Important Contest Instructions !!

Please read the following instructions carefully. They contain important information on how to run your programs and how to prepare them for submission. If you have any questions regarding these instructions, please ask a volunteer or in our Discord server.

NEW for 2022!

Removing support for JavaScript and Python 2: Programs in JavaScript or Python 2.x will no longer be accepted.

The supported languages are: Java, C, C++, and Python 3.x

Language Versions

The judges will be using the following versions for each of our supported languages.

Language	Version
Java	17.0.2
Python	3.10.0
C	11.2
C++	11.2

Java

Your program file name must be **probXX.java** and your class name must be **probXX**, where **prob** is all lower case and **XX** corresponds to the two digit problem number. (ex. `public class prob01`, `prob01.java`).

Remember to copy the entire Java source file contents into the submission box. Java programs can rely on the standard Java library.

Python

Python programs can rely on the standard python library.

C / C++

C or C++ programs must be submitted as source files and can rely on glibc/libm and the standard headers.

> Download our "C/C++ Development Environment Guide" from https://hpecodewars.org/downloads

Continued on the next page...

Program Input (all languages)

Most programs will require input. You have two options:

File Input

Your program may read input from a file named **input.txt** for all problems. The file will be in the same directory as your program. Use `"input.txt"` as the filename and do not prefix it with a path.

> Download our "Guide to data file I/O" from https://hpecodewars.org/downloads

Keyboard Input

Your program may read input from standard in (the keyboard). Do **not** include any prompts in your output. There are two options to provide input to your program via standard in (the keyboard).

- (1) The preferred option is to redirect the contents of a file to standard in of your program.

```
java prob01 < input.txt
java -jar js.jar prob01.js < input.txt
python prob01.py < input.txt
prob01.exe < input.txt
```

- (2) Otherwise you may type everything manually, or copy/paste from the packet or data set files.

> Tip: Type `Ctrl-Z <return>` to signal the end of keyboard input.

Program output (all languages)

Your program must send the output to the screen (standard out, the default for any print statement). Do **not** include any prompts for input in the output! The only output for your program should be the expected output for the given problem.

Testing your program

Test your program like a judge will with the Windows Batch script (`checkProb.bat`) in the student ZIP file. It tests your program with all student data sets for a given problem. It will redirect the data set file contents to standard in and create the **input.txt** file.

Back by popular demand

You've been invited up on stage to sing along with Beyoncé as she sings her hit single, *Formation.* There is a teleprompter at the foot of the stage with the lyrics showing you the next line.

Input

There is no data file to read in for this (**and only this**) problem.

Output

Print to the screen the following line from the song.

```
Okay, ladies, now let's get in formation, prove to me you got some coordination.
```

problem
01

Players, are you ready?

1
points

Please welcome the players, by name, as they take the court for the last game of the NBA finals.

Input

You will receive one name on a single line. It will not contain spaces or punctuation (even if the proper spelling of the name would include them).

```
TroyBrown
```

Output

Print to the screen the following sentence, as a prompt for the announcer to read as the player takes the court. **The only part of the sentence which should change is the name!**

```
Clap your hands together and give it up for TroyBrown!
```

Discussion

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1
`JavonteGreen`	`Clap your hands together and give it up for JavonteGreen!`

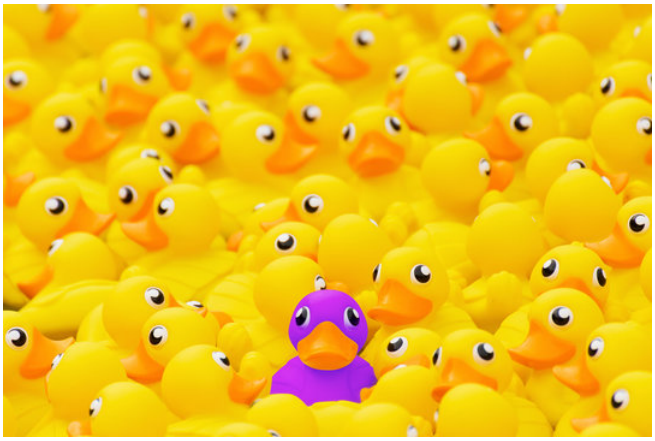
Input 2	Output 2
`GarrettTemple`	`Clap your hands together and give it up for GarrettTemple!`

problem
02

Odd Number Out

2
points

The criminal mastermind, *The Duck*, is at it again. In his ongoing efforts to stop what he (and Daffy) consider insufficient *respect* to him and his fine-feathered friends, he has sabotaged the machines at the world's largest rubber duckie factory to corrupt the factory computer's counts. (Ernie will be sad.)



Input

You will receive a list of integers (no integer will be larger than what an int-32 can hold), one per line. The minimum count of integers you might receive is four (4). When you see zero (0) on a line by itself, input has terminated. (Do not include the zero in the data set).

```
1
3
5
23
8
7
15
0
```

Output

Figure out if there is an odd number in an otherwise list of all even numbers, or vice versa. If a problem is found, print the number found along with what the problem is (odd in an even list, or even in an odd list.) If the list contains only odd or event numbers, output: NO LIST PROBLEMS FOUND.

```
8 is not odd
```

Discussion

You will not receive a list with more than one odd number in an otherwise even list, or vice versa. E.G. you do not have to code handlers for situations where you find 3 odd numbers in an otherwise even list, or 2 evens in an otherwise odd list, etc.

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1	Input 2	Output 2
2 4 6 22 18 90 102 85 0	85 is not even	3 9 17 101 33 1005 0	NO LIST PROBLEMS FOUND

Hot Shot

Basketball is full of advanced shooting metrics and one of these metrics is `true shooting percentage`.

Though it is not a perfect statistical measurement, the main idea of TS% is that it is a measure of scoring efficiency based on the number of points scored over the number of possessions in which a player attempted to score. In simple terms, true shooting percentage can be thought of as points per shot.

True shooting percentage is calculated by using the formula:

```
TS% = 100 * PTS/(2(FGA + (0.44 * FTA)))
```

Where:

- PTS = points scored
- FGA = field goal attempts (shots taken)
- FTA = free throw attempts

Write a program to calculate the true shooting percentage of a player given the PTS scored, FGA and FTA.

Input

The input consists of three whole numbers in one line separated by spaces: the total PTS, total FGA, and total FTA.

```
98 71 43
```

Output

Use the PTS, FGA, and FTA to calculate the TS%. Print the true shooting percentage, including two decimal places rounded, and append a '%' sign. [1]

```
54.49%
```

Discussion

Round to nearest hundredth. The denominator may contain a 0 but will not result in 0. The result may be more than 100%.

[1] Rounding should follow the rule of: round **up** if the last number is **5-9**, leave the hundreths place (the 2nd digit) **unchanged** if the last number is **0-4**.

E.G.

- 0.465 -> 0.47
- 0.464 -> 0.46

Additional Examples

Input 1	Output 1	Input 2	Output 2	Input 3	Output 3
60 60 0	50.00%	10000 0 10000	113.64%	8765 4321 0	101.42%

problem 04

Wait ... How Many Fingers Do They Have?

3 points

You are part of a first-contact team with the Systems Alliance working with the newly encountered Quarian species, aliens from another planet. Your team is trying to help develop translations for the Quarian language but every time calculation made with the numbers they give your team don't match the expected results.

Your chief scientist suggests that because the Quarians only have 3 fingers on each hand, the numbers they are giving the team may be in **ternary** instead of **decimal** -- after running some back-of-the-envelope calculations you confirm that when the numbers given by the Quarians are converted into base-10 (decimal) assuming base-3 (ternary) as the source -- the numbers match expectations!

Now your team needs an automated process to convert ternary numbers into decimal numbers so that work can continue on the translation device.

Input

You will receive a list of **ternary** numbers, one number per line. The last line will be signaled with a zero on a line alone. You will not be given zero (as a ternary number) to be translated into decimal. The largest ternary number you will be expected to handle will be 1212210202001. Example:

```
10
112
1000
1110
1212210202001
0
```

Output

Output the value of the ternary numbers, one per line (ignoring the 0 line).

```
3
14
27
39
1000000
```

Discussion

Reminder: have you run your solution against **all** of the student data sets? Hint: some of the values you may get as input may be larger than what an int-32 variable can hold.

Additional Examples

Input 1	Output 1	Input 2	Output 2
21 10021 101 121 22020 20202 0	7 88 10 16 222 182	222222 111111 100001 2220121 20021100110 0	728 364 244 2122 123456

problem
05

Odd Roundings

3
points

You've probably heard the number propaganda that when rounding a number one should round **up** if the number is 5-9, and round **down** if the number is 1-4. That's just what *they* want you to believe. In here, we round using the Queen's rounding rules! Watch the corners.

Input

You will receive, on one line, a single decimal number containing exactly 2 decimal values. Round the number using the rules given below.

```
1234.17
```

Output

Check the rules' conditions, in order from condition 1 to 4, stop after you have applied one rule.

Condition 1	Rule	Condition 2	Rule	Condition 3	Rule	Condition 4	Rule	Else
last digit is 7	add 0.02	last digit is odd	subtract 0.09	last digit > 7	subtract 4.00	last digit < 4	add 6.78	do nothing

```
1234.19
```

Discussion

Your output must include 2 (and only 2) decimal places, if your code produced 1234.5 as the answer, you need to supply the trailing zero to make it 1234.50.

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1	Input 2	Output 2	Input 3	Output 3	Input 4	Output 4
999.99	999.90	881.38	877.38	654.33	661.11	5.56	5.56

Saving Money

Maria just received some money and has decided she would like some more. She's going to save it in the bank and get interest back. Using the Simple Interest Formula, figure out how much interest she'll get paid.

The Simple Interest Formula is:

```
I = Prt
```

...where P = the Principal (the initial amount of money),

r = the rate of return per year,

t = the number of years.

For example, if P = $10000, r = .04 (or 4% per year), and t = 5 years, the interest I = ($10000)(.04)(5) = $2000, so she gets $2000 in interest in 5 years!

Given P, r, and t, find I.

Input

The input has 3 lines. The first is an integer showing the starting Principal amount. The second is the rate of return per year, as a decimal fraction beginning with "0.". The third is an integer, showing the number of years.

```
10000
0.04
5
```

Output

Print the Interest as a decimal fraction with 2 decimal places. Do not round up! Instead truncate any digits past 2 decimal places.

```
2000.00
```

Discussion

For Output 1, Prt = 4305.31875. We truncate to 4305.31. For Output 2, Prt = 13362.966. We truncate to 13362.96.

Additional Examples

```
Input 1   Output 1   Input 2    Output 2

12345     4305.31    54321      13362.96
0.03875              0.0123
9                    20
```

problem
07

Punt, Pass, Kick, and Compute

4
points

Every contestant is allowed one punt, pass, and kick in this thrilling competition. Scoring is based on both distance and accuracy. The score is also determined from where a contestant's ball first makes contact with the ground, therefore bounces or rolls are not part of the distance. If a contestant passes their ball 121 feet, but it went wide of the measuring tape by 30 feet, their final score for passing is 91 feet; the score is the difference between the two measures.

The distances of the balls are measured and rounded to the nearest foot. A person can not score less than 0. There are 5 contestants and no ball can go farther than 300 feet or be off the mark by more than 300 feet. A participant's final score is the total of the three events (Punt, Pass, and Kick). For example, a participant scores 30 feet for punting, 50 feet for passing and 22 feet for kicking, the participant's final score is 102 feet

Write a program that displays the results of the competition given the distance and accuracy of each category and also the contestant's name.

All input and output will be in whole feet. The range of numbers for distance and accuracy is 0 to 300.

Input

The input consists of 3 pairs of numbers and a name. For the pairs of numbers, one pair is for punting, another for passing, and last for kicking. The first number in each pair is the distance and the second number is how far off the mark the ball was.

Use the contestants' names and scores to compute a list of results.

```
40 20 61 17 52 13 Justin
30 40 60 50 100 10 Aaron
70 10 53 13 42 19 Lamar
60 20 53 10 70 20 Patrick
40 5 60 0 9 13 Tom
```

Output

Print an ordered list of results with the name in the first column and total distance in the second column.

```
Patrick 133
Lamar 123
Justin 103
Aaron 100
Tom 95
```

Additional Examples

Input	Output
101 240 110 258 134 295 Tua	Kyler 354
164 86 216 229 154 209 Josh	Russell 300
208 40 167 62 300 19 Kyler	Dak 103
60 20 53 10 40 20 Dak	Josh 78
128 300 0 300 300 0 Russell	Tua 0

problem
08

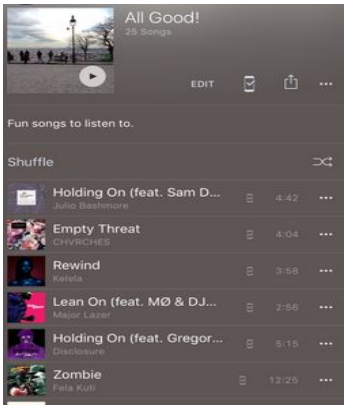
Playlist Time

4
points

Suresh creates new playlists as often as he sneezes (he's quite allergic.)

He keeps forgetting how to find the total length of each list.

Can you calculate it?



Input

The first line of input is an integer N, the number of songs in the list.

The next N lines contain the length of each song in MM:SS format (or M:SS if M is less than 10).

The total time will be less than 1 hour.

```
5
3:05
2:06
9:59
1:01
1:15
```

Output

Print the total time of the list in MM:SS format. If less than 10 minutes, print M:SS format.

```
17:26
```

Additional Examples

```
Input 1   Output 1   Input 2    Output 2
4         9:06       3          59:59
1:08                 4:04
2:59                 22:22
3:56                 33:33
1:03
```

Mustache Matching

Oh no, the mustache emojis are out of control! Quick, help the internet find the rogues before their shenanigans shut down social media as we know it!

Input

You will receive, on a single line, a series of 3-10 mustache emojis separated by spaces. Mustache emojis will always follow the same pattern of **EumuE** where **E** are the **eyes**, and **m** is the **mouth**, always a single character for each.

The eyes may use the following characters: ^, o, 0, O (caret, lowercase-O, zero, uppercase-O). The mouth will always be a single underscore (_) character. **u** == the mustache.It may not be symmetrical and may have from 1 to 5 characters on either side of the mouth character. The mustache may use the following characters: ~, =, - (tilde, equals, hyphen). Mustache characters will not be mixed in the same emoji.

```
^~_~^ ^~_~^ ^~_~~^ ^~_~^
```

Output

Find the emoji who has gone renegade with their mustache grooming, and call them out by number! A *rogue* mustache emoji will either have too many, or too few mustache parts, or may be symmetrical when the rest are asymmetrical (or vice versa). The first emoji on the line is #1.

Output as follows: #NUM EMOJI you are out of control!

Where NUM is the number of the emoji in the list, starting at 1 from the left, and EMOJI is the emoji you found which doesn't match the others.

```
#3 ^~_~~^ you are out of control!
```

Discussion

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1
`0=_=0 0==_==0 0=_=0`	`#2 0==_==0 you are out of control!`

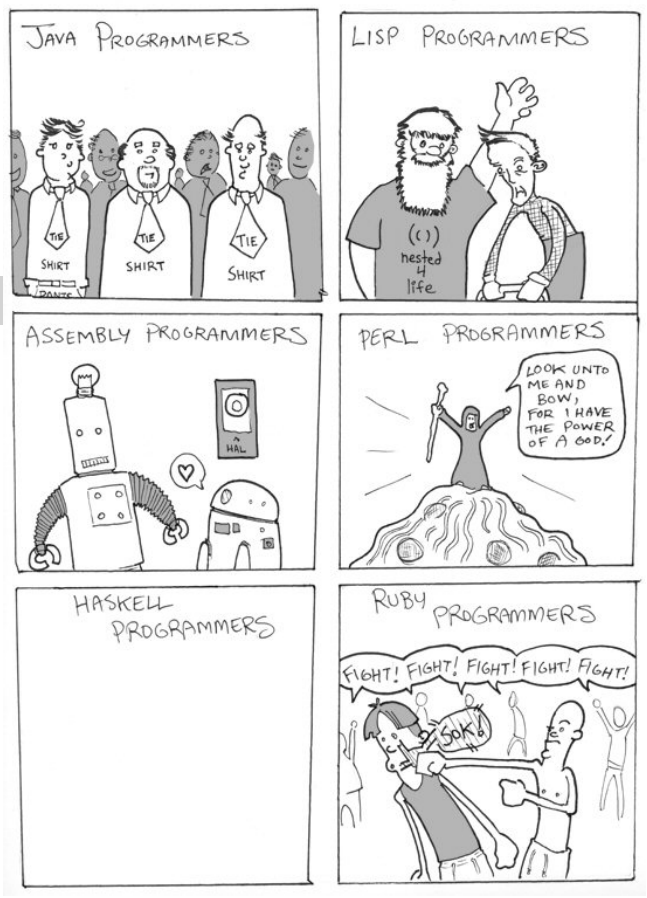
Input 2	Output 2
`o--_--O o--_--O o--_--O o-_-O`	`#4 o-_-O you are out of control!`

Input 3
`0~_~~0 0~_~~0 0~_~~0 0~_~~0 0~_~~0 0~_~~0 0~~_~~0 0~_~~0 0~_~~0 0~_~~0`

Output 3
`#7 0~~_~~0 you are out of control!`

Nested For Lyfe!

Check whether or not a given set of parenthesis match up evenly. E.G. for every Left-open parenthesis "(" there must be a Right-close ")" parenthesis.



Input

You will receive a single line (which may wrap on your screen) of sets of parenthesis.

```
(()))()(()))
```

Output

Determine if the nested parenthesis are balanced or not. That means for every opening/left parenthesis "(", there should be an equal and opposite closing/right parenthesis ")".

Print the total number of left and right parenthesis on their own lines, the total number of paired parenthesis on its own line, and finally a determination of either **Unbalanced** or **Balanced**.

The sets of parenthesis should only be labeled balanced if the left and right parenthesis match up evenly (even if the counts are equal). E.G. if you couldn't put code inside the parenthesis and get your compiler to run it, then they are **Unbalanced**

```
Total left: 6
Total right: 6
Total pairs: 5
Unbalanced
```

Discussion

In the example above, the counts match, but both the left and the right side of the parenthesis groups have an extra parenthesis which doesn't match with a partner. They are at character positions 4 and 7 (zero indexed). You don't need to find the position of any trouble parenthesis, we are just calling it out so you understand the nature of the problem. Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1	Input 2	Output 2
))))((((	Total left: 5 Total right: 5 Total pairs: 1 Unbalanced	((((((((((()(((((((((()))(((())))))))))))))	Total left: 22 Total right: 22 Total pairs: 22 Balanced

Squeak Latin

You've heard of Pig Latin, but did you know chipmunks have their own version of that? They call it Squeak Latin!

Input

You will receive between 3-9 lines of text. Each line will be no longer than 200 characters. Input stops when you encounter the word END on a line by itself.

```
How did you guys?
We're talking chipmunks, Dave.
We can get out of a cat carrier.
Not even hard to do.
END
```

Output

Change all words beginning with a consonant by moving the first letter to the end of the word, then add "-squeak" after that. If a word begins with a consonant **and** is only two letters long, leave it unchanged. Else if a word begins with a vowel (aA,eE,iI,oO,uU), simply add "-m" to the word. Do not worry about correcting the CaSe of moved letters (we're talking about chipunks here, they aren't picky). Preserve all punctuation (punctuation can include: commas (,) exclamation marks (!) periods (.) and question marks (?)).

```
owH-squeak idd-squeak ouy-squeak uysg-squeak?
e'reW-squeak alkingt-squeak hipmunksc-squeak, aveD-squeak.
We anc-squeak etg-squeak out-m of-m a-m atc-squeak arrierc-squeak.
otN-squeak even-m ardh-squeak to do.
```

Additional Examples

Input 1	Output 1
That armors too strong for blasters. Rogue Group, use your harpoons and tow cables. Go for the legs. It might be our only chance of stopping them. END	hatT-squeak armors-m oot-squeak trongs-squeak orf-squeak lastersb-squeak. ogueR-squeak roupG-squeak, use-m oury-squeak arpoonsh-squeak and-m owt-squeak ablesc-squeak. Go orf-squeak het-squeak egsl-squeak. It-m ightm-squeak be our-m only-m hancec-squeak of-m toppings-squeak hemt-squeak.

Input 2	Output 2
Before the network, there was the fleet. Before diplomacy, there were soldiers. Our influence stopped the rachni, but before that, we held the line. Our influence stopped the krogan, but before that, we held the line! Our influence will stop Saren! in the battle today, we will hold the line! END	eforeB-squeak het-squeak etworkn-squeak, heret-squeak asw-squeak het-squeak leetf-squeak. eforeB-squeak iplomacyd-squeak, heret-squeak erew-squeak oldierss-squeak. Our-m influence-m toppeds-squeak het-squeak achnir-squeak, utb-squeak eforeb-squeak hatt-squeak, we eldh-squeak het-squeak inel-squeak. Our-m influence-m toppeds-squeak het-squeak rogank-squeak, utb-squeak eforeb-squeak hatt-squeak, we eldh-squeak het-squeak inel-squeak! Our-m influence-m illw-squeak tops-squeak arenS-squeak! in-m het-squeak attleb-squeak odayt-squeak, we illw-squeak oldh-squeak het-squeak inel-squeak!

No Leftovers

Bob the Builder is trying very hard to manage his resources.

He wants to make sure that he has absolutely no extra materials lying around (like Bob-Nails, Bob-Boards and Bob-Posts, ...) unused when he finishes building his next set of Bob-Bridges. He also has a budget to keep track of, so he must use the least possible resources.

Bob is headed to the hardware store, so could you find out the exact amount of material he needs? You will need to make sure that he has zero leftover materials. He may need to build several Bob-Bridges to exhaust all of his resources.

Input

You will receive three lines of input.

The first line contains the number (from 1 to 10) of different types of materials (like Bob-Hammers, Bob-Windows, Bob-Tiles, etc.) needed for one Bob-Bridge. The second line contains the amount of each of those material needed to create one Bob-Bridge. The third line lists the count of those same materials Bob already has (in the same order as the second line).

All material counts will be more than 0 and less than 100000.

```
3
5 2 1
11 10 1
```

Output

Your output should display on a single line the amount of each resource Bob needs to purchase. The materials must be in the same order as they were in the input.

```
14 0 4
```

Discussion

Input explanation:

```
3       <- There will be 3 different materials needed for this type of Bob-Bridge.
5 2 1   <- To create one Bob-Bridge, you need 5 of the first material, 2 of the second,
and 1 of the third.
11 10 1 <- Bob already has 11 of the first material, 10 of the second, and 1 of the third.
```

A good place to start in solving this problem, is to find the fewest Bob-Bridges to make, just to use up all the materials Bob already has.

In this example, that would be 5 Bob-Bridges, because Bob already has 10 of the second material and he needs 2 of the second material to finish 1 Bob-Bridge.

To build 5 complete Bob-Bridges, Bob will still need 14 of the first material, and 4 of the third.

Things to remember

You will need to tell Bob to buy enough of all materials so that there is 0 left of all the materials after building the fewest number of Bob-Bridges.

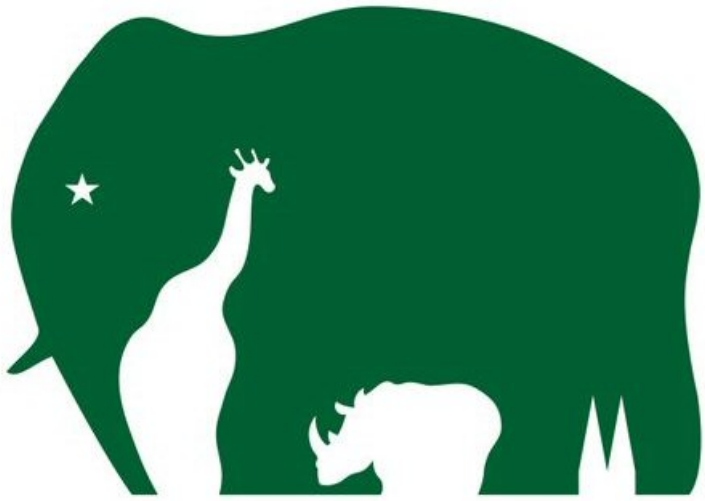
problem
13

Negative Space

7
points

You work for a graphic design company and they want you to prototype new designs they can print on t-shirts, mugs, etc.

Your prototypes need to use negative space. That means the design will be *printed* from the space that *isn't there* -- essentially you are creating a print-negative to be used like a stamp.



Input

You will receive a datafile with the following elements:

- On line one you will receive two positive, non-zero, integers separated by spaces which will give you the **width** and **height** of the design grid.
- On the following lines you will receive coordinate pairs in the format of Y: X X X_n where X and Y are all integers (they may be zero). If the *X-adjacent* number of X coordinates will be greater than 10, the range will be given as X_1-X_n [see discussion note 1]. Rows which are all solid, will be omitted from your input.
- Input is terminated when you read a line with END on it

Example:

```
50 11
2: 13 14 32 33
3: 13 14 32 33
5: 6 7 8 40 41 42
6: 7 41
7: 8 39
8: 9 37
9: 12-35
END
```

Output

Assemble an output grid using the *solid* # character in the width and height given in your input on line one. Output the *negative* **space** character at coordinates given in the file. You will have to assemble each (X,Y) coordinate from the given input. For example **2: 13 14 32 33** maps to coordinates: (13,2) (14,2) (32,2) (33,2). Coordinates for your output grid should **start with (0,0) in the upper left corner** [see discussion note 2]. Print the resulting grid as your output.

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

Discussion

[1] If there are many blank spaces in a row, we will signify that with a dash. For instance, 12-35 means that spots 12, 13, 14,...34, and 35 are all supposed to be blank.

[2] The origin (0,0) of the drawing will be the left upper corner. The X axis will go from zero on the far left and become more and more positive heading to the right. The Y axis will go from zero on the top and become more and more positive heading down

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1

```
16 7
1: 1 2 5 6 9 10 11 12 13 14
2: 1 2 5 6 11 12
3: 1 2 3 4 5 6 11 12
4: 1 2 5 6 11 12
5: 1 2 5 6 9 10 11 12 13 14
END
```

Output 1

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

Input 2

```
43 11
0: 21
1: 19 23
2: 17 25
3: 15 27
4: 13 29
5: 11-31
6: 9 13-29 33
7: 7 15-27 35
8: 5 17 18 19 20 21 22 23 24 25 37
9: 3 19 20 21 22 23 39
10: 1 21 41
END
```

Output 2

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

problem
14

Text Translator

7
points

Help the millions of people out there who need help with their social game with your new Text Translator app. Now, those who have trouble picking up on social cues can install your app and translate their polite and/or passive aggressive messages from friends, family and significant others into exactly what they mean to say.

Input

You will receive a selection of messages between two people. The first line will tell you which person has installed your app. The second line will say YES or NO. YES means the other person is the boy/girlfriend of the person who has installed your app. Input ends when you encounter END on a line by itself.

```
Daisuke
YES
Sarah: Hey..
Daisuke: ?
Sarah: we meeting up tonight?
Sarah: ...
Daisuke: um, yeah but can we end early tho? I wanted to go out with the guys later
Sarah: oh.
Sarah: sure- i guess. have fun.
END
```

Output

Use your app's translation matrix (see below, there is also a translations.txt file in your student data which **will not** be in the judge data, but you can use it to quickly copy/paste the data into your program) to change the text from the person who does not have the app installed. Pay careful attention to the column **Only Boy/Girlfriend (Only B/Gf)** -- if that is true, then those translations will apply to the other person. Else do not use them. Note, **CaSe MATTERS!** Translations only apply when the text matches the case of the translation exactly **that includes punctuation!**

The text "**have fun**" is *not* the same as "**have fun.**"

```
Sarah: Pay attention to me. Right now.
Daisuke: ?
Sarah: I want to go out.
Sarah: I am annoyed you are not responding faster.
Daisuke: um, yeah but can we end early tho? I wanted to go out with the guys later
Sarah: I am not happy about your last message.
Sarah: I am angry now. Do not do that if you want to stay in a relationship with me.
```

Discussion

Reminder: have you run your solution against **all** of the student data sets?

Translation Matrix

Phrase	Translation	Only B/Gf	Phrase	Translation	Only B/Gf
Hey..	Pay attention to me. Right now.	No	...	I am annoyed you are not responding faster.	No
we meeting up tonight?	I want to go out.	Yes	....	I am too busy to respond right now, but I am dating you, so here's some ellipses.	Yes
we meeting up tonight?	I have nothing better to do, you free?	No	have fun.	Do not do that if you want to stay in a relationship with me.	Yes
oh.	I am not happy about your last message.	Yes	have fun.	I am annoyed you didn't invite me.	No
oh.	okay	No	Its fine	It's not fine	No
yup.	yes and I am not happy with you	No	k.	That's not okay.	No
K.	Very well, but I am annoyed with you.	No	yup	Okay, I agree, but I'm not happy	No
good for you	I am partially jealous, and partially annoyed by your bragging.	No	<3u	I love you	Yes
I always love this story	Stop telling me variations of this same story, or I'm shaving your dog and painting him green.	No	<3	I approve	No
ha.	That was not funny.	No	so,,,	I am about to start an incredibly long/complex story, which you have to listen to, because you love me.	Yes
ha.	You get a pity laugh, because I <3 u	Yes	so,,,	I broke/lost/forgot something important to you, and I am about to lie about it.	No
can we call instead?	I am angry enough that I want to yell at you.	Yes	Can I let you know?	I have no idea, need to research an answer first.	No
can we call instead?	It is going to take me too long to type this out.	No	lol jk	I am a jerk who tries to pretend my hurtful jokes aren't hurtful.	No
You okay?	Worried about you.	Yes	sure- i guess.	I am angry now.	No
You okay???	You're either late, not responding, or just generally being a jerk Knock it off.	No	If you really want to	I am trying to come up with a reason not to, but I've got nothing	No

Additional Example

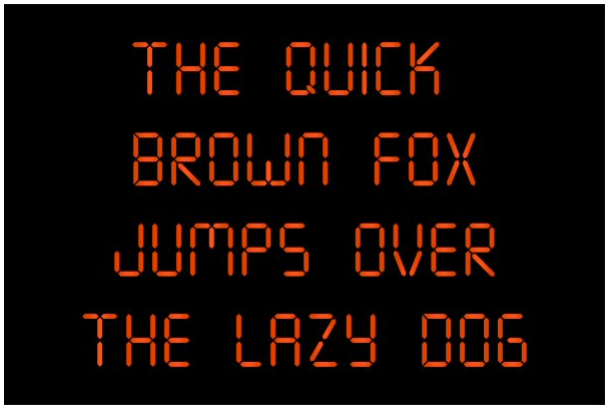
Input	Output
Miguel NO Bhavini: so,,, Miguel: yeah? Bhavini: there was this squirrel right? Miguel: um, wut? Bhavini: I only set your keys down for a minute Miguel: I don't like where this is heading Bhavini: like,,, i'm PRETTY sure squirrels cant drive,,, Miguel: stop stop stop WHERE IS MY CAR?! Bhavini: Can I let you know? Miguel: yup	Bhavini: I broke/lost/forgot something important to you, and I am about to lie about it. Miguel: yeah? Bhavini: there was this squirrel right? Miguel: um, wut? Bhavini: I only set your keys down for a minute Miguel: I don't like where this is heading Bhavini: like,,, i'm PRETTY sure squirrels cant drive,,, Miguel: stop stop stop WHERE IS MY CAR?! Bhavini: I have no idea, need to research an answer first. Miguel: yup

problem 15

CLOCK LETTERS

8 points

Your team has been tasked with coming up with a process to show small messages on a digital display which can be built into a smart watch or similar display. Management has given you a table of characters they want your team to implement in the display controller, and a short list of test messages they want to see proof-of-concept work for by the end of the day.



Input

You will receive a single sentence on a single line. It may or may not include one or more exclamation marks or periods as punctuation.

```
HI ZHU.
```

Output

Spaces in the input should be ignored. Your output must include a single space column between each "digital letter." Map the non-space characters (A-Z.!) to output as "digital letters" using the letter conversions given in the file named **digitalLetters.txt** included in your datasets. (It is too large to print here. You **must** copy that data into your program, as the file will **not** be in the judge datasets!) All "digital letter characters" are 5 units tall, and at least 4 units wide. Several characters (M,N,Q,R,V,W,X,Y,Z) are 5 units wide. Print the transformed sentence as the "digital characters." Punctuation characters are a single column wide. Each character must be either 4 or 5 units wide, including all spaces. Do not leave out any trailing spaces, especially for the last character in the sentence!

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

Discussion

Hint: This problem will be easier if you think in multiple dimensions ;) Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

```
Input 1
THE QUICK BROWN FOX

Output 1
#### #  # #### ####  #  # #### #### #  # #### ####  #### #   # #   # #### #### #   #
 #   #  # #    #  #  #  #  #   #    # #  #  # #  #  #  # #   # ##  # #    #  #  # #
 #   #### ###  #  #  #  #  #   #    ##   #### ####  #  # # # # # # # #### #  #   #
 #   #  # #    # ##  #  #  #   #    # #  #  # # #   #  # ## ## #  ## #    #  #  # #
 #   #  # #### ### # #### #### #### #  # #### #   # #### #   # #   # #    #### #   #

Input 2
JUMPS OVER LAZY DOG!

Output 2
#### #  # #   # #### #### #### #   # #### ####  #    #### ##### #   # ###  #### #### #
  #  #  # ## ## #  # #    #  # #   # #    #  #  #    #  #    #   # # #  # #  # #    #
  #  #  # # # # #### #### #  #  # #  ###  ####  #    ####   #     #   #  # #  # # ## #
# #  #  # #   # #      # #  #  # #  #    # #   #    #  #  #     #   #  # #  # #  #
###  #### #   # #    #### ####   #   #### #   # #### #  # #####   #   ###  #### #### #
```

problem
16

Secret Message

8
points

Gru is out on vacation and needed a way to communicate with his minions. They decided that at the end of each message Gru sends to the minions, they will sign it with a special cypher to verify it was sent by Gru. The only problem with this, is that the minions can't figure out how to create the cypher text hash. Help the minions by figuring out if Gru really sent the message.

Input

The first line will contain a message, the second line will contain the signature cypher text.

```
Don't forget to give Kyle water.
0348ad5d
```

Output

Print YOUR_SIG equals MESSAGE_SIG if the signatures match (where YOUR_SIG the cypher text you just calculated, and MESSAGE_SIG is the cypher text given with the message input). Then print Gru on a new line

Else, print YOUR_SIG does not equal MESSAGE_SIG if the signatures do not match. Then print Not Gru on a new line

To create the cypher text follow the following steps (see discussion section 1 for the algorithm, and section 2 an alternative)

```
reuamqol does not equal 0348ad5d
Not Gru
```

Discussion

Reminder: have you run your solution against **all** of the student data sets?

[1] Algorithm:

1. Look at the first 4 characters in the example input given above, **Don'**
2. **VERY IMPORTANT!** If the final ordinal value you get in the calculations show below is **less than 97** add the ordinal value on the right of the plus sign to the value you got from the original addition **again**! Keep doing that until the value you get is >= 97. E.G. if in step 1 below, ordinal(D) + ordinal(.) == 96 (it doesn't, this is only for illustration) then add ordinal(.) (which would be 46) again. And keep doing that until the value is >= 97.
3. Take their ASCII ordinal values and add them up as follows (where N is the last character in the message, e.g. N == index=length(message)-1):
    1. message[0] + message[N] == ordinal(D) + ordinal(.) == 68 + 46 == final ordinal value: **114**
    2. message[1] + message[0] == ord(o) + ord(D) == 111 + 68 == **179**
    3. message[2] + message[1] == ord(n) + ord(o) == 110 + 111 == **221**
    4. message[3] + message[2] == ord(') + ord(n) == 39 + 110 == **149**
4. Look at the last 5 characters in the message, in this case **ater.**
5. Take their ASCII ordinal values and add them up as follows (where N is the last character in the message, e.g. N == index=length(message)-1):
    1. message[N-3] + message[N-4] == ordinal(t) + ordinal(a) == 116 + 97 == **213**
    2. message[N-2] + message[N-3] == ord(e) + ord(t) == 101 + 116 == **217**
    3. message[N-1] + message[N-2] == ord(r) + ord(e) == 114 + 101 == **215**
    4. message[N] + message[N-1] == ord(.) + ord(r) == 46 + 114 == **160**
6. That gives us a starting point of 114 179 221 149 213 217 215 160
7. If the ordinal calculated value is <= 122 simply print the character value. E.G. 114 is character **r** in the ASCII table
8. If the ordinal calculated value is > 122, then subtract 26 until it is <= 122. E.G. 179 is > 122. 179 - 26 == 153. 153 is > 122. 153 - 26 == 127. 127 is > 122. 127 - 26 == 101. 101 is <= 122, and has the character value of **e**
9. Finish translating the ordinal calculated values into characters using the algorithm given above, then compare your calculated signature cypher text against the signature cypher text given for the message.

[2] Note: you can do this in a very similar manner using modulo 26. You will have to manipulate the number a bit to get this step to work but it removes all subtraction.

Things to help:

- The last letter will be added to the first letter in the message.
- You will need to convert the characters to int values using ord.
- spaces are involved in the adding process.

Additional Examples

Input 1	Output 1
`This is a successful message.` `hnitlzdy`	`hnitlzdy equals hnitlzdy` `Gru`

Grow Your Own

In order to be more sustainable and self sufficient, you have decided to invest in a raised garden bed. However, before you can start, you need to figure out how much this is going to cost. In particular, you need to calculate how expensive the soil is going to be.

Your task is to calculate the amount of regular and premium soil you need, and the total soil cost for the garden bed size. The input will consist of the width, height, and depth of your garden bed in feet and inches.

You will use 80% regular soil and 20% premium soil (by volume). The price of regular and premium soil (per cubic yard) is also provided in the input.

There are 12 inches in 1 foot, and there are 27 cubic feet in 1 cubic yard.

Input

The input has five lines, each prefixed by a letter. The first three lines are the width (W), length (L), and depth (D) of the bed in feet and inches. The next two lines are the price of the regular (R) and premium (P) soil per cubic yard. All numbers are integers.

```
W 8 0
L 4 6
D 0 10
R 12
P 32
```

Output

The output must include the amount of soil needed for each layer and the *total* cost for the soil. The first line of output contains "R " and the amount of regular soil in cubic yards (" cu yd"), and the second line contains "P " and the amount of premium soil in cubic yards (" cu yd"), both rounded **up** to 3 decimal places. Finally, the third line contains "T " and the total price of the soil (" dollars"), rounded **up** to 2 decimal places [1].

```
R 0.889 cu yd
P 0.223 cu yd
T 17.81 dollars
```

Discussion

[1] Pay close attention to rounding **up**! Rounding must be exact, and will be strictly enforced by the judges. With the data above, if you don't round up, your answer may be low by one cent.

Additional Examples

Input	Output	Input	Output
W 15 1 L 15 1 D 15 0 R 1 P 1	R 101.115 cu yd P 25.279 cu yd T 126.40 dollars	W 15 0 L 15 0 D 15 0 R 1 P 1	R 100.000 cu yd P 25.000 cu yd T 125.00 dollars

Newspaper Cipher

You are a member of the anti-crypto resistance fighting against the NFT-baron in the digiwars.

As digital communications were disrupted during the great 5G rollout, all communications between the brave members of the rebellion fighting to bring ownership of goods and services back to the people, without the incredible energy costs of *the blockchain* have gone back to printed newspapers.

Only you and your fellow freedom fighters know the secret ciphers with which to pull hidden messages from specific articles. Long live the fighters.

Input

You will receive up to 1000 lines of text characters of a *news article.* (May or may not be an actual news article.) Each line will be no longer than 100 characters, and no shorter than 1 character.

The first line will always be the comma-delimited cipher to be used to pull the message out of the article. Each number (or range of numbers) in the comma-delimited list matches the corresponding text line from the article, e.g. the first item in the list references the first line of text [see discussion note 1]. A zero means no character should be pulled from that row. The first character in a row is character 1 (not zero-indexed). Hyphenated numbers mean to pull that range of characters (all hyphenated ranges will be positive integers). Negative numbers mean to start counting from the end of the line up to that index. Numbers in [square;brackets] separated by semi colons mean to pull all of those characters from that line. (Negative numbers and Hyphenated ranges can appear in bracketed sets.) [See discussion note 2 for a mapping example.]

```
12,[13;18],[22-23;47-49],0,-33,[6;23;43;-1]
There were many happy clouds in the sky today.
But gray skies are on the way.
In other news the last days of Christmas sale at Crazy Johns is ending soon,
right in time for the New Year, so . . .
run, walk, skip or jump down to the location nearest you.
If that isn't enough, we'll have April joining us for Action News at 11!
```

Output

Output the message pulled out of the article with the given cipher, including punctuation. Stop processing lines if the cipher ends before the number of lines ends. Preserve case of the characters pulled from the cipher. E.G. if the characters pulled from the lines spell out "hAppY dANce!" that is how you should output your result, do not change it to "happy dance!" or "Happy Dance!" or "HAPPY DANCE!"

```
meet at dawn!
```

Newspaper Cipher

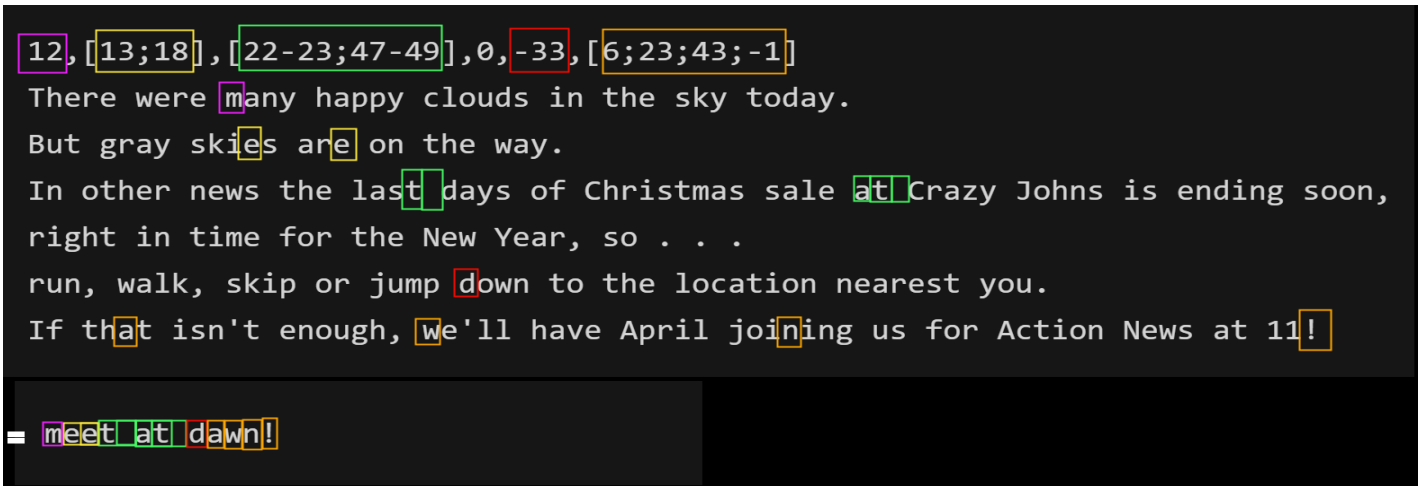
continued

Discussion

- **a zero**: means do nothing with that line (skip it)
- **a positive int**: means to pull that character from that position in the line, starting count from the left, with a start index of 1
- **a negative int**: means to pull that character from that position in the line, starting count from the right
- **a positive int range #1-#n**: works the same as a single positive int, but keep going until you reach #n given in the range (you will not be given a negative int range)
- **a square bracketed list [a;b;...n]**: can consist of any combination of this list, separated by semi-colons.

Reminder: have you run your solution against **all** of the student data sets?

[1] The cipher gives an item in the comma-delimited list to tell you which character(s) to pull [2] Here is a visual on how the cipher maps to the example input and output text.



problem 19

Square Counter

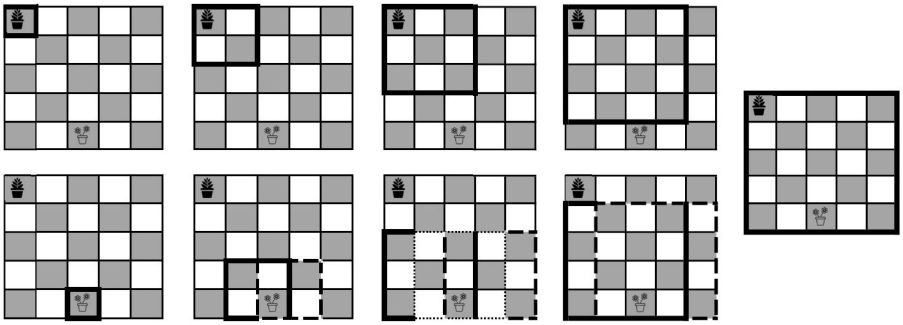
11 points

While Julie was in line at a puzzle and games store, the clerk handed her a puzzle, saying:

"Did you know an 8x8 chess board has 204 sub-squares, ranging in size from 1x1 to 8x8? Well, here's a diagram of a 5x5 board. It contains 55 sub-squares, ranging in size from 1x1 to 5x5. But we've put potted plants in two of the squares, and the challenge is to figure out how many squares can be drawn (following the gridlines, of course) that do not contain a potted plant."

The clerk went on to explain the answer was 42.

A 5x5 grid has 25 (1x1) + 16 (2x2) + 9 (3x3) + 4 (4x4) + 1 (5x5) = 55 squares. But two tiles hold plants. The top-left plant is in five different squares (one of each size). The bottom-center plant is in nine squares: 1 (1x1), 2 (2x2), 3 (3x3), 2 (4x4), and the 1 (5x5). Together, the plants are in 13 of the 55 possible squares, leaving 42 empty squares.



The store has many more similar problems to solve. Can you help Julie find the correct count for each?

Input

The first line of input is an integer N from 4 to 10, the size of the square board. The next N lines each contain N characters, representing a map of the floor tiles. A period '.' is an empty square, and '#' is a plant.

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

Output

Print the total number of completely empty squares of any size.

```
42
```

Discussion

Input 1 has 6x6 all empty tiles, so has 36 more squares than an empty 5x5 grid.

Input 2 has one 2x2 empty square in the top-left. All others 2x2 or bigger hold at least one plant. So there are 49 1x1 squares and 1 2x2 square.

Input 3 is like an empty 4x4 grid, so has one 4x4, four 3x3, nine 2x2 and sixteen 1x1.

Additional Examples

Input 1
```
6
......
......
......
......
......
......
```

Output 1
```
91
```

Input 2
```
8
..#.#.#.
........
#.#.#.#.
........
#.#.#.#.
........
#.#.#.#.
........
```

Output 2
```
50
```

Input 3
```
6
######
#....#
#....#
#....#
#....#
######
```

Output 3
```
30
```

Ring Combos

Amanda wants to decorate her fingers of her left hand differently each day. She has a collection of Red, Blue, and Green rings to choose from.

Given her requirements below, determine how many different ways she can decorate her hand.

Amanda will wear no more than one ring on a finger at a time. (The picture is not accurate here.) She can wear the same ring on different days, as long as the complete hand-decoration is different than previous days. She will decorate only her four fingers, not her thumb. Before she starts, she will decide how many fingers will have a ring (the others will not.)

Input

The input will be one line of four integers, separated by spaces. The first is the number of fingers to have a ring each day. The next three are the number of Red, Blue, and Green rings she has available. For example, if she will decorate exactly 1 finger every day and has only 1 Red and 3 Blue rings, this is the input:

```
1 1 3 0
```

Output

Print the total number of ways she can decorate her fingers. For the above input, she could wear either a Red or Blue ring on a single finger, and she could choose any of her 4 fingers, for a total of 8 ways to wear the rings. (The Blue rings are identical, so having more than one doesn't affect the result.)

```
8
```

Discussion

Input 1 has 3 rings, but she wants rings on 4 fingers. This can't be done, so there are zero solutions.

Input 2 has one pair of Red rings. These can be placed in 6 ways on 4 fingers. The remaining Blue and Green rings can be placed in 2 ways, for 12 total.

For Input 3: Similar to Input 2, she can wear 2 of one color and 1 each of the others in 12 ways. With 3 color choices for the common pair, that's 36 ways. If she instead wears 2 pairs of 2 colors, there are 6 ways to place them, and 3 different ways to pick the 2 colors, for 18 more ways, giving a total of 54.

For Input 4: On two fingers, she can choose a pair of different colors 6 ways, and she can pick a Green pair 1 way, for 7 ways to decorate two fingers. She can pick a pair of fingers in 6 ways, for a total of 42.

For input 5: There is always only one way to decorate 0 fingers.

Additional Examples

```
Input 1   Output 1   Input 2   Output 2   Input 3   Output 3   Input 4   Output 4
4 1 1 1   0          4 2 1 1   12         4 2 2 2   54         2 1 1 2   42

Input 5   Output 5
0 1 2 3   1
```

Round of 16

You've been tasked with writing the logic that helps calculate which two soccer teams in a group of four will advance further in a world cup tournament.

Each of the four teams in a group will play every other team in the group once, resulting in a total of six games. For every game a team plays, the results of the game are recorded -- win, lose, or tie. In addition, for every game, the number of goals that team scored against their opponent ("goals for") are recorded along with the number of goals that the opponent scored against the team ("goals conceded" or "goals against"). This data is used to calculate points for each team and the two teams with the most points will advance to the next round.

POINTS are awarded to teams as follows,

- 3 points for a **win**
- 1 point for a **tie**
- 0 points for a **loss**

If two teams are tied on POINTS as calculated above, the tie breaker will be the GOAL DIFFERENCE as calculated for each team. The GOAL DIFFERENCE is calculated as,

- GOAL DIFFERENCE = (All of the team's "goals for" added up) - (All of the team's "goals against" added up).

Input

The input will consist of 2 sections. The first section will contain the 4 team identifiers, 1 per line, 3 characters each. The next section will contain the 6 matches and their scores. Each match will consist of the first team identifier, a space, the score (separated by a colon), another space and then the second team identifier.

```
GER
GHA
AUS
SER
SER 0:1 GHA
GER 4:0 AUS
GER 0:1 SER
GHA 1:1 AUS
GHA 0:1 GER
AUS 2:1 SER
```

Let's evaluate the games for team GHA to see how they are counted and what the standing is after each game. The standing is the POINTS and GOAL DIFFERENCE for the team.

- The **1st game** is SER vs GHA with a score of 0:1
  - **which counts as**: +3 POINTS (a win), +1 GOALS FOR, and +0 GOALS AGAINST
  - **creating a game standing of**: 3 1
- The **2nd game** is GHA vs AUS with a score of 1:1
  - **which counts as**: +1 POINTS (a tie), +1 GOALS FOR, and +1 GOALS AGAINST
  - **creating a game standing of**: 1 0
- The **3rd game** is GHA vs GER 0:1
  - **which counts as**: +0 POINTS (a loss), +0 GOALS FOR, and +1 GOALS AGAINST
  - **creating a game standing of**: 0 -1
- The **final standing** for team GHA is the **sum of all their game standings**: 4 0

Output

The output will be the final ranking of the four teams in the group, in descending order, starting with the top team based on POINTS, and using GOAL DIFFERENCE as a tie-breaker.

For each team print the team identifier, the number of POINTS and the GOAL DIFFERENCE (which can be negative!).

```
GER 6 4
GHA 4 0
AUS 4 -3
SER 3 -1
```

Discussion

The datasets will never result in a tie in both POINTS and GOAL DIFFERENCE.

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1	Input 2	Output 2
GER GHA POR USA GER 4:0 POR GHA 1:2 USA GER 2:2 GHA USA 2:2 POR USA 0:1 GER POR 2:1 GHA	GER 7 5 USA 4 0 POR 4 -3 GHA 1 -2	USA SWE CHI THI CHI 0:2 SWE USA 13:0 THI SWE 5:1 THI USA 3:0 CHI SWE 0:2 USA THI 0:2 CHI	USA 9 18 SWE 6 4 CHI 3 -3 THI 0 -19

Baby Yotta

The Child demands written numbers in all communications about data. Using SI units (base 10) for data bytes, translate the given data usage for the Mandalorian's communications to words from numbers using the following SI-byte scale[1]:

byte 10^0	deca 10^1	hecto 10^2	kilo 10^3	mega 10^6	giga 10^9
1	10	100	1000	1000000	1000000000

tera 10^{12}	peta 10^{15}	exa 10^{18}
1000000000000	1000000000000000	1000000000000000000

zetta 10^{21}	yotta 10^{24}
1000000000000000000000	1000000000000000000000000

bronto 10^{27}	gego 10^{30}
1000000000000000000000000000	1000000000000000000000000000000

Input

You will receive one (possibly extremely large) integer on one line, representing the Mandalorian's data usage from his wrist-communicator that month in SI-bytes through Sithrizon Inc.™ Data plans through Sithrizon Inc. max out at 999999999999999999999999999999 bytes used per month (because that is literally how high their accounting systems can count, else they would try to charge for more). (**Note** included in your student data for this problem is a **numbers.txt** file with spelling and spacing examples of SI unit numbers written as words. You can copy/paste that data into your program to speed up your solution -- the file *will not* be present in the judges data, and *does not* write out every single number between 0-999.) Example:

```
6000009004
```

Output

If zero bytes were used, output "zero bytes". Translate the bytes to words, down to the single bytes. Use lowercase for your output. Insert a comma between each SI unit's output, unless you are only outputting one unit. Insert the word "and" before your final unit. If you only have one unit to output, you can omit the "and". If inserting the word "and" before the final unit **do not** include a leading (Oxford) comma. E.G. *... kilobytes, and seven bytes* is **wrong**, write that as *... kilobytes and seven bytes*.

Example output from input:

```
six gigabytes, nine kilobytes and four bytes
```

Discussion

Most programming languages are not going to comfortably store the value of 999999999999999999999999999999 as an **int** (or even as a **long**). You will need to get creative (using standard libraries allowed for the contest) to parse the data.

[1] Note, bronto and gego are proposed, but not *actual* SI-byte units. We included them just for fun. **DO NOT** start walking around at school telling people that brontobyte and gegobytes are official SI units, citing us as our source :P

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1
999999999999999999999999999999	nine gegobytes, nine hundred ninety nine brontobytes, nine hundred ninety nine yottabytes, nine hundred ninety nine zettabytes, nine hundred ninety nine exabytes, nine hundred ninety nine petabytes, nine hundred ninety nine terabytes, nine hundred ninety nine gigabytes, nine hundred ninety nine megabytes, nine hundred ninety nine kilobytes, nine hectobytes, nine decabytes and nine bytes

Input 2	Output 2
8	eight bytes

Input 3	Output 3
0	zero bytes

Input 4	Output 4
8000050007600000020000	eight zettabytes, fifty petabytes, seven terabytes, six hundred gigabytes and twenty kilobytes

Input 5	Output 5
7020000800713400560180 49	seven hundred two zettabytes, eighty petabytes, seventy one terabytes, three hundred forty gigabytes, fifty six megabytes, eighteen kilobytes, four decabytes and nine bytes

problem 23

Champ R. Nowne's Cypher

13 points

The dubiously infamous Champ R. Nowne has created a cypher difficult to crack, but we've discovered a secret. He has a strange fascination with the mathematician Champernowne, who proposed an irrational decimal fraction, created by concatenating ALL of the positive integers in order:

```
0. 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 ...
(remove spaces)
0.12345678910111213141516171819202122232425262728293031323334353637...
Position 15 ----^ is "21"=U         Position 51 ----^ is "03"=C
...838485868788899091929394959697989910010110210310410510610710810 9...
 Position 171 ----^ is "09"=I      Position 202 ----^ is "10"=J
```

Nowne's cyphers are full of integers, and we think there's a connection.

Notice in the decimal, at the 15th position, we see the 2-digit number 21. And the 21st letter of the alphabet is U.

Taking each number in the codes below as a position in the decimal, we can find 2-digit numbers and convert them to letters of the alphabet (01=A, 26=Z). For example, the five positions (24 15 20 30 11) in the decimal fraction provide the 2-digit numbers (17 21 15 20 01). Converting these to letters of the alphabet provides the 5-letter word "QUOTA".

Input

Each line includes five positive integers less than one million, separated by spaces. The last line is five zeros, marking the end of the input.

```
24 15 20 30 11
11 1 31 15 16
959414 39968 407973 625995 751850
11 31 51 71 91
111 131 151 171 202
0 0 0 0 0
```

Output

Find the 2-digit number in the Champernowne fraction at all five locations. Convert the 2-digit numbers into capital letters, and print each 5-letter word on its own line.

```
QUOTA
ALBUM
TOPAZ
ABCDE
FGHIJ
```

Additional Example

```
Input                                Output

323928 828602 928724 39418 389963    GREEN
29853 250483 436898 232988 463249    GRASS
43958 502875 233948 299418 762008    MAKES
764312 92926 405438 138299 895118    CHILD
478299 468458 706748 603667 313418   SMILE
0 0 0 0 0
```

Inspiration for this problem comes from problem 40 at projecteuler.net

problem 24

Squirrel!

14 points

Oh no! You worked hard for the last 47 hours writing your student council speech, and copying it down onto note cards . . . but apparently those squirrels outside your window were more distracting than you realized.

During a last minute check of your cards you see that you kept trailing off words and writing SQUIRREL instead.

Even worse, your speech is out of order too. You didn't write numbers on the cards, so you will have to refer back to your written draft of your speech to fix it. You're up next, so hurry!

Input

On the first line you will be given a comma delimited list of possible words which may have gotten squirrelified (*yes that's a real word, why are you looking at us like that?*).

On the second line you will be given a space delimited list of the first and last words on each note card in square [bracket, pairs]. On the remaining lines you will be given the text of the speech from your most recent draft. It was, unfortunately, saved out of order.

```
sushi,short.,atrocity,pizza,march,shout,president,injustice,students,vote,Join,wrong.,corre
cted,protest!
[Vote,short.] [Today,cafeteria] [Now,wrong.] [But,protest!] [And,Jenny!]
And as your class pre-SQUIRREL I will lead the way.
Jo-SQUIRREL me, and together, we'll never go hungry for sushi again!
Thank you, and don't forget to vo-SQUIRREL for Jenny!
But for the rest of us, we must march, we must shout, we must take to the streets and pro-
SQUIRREL
Today I want to speak about an atrocity, nay, an in-SQUIRREL which must be corrected.
Of course I am referring to the lack of su-SQUIRREL options in this, our beloved cafeteria.
Now, I know that some of you prefer pi-SQUIRREL to sushi, and that's okay. You're wrong,
but that's okay. People have a right to be wr-SQUIRREL
Vote for Jenny,
My fellow st-SQUIRREL I know you don't want to sit through any more long speeches, so I'll
keep this sho-SQUIRREL
```

Output

Your task is to reassemble the last workable draft of the speech before it was squirrelified.

First find the squirrely words and cross-reference them with the known-good list of words. (Note, punctuation must be included. Any punctuation touching a word is considered to be part of that word when looking for squirrelified words. All possible punctuation you will encounter in squirrelified words will be the: period and exclamation mark.) All squirrelified words will begin with the correct 2-3 letters before the squirrel madness took over.

Next, refer to the speech note-card page order of the bracketed words list. The order given of the start-stop words, and their order in the list is correct. Also note, the speech was saved so that all lines which need to stay on the same card, are in the correct order, and together. The groups of lines, however, are in the wrong order. Put the lines into the order given in the [square, bracket] start-end sequence, and then print out the final (de-squirrelified) speech.

```
Vote for Jenny,
My fellow students I know you don't want to sit through any more long speeches, so I'll
keep this short.
Today I want to speak about an atrocity, nay, an injustice which must be corrected.
Of course I am referring to the lack of sushi options in this, our beloved cafeteria.
Now, I know that some of you prefer pizza to sushi, and that's okay. You're wrong, but
that's okay. People have a right to be wrong.
But for the rest of us, we must march, we must shout, we must take to the streets and
protest!
And as your class president, I will lead the way.
Join me, and together, we'll never go hungry for sushi again!
Thank you, and don't forget to vote for Jenny!
```

Discussion

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1

```
dishonored,saved,impersonated,officer,armor,palace,victory,difference,rice,person,Army
[A,defeat.] [I've,officer] [dishonored,all.]
dishonored the Chinese Ar-SQUIRREL
destroyed my pa-SQUIRREL
and you have sa-SQUIRREL us all.
I've heard a great deal about you, Fa Mulan.
You stole your father's ar-SQUIRREL
ran away from home, im-SQUIRREL a soldier
deceived your commanding off-SQUIRREL
A single grain of ri-SQUIRREL can tip the scale.
One person may be the dif-SQUIRREL between victory and defeat.
```

Output 1

```
A single grain of rice can tip the scale.
One person may be the difference between victory and defeat.
I've heard a great deal about you, Fa Mulan.
You stole your father's armor
ran away from home, impersonated a soldier
deceived your commanding officer
dishonored the Chinese Army
destroyed my palace
and you have saved us all.
```

Input 2

```
halls,Friends,Join,better,Elizabeth,reports,invite,Vote,class,simple,sparkly,vampires,purchasing,Nonieqa,but,Renegade,shelves
[Friends,reason.] [We,person] [but,Ramos] [Azura,me.] [Join,you.]
but I think we can do better.
Our school's pur-SQUIRREL patterns are sus.
Until I see something like Well Beauty Woke by Non-SQUIRREL Ramos
We need be-SQUIRREL books in our library.
I mean, listen, I love a cringe-fest about sparkly vam-SQUIRREL as much as the next person
Join me, and together we can usher in a new age of better book reports for everyone!
Vo-SQUIRREL for Elizabeth.
Thank you.
Fr-SQUIRREL students, those of you whom I've waved to in the ha-SQUIRREL
I am running for cl-SQUIRREL president for one simple reason.
Azura Ghost by Essa Hansen or Renegade Moon by T A Chan on those she-SQUIRREL
I will keep pushing for more and better, and I inv-SQUIRREL you all to stand with me.
```

Output 2

```
Friends, students, those of you whom I've waved to in the halls
I am running for class president for one simple reason.
We need better books in our library.
I mean, listen, I love a cringe-fest about sparkly vampires as much as the next person
but I think we can do better.
Our school's purchasing patterns are sus.
Until I see something like Well Beauty Woke by Nonieqa Ramos
Azura Ghost by Essa Hansen or Renegade Moon by T A Chan on those shelves
I will keep pushing for more and better, and I invite you all to stand with me.
Join me, and together we can usher in a new age of better book reports for everyone!
Vote for Elizabeth.
Thank you.
```

problem 25

YouTube Pie Throwing Contest

15 points

Jessica follows a few YouTubers. She particularly likes those that offer prizes. One of her favorites, Mohrgreene Fouryu, holds a contest whenever he passes a "more followers" milestone. The next contest's format is a virtual pie-throwing battle, with the winner getting a cash prize!

When the registration opens, Mohrgreene will announce the Maximum number of players (not more than 20,000). Sometimes there's an advantage to sign up early, sometimes later, but don't be so late you miss it! Mohrgreene's contests always fill up. The rules for the virtual battle are:

- Each player will be assigned a PlayerID number based on the order they sign up (from 1 to the Max).
- Each player's initial Health Points (HP) will be 10 plus their PlayerID. So the first player will start with 11 HP, and the last will have 10+Max.
- Players will be arranged into a (virtual) circle in order of their PlayerID (1 to Max) clockwise
- Players will play in order of their Player ID (1 to Max), repeating until only one remains.
- The grand prize will go to the one player remaining.

The play is very simple. - On their turn, each player becomes the Attacker and will throw their (virtual) pie at another player, the Target. - The Target will LOSE as many HP as the Attacker has. - When a player's HP drops below 1, they are eliminated from the contest.

Jessica decides the best strategy is to attack the player with the most HP, so they aren't as powerful when they take their turn, in case they fire back at her. Somehow, **every player chooses the same approach**: - Searching clockwise from their position, they aim at the first other player with the highest HP. (If two players share the highest HP, the player that comes first in a clockwise direction will be the one selected for an attack.)

For example, with 15 people, on the first turn, PlayerID 1 has 11 HP and aims at PlayerID 15 (with 25 HP). After that pie toss, Player 15 has 14 HP remaining. Then, PlayerID 2 chooses PlayerID 14 (with 24 HP) as the next Target.

Jessica needs your help. When the maximum number of competitors is announced and the contest opens, people won't want to be left out, so they will start registering, and the player counter will start climbing. Jessica needs to know quickly when to register to get the grand prize.

Input

The only line of input is an integer N (from 11 to 4000), the number of players in the contest.

```
100
```

Output

On their own lines, print the PlayerIDs of the first player eliminated, tenth player eliminated, and the grand prize winner.

```
47
70
25
```

Additional Examples

```
11      13      20      4000      2511      789

5       6       2       1643      2255      673
9       8       9       448       1611      392
7       7       16      339       10        144
```

Discussion

Your solution must print the grand prize winner PlayerID in under 1 minute, or it will be marked as a failure. Measure your time before submitting. Improve your algorithm if it's too long.

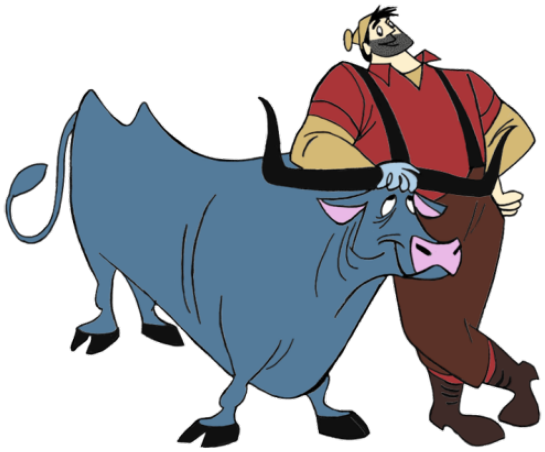
Look at ALL the student dataset output files for some complete lists of contest attacks.

For example, for a contest with 11 people, these are all the attacks and targets, and the new HP for each target.

```
Attacker/HP pies Target/HP ==> Target/newHP

1/11 pies 11/21 ==> 11/10
2/12 pies 10/20 ==> 10/8
3/13 pies 9/19 ==> 9/6
4/14 pies 8/18 ==> 8/4
5/15 pies 7/17 ==> 7/2
6/16 pies 5/15 ==> 5/-1  5 eliminated.
7/2 pies 6/16 ==> 6/14
8/4 pies 4/14 ==> 4/10
9/6 pies 6/14 ==> 6/8
10/8 pies 3/13 ==> 3/5
11/10 pies 2/12 ==> 2/2
1/11 pies 4/10 ==> 4/-1  4 eliminated.
2/2 pies 1/11 ==> 1/9
3/5 pies 11/10 ==> 11/5
6/8 pies 1/9 ==> 1/1
7/2 pies 10/8 ==> 10/6
8/4 pies 6/8 ==> 6/4
9/6 pies 10/6 ==> 10/0  10 eliminated.
11/5 pies 9/6 ==> 9/1
1/1 pies 3/5 ==> 3/4
2/2 pies 11/5 ==> 11/3
3/4 pies 6/4 ==> 6/0  6 eliminated.
7/2 pies 8/4 ==> 8/2
8/2 pies 3/4 ==> 3/2
9/1 pies 11/3 ==> 11/2
11/2 pies 2/2 ==> 2/0  2 eliminated.
1/1 pies 3/2 ==> 3/1
3/1 pies 7/2 ==> 7/1
7/1 pies 8/2 ==> 8/1
8/1 pies 11/2 ==> 11/1
9/1 pies 11/1 ==> 11/0  11 eliminated.
1/1 pies 3/1 ==> 3/0  3 eliminated.
7/1 pies 8/1 ==> 8/0  8 eliminated.
9/1 pies 1/1 ==> 1/0  1 eliminated.
7/1 pies 9/1 ==> 9/0  9 eliminated.
```

Falling Trees



Paul Bunyan is just starting out as a lumberjack and needs your help. He needs to clear out a grid of trees by cutting them down. The hard part is that he cannot have any trees fall on top of each other or work gets much more difficult. The biggest problem is that he has no idea how to cut down a tree to make it fall in the direction he wants it to. The trees can fall up, down, left or right! Luckily the grid is pretty scarce already and there is only 1 set of trees max that can fall on each other.

When the trees fall, they fall in any direction and take up height - 1 spaces. Tress will be no taller than 9 in height, and no smaller than 1 (if there is no tree, there will be an x). If a tree of height 1 falls, it will fall in no direction, it stays in place. We only care about the part of trees that fall inside the grid.

Could you please tell Paul Bunyan if there is a pair of trees that can cause overlap? If there is no overlap print "None"

Below is an example of a tree falling where "-" is a space where a tree could possibly fall.

```
Before:
|x|x|x|x|x|x|x|
|x|x|x|x|x|x|1|
|x|x|x|x|x|x|x|
|x|x|3|x|x|x|x|
|x|x|x|x|x|x|x|
|x|x|x|x|x|x|x|

After:
|x|x|x|x|x|x|x|
|x|x|-|x|x|x|-|
|x|x|-|x|x|x|x|
|-|-|-|-|-|x|x|
|x|x|-|x|x|x|x|
|x|x|-|x|x|x|x|
```

Input

The first line of input will tell you how wide the grid is, and the second line of input will tell you how tall the grid is. All lines after that will contain the grid specified by the first two lines.

```
13
5
xx2xxxxx2xxx1
x4xxxx1xxx2xx
xxxx1xxxx1xx1
xxxxx1xx1xx2x
xxx1xxxxxx1xx
```

Output

If there is a pair of trees that can cause overlap, you will need to print out the coordinates of each tree. If there is overlap, print "None" (without the quotes). Coordinates start at 1. As these are coordinates, the set is (x,y) where x is left and right, and y is up and down. Print the coordinate with the lowest x value first, if this value is the same, print the value with the lowest y value first.

```
(2,2)(3,1)
```

Discussion

A tree of size 1 only falls in the same square it was cut, it does not really "fall" in any direction. Anything that is outside of the grid, can be ignored.

Additional Examples

Input	Output
13 5 xx2xxxxxxxx1 x1xxxx3xxx3xx xxxx1xxxx1xx1 xxxxx1xx1xx2x xxx2xxx2xx1xx	(7,2)(11,2)
13 5 xx2xxxxxxxx1 x1xxxxxxx2xx xxxx1xxxx1xx1 xxxxx1xx1xx2x xxx2xxx2xx1xx	None

Blocked Chains

You have been assigned to a Cyber Command task force.

Your mission is to prove that Illiad, the notorious donut ganster, has been laundering money through *blockchain* investments of crypto-currency.

All blockchains are, at their core, a public ledger of transactions tied to a unique ID and added to the ledger with public key signing cryptography in various ways.

Fortunately for your task force, the AGlazingCoins Blockchain Illiad is using for their money laundering cover isn't very sophisticated.

Input

You will receive a section of the public ledger of the blockchain which is suspected to contain evidence of money laundering. While the task force is capturing a copy of the ledger in real time, it is terabytes in size, and growing by the hour. So we're only examining it in pieces.

At the top of the file will be three pieces of critical information. On the first line will be Illiad's actual bank account ID he tells the IRS about. On the second will be the blockchain wallet ID Illiad uses for his *investments*. On the third line will be a list (comma delimited) of all suspected wallet IDs Illiad and his associates control. The rest of the data is the captured transaction ledger. Unfortunately, the ledger is not in chronological order - some entries are added later than others due to processing delays.

```
EagleBank-123-987-1029
0f8c2990-23e8920708dc
c1730fac-92cb6e9070f8,65b3db3e-8c98e3abc84f
{
{"d":"2022-07-23 06:50:51","f":"f995f3bd-c5704065bd46","t":"0f610683-75a94bfb8b79","a":"6264.81","c":"GZC","h":"NfMmsR24Ew24uA=="}
,{"d":"2022-01-15 06:34:53","f":"EagleBank-123-987-1029","t":"0f8c2990-23e8920708dc","a":"500.00","c":"GZC","h":"evY+lsOJ3fft0g=="}
,{"d":"2022-01-21 11:40:50","f":"0f8c2990-23e8920708dc","t":"c1730fac-92cb6e9070f8","a":"200.00","c":"GZC","h":"WeHXwblF/WXaYQ=="}
,{"d":"2022-03-19 09:16:49","f":"0f8c2990-23e8920708dc","t":"65b3db3e-8c98e3abc84f","a":"100.00","c":"GZC","h":"9oLAtJ/9aFpx+Q=="}
,{"d":"2022-07-26 00:37:36","f":"20fc161a-b70844f3b9c9","t":"e4c0908b-d9a84a58b232","a":"4590.10","c":"GZC","h":"Nam7ktPX61ZlaQ=="}
,{"d":"2022-06-28 09:32:01","f":"ShadyBank-9283-6593-6345","t":"ShadyBank-6448-3115","a":"4192.50","c":"GZC","h":"NEphuF+stnroJA=="}
,{"d":"2022-03-21 11:40:50","f":"0f8c2990-23e8920708dc","t":"a5390494-bdd8295c7246","a":"200.00","c":"GZC","h":"WeHXwxlF/WXaYQ=="}
,{"d":"2022-03-10 03:38:00","f":"c1730fac-92cb6e9070f8","t":"9c5128fe-c3b56d0bbdf5","a":"200.00","c":"GZC","h":"gD6lgCDX59BbLQ=="}
,{"d":"2022-04-18 17:53:35","f":"65b3db3e-8c98e3abc84f","t":"fcb1af3b-06e024132bfb","a":"100.00","c":"GZC","h":"oExtjQpjy6BBDg=="}
,{"d":"2022-03-27 10:25:04","f":"a5390494-bdd8295c7246","t":"fcb1af3b-06e024132bfb","a":"200.00","c":"GZC","h":"f0hGl5EDLhg=="}
,{"d":"2022-07-11 18:12:53","f":"fcb1af3b-06e024132bfb","t":"06ab75c9-bcb84ed2bf56","a":"300.00","c":"GZC","h":"cBNePbq56NCL7Q=="}
,{"d":"2022-01-11 00:19:14","f":"ce5cb39e-24aa4838bf8d","t":"7f3caa9c-df664a0f846c","a":"1732.88","c":"GZC","h":"ZD5AU7I5KBrBWA=="}
,{"d":"2022-07-22 13:13:42","f":"9c5128fe-c3b56d0bbdf5","t":"06ab75c9-bcb84ed2bf56","a":"200.00","c":"GZC","h":"m8ygR0dPRV61Kw=="}
,{"d":"2022-10-21 14:02:49","f":"HiddenBank-9220-142","t":"637781b3-c32d4c608612","a":"7811.64","c":"GZC","h":"Z7QhyoiDLe1zQg=="}
,{"d":"2022-09-22 05:23:36","f":"06ab75c9-bcb84ed2bf56","t":"EagleBank-123-987-1029","a":"137.16","c":"USD","h":"l3Sl9E9ovITlhA=="}
,{"d":"2022-05-01 15:03:50","f":"076b27ed-677544ce8f02","t":"44e0ab94-130546138c2c","a":"4147.89","c":"GZC","h":"ex0fX32GtrgH4w=="}
,{"d":"2022-07-15 10:16:58","f":"06ab75c9-bcb84ed2bf56","t":"EagleBank-123-987-1029","a":"362.82","c":"USD","h":"IGs6ulKX7KNa3A=="}
}
```

Output

Illiad is crafty. To launder $500 (**USD**) of ill-gotten donuts gains, he isn't just going to make a purchase of $500 worth of GlazeCoins (**GZC**) and immediately transfer back into his bank account. (Luckily, the value of GlazeCoins (GZC) is tied to the dollar, so at least you won't have to calculate *that* as well.) He is going to split it up into smaller transactions. You must search for all of the transactions Illiad makes, from one wallet ID to another, to find **all** transactions that link back to Illiad. That means, as soon as a known wallet or bank ID linked to Illiad touches another (previously unknown) wallet or bank ID, that ID gets added to the suspect list, and that transaction must be part of your output (even if it dead-ends and ultimately is not involved with moving the USD money in and out of Illiad's bank account).

In order to send in the U.S. Marshals to arrest Illiad and his gang, your task force needs to build an iron-clad case against him. You need to output **all** of the transactions (in chronological order) found in the AGlazingCoins Blockchain that have any link to Illiad's Bank Account ID through various wallet IDs. Then print the total in and out amounts for Illiad's bank account **rounded to 2 decimal places**.

```
2022-01-15 06:34:53 from: EagleBank-123-987-1029 to: 0f8c2990-23e8920708dc amount:500.00GZC
2022-01-21 11:40:50 from: 0f8c2990-23e8920708dc to: c1730fac-92cb6e9070f8 amount:200.00GZC
2022-03-10 03:38:00 from: c1730fac-92cb6e9070f8 to: 9c5128fe-c3b56d0bbdf5 amount:200.00GZC
2022-03-19 09:16:49 from: 0f8c2990-23e8920708dc to: 65b3db3e-8c98e3abc84f amount:100.00GZC
2022-03-21 11:40:50 from: 0f8c2990-23e8920708dc to: a5390494-bdd8295c7246 amount:200.00GZC
2022-03-27 10:25:04 from: a5390494-bdd8295c7246 to: fcb1af3b-06e024132bfb amount:200.00GZC
2022-04-18 17:53:35 from: 65b3db3e-8c98e3abc84f to: fcb1af3b-06e024132bfb amount:100.00GZC
2022-07-11 18:12:53 from: fcb1af3b-06e024132bfb to: 06ab75c9-bcb84ed2bf56 amount:300.00GZC
2022-07-15 10:16:58 from: 06ab75c9-bcb84ed2bf56 to: EagleBank-123-987-1029 amount:362.82USD
2022-07-22 13:13:42 from: 9c5128fe-c3b56d0bbdf5 to: 06ab75c9-bcb84ed2bf56 amount:200.00GZC
2022-09-22 05:23:36 from: 06ab75c9-bcb84ed2bf56 to: EagleBank-123-987-1029 amount:137.16USD
Total Out: 500.00USD
Total In: 499.98USD
```

Discussion

Supply the trailing zero in the 2-decimal rounded total in/out amounts. 1234.5 is wrong. 1234.50 is what the judges will expect.

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Too large to show here, check your student datasets.

problem
28

No Time Left

20
points

Squidward has been ordered to bring in a truck load of meat for crabby patties as fast as possible. There is a huge distance between the truck and the kitchen; Squidward wants to know exactly how far he is going to have to walk to get this done.

He has a map that shows the position of the truck (T), the position of walls(W), the drop off location (D), and also the empty spaces that have no obstacles (-). Each of these sections represents 1 distance that Squidward will have to walk. Keep in mind that he can only walk up, left, down, and right, **he cannot walk diagonally.** can you tell him the total distance he will need to travel?

Input

The first line of input will tell you how wide the grid is, the second line of input, will tell you how tall the grid is. All lines after that will contain the grid specified by the first two lines.

The width and height are greater than 0 but less than 20.

```
7
4
-------
---WWW-
-T-WDW-
---W---
```

Output

The output should be a single number representing the least amount of squares needed to get from the Start to the Finish.

```
13
```

Discussion

Squidward can only move left, right, up, and down. He cannot move diagonally to solve this problem. Each Square counts as one "distance" traveled. There is always at least one (and only one) solution to get Squidward from the Truck to the Drop of point.

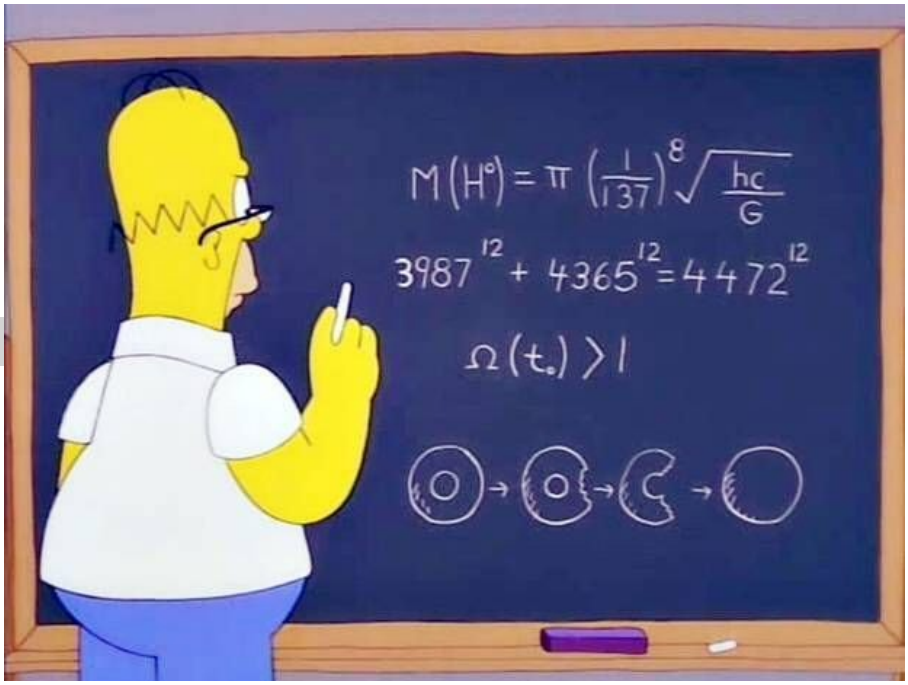
Character Mappings:

Character	Meaning
-	Walkable Space
W	Wall
S	Starting Space
D	Drop Off Space

Note: The letters are capitals, not lowercase.

Long Division

What do you do when you don't have a calculator, and you need to divide some numbers? Long division!



Input

You will receive two numbers, on a single line, separated by a forward slash, with no spaces between them.

```
250501/25
```

Output

Setup the numbers with the divisor on the left, and the dividend to the right of a vertical bar (|). Above the dividend (starting at the same column from which the dividend starts) output 10 underscore characters (_). With the setup done, carry out the division until you have a remainder, or you finish determining the number of times the divisor can be evenly divided into the dividend. Output your answer above the top bar you made with the underscores.

If (and only if) there will be no remainder from the division, you can simply show the answer **without showing work**. But, **If there will be a remainder,** include a space, then the uppercase letter R, another space, and then the remainder. If there will be a remainder you must **SHOW YOUR WORK** below the initial setup of the problem. You need to show only the subtraction step(s) followed by the "carry down" steps each time you place a non-zero number into the answer.

Stop your output when you have exhausted all "carry down" steps, or have finished evenly dividing the divisor into the dividend.

```
    10020 R 1
   __________
25|250501
  -25
   00050
     -50
      001
```

Discussion

Please see the included **longDivision.pdf** file in your datasets if you need a reminder on how long division works. All dividends you receive will be divisible at least one time by the divisor given. You will not be given any "undividable" problems such as 0/1.

Output formatting rules (short version):

- Align answer at top to columns where even division started (print nothing until at least one digit can be put into the answer).
- Print all subtraction steps
- Show results of subtraction with leading zeros to match column position, but only starting from last subtraction/carry-down step
- Print all "carry down" steps
- Stop printing steps when the remaining value left in the dividend is either equal to or less than the divisor

Output formatting discussion (long version):

	A	B	C	D	E	F	G	H	I	J	K	L	M
1					1	0	0	2	0		R		1
2				_	_	_	_	_	_	_	_	_	_
3	2	5	|	2	5	0	5	0	1				
4			-	2	5								
5				0	0	5	0						
6					-	5	0						
7						0	0	1					

1. Start with your problem setup. As you can see in the illustration, the divisor will be the leftmost number on the equation line, starting in the **A** column. It is 2 lines down from the *top* to give us enough room for the answer, and the long division bar of 10 underscores on the top (which don't look that good when zoomed in like this, but look fine on the screen).
2. After the divisor put a vertical bar (|) to the right of it. In our example it is in **C3**, because that is the most immediate position to the right of the divisor.
3. Immediately after the vertical bar, output the dividend. In our example that starts at **C4**, because that is the most immediate position to the right of the vertical bar.
4. One row above the vertical bar, and one position to the right, at **D2** in our example, output 10 underscore (_) characters. Leave a row free above this row, that will be row 1, where you will output your answer.
5. With the setup complete, start working through the long division. In our example, 25 does not evenly divide into 2, the first digit of 250501, so we place nothing in our answer row (do not place anything into the answer row until you have a non-zero number to place. Once one number has been placed, output a value for each operation after that), and add the next digit to the value we are checking comprising columns **D** and **E** in our example. That creates a value of 25, which *can* be evenly divided by our divisor 1 time.
6. Output a 1 on the answer row starting at column **E**. We start at E because we couldn't evenly divide into the number in column **D**.
7. Multiply 1 against our divisor, giving us 25 to subtract from our dividend.
8. Output a hyphen/minus sign (-) to the left of where we will be lining up our columns so they match where we are subtracting from. In this case, that will be directly under the vertical bar at position **C4** in our example.
9. Place the 25 we got from multiplying 1 against our divisor of 25 to the right of the hyphen so the columns line up with the right side of the column where we placed our answer. In this case, column **E**.
10. Carry out the subtraction and "bring down" the result. In this case zeros in both columns **D** and **E**.

11. Check the next number in the dividend. It is at position **F3** and it is zero. We cannot divide into zero, so we add zero to our answer row in position **F1**, we do not bring the zero down to row 5, as it isn't the result of subtraction and adding zero to zero doesn't change our value.
12. Add the next digit from the dividend into the value we are checking, making the value five.
13. Carry the 5 down to position **F5** which is where we are keeping track of our new running division.
14. We cannot divide 25 into 5, so we add zero to our answer row in position **G1**, then add the next digit from the dividend into the value we are checking, and carry that down to position **G5**, making the new value 50.
15. We *can* divide 25 evenly into 50, a total of 2 times. Place a 2 in our answer row at position **H1**.
16. Multiply 2 against our divisor of 25, giving a total of 50 to subtract.
17. Output a hyphen/minus sign (-) to the left of where we will be lining up our columns so they match where we are subtracting from. In this case, that will be at position **E6** in our example.
18. Output the 50 we got from multiplying our 2 from our answer against our divisor of 25 starting at position **F6**.
19. Carry out the subtraction and "bring down" the result. In this case zeros in both columns **F** and **G**.
20. We cannot divide 25 into zero, so we add zero to our answer row in position **I1**, then add the next digit from the dividend into the value we are checking, and carry that down to position **H7**, making the new value 1.
21. We stop here as we cannot evenly divide 25 into 1, and we have no more digits to carry down.
22. Add a space to the right of our whole number division answer at position **J1** then an **R** (for remainder) at position **K1**, another space after that at position **L1** and then our remainder after that, starting at position **M1** (if our remainder was two digits wide, then it would occupy positions **M1-N1**)

Reminder: have you run your solution against **all** of the student data sets?

Additional Examples

Input 1	Output 1	Input 2	Output 2

`100/24`	`  4 R 4`
	`  ____`
	`24|100`
	`  -96`
	`   04`
		`99999/3`	`  33333`
			`  _____`
			`3|99999`

Input 3	Output 3	Input 4	Output 4

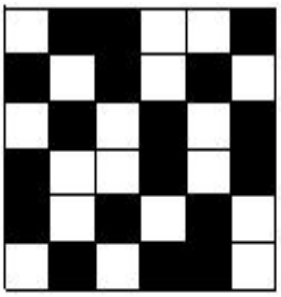
`33363/333`	`  100 R 63`
	`  _____`
	`333|33363`
	`  -333`
	`   00063`
		`1345/103`	`  13 R 6`
			`  _____`
			`103|1345`
			`  -103`
			`   0315`
			`   -309`
			`    006`

The Balanced Baker

After a year contemplating the difficulties he created, the Baker has returned with a simpler scenario. He still creates party cakes by placing many cupcakes side-by-side to make a single large cupcake-square. All of his individual cupcakes are either Vanilla or Chocolate flavored. The reformed Baker has developed a new set of rules for each cupcake-square. The Vanilla and Chocolate cupcakes are spread out and balanced across the full square:

- No more than 2 neighboring cupcakes in a line (row or column) are the same type. So, there will never be 3 of the same type together in a line.
- In every row and column, there are an equal number of Vanilla and Chocolate cupcakes.

Following the above rules, the baker has supplied partial diagrams of his full squares, showing just enough information to construct the layout. For example, in the following diagram, black and white are known to be Chocolate and Vanilla cupcakes. The rest of the cupcakes are unknown, but can be determined by following the rules.



Input

The first line of input is an even integer N (from 6 to 14), the size of the cupcake-square. The next N lines contain N characters, each representing:

- 'v' - for Vanilla cupcakes
- 'C' - for Chocolate cupcakes
- '.' - a period for Unknown cupcakes

```
6
.CC...
...v..
.Cv..C
..v.v.
......
v....v
```

Output

Print the fully defined grid, showing all Vanilla (v) and Chocolate (C) cupcakes. There is only one solution.

```
vCCvvC
CvCvCv
vCvCvC
CvvCvC
CvCvCv
vCvCCv
```

Additional Examples

Input 1	Output 1	Input 2	Output 2
6		8	
..C...	vCCvCv	C.C....C	CvCvCvvC
C.C...	CvCvvC	....	vvCCvCCv
C...C.	CvvCCv	..v..vv.	CCvvCvvC
...v.v	vCCvCv	..v.....	vCvCvCCv
......	CvvCvC	....	CvCCvvCv
v...v.	vCvCvC	vv.v.C..	vvCvCCvC
		.....C.C	vCvCvCvC
		...v....	CCvvCvCv

Inspiration for this problem comes from "Unruly" in the App "Genius Puzzles"