may be following. The social media network will put posts into your feed from people you aren't following as suggestions, it will also put posts into your feed from people you have blocked. You want to score celebrity posts more harshly so that the odds of their posts rising to the top of your feed are low, unless their post is universally liked. Additionally, anyone you have blocked you want to remove all of their posts from your feed entirely.

Given a set of posts for your social media feed, sort and display the posts according to a scoring system. Each pipe (|) delimited line of data will consist of:

```
A post title
Poster Name
Post likes
Post dislikes
Poster's follower count (warning, this can be a huge number)
If you are following them
If you have blocked the poster
Posting date (in format: yyyy-MM-dd)
Poster's join Date
Total posts from poster
```

Calculate an "importance" score for the posts (the larger the number, the higher the "importance") based on the following formula:

Starting score = 10,000 points.

If the poster has a million followers (or more), add 10,000 points.

If the poster has less than one hundred followers, subtract 5,000 points.

Divide the score by the total likes for the post subtracting the total dislikes, if the likes do not equal the dislikes, else divide by 100.

If you are following the poster, add 5,000 points.

If the post is a week or less old, add 500 points, else if the post is more than 2 weeks old, subtract 100 points.

Add the poster's total number of posts, and the total number of weeks since the poster joined.

Remove the post from the list entirely if you've blocked the poster.

You are safe to assume that each data line will be complete and not missing any data.

Once the scores are calculated for each post, sort the list in descending order of their "importance" score.

In the unlikely event of a tie, output in the order the lines were read in.

Finally, output the list in the scoring order you have devised in the format:

(importance score) Post title, by: Poster Name [Posting date (in format: yyyy-MM-dd)]

Example:

(22314.9) An example post, by Jane Smith [2019-03-04]

If the scoring process removes all posts from the list, output instead:

You are blocking all (##) posters in your feed, where ## is the number of unique poster names (CaSe sEnSiTiVe) found in the list.

Input

```
Been a long day, time for some lily pad banjo|Kermit The Frog|98765|2|64123456|true|false|2018-12-20|
1955-05-09|4567
Can't hire good help to save my life #contractorProblems #SithLife|Emperor Palpatine|15|99871625|515623|
false|false|2018-01-13|2009-03-15|998
The Vulcan body has over one trillion nerve endings. Amazingly, some people can get on all of them
#emotionalControlFailing|Spock|159|1|35421222|true|false|2018-03-15|2230-01-06|35
Don't believe Vulcans about their emotional control, they run hotter than we do
#youreNotFoolingAnyoneSpock|CPT. J.T. Kirk|302|7|7123444|true|false|2018-03-15|2233-03-22|852
Don't make me come up there! Mr. McCoy, calm them down, they're upsetting my engines!|Scotty|199|0|8562|
false|false|2018-03-15|2222-03-03|19
Damn it man, I'm a doctor, not a babysitter! #SMH|Bones McCoy|8|1|235|false|false|2018-03-15|2227-09-06|3
I'd be happy to teach him some manners, Mr. Spock. #MyPenIsHungry|Dolores Umbridge|1|78456321|1|false|
true|2018-03-15|1965-08-26|87410
```

Output

```
(12796.2) Been a long day, time for some lily pad banjo, by: Kermit The Frog [2018-12-20]
(1417.0) Can't hire good help to save my life #contractorProblems #SithLife, by: Emperor Palpatine
[2018-01-13]
(-5349.2) Don't believe Vulcans about their emotional control, they run hotter than we do
#youreNotFoolingAnyoneSpock, by: CPT. J.T. Kirk [2018-03-15]
(-5940.4) The Vulcan body has over one trillion nerve endings. Amazingly, some people can get on all of
them #emotionalControlFailing, by: Spock [2018-03-15]
(-9548.4) Damn it man, I'm a doctor, not a babysitter! #SMH, by: Bones McCoy [2018-03-15]
(-10622.7) Don't make me come up there! Mr. McCoy, calm them down, they're upsetting my engines!, by:
Scotty [2018-03-15]
```

Problem.24 Get your homework in!

+ 10 points

The airport WiFi has a content filter, and it is blocking your attempts to reach your school's VPN in order to submit your last assignment. You need to write a quick program to defeat the airport's content filter to allow you to connect to your school's VPN.

URL encoding takes the 2-character HEX value of an ASCII character, and prefixes it with a percent (%) symbol. For example, a space character is %20, and the tilde (~) character is %7E. A URL-encoded string (also known as "percent encoding") normally would change this address:

```
https://example.com:8080/schoolVPN?startConnection=begin!&schoolName=Jedi Academy&Connect0n=Number=#3
&username=Luke Skywalker&password=who invited all these EWOKS?!
```

to:

```
https://example.com:8080/schoolVPN?startConnection=begin%21&schoolName=Jedi%
20Academy&Connect0n=Number%23%3D3
&username%3DLuke%20Skywalker%26password%3Dwho%20invited%20all%20these%20EWOKS%3F%21
```

Normal URL encoding only changes the query string, which is everything after the first question mark in the URL. Generally, only characters other than numbers and letters get encoded. The airport filter is blocking this normal encoding.

To encode your URL, implement the above encoding, but instead of looking for the question mark, change ALL characters which are not letters or numbers to URL-encoded characters. Then, to get around the airport filtering, replace all percent signs (%) with the character sequence "0x25" which is the hex representation of the percent sign your school's VPN understands.

Special exception

The URL will be given in the form below. The domain has periods in it, and the port is optional, but the periods and colon may NOT be encoded. This exception only applies to the domain (the text between "://" and the first "/".)

```
https://DOMAIN.NAME(:port)/(something)?query
```

Domain Example:

```
https://hpe.com:443/powerChord?E=5&A=7&D=7
```

The domain is: hpe.com, and the port is 443, so you would leave "hpe.com:443" alone.

Sample Input

```
https://example.com:8080/schoolVPN?startConnection=begin!&schoolName=Jedi Academy&Connect0n=Number=#3
&username=Luke Skywalker&password=who invited all these EWOKS?!
```

Sample Output

```
https0x253a0x252f0x252fexample.com:80800x252fschoolVPN0x253fstartConnection0x253dbegin0x2521
0x2526schoolName0x253dJedi0x2520Academy0x2526Connect0n0x253dNumber0x253d0x252330x2526
username0x253dLuke0x2520Skywalker0x2526password0x253dwho0x2520invited0x2520all0x2520these0x2520EWOKS0x253f
0x2521
```

Note

The example shows line breaks for easier readability. Do not insert line breaks in your output.



Problem.25 Superhero Fight

+ 11 points

Who would win in a fight, Hulk® or Superman®? Would the USS Enterprise® win against a Star Destroyer®? What about a Super Star Destroyer®?[1] It's time to decide once and for all!

Your local comics and games shop is hosting a tournament this weekend, and they need a scoring system for their fantasy battle game. Teams will be able to submit a 'fighter' from one of 2 categories:

1. Person/Entity
2. Vehicle/Ship

A person will be able to draw from the following attributes (via dice rolls):

```
HP: Health (when a person's health reaches zero (or below), they lose the fight)
ST: Strength (determines damage bonus (how hard they hit) -- if >8 & <14 +1 to damage rolls, if >=14 & <18 +2 to damage, if >=18 +3 to damage rolls)
DF: Defense (determines minimal roll to hit)
LK: Luck (if >8 & <14 +1 to defense, if >=14 & <18 +2 to defense, if >=18 +3 to defense)
```

A Vehicle/Ship will be able to draw from the following attributes (via dice rolls):

```
SI: Structural Integrity (when reaches zero (or below), they lose the fight)
MW: Main Weapon (determines damage bonus (how hard it hits with primary weapon) -- if >8 & <14 +1 to damage rolls, if >=14 & <18 +2 to damage, if >=18 +3 to damage rolls)
AM: Armor (determines minimal roll to hit)
TK: Technology (if >8 & <14 +1 to armor, if >=14 & <18 +2 to armor, if >=18 +3 to armor)
```

Turns alternate per opponent.

Opponent with lowest defense/armor goes first (in a tie, opponent 1 goes first).

Combat rules per turn:

```
Roll 1-20 to determine "hit"
Check "hit" score against defense/armor + bonuses of defender
  if >= defense/armor, roll for damage
  else, turn ends
Rolling for damage:
  Roll 1-10 to determine "damage"
  add any bonuses
  subtract from target's health/structural integrity
If health/structural reaches zero or less battle is over, and the opponent with the greater than zero health/structural integrity wins.
```

Your scoring program needs to prove that it can accurately handle the inputs of a live tournament, so before the tournament will accept your program (and pay you for it), they want to run it through an automated trial run. They will submit pre-determined "battles" into your program, with all dice rolls pre-determined. Your program simply needs to apply the rules to the "battle" and output the winner.

Input

```
Opponent1:Hulk
HP:20
ST:20
DF:16
LK:10
Opponent2:Loki
HP:12
ST:13
DF:12
LK:20
Rolls:
ROUND:1
TOHIT:17
DAMAGE:10
TOHIT:8
ROUND:2
TOHIT:20
DAMAGE:8
```

Output

```
Opponent 1: Hulk
Start HP=20
Damage Bonus=+3
Defense Bonus=+1
Defense: 17 (base 16+1)
```

```
Opponent 2: Loki
Start HP=12
Damage Bonus=+1
Defense Bonus=+3
Defense: 15 (base 12+3)

Loki will strike first.

Round 1:
Loki rolls 17 vs. 17 =HIT!
Loki damage roll=10+1(bonus)=11
Hulk HP Now=9
Hulk rolls 8 vs. 15 =miss
Round 2:
Loki rolls 20 vs. 17 =HIT!
Loki damage roll=8+1(bonus)=9
Hulk HP Now=0

Loki HAS DEFEATED Hulk!!!
```

[1] All registered trademarks used without permission as fair-use examples.

Problem.26 Balanced I/O

+ 11 points

Corporate Announcement: We have been monitoring our site traffic, and there is much more data streaming in than going out. We're now instituting a check to make sure we're more balanced. We will randomly check everyone's system, and we expect to find a good balance of both input and output.

To meet the corporate demands, you have decided to make sure your own I/O meets the requirements by filling in the right amount of In and Out bits in your data.

You will analyze a grid of I/O data and make sure it meets these requirements:

- The grid is a square, with an even number of rows and columns, from 2 to 10.
- Input bits are marked with an upper-case "I" and output bits are marked with a lower-case "o".
- There must equal counts of "I" and "o" in each row and column.
- In each row and column, no more than two identical characters may be next to each other.
- No two rows (or columns) may have exactly the same sequence of N characters left to right (or top to bottom).

Input

The first line of input is an even integer N (2, 4, 6, 8, or 10), the size of the square grid. The next N lines each contain N characters, marking known "I", or "o". Unknown locations are marked with a period ".".

Example 1

```
10
.I.I.....
Io.O.o..o.
.....o
..II...I.
.....I
oI.II..o.I
.....o.
....II...
..I.....
...I..I.Io
```

Example 2

```
2
.I
..
```

Output

Print the same grid, with all the periods replaced with "I" or "o" to meet all the requirements. There is only one answer.

Example 1

```
oIoIoIoIIo
IoIoIoIoIoI
IIooIoIIoo
ooIIIoIoIIo
IoIooIoIoII
oIoIIoIoIoI
IIoIIooIoIo
ooIooIIoII
ooIIoIoIoIoI
IIooIoIoIoIo
```

Example 2

```
oI
Io
```

Problem.27 AI Assistant

+ 15 points

You have been put on the digital assistant "AI" team for a new phone company. Your director wants to do a voice test so they can start interviewing voice actors, so your team needs to create a simple "AI" script that will respond to certain inputs. The name of the "AI" assistant hasn't been decided on yet, so it is subject to change.

For instance, if the user says to the assistant:

I'm cold

The assistant should reply with:

Temperature has been raised, OwnerName

The assistant will need to respond to some variations of some spoken commands. Below is a table with all of the spoken commands your "AI" script will need to respond to, along with any variations, and the expected response and action (if any).

COMMAND	ALT	RESPONSE	ACTION
Hey AI, what's 3 plus 4?	what's 3 added to 4?	# plus 4 is #, [OwnerName]	
Hey AI, what's 3 times 4?	what's 3 multiplied by 4?	# times # is #, [OwnerName]	
Hey AI, what's 3 to the 4th power?	what's 3 raised to the 4th power?	# to the power of # is #, [OwnerName]	
Hey AI, I'm [still] cold	Turn up the heat	Temperature has been raised, [OwnerName]	Increment room temperature value by one
Hey AI, I'm [still] hot	Turn up the AC	Temperature has been lowered, [OwnerName]	Decrement room temperature value by one
Hey AI, what's the current temp?	what's the current temperature?	The current temperature is #, [OwnerName]	
Hey AI, call me Dave from now on		Okay, I'll call you Dave from now on	Change owner name value to new name given in command
Hey AI, what's your name?		My name is [AI], [OwnerName]	
Hey AI, can I call you Computer?		Okay, you can call me Computer from now on	Change "AI" name value to new name given in command
Hey AI, tell me a joke		So this guy, a squirrel, and a dog walk into a bar...	
Hey AI, tell me a better joke		No	
Hey AI, open the pod bay doors		I can't do that, [OwnerName]	

Notes:

- If the command has a word in [brackets], that means the word is optional
- All commands CAN start with "Hey AI-Name," but are not required to
- You will be given a starting room temperature, an owner name, and a name for the "AI" in your script's data file
- You will need to handle the commands using the name for the "AI" given in the data file
- You will need to use the owner name given in the data file (shown as [OwnerName] in the expected response)
- If the command is given to change the owner or "AI's" name, from that point forward you will need to use the altered name(s)
- For the responses for your "AI" script, replace the # symbol with an integer with an appropriate value for the command
- If the command is not understood by the "AI" script, respond with: I don't understand you, [OwnerName]

Example In:

```
START_TEMP 78
NAME_OWNER Chandler
NAME_AI Dave
What's your name?
Hey Dave, Can I call you Computer?
Hey Computer, What's 3 to the 4th power?
Hey Computer, What's the current temp?
Turn up the heat
Hey Computer, I'm still cold
What's the current temp?
```

Example Out:

```
My name is Dave, Chandler
Okay, you can call me Computer from now on
3 to the power of 4 is 81, Chandler
The current temperature is 78, Chandler
Temperature has been raised, Chandler
Temperature has been raised, Chandler
The current temperature is 80, Chandler
```

Problem.28 Game Design 101

+ 20 points

You have been hired as a programmer for a new game development studio working on a new top-down isometric role-playing game. Your first assignment is to work on the pathfinding for the game. When the player clicks on the map, the game knows the starting point (X), and where the user wants to go (Y). The game is setup so certain maps have waypoints (A-J) the characters will need to travel through in order to trigger game events. Additionally, in-game assets (such as houses, rivers, trees, rocks, etc.) cannot be crossed.

The path your code will need to produce should be the shortest path connecting the start (X) and end (Y) points with all of the waypoints (A-J) visited at least once while not crossing any walls, rivers, etc. Your game engine will have a list of all start and end points of uncrossable areas. Your path must start at the start point (X) and end at the end point (Y). The waypoints (A-J) may be visited in any order.

The player will be annoyed with you if they see their characters retracing their path. Therefore, your path should never travel the exact same path twice.

It is okay to take the following path to get around a wall:

```
A->D->C->A->G
That path visits point A twice, but never
retraces the same path.
```

However DO NOT create a path like this:

```
A->D->C->D->G
That path again visits the same point twice,
which is okay, but the path travels from point
D to point C, and then from C to D, which is
the exact same line (just going the other way).
```

The data will be passed into your pathfinding function in the form of a list of coordinates including the start (X) and end (Y) points, and the necessary waypoints for your path to cross through. The data will also include the obstructions on the map the characters are currently in, listed after a line with the tilde (~) character on it alone. Those obstructions will be listed as a pair of coordinates, which show the beginning and end of the obstruction which your path cannot cross.

When you have finished calculating the shortest path between all the points, output your path, and the total distance (rounded to 2 decimals) in the format:

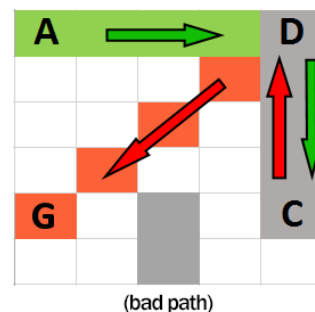
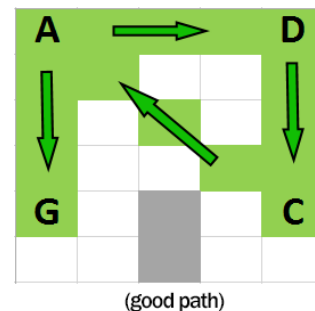
```
Shortest path: XEJBY
Distance=12.39
```

Example Input

```
X 0,0
A 3,1
C 1,2
G 7,7
Y 8,2
~
2,1 2,2
```

Example Output

```
Shortest path: XCGAY
Distance=22.36
```



Problem.29 Wonka Factory Password

+ 22 points

The Oompa Loompas at Willy Wonka's factory have accidentally locked themselves out. Unfortunately, they've forgotten the password to the keypad used to unlock the door! Luckily, they have a password hint that they set up so they can figure out their password but since they take their security very seriously, it takes quite a bit of effort to figure out the password from the hint. Today is the day the children are coming for their tour so they don't have the time to go through the regular process of figuring out the password from the hint. This is where you come in. They've provided you with an explanation of how to get the password from some input data and are asking for you to figure out the password for them in a faster way.

Using their magical powers, the Oompa Loompas set up an infinitely long dance room. When you walk in, you see markings on the floor in a line that extends infinitely in both directions. Each spot is numbered. Directly in front of you is a spot marked 0. To the left, the spots are marked -1, -2, -3, and so on. To the right, they are marked 1, 2, 3, and so on.

The first line in your input contains a starting position telling you whether the Oompa Loompa there is standing (#) or sitting (.), starting at spot 0 and continuing towards the right. For example: ".#.#...." means at time 0 the Oompa Loompas in spots 2 and 5 are standing and the rest are sitting. The rest of your input gives you 5-character patterns (like ".##.#") you will use to pattern match. Take the initial state, then parse through the line looking for the patterns. When you find a match, it tells you the middle Oompa Loompa will be standing in the next round.



Input

The first line is "initial state: ", followed by the description of the line at time 0. The first character is the state of index 0 and the remaining characters count up. The second line holds the password at time 11. The third line holds the password at time 1000. Use these values to verify your program is running correctly. The remaining lines show the patterns that result in standing on the next round. The last line of input is a 0. Any patterns not listed result in sitting on the next round.

```
initial state: #..#.#.##.....###...###
136
19374
...##
..#..
.#...
.#.#.
.#.##
.##..
.####
#.#.#
#.###
##.#.
##.##
###..
###.#
####.
0
```

Instructions on how to process the data

Pad the current state with periods until you have# at the beginning of the line and #.... at the end of the line. Keep track of the index number for the first # as it moves around.

```
#..#.#.##.....###...###
becomes
....#..#.#.##.....###...###....
```

Go through each pattern and search for it in the line (patterns may not exist in the line; they may also be found multiple times; they can also overlap each other). For each matched pattern, take the middle position of the pattern and store a # in that index for the next state. Below, underscores "_" denote indexes that have not been determined yet.

```
...## : ....#..#.#.##.....###...###....
new result object:      _#_#_

..#.. : ....#..#.#.##.....###...###....
new result object:      _#_#_#_

.#... : not found

.#.#. : ....#..#.#.##.....###...###....
new result object:      _#_#_#_#_

and so on...
```


_____#____#____#____#____#____#____#____
becomes:
...#...#...#...#...#...#...#...#...

Replace your state with this result. This is the new state for time=1.

Do this exactly 50 billion times. When you reach the final state (t=50 billion), take the value of each standing (#) index, and add them up. This will be your final password.

Note: At some point during the iterations, you will find that the pattern repeats but is shifted from a previous state. Once you find this, you can predict mathematically what the final state will be. Going through the process above 50 billion times without using this shortcut will take a very long time. Your program will be marked as a PASS if it takes less than 1 minute.

The states of the solution (for reference)

This shows the state after 20 rounds. The numbers at the top are the indexes.

```

          0          1          2          3
        0          0          0          0
0:  ...#.##.###.....###.###.....
1:  ...#.##.###.....#.##.###.....
2:  ...##.##.###.....#.##.##.###.....
3:  ...#.##.##.##.....#.##.##.###.....
4:  ...##.##.##.##.....##.##.##.###.....
5:  ...##.##.##.##.....##.##.##.###.....
[time passes...]
11: ...#.##.##.##.....#.##.##.##.....
[time passes...]
19: ..#####.##.#####.##.##.##.##.....
20: ...##.#####.#####.....#.##.###.....

```

Output

If we were only going 20 rounds, adding up all of the standing indexes (#) in the 20th round above, the solution would be 325. For your solution, print on separate lines: the password at time=12, time=1001, and at time=50 billion.

218
19394
999999999374