Title: [easy] challenge #1
Text: create a program that will ask the users name, age, and reddit username. have it tell them the information back, in the format:

your name is (blank), you are (blank) years old, and your username is (blank)

for extra credit, have the program log this information in a file to be accessed later.

Title: [easy] challenge #2
Text: Hello, coders! An important part of programming is being able to apply your programs, so your challenge for today is to create a calculator application that has use in your life. It might be an interest calculator, or it might be something that you can use in the classroom. For example, if you were in physics class, you might want to make a F = M * A calc.

EXTRA CREDIT: make the calculator have multiple functions! Not only should it be able to calculate F = M * A, but also A = F/M, and M = F/A!

Title: [2/11/2012] Challenge #3 [easy]
Text: Welcome to cipher day!

write a program that can encrypt texts with an alphabetical caesar cipher. This cipher can ignore numbers, symbols, and whitespace.

for extra credit, add a "decrypt" function to your program!

Title: [2/12/2012] Challenge #4 [easy]
Text: You're challenge for today is to create a random password generator!

For extra credit, allow the user to specify the amount of passwords to generate.

For even more extra credit, allow the user to specify the length of the strings he wants to generate!

Title: [2/12/2012] Challenge #5 [easy]
Text: Your challenge for today is to create a program which is password protected, and wont open unless the correct user and password is given.

For extra credit, have the user and password in a seperate .txt file.

for even more extra credit, break into your own program :)

Title: [2/14/2012] Challenge #6 [easy]
Text: You're challenge for today is to create a program that can calculate pi accurately to at least 30 decimal places.

Try not to cheat :)

Title: [2/15/2012] Challenge #7 [easy]
Text: Write a program that can translate Morse code in the format of ...---...

A space and a slash will be placed between words. ..- / --.-

For bonus, add the capability of going from a string to Morse code.

Super-bonus if your program can flash or beep the Morse.

This is your Morse to translate:

.... . .-.. .-.. --- / -.. .- .. .-.. -.-- / .--. .-. --- --. .-. .- -- -- . .-. / --. --- --- -.. / .-.. ..- -.-. -.- / --- -. / - .... . / -.-. .... .- .-.. .-.. . -. --. . ... / - --- -.. .- -.--

Title: [2/16/2012] Challenge #8 [easy]
Text: write a program that will print the song "99 bottles of beer on the wall".

for extra credit, do not allow the program to print each loop on a new line.

Title: [2/17/2012] Challenge #9 [easy]
Text: write a program that will allow the user to input digits, and arrange them in numerical order.

for extra credit, have it also arrange strings in alphabetical order

Title: [2/18/2012] Challenge #10 [easy]
Text: The exercise today asks you to validate a telephone number, as if written on an input form. Telephone numbers can be written as ten digits, or with dashes, spaces, or dots between the three segments, or with the area code parenthesized; both the area code and any white space between segments are optional.

Thus, all of the following are valid telephone numbers: 1234567890, 123-456-7890, 123.456.7890, (123)456-7890, (123) 456-7890 (note the white space following the area code), and 456-7890.

The following are not valid telephone numbers: 123-45-6789, 123:4567890, and 123/456-7890.

Title: [2/19/2012] Challenge #11 [easy]
Text: The program should take three arguments. The first will be a day, the second will be month, and the third will be year. Then, your program should compute the day of the week that date will fall on.

Title: [2/20/2012] Challenge #12 [easy]
Text: Write a small program that can take a string:

"hi!"

and print all the possible permutations of the string:

"hi!"

"ih!"

"!hi"

"h!i"

"i!h"

etc...

Title: [2/21/2012] Challenge #13 [easy]
Text: Find the number of the year for the given date. For example, january 1st would be 1, and december 31st is 365.

Title: [2/23/2012] Challenge #14 [easy]
Text: Input: list of elements and a block size k or some other variable of your choice

Output: return the list of elements with every block of k elements reversed, starting from the beginning of the list.

For instance, given the list 12, 24, 32, 44, 55, 66 and the block size 2, the result is 24, 12, 44, 32, 66, 55.

Title: [2/24/2012] Challenge #15 [easy]
Text: Write a program to left or right justify a text file

Title: [2/27/2012] Challenge #16 [easy]
Text: Hi folks! We are in the midst of discussing how this subreddit will go about but for now how about we just concentrate on challenges!

Write a function that takes two strings and removes from the first string any character that appears in the second string. For instance, if the first string is “Daily Programmer” and the second string is “aeiou ” the result is “DlyPrgrmmr”.
note: the second string has [space] so the space between "Daily Programmer" is removed

Title: [3/4/2012] Challenge #17 [easy]
Text: write an application which will print a triangle of stars of user-specified height, with each line having twice as many stars as the last.
sample output:

@

@@

@@@@

hint: in many languages, the "+" sign concatenates strings.

bonus features: print the triangle in reverse, print the triangle right justified

Title: [3/5/2012] Challenge #18 [easy]
Text: Often times in commercials, phone numbers contain letters so that they're easy to remember (e.g. 1-800-VERIZON). Write a program that will convert a phone number that contains letters into a phone number with only numbers and the appropriate dash. Click [here](http://en.wikipedia.org/wiki/Telephone_keypad) to learn more about the telephone keypad.

Example Execution:
Input: 1-800-COMCAST
Output: 1-800-266-2278

Title: [3/7/2012] Challenge #19 [easy]
Text: Challenge #19 will use [The Adventures of Sherlock Holmes](http://www.gutenberg.org/cache/epub/1661/pg1661.txt) from [Project Gutenberg](http://www.gutenberg.org).

Write a program that counts the number of alphanumeric characters there are in The Adventures of Sherlock Holmes. Exclude the Project Gutenberg header and footer, book title, story titles, and chapters. Post your code and the alphanumeric character count.

Title: [3/8/2012] Challenge #20 [easy]
Text: create a program that will find all prime numbers below 2000

Title: [3/9/2012] Challenge #21 [easy]
Text: Input: a number

Output : the next higher number that uses the same set of digits.

Title: [3/10/2012] Challenge #22 [easy]
Text: Write a program that will compare two lists, and append any elements in the second list that doesn't exist in the first.

input: ["a","b","c",1,4,], ["a", "x", 34, "4"]

output: ["a", "b", "c",1,4,"x",34, "4"]

Title: [3/13/2012] Challenge #23 [easy]
Text: Input: a list

Output: Return the two halves as different lists.

If the input list has an odd number, the middle item can go to any of the list.

Your task is to write the function that splits a list in two halves.

Title: [3/15/2012] Challenge #25 [easy]
Text: In an election, the person with the majority of the votes is the winner. Sometimes due to similar number of votes, there are no winners.

Your challenge is to write a program that determines the winner of a vote, or shows that there are no winners due to a lack of majority.

Title: [3/16/2012] Challenge #26 [easy]
Text: you have a string "ddaaiillyyprrooggrraammeerr". We want to remove all the consecutive duplicates and put them in a separate string, which yields two separate instances of the string "dailyprogramer".

use this list for testing:

input: "balloons"

expected output: "balons" "lo"

input: "ddaaiillyyprrooggrraammeerr"

expected output: "dailyprogramer" "dailyprogramer"

input: "aabbccddeded"

expected output: "abcdeded" "abcd"

input: "flabby aapples"

expected output: "flaby aples" "bap"

Title: [3/17/2012] Challenge #27 [easy]
Text: Write a program that accepts a year as input and outputs the century the year belongs in (e.g. 18th century's year ranges are 1701 to 1800) and whether or not the year is a leap year. Pseudocode for leap year can be found [here](http://en.wikipedia.org/wiki/Leap_year#Algorithm).

Sample run:

Enter Year: 1996

Century: 20

Leap Year: Yes

Enter Year: 1900

Century: 19

Leap Year: No

Title: [3/20/2012] Challenge #28 [easy]
Text: The array duplicates problem is when one integer is in an array for more than once.

If you are given an array with integers between 1 and 1,000,000 or in some other interval and one integer is in the array twice. How can you determine which one?

Your task is to write code to solve the challenge.

Note: try to find the most efficient way to solve this challenge.

Title: [3/22/2012] Challenge #29 [easy]
Text: A [Palindrome](http://en.wikipedia.com/wiki/Palindrome) is a sequence that is the same in reverse as it is forward.

I.e. hannah, 12321.

Your task is to write a function to determine whether a given string is palindromic or not.

Bonus: Support multiple lines in your function to validate Demetri Martin's [224 word palindrome poem](http://www.pastemagazine.com/articles/2009/02/demetri-martins-palindrome-poem.html).

Title: [3/26/2012] Challenge #30 [easy]
Text: Write a program that takes a list of integers and a target number and determines if any two integers in the list sum to the target number. If so, return the two numbers. If not, return an indication that no such integers exist.

Title: [3/27/2012] Challenge #31 [easy]
Text: Write a function that takes two base-26 numbers in which digits are represented by letters with A=0, B=1, … Z=25 and returns their product using the same notation. As an example, CSGHJ × CBA = FNEUZJA.

Your task is to write the base-26 multiplication function.

Try to be very efficient in your code!

Title: [3/28/2012] Challenge #32 [easy]
Text: lets simulate a roulette wheel!
a program that takes as input your bet, and gives as output how much you won, with the appropriate probability

write a program that will take a players bet and output the resulting spin and payout.
try using an american roulette wheel (which has the 00 slot) to add a slight twist. and try to incorporate as many complex bets as you can to. a comprehensive list can be found [here](http://en.wikipedia.org/wiki/Roulette#Bet_odds_table)

Title: [3/30/2012] Challenge #33 [easy]
Text: This would be a good study tool too. I made one myself and I thought it would also be a good challenge.

Write a program that prints a string from a list at random, expects input, checks for a right or wrong answer, and keeps doing it until the user types "exit". If given the right answer for the string printed, it will print another and continue on. If the answer is wrong, the correct answer is printed and the program continues.

Bonus: Instead of defining the values in the program, the questions/answers is in a file, formatted for easy parsing.

Example file:
12 * 12?,144
What is reddit?,website with cats
Translate: hola,hello

Title: [3/31/2012] Challenge #34 [easy]
Text: A very basic challenge:
In this challenge, the

input is are : 3 numbers as arguments

output: the sum of the squares of the two larger numbers.

Your task is to write the indicated challenge.

Title: [4/3/2012] Challenge #35 [easy]
Text: Write a program that will take a number and print a right triangle attempting to use all numbers from 1 to that number.

Sample Run:
Enter number: 10

Output:
7 8 9 10

4 5 6
2 3
1

Enter number: 6
Output:
4 5 6
2 3
1

Enter number: 3
Output:
2 3
1

Enter number: 12
Output:
7 8 9 10
4 5 6
2 3
1

Title: [4/5/2012] Challenge #36 [easy]
Text: 1000 Lockers Problem.

In an imaginary high school there exist 1000 lockers labelled 1, 2, ..., 1000. All of them are closed. 1000 students are to "toggle" a locker's state. * The first student toggles all of them * The second one toggles every other one (i.e, 2, 4, 6, ...) * The third one toggles the multiples of 3 (3, 6, 9, ...) and so on until all students have finished.

To toggle means to close the locker if it is open, and to open it if it's closed.

How many and which lockers are open in the end?

Title: [4/8/2012] Challenge #37 [easy]
Text: write a program that takes

input : a file as an argument

output: counts the total number of lines.

for bonus, also count the number of words in the file.

Title: [4/10/2012] Challenge #38 [easy]
Text: Implement [Dijkstra's algorithm](http://en.wikipedia.org/wiki/Dijkstra's_algorithm) in any way you can :)

Title: [4/12/2012] Challenge #39 [easy]
Text:

You are to write a function that displays the numbers from 1 to an input parameter n, one per line, except that if the current number is divisible by 3 the function should write “Fizz” instead of the number, if the current number is divisible by 5 the function should write “Buzz” instead of the number, and if the current number is divisible by both 3 and 5 the function should write “FizzBuzz” instead of the number.

For instance, if n is 20, the program should write 1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, FizzBuzz, 16, 17, Fizz, 19, and Buzz on twenty successive lines.

Title: [4/16/2012] Challenge #40 [easy]
Text: Print the numbers from 1 to 1000 without using any loop or conditional statements.

Don't just write the printf() or cout statement 1000 times.

Be creative and try to find the most efficient way!

Title: [4/19/2012] Challenge #41 [easy]
Text: Write a program that will accept a sentence as input and then output that sentence surrounded by some type of an ASCII decoratoin banner.

Sample run:
Enter a sentence: So long and thanks for all the fish

Output

```
****************************************
*                                      *
*  So long and thanks for all the fish  *
*                                      *
****************************************
```

Bonus: If the sentence is too long, move words to the next line.

Title: [4/23/2012] Challenge #42 [easy]
Text: Write a program that prints out the lyrics for "Ninety-nine bottles of beer", "Old McDonald had a farm" or "12 days of Christmas".

If you choose "Ninety-nine bottles of beer", you need to spell out the number, not just write the digits down. It's "Ninety-nine bottles of beer on the wall", not "99 bottles of beer"!

For Old McDonald, you need to include at least 6 animals: a cow, a chicken, a turkey, a kangaroo, a T-Rex and an animal of your choosing (Old McDonald has a weird farm). The cow goes "moo", the chicken goes "cluck", the turkey goes "gobble", the kangaroo goes "g'day mate" and the T-Rex goes "GAAAAARGH". You can have more animals if you like.

Make your code shorter than the song it prints out!

Title: [4/24/2012] Challenge #43 [easy]
Text: Today is a common interview question.

Given a binary tree t and two elements of the tree, m and n, with m < n, find the lowest element of the tree (farthest from the root) that is an ancestor of both m and n.

Title: [4/25/2012] Challenge #44 [easy]

Text: Write a program that divides up some input text into sentences and then determines which sentence in the input has the most words. Print out the sentence with the most words and the number of words that are in it. Optionally, also print out all words in that sentence that are longer than 4 characters.

Sentences can end in periods, exclamation points and question marks, but not colons or semi-colons.

If you need something to input, try Shylock's famous speech from Shakespeare's *The Merchant of Venice*:

>If it will feed nothing else, it will

feed my revenge. He hath disgrac'd me and hind'red me half a million; laugh'd at my losses, mock'd at my gains, scorned my nation, thwarted my bargains, cooled my friends, heated mine enemies. And what's his reason? I am a Jew. Hath not a Jew eyes? Hath not a Jew hands, organs, dimensions, senses, affections, passions, fed with the same food, hurt with the same weapons, subject to the same diseases, healed by the same means, warmed and cooled by the same winter and summer, as a Christian is? If you prick us, do we not bleed? If you tickle us, do we not laugh? If you poison us, do we not die? And if you wrong us, shall we not revenge? If we are like you in the rest, we will resemble you in that. If a Jew wrong a Christian, what is his humility? Revenge. If a Christian wrong a Jew, what should his sufferance be by Christian example? Why, revenge. The villainy you teach me I will execute; and it shall go hard but I will better the instruction.

Title: [4/27/2012] Challenge #45 [easy]

Text: Your challenge today is to write a program that can draw a checkered grid (like a chessboard) to any dimension. For instance, a 3 by 8 board might look like this:

```
********************************
* *###* *###* *###* *###*
* *###* *###* *###* *###*
* *###* *###* *###* *###*
********************************
*###* *###* *###* *###* *
*###* *###* *###* *###* *
*###* *###* *###* *###* *
********************************
* *###* *###* *###* *###*
* *###* *###* *###* *###*
* *###* *###* *###* *###*
********************************
```

Yours doesn't have to look like mine, you can make it look any way you want (now that I think of it, mine looks kinda bad, actually). Also try to make it scalable, so that if you want to make a 2 by 5 board, but with bigger squares, it would print out:

```
******************************
*  *#####*  *#####*  *
*  *#####*  *#####*  *
*  *#####*  *#####*  *
*  *#####*  *#####*  *
*  *#####*  *#####*  *
******************************
*#####*  *#####*  *#####*
*#####*  *#####*  *#####*
*#####*  *#####*  *#####*
*#####*  *#####*  *#####*
*#####*  *#####*  *#####*
******************************
```

Title: [4/30/2012] Challenge #46 [easy]
Text: The population count of a bitstring is the number of set bits (1-bits) in the string. For instance, the population count of the number 23, which is represented in binary as 10111 is 4.

Your task is to write a function that determines the population count of a number representing a bitstring

Title: [5/2/2012] Challenge #47 [easy]
Text: Your task today is to implement one of the oldest ciphers known, the so-called [Caesar cipher](http://en.wikipedia.org/wiki/Caesar_cipher) (or *Caesar shift*, as it is sometimes called). It works like this: for every letter you want to encrypt, you shift it some number of places down the alphabet to get the letter in the cipher.

So, for instance, in a Caesar cipher with a shift of 3, "A" becomes "D", "B" becomes "E", "C" becomes "F", and so on. At the end of the alphabet it wraps around, so "W" becomes "Z", "X" becomes "A", "Y" becomes "B" and "Z" becomes "C". If you encrypt "Hello" with a shift of 3, you get "Khoor".

One interesting thing about this cipher is that you can use the same algorithm to decode a cipher as you can to encode it: if you wish to decrypt some text that has been Caesar-shifted 6 places, you simply shift it another 20 places to get back the original text. For example, if you encrypt "Daily programmer" with a shift of 6 you get "Jgore vxumxgsskx", and if you encrypt "Jgore vxumxgsskx" with a shift of 20 you get "Daily programmer".

Implement the cipher and encrypt a bit of text of your choice!

Bonus: Using your program, become a code-cracker and decrypt this cipher (posted in honor of Mayday):

> Spzalu - zayhunl dvtlu sfpun pu wvukz kpzaypibapun zdvykz pz uv ihzpz mvy h
> zfzalt vm nvclyutlua. Zbwyltl leljbapcl wvdly klypclz myvt h thukhal myvt aol
> thzzlz, uva myvt zvtl mhyjpjhs hxbhapj jlyltvuf. Fvb jhu'a lewlja av dplsk
> zbwyltl leljbapcl wvdly qbza 'jhbzl zvtl dhalyf ahya aoyld h zdvyk ha fvb! P
> tlhu, pm P dlua hyvbuk zhfpu' P dhz hu ltwlylyvy qbza iljhbzl zvtl tvpzalulk
> ipua ohk sviilk h zjptpahy ha tl aolf'k wba tl hdhf!... Ho, huk uvd dl zll aol
> cpvslujl puolylua pu aol zfzalt! Jvtl zll aol cpvslujl puolylua pu aol zfzalt!
> Olsw! Olsw! P't ilpun ylwylzzlk!

Title: [5/4/2012] Challenge #48 [easy]
Text: Take an array of integers and partition it so that all the even integers in the array precede all the odd integers in the array. Your solution must take linear time in the size of the array and operate in-place with only a constant amount of extra space. Your task is to write the indicated function.

Title: [5/7/2012] Challenge #49 [easy]
Text: The [Monty Hall Problem](http://en.wikipedia.org/wiki/Monty_Hall_problem) is a probability brain teaser that has a rather unintuitive solution.

The gist of it, taken from Wikipedia:
>Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1 [but the door is not opened], and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice? (clarification: the host will always reveal a goat)

Your task is to write a function that will compare the strategies of *switching* and *not switching* over many random position iterations. Your program should output the proportion of successful choices by each strategy. Assume that if both unpicked doors contain goats the host will open one of those doors at random with equal probability.

If you want to, you can for simplicity's sake assume that the player picks the first door every time. The only aspect of this scenario that needs to vary is what is behind each door.

Title: [5/9/2012] Challenge #50 [easy]
Text: Hello everyone! As of today, we have finished our 50th challenge and it has been a pleasure giving out these challenges to you all. You have all been amazing with the solutions and seeing you all i hope i become a good programmer like you all one day :D

If i did any mistakes in challenges please forgive me and as you may have noticed we post once in two days or so to give you time to complete these. Really sorry if you wanted everyday posts .. but due to our busy lives, maybe sometime in future or maybe when i leave this subreddit, you may have that in the new management :) Thank You one and all ... As for now I have given today's two challenges are from [Google Code Jam Qualification Round Africa 2010](http://code.google.com/codejam/contest/dashboard?c=351101#s=p0)

Store Credit:

You receive a credit C at a local store and would like to buy two items. You first walk through the store and create a list L of all available items. From this list you would like to buy two items that add up to the entire value of the credit. The solution you provide will consist of the two integers indicating the positions of the items in your list (smaller number first).

For instance, with C=100 and L={5,75,25} the solution is 2,3; with C=200 and L={150,24,79,50,88,345,3} the solution is 1,4; and with C=8 and L={2,1,9,4,4,56,90,3} the solution is 4,5.

PROBLEM A IN THE LINK. PLEASE USE IT TO CLARIFY YOUR DOUBTS

Title: [5/11/2012] Challenge #51 [easy]
Text: Write a program that given an array A and a number N, generates all combinations of items in A of length N.

That is, if you are given the array [1,2,3,4,5] and 3, you're supposed to generate

* [1,2,3]
* [1,2,4]
* [1,2,5]
* [1,3,4]
* [1,3,5]
* [1,4,5]
* [2,3,4]
* [2,3,5]
* [2,4,5]
* [3,4,5]

Note that order doesn't matter when counting combinations, both [1,2,3] and [3,2,1] are considered the same. Order also doesn't matter in the output of the combinations, as long as you generate all of them, you don't have to worry about what order they pop out. You can also assume that every element of the array is distinct.

Title: [5/14/2012] Challenge #52 [easy]
Text: Imagine each letter and its position within the alphabet. Now assign each letter its corresponding value ie a=1, b=2,... z=26. When given a list of words, order the words by the sum of the values of the letters in their names.

Example: Shoe and Hat

Hat: 8+1+20 = 29

Shoe: 19+8+15+5 = 47

So the order would be Hat, Shoe.

For extra points, divide by the sum by the number of letters in that word and then rank them.

Title: [5/16/2012] Challenge #53 [easy]
Text: Write a function that given two sorted lists, returns a list whith the two lists merged together into one sorted list.

So, for instance, for inputs [1,5,7,8] and [2,3,4,7,9] it should return [1,2,3,4,5,7,7,8,9].

Try and make your code as efficient as possible.

Title: [5/19/2012] Challenge #54 [easy]
Text: A transposition cipher we'll call the "matrix cipher" can be defined as follows: write each character in the text that you want to encrypt in a matrix of some specified width, where the width is the key of the cipher. So, for instance, if you wanted to encrypt "The cake is a lie!" with the key 3, you would write it like so (the spaces are replaced with underscores for clarity):

```
T h e
_ c a
k e _
i s _
a _ l
i e !
```

Then to get the ciphertext, you simply read off the columns one by one. So in this case, the ciphertext would be "T_kiaihces_eea__l!", or "T kiaihces eea l!" if you put the spaces back in.

If the text doesn't fit the matrix perfectly, you add in random letters to fill in the last row. So, if we wanted to encode "The cake is a lie!" with key 7, we'd construct this matrix:

```
T h e _ c a k
e _ i s _ a _
l i e ! v m z
```

Here "v", "m" and "z" have been added in to fill the last row, and the ciphertext is "Telh ieie s!c vaamk z".

Write an implementation of the matrix cipher that can both encode and decode text given the correct key.

***

BONUS: One of the major tricks code-crackers have used throughout history is the fact that the first parts of many messages often follow a regular pattern. They start with "Hello" or "Greetings", "Transmission from" or something like that (Allied codebreakers during World War II took advantage of the fact that Nazi messages often began with "Heil Hitler").

Use this trick to construct a way to automatically crack messages encrypted with the matrix cipher. That is, given a certain ciphertext to crack and the first few characters of the cleartext, figure out what the entire message is without human input. Your code should just return the correct answer and optionally the key, but nothing else.

Try your code-cracker on this text, using the clue that the message starts with "It seems" (or "It_seems", if you use the underscore):

```
I_rso_wotTe,taef_h__hl__socaeihtemonraaheamd_svemsp_l_ems_ayiN___Anofeadt.yueo_o
h_..__leaA_.iaastnY.snw__do__d_nyeuhl_foor_eiaotushlvrr.'oapee.avnv_d__he,ey_gOf
___oiunrbpaunieeer_r_l_geos_ctoingoloyfq_rcam__ilainpotlimadufhjv_llt_emiw_aevsd
nrsdriengieysr_p_tc_,tlfteuc_uitwrrawavzo_irhlez_ftrelszloyyry_bir__e_huv_no_ead
eauuyvsbs_mtoe_l.rb_urat_eeh_y_pOsreg_fjnp,rocucee___otn_cpgbmujltaayprgiayr_uep
fb_btt,velyahe_s,eogeraq__ue__ncysr.hcdzoo__ar_duftTcioi'tahkmnarwxeeeegeae_r__j
```

As you can see, there's plenty of punctuation in this text, but there are no new-lines, it is just one chunk of text. And again, all spaces have been replaced with underscores for clarity, but you should remove those to make the cleartext readable. If you do solve it, please put four spaces before the cleartext if you post it here, to hide it for people who want to solve it themselves.

Title: [5/21/2012] Challenge #55 [easy]
Text: Write a program to solve the sliding window minimum problem using any of the methods possible. [This](http://home.tiac.net/~cri/2001/slidingmin.html) could be a helpful link.

Title: [5/23/2012] Challenge #56 [easy]
Text: The ABACABA sequence is defined as follows: start with the first letter of the alphabet ("a"). This is the first iteration. The second iteration, you take the second letter ("b") and surround it with all of the first iteration (just "a" in this case). Do this for each iteration, i.e. take two copies of the previous iteration and sandwich them around the next letter of the alphabet.

Here are the first 5 items in the sequence:

a
aba
abacaba
abacabadabacaba
abacabadabacabaeabacabadabacaba

And it goes on and on like that, until you get to the 26th iteration (i.e. the one that adds the "z"). If you use one byte for each character, the final iteration takes up just under 64 megabytes of space.

Write a computer program that prints the 26th iteration of this sequence to a file.

***
BONUS: try and limit the amount of memory your program needs to finish, while still getting a reasonably quick runtime. Find a good speed/memory tradeoff that keeps both memory usage low (around a megabyte, at most) and the runtime short (around a few seconds).

Title: [5/25/2012] Challenge #57 [easy]
Text: ~~Your task is to implement [Ackermann Function](http://en.wikipedia.org/wiki/Ackermann_function) in the most efficient way possible.~~

~~Please refer the wiki page link given for its explanation.~~

Since many did not like the previous challenge because it was quite unsatisfactory here is a new challenge ...

Input: A sequence of integers either +ve or -ve

Output : a part of the sequence in the list with the maximum sum.

Title: [5/28/2012] Challenge #58 [easy]
Text: As computer programmers are well aware, it can be very useful to write numbers using numerical bases other than the familiar base 10 notation we use in everyday life. In computer programming, [base 2](http://en.wikipedia.org/wiki/Binary_number) and [base 16](http://en.wikipedia.org/wiki/Hexadecimal) are especially handy. In base 2, the number 1234 becomes 10011010010 and in base 16 it becomes 4D2.

Because there are only 10 regular digits, when numbers are written in base 16, the first few letters of the alphabet are added to represent the remaining required digits, so 'A' stands in for 10, 'B' for 11, 'C' for 12, 'D' for 13, 'E' for 14 and 'F' for 15.

Of course, this trick of adding letters to stand in for numbers allows us to represent higher bases than 16; if you can use all letters of the alphabet, you can represent bases up to base 36 (because there are ten regular digits and 26 "letter-digits"). So for instance, 12345678 becomes 1L2FHE in base 23 and 4IDHAA in base 19.

Write a program that will take a number and convert it to any base between 2 and 36. Have the program print out 19959694 in base 35 and 376609378180550 in base 29.

NOTE: Many languages have this built in as a library function. For instance, in Java, the function Integer.toString(i, radix) does exactly this. However, the entire point of this challenge is to write the program yourself, so you **are not** allowed to use any library functions like this.

***

BONUS: A number is said to be "palindromic in base N" if, when written in base N the number is the same backwards and forwards. So, for instance, the number 16708 is palindromic in base 27, because in base 27 the number is written as MOM, obviously a palindrome. The number 12321 is a palindrome in in base 10, because 12321 written backwards is 12321. Some numbers are palindromic in several bases, the number 15167 for instance is palindromic in bases 9, 27, 28, 35 and 36.

In what bases is the number 10858 palindromic?

Title: [6/2/2012] Challenge #59 [easy]
Text: Write a program that given two strings, finds out if the second string is contained in the first, and if it is, where it is.

I.e. given the strings "Double, double, toil and trouble" and "il an" will return 18, because the second substring is embedded in the first, starting on position 18.

NOTE: Pretty much every language have this functionality built in for their strings, sometimes called find() (as in Python) or indexOf() (as in Java). But the point of this problem is to write the program yourself, so you **are not** allowed to use functions like this!

Title: [6/4/2012] Challenge #60 [easy]
Text: A polite number n is an integer that is the sum of two or more consecutive nonnegative integers in at least one way.

[Here](http://en.wikipedia.org/wiki/Polite_number) is an article helping in understanding Polite numbers

Your challenge is to write a function to determine the ways if a number is polite or not.

Title: [6/6/2012] Challenge #61 [easy]
Text: The number 19 is can be represented [in binary](http://en.wikipedia.org/wiki/Binary_numeral_system) as 10011. Lets define the operation of "rotating a number" as taking the last binary digit of that number and moving it so it becomes the first binary digit, and moving the other digits one step forward. I.e. if you rotate 10011, you get 11001 (i.e. 25), because the 1 that was in the last position has now moved to the first position. If you rotate it again, you get 11100 (i.e. 28).

If you rotate it again, something curious happens: you get 01110, which is the same as 1110 (i.e. 14) since leading zeroes don't count in a binary representation. That is to say, when you rotate it this time, the zero disappears. If you rotate it once more, you get 0111, which is the same as 111 (i.e. 7). Again, the zero has disappeared.

After that, the number remains the same regardless of how much you rotate it, since the binary number representation of 7 only has 1's in it.

This gives us a sequence of numbers. Starting with 19 and then rotating it step by step until we get a number with only 1's in the binary representation, we get

19 -> 25 -> 28 -> 14 -> 7

Lets call this a "binary rotation sequence" for 19. Here are the binary rotation sequences for the numbers 69, 205 and 357, with the numbers written first in decimal and then in binary to show what is going on:

69 -> 98 -> 49 -> 56 -> 28 -> 14 -> 7
1000101 -> 1100010 -> 110001 -> 111000 -> 11100 -> 1110 -> 111

205 -> 230 -> 115 -> 121 -> 124 -> 62 -> 31
11001101 -> 11100110 -> 1110011 -> 1111001 -> 1111100 -> 111110 -> 11111

357 -> 434 -> 217 -> 236 -> 118 -> 59 -> 61 -> 62 -> 31
101100101 -> 110110010 -> 11011001 -> 11101100 -> 1110110 -> 111011 -> 111101 -> 111110 -> 11111

Write a program that given a number will print out the binary rotation sequence for that number (you only need to print out the sequence in decimal).

What is the binary rotation sequence for 54321?

Title: [6/8/2012] Challenge #62 [easy]
Text: Give the [Ullman's Puzzle](http://regator.com/p/246306389/ullmans_puzzle/)

Write a function that makes that determination

Title: [6/11/2012] Challenge #63 [easy]
Text: Write a procedure called reverse(N, A), where N is an integer and A is an array which reverses the N first items in the array and leaves the rest intact.

For instance, if N = 3 and A = [1,2,3,4,5], then reverse(N,A) will modify A so that it becomes [3,2,1,4,5], because the three first items, [1,2,3], have been reversed. Here are a few other examples:

reverse(1, [1, 2, 3, 4, 5]) -> A = [1, 2, 3, 4, 5]
reverse(2, [1, 2, 3, 4, 5]) -> A = [2, 1, 3, 4, 5]
reverse(5, [1, 2, 3, 4, 5]) -> A = [5, 4, 3, 2, 1]
reverse(3, [51, 41, 12, 62, 74]) -> A = [12, 41, 51, 62, 74]

So if N is equal to 0 or 1, A remains unchanged, and if N is equal to the size of A, all of A gets flipped.

Try to write reverse() so that it works *in-place*; that is, it uses only a constant amount of memory in addition to the list A itself. This isn't necessary, but it is recommended.

Title: [6/13/2012] Challenge #64 [easy]
Text: The divisors of a number are those numbers that divide it evenly; for example, the divisors of 60 are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, and 60. The sum of the divisors of 60 is 168, and the number of divisors of 60 is 12.

The totatives of a number are those numbers less than the given number and coprime to it; two numbers are coprime if they have no common factors other than 1. The number of totatives of a given number is called its totient. For example, the totatives of 30 are 1, 7, 11, 13, 17, 19, 23, and 29, and the totient of 30 is 8.

Your task is to write a small library of five functions that compute the divisors of a number, the sum and number of its divisors, the totatives of a number, and its totient.

Title: [6/15/2012] Challenge #65 [easy]

Text: Write a program that given a floating point number, gives the number of American dollar coins and bills needed to represent that number (rounded to the nearest 1/100, i.e. the nearest penny). For instance, if the float is 12.33, the result would be 1 ten-dollar bill, 2 one-dollar bills, 1 quarter, 1 nickel and 3 pennies.

For the purposes of this problem, these are the different denominations of the currency and their values:

* Penny: 1 cent
* Nickel: 5 cent
* Dime: 10 cent
* Quarter: 25 cent
* One-dollar bill
* Five-dollar bill
* Ten-dollar bill
* Fifty-dollar bill
* Hundred-dollar bill

Sorry Thomas Jefferson, JFK and Sacagawea, but no two-dollar bills, half-dollars or dollar coins!

Your program can return the result in whatever format it wants, but I recommend just returning a list giving the number each coin or bill needed to make up the change. So, for instance, 12.33 could return [0,0,1,0,2,1,0,1,3] (here the denominations are ordered from most valuable, the hundred-dollar bill, to least valuable, the penny)

Title: [6/18/2012] Challenge #66 [easy]

Text: Write a function that takes two arguments, x and y, which are two strings containing Roman Numerals without prefix subtraction (so for instance, 14 is represented as XIIII, not XIV). The function must return true if and only if the number represented by x is less than the number represented by y. Do it without actually converting the Roman numerals into regular numbers.

Challenge: handle prefix subtraction as well.

Title: [6/20/2012] Challenge #67 [easy]

Text: As we all know, when computers do calculations or store numbers, they don't use decimal notation like we do, they use binary notation. So for instance, when a computer stores the number 13, it doesn't store "1" and "3", it stores "1101", which is 13 in binary.

But more than that, when we instruct it to store an integer, we usually tell it to store it in a certain datatype with a certain length. For (relatively small) integers, that length is usually as 32 bits, or four bytes (also called "one word" on 32-bit processors). So 13 isn't really stored as "1101", it's stored as "00000000000000000000000000001101".

If we were to reverse that bit pattern, we would get "10110000000000000000000000000000", which written in decimal becomes "2952790016".

Write a program that can do this "32-bit reverse" operation, so when given the number 13, it will return 2952790016.

Note: just to be clear, for all numbers in this problem, we are using *unsigned* 32 bit integers.

Title: [6/22/2012] Challenge #68 [easy]

Text: [Emirp](http://mathworld.wolfram.com/Emirp.html) is an interesting concept. The explanation about it is provided in the link i just gave.

Your task is to implement a function which prints out the emirps below a number(input) given by the user.

Title: [6/26/2012] Challenge #69 [easy]
Text: Write a program that takes a title and a list as input and outputs the list in a nice column. Try to make it so the title is centered. For example:

```
title: 'Necessities'
input: ['fairy', 'cakes', 'happy', 'fish', 'disgustipated', 'melon-balls']

output:

  +---------------+
  |  Necessities  |
  +---------------+
  | fairy         |
  | cakes         |
  | happy         |
  | fish          |
  | disgustipated |
  | melon-balls   |
  +---------------+
```

Bonus: amend the program so that it can output a two-dimensional table instead of a list. For example, a list of websites:

```
titles: ['Name', 'Address', 'Description']
input:  [['Reddit', 'www.reddit.com', 'the frontpage of the internet'],
         ['Wikipedia', 'en.wikipedia.net', 'The Free Encyclopedia'],
         ['xkcd', 'xkcd.com', 'Sudo make me a sandwich.']]

output:

  +-----------+-----------------+-----------------------------+
  |   Name    |     Address     |         Description         |
  +-----------+-----------------+-----------------------------+
  | Reddit    | www.reddit.com  | the frontpage of the internet |
  +-----------+-----------------+-----------------------------+
  | Wikipedia | en.wikipedia.net | The Free Encyclopedia       |
  +-----------+-----------------+-----------------------------+
  | xkcd      | xkcd.com        | Sudo make me a sandwich     |
  +-----------+-----------------+-----------------------------+
```

Title: [6/29/2012] Challenge #70 [easy]
Text: Write a program that takes a filename and a parameter n and prints the n most common words in the file, and the count of their occurrences, in descending order.

Title: [7/2/2012] Challenge #71 [easy]
Text: If a right angled triangle has three sides A, B and C (where C is the hypothenuse), the pythagorean theorem tells us that A^2 + B^2 = C^2

When A, B and C are all integers, we say that they are a *pythagorean triple*. For instance, (3, 4, 5) is a pythagorean triple because 3^2 + 4^2 = 5^2 .

When A + B + C is equal to 240, there are four possible pythagorean triples: (15, 112, 113), (40, 96, 104), (48, 90, 102) and (60, 80, 100).

Write a program that finds all pythagorean triples where A + B + C = 504.

Title: [7/4/2012] Challenge #72 [easy]
Text: The one-dimensional simple cellular automata [Rule 110] (http://en.wikipedia.org/wiki/Rule_110)
is the only such cellular automata currently known to be turing-complete, and many people say it is the simplest known turing-complete
system.

Implement a program capable of outputting an ascii-art representation of applying Rule 110 to some initial state. How many iterations and what your initial state is is up to you!

You may chose to implement rule 124 instead if you like (which is the same thing, albeit backwards).

Bonus points if your program can take an arbitrary rule integer from 0-255 as input and run that rule instead!

Title: [7/6/2012] Challenge #73 [easy]
Text: During the 70s and 80s, some handheld calculators used a very different notation for arithmetic called [Reverse Polish notation](http://en.wikipedia.org/wiki/Reverse_Polish_notation) (RPN). Instead of putting operators (`+`, `*`, `-`, etc.) between their operands (as in `3 + 4`), they were placed behind them: to calculate `3 + 4`, you first inputted the operands (`3 4`) and then added them together by pressing `+`.

Internally, this was implemented using a stack: whenever you enter a number, it's pushed onto the stack, and whenever you enter an operator, the **top two elements** are popped off for the calculation. Here's an example of a RPN calculator calculating `3 4 * 6 2 - +`:

```
[3] --> 3
[4] --> 3 4
[*] --> 12      ( 3 * 4 = 12)
[6] --> 12 6
[2] --> 12 6 2
[-] --> 12 4    ( 6 - 2 =  4)
[+] --> 16      (12 + 4 = 16)
```

Your task is to implement a program that reads a string in Reverse Polish notation and prints the result of the calculation. Your program should support positive and negative integers and the operators `+`, `-`, `*`. (For extra credit, you can implement extra functions, such as decimal numbers, division, exponentiation, etc.)

Title: [7/9/2012] Challenge #74 [easy]
Text: The Fibonacci numbers, which we are all familiar with, start like this:

0,1,1,2,3,5,8,13,21,34,...

Where each new number in the sequence is the sum of the previous two.

It turns out that by summing different Fibonacci numbers with each other, you can create every single positive integer. In fact, a much stronger statement holds:

Every single positive integer can be represented in one *and only one* way as a sum of *non-consecutive* Fibonacci numbers. This is called the number's ["Zeckendorf representation"](http://en.wikipedia.org/wiki/Zeckendorf%27s_theorem).

For instance, the Zeckendorf representation of the number 100 is 89 + 8 + 3, and the Zeckendorf representation of 1234 is 987 + 233 + 13 + 1. Note that all these numbers are Fibonacci numbers, and that they are non-consecutive (i.e. no two numbers in a Zeckendorf representation can be next to each other in the Fibonacci sequence).

There are other ways of summing Fibonacci numbers to get these numbers. For instance, 100 is also equal to 89 + 5 + 3 + 2 + 1, but 1, 2, 3, 5 are all consecutive Fibonacci numbers. If no consecutive Fibonacci numbers are allowed, the representation is unique.

Finding the Zeckendorf representation is actually not very hard. Lets use the number 100 as an example of how it's done:

First, you find the largest fibonacci number less than or equal to 100. In this case that is 89. This number will always be of the representation, so we remember that number and proceed recursively, and figure out the representation of 100 - 89 = 11.

The largest Fibonacci number less than or equal to 11 is 8. We remember that number and proceed recursively with 11 - 8 = 3.

3 is a Fibonacci number itself, so now we're done. The answer is 89 + 8 + 3.

Write a program that finds the Zeckendorf representation of different numbers.

What is the Zeckendorf representation of 3^15 ?

Title: [7/12/2012] Challenge #75 [easy] (Function Transformation)
Text: First off, I'd like to apologize for posting this 12 hours late, I'm a little new to my mod responsibilities. However, with your forgiveness, we can go onward!

Everyone on this subreddit is probably somewhat familiar with the C programming language.
Today, all of our challenges are C themed! Don't worry, that doesn't mean that you have to solve the challenge in C, you can use whatever language you want.

You are going to write a home-work helper tool for high-school students who are learning C for the first time. These students are in the advanced placement math course,
but do not know anything about programming or formal languages of any kind. However, they do know about functions and variables!

They have been given an 'input guide' that tells them to write simple pure mathematical functions like they are used to from their homework with a simple subset grammar, like this:

```
f(x)=x*x
big(x,y)=sqrt(x+y)*10
```

They are allowed to use
sqrt,abs,sin,cos,tan,exp,log, and the mathematical arithmetic operators +*/-, they can name their functions and variables any lower-case alphanumeric name and functions can have between 0 and 15 arguments.

In the this challenge, your job is to write a program that can take in their "simple format" mathematical function and output the correct C syntax for that function. All arguments should
be single precision, and all functions will only return one float.

As an example, the input

```
L0(x,y)=abs(x)+abs(y)
```

should output

```
float L0(float x,float y)
{
   return fabsf(x)+fabsf(y);
}
```

Bonus points if you support exponentiation with "\^", as in "f(x)=x\^2"

Title: [7/13/2012] Challenge #76 [easy] (Title case)
Text: Write a function that transforms a string into [title case](http://en.wikipedia.org/wiki/Letter_case#Headings_and_publication_titles). This mostly means: capitalizing only every first letter of every word in the string. However, there are some non-obvious exceptions to title case which can't easily be hard-coded. Your function must accept, as a second argument, a set or list of words that should **not** be capitalized. Furthermore, the **first word** of every title should always have a capital leter. For example:

```
exceptions = ['jumps', 'the', 'over']
titlecase('the quick brown fox jumps over the lazy dog', exceptions)
```

This should return:

```
The Quick Brown Fox jumps over the Lazy Dog
```

An example from the Wikipedia page:

```
exceptions = ['are', 'is', 'in', 'your', 'my']
titlecase('THE vitamins ARE IN my fresh CALIFORNIA raisins', exceptions)
```

Returns:

```
The Vitamins are in my Fresh California Raisins
```

Title: [7/16/2012] Challenge #77 [easy] (Enumerating Morse code sequences)
Text: Morse code, as we are all aware, consists of dots and dashes. Lets define a "Morse code sequence" as simply a series of dots and dashes (and nothing else). So ".--.-.--" would be a morse code sequence, for instance.

Dashes obviously take longer to transmit, that's what makes them dashes. Lets say that a dot takes 1 unit of time to transmit, and a dash takes 2 units of time. Then we can say that the "size" of a certain morse code sequence is the sum of the time it takes to transmit the dots and dashes. So, for instance "..-." would have a size of 5 (since there's three dots taking three units of time and one dash taking two units of time, for a total of 5). The sequence "-.-" would also have a size of 5.

In fact, if you list all the the possible Morse code sequences of size 5, you get:

```
..... ...- ..-. .-.. -... .-- -.- --.
```

A total of 8 different sequences.

Your task is to write a function called Morse(X) which generates all morse code sequences of size X and returns them as an array of strings (so Morse(5) should return the 8 strings I just mentioned, in some order).

Use your function to generate and print out all sequences of size 10.

***
Bonus: Try and write your code so that it can generate Morse(35) (or even Morse(36) or higher, but that takes a significant amount of memory) in a "reasonable" amount of time. "Reasonable" obviously depend on what computer and programming language you are using, but a good rule of thumb should be that it should finish in less than a minute.

Title: [7/18/2012] Challenge #78 [easy] (Keyboard Locale Simulator)
Text: This one is inspired by an actual problem my friend had to deal with recently. Unfortunately, its a little bit keyboard-locale specific, so if you don't happen to use a us-EN layout keyboard you might want to get a picture of one.

The en-us keyboard layout [pictured here](http://en.wikipedia.org/wiki/File:KB_United_States-NoAltGr.svg) is one common layout for keys. There are character-generating keys such as '1' and 'q', as well as modifier keys like 'ctrl' and 'shift', and 'caps-lock'

If one were to press every one of the character-generating keys in order from top to bottom left-to-right, you would get the following string:

    `1234567890-=qwertyuiop[]\asdfghjkl;'zxcvbnm,./

plus the whitespace characters TAB,RETURN,SPACE.

Your job is to write a function that takes in a character representing a keypress, as well as a boolean for each 'modifier' key like ctrl,alt,shift,and caps lock, and converts it properly into the ascii character for which the key gets output.

For example, my python implementation keytochar(key='a',caps=True) returns 'A'. However, keytochar(key='a',caps=True,shift=True) returns 'a'.

BONUS:
Read in a string containing a record of keypresses and output them to the correct string. A status key change is indicated by a ^ character..if a ^ character is detected, then the next character is either an 's' or 'S' for shift pressed or shift released, respectively, a 'c' or 'C' for caps on or caps off respectively, and a 't' 'T' for control down or up, and 'a' 'A' for alt down or up.

For example on the bonus, given the input

    ^sm^Sy e-mail address ^s9^Sto send the ^s444^S to^s0^S is ^cfake^s2^Sgmail.com^C

you should output

    My e-mail address (to send the $$$ to) is FAKE@GMAIL.COM

Title: [7/18/2012] Challenge #79 [easy] (Counting in steps)
Text: Write a function `step_count(a, b, steps)` that returns a list or array containing `steps` elements, counting from `a` to `b` in steps of an equal size. `steps` is a positive integer greater than or equal to 2, `a` and `b` are floating point numbers.

For example:
```
step_count(18.75, -22.00, 5)
==> [18.75, 8.5625, -1.625, -11.8125, -22.0]

step_count(-5.75, 12.00, 5)
==> [-5.75, -1.3125, 3.125, 7.5625, 12.0]

step_count(13.50, -20.75, 3)
==> [13.5, -3.625, -20.75]

step_count(9.75, 3.00, 9)
==> [9.75, 8.90625, 8.0625, 7.21875, 6.375, 5.53125, 4.6875, 3.84375, 3.0]
```

Title: [7/23/2012] Challenge #80 [easy] (Anagrams)
Text: As all of us who have read "Harry Potter and the Chamber of Secrets" knows, the reason He-Who-Must-Not-Be-Named chose his creepy moniker is that "I Am Lord Voldemort" is an anagram for his birthname, "Tom Marvolo Riddle".

I've never been good at these kinds of word-games (like anagrams), I always find it hard to figure out that stuff manually. I find it much more enjoyable to write computer programs to solve these problems for me. In the spirit of that, today's problem is to find simple one-word anagrams for other words.

Write a program that given a word will find all one-word anagrams for that word. So, for instance, if you put in "LEPROUS", it should return "PELORUS" and "SPORULE". As a dictionary, use [this file](http://code.google.com/p/dotnetperls-

controls/downloads/detail?name=enable1.txt), which is a 1.8 mb text-file with one word listed on each line, each word listed in lower-case. In this problem description, I've used upper-case for all words and their anagrams, but that is entirely optional, it's perfectly all right to use lower-case if you want to.

Using your program, find all the one-word anagrams for "TRIANGLE".

***

(by the way, in case anyone is curious: a "PELORUS" is "a sighting device on a ship for taking the relative bearings of a distant object", which I imagine basically is a telescope bolted onto a compass, and a "SPORULE" is "a small spore")

***

Bonus: if you looked up the anagrams for "PAGERS", you'd find that there was actually quite a few of them: "GAPERS", "GASPER", "GRAPES", "PARGES" and "SPARGE". Those five words plus "PAGERS" make a six-word "anagram family".

Here's another example of an anagram family, this time with five words: "AMBLERS", "BLAMERS", "LAMBERS", "MARBLES" and "RAMBLES".

What is the largest anagram family in the dictionary I supplied? What is the second largest?

Title: [7/25/2012] Challenge #81 [easy] (Numerical Calculus I)
Text: For a lot of the questions today we are going to be doing some simple numerical calculus. Don't worry, its not too terrifying.

For the easy problem, write a function that can take in a list of y-values that represents a function sampled on some domain. The domain can be specified as a list of x-values or two values for the x-minimum and x-maximum (the x-coordinates of the endpoints)

This function should output another list of values that represents the derivative of that function over the same domain.

Python example:

```
print derivative(xmin=-1,xmax=1,y=[-1.0,-.5,0,.5,1.0])
```

outputs:

```
[1.0,1.0,1.0,1.0,1.0]
```

Bonus 1) Write the same function but for the indefinite integral.

Bonus 2) This is sort-of an alternate version of the problem... if your language supports first-order functions (that is, functions as data), then write a function that takes a function A as an argument and returns a function A'.
When A' is evaluated at x, it computes an approximation of the derivative at x

EDIT: devil's assassin gave a decently good explaination of the problem, I'm stealing it here and modifying it a bit.

>for those of you who don't know, the derivative can be defined as the slope of the tangent line at a particular point. I'm assuming he wants a numerical derivative, making this a programming exercise and not a math one. We need to interpolate those values into a derivative. If you have some set of numbers N = {a1,a2,a3,a4,...,an} and some domain set S={b1,b2,...,bn} you can find the slope between two points on this line. Now this is a /bad/ approximation, but it can be made better through limiting the step size.

>Basically, here is the formula:

>f'(x) = lim h->0(f(x+h) - f(x))/h

>the "lim" part, means that this only works when h is REALLY small (pretty much 0, but without being exactly 0 so there is no dividing by 0). So you can modify this:

>f'(x) ~= f(x+h)-f(x)/h

Basically, what you do here is use compute the slope between the current point and the next point for each point. Use the slope equation from two points.

Title: [7/27/2012] Challenge #82 [easy] (Substring list)
Text: Write a function that takes a number *n* as an argument and returns (or outputs) every possible unique substrings (not counting "") of the first *n* letters of the alphabet (in any order you like). For example, `substrings(5)` could be:

a
ab
abc
abcd
abcde
b
bc
bcd
bcde
c
cd
cde
d
de
e

BONUS 1: Find a way to quickly determine how many strings this function returns for a given input. If the alphabet were 500 letters long, how much possible substrings would it have?

BONUS 2: Modify your function to take a string as an argument. Make sure all substrings in your output are still unique (i.e., there are two "l" substrings in "hello", but you should output only one).

Title: [7/30/2012] Challenge #83 [easy] (Long scale and short scale)
Text: One of the most annoying and confusing differences between English and basically every other language in the world is that English uses a weird way to name very large numbers.

For instance, if you wanted to translate "one trillion" from English to French, you might think it would be "un trillion", but that is wrong. The correct translation of "one trillion" to French is "un billion". Well, then, you might ask, what is "one billion" in French? It is, in fact, "un milliard". And "un trillion" in French is equal to english "one quintillion". Confusing, no?

The reason for this is that there are two so-called scales for large numbers, [the long scale and the short scale](http://en.wikipedia.org/wiki/Long_and_short_scales). English uses the short scale, almost everyone else uses the long scale (though the Arabic languages also apparently use the short scale). The two systems can be summarized as follows:

* In the *short scale*, you get a "new word" for the numbers every time the number increases by a factor of 1,000. So "a thousand millions" is "one billion" and "a thousand billions" is "one trillion".

* In the *long scale*, you get a "new word" for the numbers every time the number increases by a factor of 1,000,000. So "a million millions" is the same as "one billion" and "a million billions" is the same as "one trillion". If we just increase by a factor of 1,000, the "-on" ending on the word is replaced by "-ard". So "a thousand millions" is "one milliard", "a thousand billions" is "one billiard".

Here's a table summarizing the words for different numbers:

Actual number	Short scale name	Long scale name
10^6	million	million
10^9	billion	milliard
10^12	trillion	billion
10^15	quadrillion	billiard
10^18	quintillion	trillion
10^21	sextillion	trilliard

And it goes on like that.

Your task today is to write a program that will print out the name of a number in both long-scale and short-scale. So, for instance, given 1234567891111, it should print out

Short scale: 1 trillion, 234 billion, 567 million, 891 thousand and 111
Long scale: 1 billion, 234 milliard, 567 million, 891 thousand and 111

Bonus points if it prints out everything in letters, i.e.:
Short scale: one trillion, two hundred and thirty-four billion, five hundred and sixty-seven million, eight hundred and ninety-one thousand and one hundred and eleven

Long scale: one billion, two hundred and thirty-four milliard, five hundred and sixty-seven million, eight hundred and ninety-one thousand and one hundred and eleven

The program should be able to handle all numbers that can fit into an unsigned 64-bit integers, i.e. all numbers up to 2^64 - 1 ("18 trillion, 446 billiard, 744 billion, 73 milliard, 709 million, 551 thousand and 615" in long scale, though it's something completely different in short scale.), or 2^63 - 1 if you're using signed 64-bit integers. Of course, you can write your program so it can handle bigger numbers if you want, but it's not necessary.

Title: [8/1/2012] Challenge #84 [easy] (Searching Text Adventure)

Text: Like many people who program, I got started doing this because I wanted to learn how to make video games.

As a result, my first ever 'project' was also my first video game. It involved a simple text adventure I called "The adventure of the barren moor"

In "The adventure of the barren moor" the player is in the middle of an infinite grey swamp. This grey swamp has few distinguishing characteristics, other
than the fact that it is large and infinite and dreary. However, the player DOES have a magic compass that tells the player how far away the next feature of interest is.

The player can go north,south,east,or west. In my original version of the game, there was only one feature of interest, a treasure chest at a random point in the world.

Here is an example playthrough of my old program:

You awaken to find yourself in a barren moor. Try "look"
> look
Grey foggy clouds float oppressively close to you,
reflected in the murky grey water which reaches up your shins.
Some black plants barely poke out of the shallow water.
Try "north","south","east",or "west"
You notice a small watch-like device in your left hand.
It has hands like a watch, but the hands don't seem to tell time.

The dial reads '5m'

>north
The dial reads '4.472m'
>north
The dial reads '4.123m'
>n
The dial reads '4m'
>n
The dial reads '4.123m'
>south
The dial reads '4m'
>e
The dial reads '3m'
>e
The dial reads '2m'
>e
The dial reads '1m'
>e

You see a box sitting on the plain. Its filled with treasure! You win! The end.

The dial reads '0m'

Obviously, you do not have to use my flavor text, or my feature points. As a matter of fact, its probably more interesting if you don't!

Title: [8/3/2012] Challenge #85 [easy] (Row/column sorting)
Text: Write a program that reads a matrix of numbers separated by newlines and whitespace, like this:

```
10 5 4 20
9 33 27 16
11 6 55 3
```

then calculates the sums for each row and column, optionally outputting them...

```
Rows: 39 85 75
Columns: 30 44 86 39
```

then prints two new matrices:

* first, print the matrix with its rows sorted by their sums
* then, print the matrix with its columns sorted by their sums.

Like this:

```
10 5 4 20
11 6 55 3
9 33 27 16

10 20 5 4
9 16 33 27
11 3 6 55
```

Here's a large input matrix to test your program on.

```
5 58 88 60 11 23 97 48 59 82 95 24 6 67 47
45 14 36 99 16 70 77 18 43 39 97 54 11 53 98
```

85 14 96 66 34 86 95 49 4 49 72 76 45 49 37
72 88 20 56 37 16 20 97 71 11 91 33 90 5 96
15 53 54 95 61 93 75 95 51 83 71 70 2 57 83
54 29 56 80 79 93 40 55 40 14 63 94 77 12 90
96 97 3 47 2 43 12 2 82 92 1 99 90 13 35
24 19 54 96 82 96 10 40 62 30 35 16 70 83 64
59 81 8 84 14 46 32 45 41 35 98 66 87 51 49
13 49 12 51 34 82 36 77 88 14 84 41 66 18 56
6 68 82 63 77 72 67 36 85 53 66 70 21 86 80
40 51 87 5 78 56 99 44 39 48 78 56 19 55 40
5 94 62 46 85 73 24 67 95 63 42 95 43 53 4
14 99 7 36 25 65 22 71 20 80 16 10 71 97 23
99 77 85 53 13 32 37 19 61 32 45 62 25 18 32
98 79 35 17 26 96 22 3 76 20 81 9 40 95 72
18 39 55 99 96 63 90 78 77 81 2 99 68 6 84
53 27 68 43 48 29 27 14 50 29 53 65 5 56 46
94 36 17 64 2 93 5 95 47 78 90 3 85 26 32
46 62 70 63 81 6 86 51 44 96 47 83 33 28 28

For bonus points, format your output matrices nicely (align the columns, draw boxes with `-` and `|`...)

Title: [8/8/2012] Challenge #86 [easy] (run-length encoding)
Text: Run-Length encoding is a simple form of compression that detects 'runs' of repeated instances of a symbol in a string and compresses them to a list of pairs of 'symbol' 'length'. For example, the string

"Heeeeellllloooooo nurse!"

Could be compressed using run-length encoding to the list of pairs
[(1,'H'),(5,'e'),(5,'l'),(5,'o'),(1,'n'),(1,'u'),(1,'r'),(1,'s'),(1,'e')]

Which seems to not be compressed, but if you represent it as an array of 18bytes (each pair is 2 bytes), then we save 5 bytes of space compressing this string.

Write a function that takes in a string and returns a run-length-encoding of that string. (either as a list of pairs or as a 2-byte-per pair array)

BONUS: Write a decompression function that takes in the RLE representation and returns the original string

Title: [8/10/2012] Challenge #87 [easy] (Rectangle intersection)
Text: Write a function that calculates the [intersection](http://en.wikipedia.org/wiki/Intersection_(set_theory\)) of two rectangles, returning either a new rectangle or some kind of null value.

You're free to represent these rectangles in any way you want: tuples of numbers, class objects, new datatypes, anything goes. For this challenge, you'll probably want to represent your rectangles as the *x* and *y* values of the top-left and bottom-right points. (`Rect(3, 3, 10, 10)` would be a rectangle from `(3, 3)` (top-left) to `(10, 10)` (bottom-right).)

As an example, `rectIntersection(Rect(3, 3, 10 10), Rect(6, 6, 12, 12))` would return `Rect(6, 6, 10, 10)`, while `rectIntersection(Rect(4, 4, 5, 5), Rect(6, 6, 10 10))` would return null.

Title: [8/13/2012] Challenge #88 [easy] (Vigenère cipher)
Text: The easy challenge today is to implement the famous [Vigenère cipher](http://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher). The Wikipedia article explains well how it works, but here's a short description anyway:

You take a message that you want to encrypt, for instance "THECAKEISALIE" (lets assume that all characters are upper-case and there are no spaces in the messages, for the sake of simplicity), and a key you want to encrypt it with, for instance "GLADOS". You then write the message with the key repeated over it, like this:

```
GLADOSGLADOSG
THECAKEISALIE
```

The key is repeated as often as is needed to cover the entire message.

Now, one by one, each letter of the key is "added" to the letter of the clear-text to produce the cipher-text. That is, if A = 0, B = 1, C = 2, etc, then E + G = K (because E = 4 and G = 6, and 4 + 6 = 10, and K = 10). If the sum is larger than 25 (i.e. larger than Z), it starts from the beginning, so S + K = C (i.e. 18 + 10 = 28, and 28 - 26 is equal to 2, which is C).

For a full chart of how characters combine to form new characters, see
[here](http://en.wikipedia.org/wiki/File:Vigen%C3%A8re_square_shading.svg)

The cipher text then becomes:

```
GLADOSGLADOSG
THECAKEISALIE
-------------
ZSEFOCKTSDZAK
```

Write funtions to both encrypt and decrypt messages given the right key.

As an optional bonus, decrypt the following message, which has been encrypted with a word that I've used in this post:

```
HSULAREFOTXNMYNJOUZWYILGPRYZQVBBZABLBWHMFGWFVPMYWAVVTYISCIZRLVGOPGBRAKLUGJUZGLN
BASTUQAGAVDZIGZFFWVLZSAZRGPVXUCUZBYLRXZSAZRYIHMIMTOJBZFZDEYMFPMAGSMUGBHUVYTSABB
AISKXVUCAQABLDETIFGICRVWEWHSWECBVJMQGPRIBYYMBSAPOFRIMOLBUXFIIMAGCEOFWOXHAKUZISY
MAHUOKSWOVGBULIBPICYNBBXJXSIXRANNBTVGSNKR
```

Title: [8/20/2012] Challenge #89 [easy] (Simple statistical functions)
Text: For today's challenge, you should calculate some simple statistical values based on a list of values. Given [this data set](http://pastebin.com/vqzTrVK2), write functions that will calculate:

* [The mean value](http://en.wikipedia.org/wiki/Arithmetic_mean)
* [The variance](http://en.wikipedia.org/wiki/Variance)
* [The standard deviation](http://en.wikipedia.org/wiki/Standard_deviation)

Obviously, many programming languages and environments have standard functions for these (this problem is one of the few that is *really* easy to solve in Excel!), but you are not allowed to use those! The point of this problem is to write the functions yourself.

Title: [8/22/2012] Challenge #90 [easy] (Walkaround Rasterizer)
Text: In this challenge, we propose a simple image file format for binary (2 color) black-and-white images.
Rather than describing the image as a sequence of bits in a row, instead we describe it in a little bit of a non-standard way.

Imagine a grid of white squares. On this grid, a single man carrying a large black stamp stands on the square at 0,0. You can tell him 5 commands: walk N,S,E,W, and stamP. This will cause him to wander around the grid, and when he recieves a stamp command, he will change the white square there to black. By giving him the sequence of commands
of how to move, you can render an arbitrary b+w image.

The input file will have two integers describing the size of the grid. Then, it will contain a sequence of characters. These characters describe the command sequence to execute to create the image. The program should output the image in some way. For example, it might print it to a png file or print it in ascii art to the screen.

As an example, the input file

```
5 5 PESPESPESPESPNNNNPWSPWSPWSPWSP
```

would output a 5x5 grid with an X in it.

SUPER BONUS: implement a program that can convert an arbitrary image to the walkaround rasterizer format.

Title: [8/24/2012] Challenge #91 [easy] (Sleep sort)

Text: An anonymous user on world4ch's programming text board posted a thread in early 2011 in which he describes an ingenious *O(n)* sorting algorithm. This means it's, supposedly, more efficient than any sorting algorithm ever invented. Some bloggers picked up on it, and dubbed the algorithm [sleep sort](http://beust.com/weblog/2011/06/15/sleep-sort/):

```
#!/bin/bash
function f() {
  sleep "$1"
  echo "$1"
}
while [ -n "$1" ]
do
  f "$1" &
  shift
done
wait
```

This program takes some command line arguments, like `./sleepsort.sh 3 1 4 1 5 9`, and starts a new [thread](http://en.wikipedia.org/wiki/Thread_(computing\)) for each number in the list, which first sleeps for *n* seconds, then prints *n*. After 1 second, both 1s are printed, then after 2 more seconds the 3 follows, etc. Because it only loops through the list of numbers once, the algorithm appears to run in linear time.

Your task is to **re-implement sleep sort** in a language of your choice (which might look trivial, but this challenge is all about learning how your language handles multithreading.)

BONUS\: at first glance, this algorithm appears to be *O(n)*. Can you prove this isn't true? (This bonus requires some knowledge of both algorithms and concurrency.)

Title: [8/27/2012] Challenge #92 [easy] (Digital number display)

Text: Today's easy challenge is to write a program that draws a number in the terminal that looks like one of those old school [seven segment displays](http://en.wikipedia.org/wiki/Seven-segment_display) you find in alarm clocks and VCRs. For instance, if you wanted to draw the number 5362, it would look somthing like:

```
+--+ +--+ +--+ +--+
|     | |     |
|     | |     |
+--+ +--+ +--+ +--+
  |   | | | |
  |   | | | |
+--+ +--+ +--+ +--+
```

I've added some +'s to the joints to make it a bit more readable, but that's optional.

Bonus: Write the program so that the numbers are scalable. In other words, that example would have a scale of 2 (since every line is two terminal characters long), but your program should also be able to draw them in a scale of 3, 4, 5, etc.

Title: [8/30/2012] Challenge #93 [easy] (Two-Way Morse Code Translator)
Text: This challenge courtesy of user [nagasgura](http://www.reddit.com/user/nagasgura)

In this challenge, we read in a string from standard input and output the translation to or from morse code on standard output.

When translating to Morse code, one space should be used to separate morse code letters, and two spaces should be used to separate morse code words. When translating to English, there should only be one space in between words, and no spaces in between letters.

Here's a chart of the morse code symbols: [1] http://www.w1wc.com/pdf_files/international_morse_code.pdf

Example input and output:
'sos' -> '... --- ...'
'... --- ...' -> 'sos'

Title: [9/01/2012] Challenge #94 [easy] (Elemental symbols in strings)
Text: If you've ever seen *Breaking Bad*, you might have noticed how some [names in the opening credit sequence](http://i.imgur.com/qnul0.jpg) get highlights according to symbols of elements in the [periodic table](http://en.wikipedia.org/wiki/Periodic_table). Given a string as input, output every possible such modification with the element symbol enclosed in brackets and capitalized. The elements can appear anywhere in the string, but you must only highlight one element per line, like this:

```
$ ./highlight dailyprogrammer
dailypr[O]grammer
daily[P]rogrammer
dail[Y]programmer
da[I]lyprogrammer
dailyprog[Ra]mmer
daily[Pr]ogrammer
dailyprogramm[Er]
dailyprogr[Am]mer
```

Title: [9/03/2012] Challenge #95 [easy] (Reversing text in file)
Text: Write a program that reads text from a file, and then outputs the text to another file but with all the lines reversed and all the words in each line reversed.

So, for instance, if you had one file called the "thetyger.txt" which contained the two first verses of William Blake's The Tyger:

```
Tyger! Tyger! burning bright
In the forests of the night,
What immortal hand or eye
Could frame thy fearful symmetry?

In what distant deeps or skies
Burnt the fire of thine eyes?
On what wings dare he aspire?
What the hand dare sieze the fire?
```

Your program would output this to "thetyger2.txt" (or whatever you want to call the file):

```
fire? the sieze dare hand the What
aspire? he dare wings what On
eyes? thine of fire the Burnt
skies or deeps distant what In

symmetry? fearful thy frame Could
eye or hand immortal What
```

night, the of forests the In
bright burning Tyger! Tyger!

Title: [9/05/2012] Challenge #96 [easy] (Controller Chains)
Text: It's 2001 all over again, and you just got a brand new ps2 in the mail. Unfortunately, it only has 2 controller ports, and you have N friends who all want to play at the same time.

Fortunately, however, the ps2 has an accessory called a 'multitap' that multiplexes one controller port into four controller ports, to allow more than 2 controllers at once.

Pretend you don't know that only one multitap can be used in a given PS2 at once. By connecting multitaps to multitaps, you could easily create a complicated tree architecture to get as many ports as you need. However, you also have limited resources at your disposal.

Given that a controller costs $20, and a multitap costs $12, write a function that takes in an integer D for the amount of money you have (in dollars) and returns the total maximum number of people you could afford to get to play with you on one ps2 tree.

For example, the ps2 has 2 ports to start with and comes with 1 controller, so if D < 20, then the function should return 1.
However, when you add another $20, you can afford another controller, so for D = 20, the function should return 2.
Adding another controller costs you not only another $20 for the controller, but also $12 for the first multitap to go into the system, so for 20<=D<(40+12), you should return N=3.

This is tricky because once you get >5 controllers, you need ANOTHER multitap...and greater than 8 controllers you need 3+ multitaps.

Title: [9/08/2012] Challenge #97 [easy] (Concatenate directory)
Text: Write a program that concatenates all text files (`*.txt`) in a directory, numbering file names in alphabetical order. Print a header containing some basic information above each file.

For example, if you have a directory like this:

```
~/example/abc.txt
~/example/def.txt
~/example/fgh.txt
```

And call your program like this:

```
nooodl:~$ ./challenge97easy example
```

The output would look something like this:

```
=== abc.txt (200 bytes)
(contents of abc.txt)

=== def.txt (300 bytes)
(contents of def.txt)

=== ghi.txt (400 bytes)
(contents of ghi.txt)
```

For extra credit, add a command line option '`-r`' to your program that makes it recurse into subdirectories alphabetically, too, printing larger headers for each subdirectory.

Title: [9/15/2012] Challenge #98 [easy] (Arithmetic tables)
Text: Write a program that reads two arguments from the command line:

* a symbol, `+`, `-`, `*`, or `/`
* a natural number *n* (≥ 0)

And uses them to output a nice table for the operation from 0 to *n*, like this (for "+ 4"):

```
+ | 0 1 2 3 4
------------------
0 | 0 1 2 3 4
1 | 1 2 3 4 5
2 | 2 3 4 5 6
3 | 3 4 5 6 7
4 | 4 5 6 7 8
```

If you want, you can format your output using the reddit table syntax:

```
|+|0|1
|:|:|:
|**0**|0|1
|**1**|1|2
```

Becomes this:

|+|0|1
|:|:|:
|**0**|0|1
|**1**|1|2

Title: [9/17/2012] Challenge #99 [easy] (Words with letters in alphabetical order)
Text: How many words contained in [this dictionary](http://code.google.com/p/dotnetperls-controls/downloads/detail?name=enable1.txt) have their letters in alphabetical order? So, for instance the letters in "ghost" and "bee" is in alphabetical order, but the letters in "cab" are not.

Title: [9/20/2012] Challenge #100 [easy] (Sleep Cycle Estimator)
Text: This challenge comes to us from nagasgura

The human body goes through 90 minute sleep cycles during the night, and you feel more refreshed if you wake up at the end of a sleep cycle than if you wake up during a sleep cycle. The challenge is to make a program that takes a wake-up time and outputs the possible times to fall asleep so that you will wake up at the end of a sleep cycle.

Example:
Input (Wake-up time): 6:15 AM

Output (when to go to sleep): 9:15 PM, 10:45 PM, 12:15 AM, or 1:45 AM

Bonus 1: Be able to input a sleep time and output potential wake-up times

Bonus 2: Account for how long it takes to fall asleep

Title: [9/27/2012] Challenge #101 [easy] (Non-repeating years)
Text: This challenge comes to us from user [skeeto](/u/skeeto)

Write a program to count the number years in an inclusive range of years that have no repeated digits.

For example, 2012 has a repeated digit (2) while 2013 does not. Given the range [1980, 1987], your program would return 7 (1980, 1982, 1983, 1984, 1985, 1986, 1987).

Bonus: Compute the longest run of years of repeated digits and the longest run of years of non-repeated digits for [1000, 2013].

Title: [9/30/2012] Challenge #102 [easy] (Dice roller)
Text: In tabletop role-playing games like Dungeons & Dragons, people use a system called [dice notation](http://en.wikipedia.org/wiki/Dice_notation) to represent a combination of dice to be rolled to generate a random number. Dice rolls are of the form **A**d**B** *(+/-)* **C**, and are calculated like this:

1. Generate **A** random numbers from 1 to **B** and add them together.
2. Add or subtract the modifier, **C**.

If **A** is omitted, its value is 1; if (+/-)**C** is omitted, step 2 is skipped. That is, `"d8"` is equivalent to `"1d8+0"`.

Write a function that takes a string like `"10d6-2"` or `"d20+7"` and generates a random number using this syntax.

Here's a hint on how to parse the strings, if you get stuck:

> Split the string over 'd' first; if the left part is empty, A = 1,
> otherwise, read it as an integer and assign it to A. Then determine
> whether or not the second part contains a '+' or '-', etc.

Title: [10/13/2012] Challenge #103 [easy-difficult] (Text transformations)
Text: ###Easy

Back in the 90s (and early 00s) people thought it was a cool idea to \\/\\/|2][73 |_1|<3 7H15 to bypass text filters on BBSes. They called it [Leet (or 1337)](http://en.wikipedia.org/wiki/Leet), and it quickly became popular all over the internet. The habit has died out, but it's still quite interesting to see the various replacements people came up with when transforming characters.

Your job's to write a program that translates normal text into Leet, either by hardcoding a number of translations (e.g. A becomes either 4 or /-\\, randomly) or allowing the user to specify a random translation table as an input file, like this:

```
A   4 /-\
B   |3 [3 8
C   ( {
(etc.)
```

Each line in the table contains a single character, followed by whitespace, followed by a space-separated list of possible replacements. Characters should have some non-zero chance of not being replaced at all.

Title: [10/18/2012] Challenge #104 [Easy] (Powerplant Simulation)
Text: **Description:**

A powerplant for the city of Redmond goes offline every third day because of local demands. Ontop of this, the powerplant has to go offline for maintenance every 100 days. Keeping things complicated, on every 14th day, the powerplant is turned off for refueling. Your goal is to write a function which returns the number of days the powerplant is operational given a number of days to simulate.

Formal Inputs & Outputs:

Input Description:

Integer days - the number of days we want to simulate the powerplant

Output Description:

Return the number of days the powerplant is operational.

Sample Inputs & Outputs:

The function, given 10, should return 7 (3 days removed because of maintenance every third day).

Title: [10/20/2012] Challenge #105 [Easy] (Word unscrambler)
Text: Given a wordlist of your choosing, make a program to unscramble scrambled words from that list. For sanity and brevity, disregard any words which have ambiguous unscramlings, such as "dgo" unscrambling to both "dog" and "god."

Input:

A file which contains scrambled words and a wordlist to match it against

Output:
The unscrambled words which match the scrambled ones

Title: [10/23/2012] Challenge #106 [Easy] (Random Talker, Part 1)
Text: Your program will be responsible for analyzing a small chunk of text, the text of this ***entire*** dailyprogrammer description. Your task is to output the distinct words in this description, from highest to lowest count with the number of occurrences for each. Punctuation will be considered as separate words where they are not a part of the word.

The following will be considered their own words: **"** **.** **,** **:** **;** **!** **?** **(** **)** **[** **]** **{** **}**

For anything else, consider it as part of a word.

Extra Credit:

Process the text of the ebook [Metamorphosis, by Franz Kafka](http://www.gutenberg.org/cache/epub/5200/pg5200.txt) and determine the top 10 most frequently used words and their counts. (This will help for part 2)

Title: [10/25/2012] Challenge #107 [Easy] (All possible decodings)
Text: Consider the translation from letters to numbers `a -> 1` through `z -> 26`. Every sequence of letters can be translated into a string of numbers this way, with the numbers being mushed together. For instance `hello -> 85121215`. Unfortunately the reverse translation is not unique. `85121215` could map to `hello`, but also to `heaubo`. Write a program that, given a string of digits, outputs every possible translation back to letters.

Sample input:

` 123`

Sample output:

` abc`

` aw`

` lc`

Title: [10/27/2012] Challenge #108 [Easy] (Scientific Notation Translator)
Text: If you haven't gathered from the title, the challenge here is to go from decimal notation -> scientific notation. For those that don't know, scientific notation allows for a decimal less than ten, greater than zero, and a power of ten to be multiplied.

For example: 239487 would be 2.39487 x 10^5

And .654 would be 6.54 x 10^-1

Bonus Points:

* Have you program randomly generate the number that you will translate.

* Go both ways (i.e., given 0.935 x 10^3, output 935.)

Title: [10/30/2012] Challenge #109 [Easy] Digits Check
Text: **Description:**
Write a function, where given a string, return true if it only contains the digits from 0 (zero) to 9 (nine). Else, return false.

Formal Inputs & Outputs:
Input Description:
string data - a given string that may or may not contains digits; will never be empty
Output Description:

Return True or False - true if the given string only contains digits, false otherwise
Sample Inputs & Outputs:
"123" should return true. "123.123" should return a false. "abc" should return a false.

Notes:

This is a trivial programming exercise, but a real challenge would be to optimize this function for your language and/or environment. As a recommended reading, look into how [fast string-searching](http://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm) works.

Title: [11/3/2012] Challenge #110 [Easy] Keyboard Shift
Text: **Description:**

You and a friend are working on a very important, bleeding-edge, research paper: "Computational Complexity of Sorting Pictures of Cats with Funny Text on the Web". The catch though is your friend wrote his part of the paper with his hands shifted to the right, meaning the top row of keys he used weren't "QWERTYUIOP" ([regular US keyboard](http://www.goodtyping.com/teclatUSok.png)), but instead "WERTYUIOP{".

Your goal is to take what your friend wrote, and convert it from his broken shifted text back into regular english!

Formal Inputs & Outputs:

Input Description:

String ShiftedText - The shifted text in question. The only chracters you have to deal with are letters, in both cases, and the following symbols: '{', '[', ':', ';', '<', ','. The space character may be present, but you do not have to shift that.

Output Description:

Print the correct text.

Sample Inputs & Outputs:

The string "Jr;;p ept;f" should shift back, through your function, into "Hello World". Another example is: "Lmiyj od ,u jrtp", which corrects to "Knuth is my hero"

Title: [11/6/2012] Challenge #111 [Easy] Star delete
Text: Write a function that, given a string, removes from the string any `*` character, or any character that's one to the left or one to the right of a `*` character. Examples:

`"adf*lp" --> "adp"`
`"a*o" --> ""`
`"*dech*" --> "ec"`
`"de**po" --> "do"`
`"sa*n*ti" --> "si"`
`"abc" --> "abc"`

Title: [11/14/2012] Challenge #112 [Easy]Get that URL!
Text: **Description:**

[Website URLs](http://en.wikipedia.org/wiki/Uniform_resource_locator), or Uniform Resource Locators, sometimes embed important data or arguments to be used by the server. This entire string, which is a URL with a [Query String](http://en.wikipedia.org/wiki/Query_string) at the end, is used to "[GET](http://en.wikipedia.org/wiki/GET_(HTTP)#Request_methods)" data from a web server.

A classic example are URLs that declare which page or service you want to access. The Wikipedia log-in URL is the following:

    http://en.wikipedia.org/w/index.php?title=Special:UserLogin&returnto=Main+Page

Note how the URL has the Query String "?title=..", where the value "title" is "Special:UserLogin" and "returnto" is "Main+Page"?

Your goal is to, given a website URL, validate if the URL is well-formed, and if so, print a simple list of the key-value pairs! Note that URLs only allow specific characters ([listed here](http://en.wikipedia.org/wiki/Uniform_resource_locator#List_of_allowed_URL_characters)) and that a Query String must always be of the form "<base-URL>[?key1=value1[&key2=value2[etc...]]]"

Formal Inputs & Outputs:

Input Description:

String GivenURL - A given URL that may or may not be well-formed.

Output Description:

If the given URl is invalid, simply print "The given URL is invalid". If the given URL is valid, print all key-value pairs in the following format:

    key1: "value1"
    key2: "value2"
    key3: "value3"
    etc...

Sample Inputs & Outputs:

Given "http://en.wikipedia.org/w/index.php?title=Main_Page&action=edit", your program should print the following:

    title: "Main_Page"
    action: "edit"

Given "http://en.wikipedia.org/w/index.php?title= hello world!&action=é", your program should print the following:

The given URL is invalid

(To help, the last example is considered invalid because space-characters and unicode characters are not valid URL characters)

Title: [11/20/2012] Challenge #113 [Easy] String-type checking
Text: **Description:**

You and a few co-workers are implementing a cool new technology called "blue-steel" (not to be confused with this [awesome feat of technology](http://www.youtube.com/watch?v=D519hT7-ytY)). Part of this technology, specifically the part assigned to you, is to check what "type" a given string of information is. Blue-steel currently must distinguish between a signed integer, signed float, a date, and a text-string.

Your goal is to write a function which, given a string of text, will echo out what "type" the string is. The string could be a signed integer (a series of digits with either a + or - at the front, though the + is optional), a signed float (a series of digits with either a + or - at the front, though the + is optional, and a . to distinguish the left and right hand digits), a date (which will be in the format of "day-month-year"), and finally a string of text (any other data). The given string will always be just one type at a time.

Formal Inputs & Outputs:

Input Description:

String TypeString - A string to test what type it is.

Output Description:

You must print either "int", "float", "date", or "text" after identifying what string type this is.

Sample Inputs & Outputs:

"123" should print "int", so should "+123", "-123", "0", etc. "123.456" should print "float", while "20-11-2012" should print "date", and finally "Hello, World!" should print "text". Again, you are not expected to handle a multi-type string such as "Hello 123".

Title: [12/4/2012] Challenge #114 [Easy] Word ladder steps
Text: A word ladder is a sequence of words made by changing one letter at a time. For example:

cold → cord → card → ward → warm

Given a word, list all the words that can appear next to it in a word ladder, using [this list of 3,807 four-letter words](http://pastebin.com/zY4Xt7iB). Sample input:

puma

Sample output:

duma
pima
puja
pula
pump
puna
pupa

How many words from the list can appear next to the word `best` in a word ladder?

__Bonus 1:__ One word in the list has 33 other words that can appear next to it. What is this word?

__Bonus 2:__ How many different words can be reached, starting from `best`, in 3 or fewer steps

Title: [1/2/2013] Challenge #115 [Easy] Guess-that-number game!
Text: # [](#EasyIcon) *(Easy)*: Guess-that-number game!

A "guess-that-number" game is exactly what it sounds like: a number is guessed at random by the computer, and you must guess that number to win! The only thing the computer tells you is if your guess is below or above the number.

Your goal is to write a program that, upon initialization, guesses a number between 1 and 100 (inclusive), and asks you for your guess. If you type a number, the program must either tell you if you won (you guessed the computer's number), or if your guess was below the computer's number, or if your guess was above the computer's number. If the user ever types "exit", the program must terminate.

Formal Inputs & Outputs

Input Description

At run-time, expect the user to input a number from 1 to 100 (inclusive), or the string "exit", and treat all other conditions as a wrong guess.

Output Description

The program must print whether or not your guess was correct, otherwise print if your guess was below or above the computer's number.

Sample Inputs & Outputs

Let "C>" be the output from your applicatgion, and "U>" be what the user types:

```
C> Welcome to guess-that-numbers game! I have already picked a number in [1, 100]. Please make a guess. Type "exit" to quit.
U> 1
C> Wrong. That number is below my number.
U> 50
C> Wrong. That number is above my number.
...
U> 31
C> Correct! That is my number, you win! <Program terminates>
```

Title: [01/07/13] Challenge #116 [Easy] Permutation of a string
Text:
[](#EasyIcon) *(Easy)*: Permutation of a string
Write a function that prints all of the permutatons of the unique characters of a given string. For example, permute("baz") would print:

```
baz
bza
abz
azb
zba
zab
```

Find all the permutations of daily.

Author: skeeto
Formal Inputs & Outputs
Input Description
Your function should accept a single string variable which is guaranteed to be at least 1 character long.
Output Description
Print all permutations of the given string variable.

Sample Inputs & Outputs
Sample Input
Let the string argument be "ab"
Sample Output
All permutations of "ab" would be ["ab", "ba"]
Challenge Input
Let the string argument be "abbccc"
Challenge Input Solution
abbccc abcbcc abccbc abcccb acbbcc acbcbc acbccb accbbc accbcb acccbb babccc bacbcc baccbc bacccb bbaccc bbcacc bbccac bbccca bcabcc bcacbc bcaccb bcbacc bcbcac bcbcca bccabc bccacb bccbac bccbca bcccab bcccba cabbcc cabcbc cabccb cacbbc cacbcb caccbb cbabcc cbacbc cbaccb cbbacc cbbcac cbbcca cbcabc cbcacb cbcbac cbcbca cbccab cbccba ccabbc ccabcb ccacbb ccbabc ccbacb ccbbac ccbbca ccbcab ccbcba cccabb cccbab cccbba
Note
* Bonus 1: Instead of printing, be functional. Return a collection (array, etc.) of the permutations.

* Bonus 2: Correctly handle the case when the input contains a character multiple times. That is, don't output repeats and ensure the output contains the same number of characters as the input. For example, there are three permutations of foo: foo, ofo, oof.

* Note that this challenge is a near-duplicate of challenge #12, hence why there is the above "bonus" challenges

Title: [01/14/13] Challenge #117 [Easy] Hexdump to ASCII
Text: # [](#EasyIcon) *(Easy)*: Hexdump to ASCII
Hexadecimal is a base-16 representation of a number. A single byte of information, as an unsigned integer, can have a value of 0 to 255 in decimal. This byte can be represented in hexadecimal, from a range of 0x0 to 0xFF in hexadecimal.

Your job is to open a given file (using the given file name) and print every byte's hexadecimal value.

Author: PoppySeedPlehzr
Formal Inputs & Outputs
Input Description
As a program command-line argument to the program, accept a valid file name.
Output Description
Print the given file's contents, where each byte of the file must be printed in hexadecimal form. Your program must print 16 bytes per line, where there is a space between each hexadecimal byte. Each line must start with the line number, starting from line 0, and must also count in hexadecimal.
Sample Inputs & Outputs
Sample Input
"MyFile.txt" (This file is an arbitrary file as an example)
Sample Output

```
00000000 37 7A BC AF 27 1C 00 03 38 67 83 24 70 00 00 00
00000001 00 00 00 00 49 00 00 00 00 00 00 00 64 FC 7F 06
00000002 00 28 12 BC 60 28 97 D5 68 12 59 8C 17 8F FE D8
00000003 0E 5D 2C 27 BC D1 87 F6 D2 BE 9B 92 90 E8 FD BA
00000004 A2 B8 A9 F4 BE A6 B8 53 10 E3 BD 60 05 2B 5C 95
00000005 C4 50 B4 FC 10 DE 58 80 0C F5 E1 C0 AC 36 30 74
00000006 82 8B 42 7A 06 A5 D0 0F C2 4F 7B 27 6C 5D 96 24
00000007 25 4F 3A 5D F4 B2 C0 DB 79 3C 86 48 AB 2D 57 11
00000008 53 27 50 FF 89 02 20 F6 31 C2 41 72 84 F7 C9 00
00000009 01 04 06 00 01 09 70 00 07 0B 01 00 01 23 03 01
0000000A 01 05 5D 00 00 01 00 0C 80 F5 00 08 0A 01 A8 3F
0000000B B1 B7 00 00 05 01 11 0B 00 64 00 61 00 74 00 61
0000000C 00 00 00 14 0A 01 00 68 6E B8 CF BC A0 CD 01 15
0000000D 06 01 00 20 00 00 00 00 00
```

Challenge Input
Give your program its own binary file, and have it print itself out!
Challenge Input Solution

This is dependent on how you write your code and what platform you are on.
Note
* As an added bonus, attempt to print out any ASCII strings, if such data is found in your given file.

Title: [01/21/13] Challenge #118 [Easy] Date Localization
Text:
[](#EasyIcon) *(Easy)*: Date Localization
Localization of software is the process of adapting code to handle special properties of a given language or a region's standardization of date / time formats.

As an example, in the United States it is common to write down a date first with the month, then day, then year. In France, it is common to write down the day and *then* month, then year.

Your goal is to write a function that takes a given string that defines how dates and times should be ordered, and then print off the current date-time in that format.

Author: nint22
Formal Inputs & Outputs
Input Description
Your function must accept a string "Format". This string can have any set of characters or text, but you must explicitly replace certain special-characters with their equivalent date-time element. Those special characters, and what they map to, are as follows:

"%l": Milliseconds (000 to 999)
"%s": Seconds (00 to 59)
"%m": Minutes (00 to 59)
"%h": Hours (in 1 to 12 format)
"%H": Hours (in 0 to 23 format)
"%c": AM / PM (regardless of hour-format)
"%d": Day (1 up to 31)
"%M": Month (1 to 12)
"%y": Year (four-digit format)
Output Description
The output must be the given string, but with the appropriate date-time special-characters replaced with the current date-time of your system. All other characters should be left untouched.
Sample Inputs & Outputs
Sample Input
"%s.%l"
"%s:%m:%h %M/%d/%y"
"The minute is %m! The hour is %h."
Sample Output
"32.429"
"32:6:9 07/9/2013"
"The minute is 32! The hour is 6."
Challenge Input
None needed
Challenge Input Solution
None needed
Note
There are several standards for this kind of functionality in many software packages. ISO has a well documented standard that follows similar rules, which this exercise is based on.

Title: [01/28/13] Challenge #119 [Easy] Change Calculator
Text:
[](#EasyIcon) *(Easy)*: Change Calculator
Write A function that takes an amount of money, rounds it to the nearest penny and then tells you the *minimum* number of coins needed to equal that amount of money. For Example: "4.17" would print out:

```
Quarters: 16
Dimes: 1
Nickels: 1
Pennies: 2
```

Author: nanermaner
Formal Inputs & Outputs
Input Description
Your Function should accept a decimal number (which may or may not have an actual decimal, in which you can assume it is an integer representing dollars, not cents). Your function should round this number to the nearest hundredth.
Output Description
Print the minimum number of coins needed. The four coins used should be 25 cent, 10 cent, 5 cent and 1 cent. It should be in the following format:

```
Quarters: <integer>
Dimes: <integer>
Nickels: <integer>
Pennies: <integer>
```

Sample Inputs & Outputs
Sample Input
1.23
Sample Output
```
Quarters: 4
Dimes: 2
Nickels: 0
Pennies: 3
```
Challenge Input
```
10.24
0.99
5
00.06
```
Challenge Input Solution
Not yet posted
Note
This program may be different for international users, my examples used quarters, nickels, dimes and pennies. Feel free to use generic terms like "10 cent coins" or any other unit of currency you are more familiar with.

* Bonus: Only print coins that are used at least once in the solution.

Title: [02/04/13] Challenge #120 [Easy] Log throughput counter
Text:
[](#EasyIcon) *(Easy)*: Log throughput counter
You are responsible for a search engine of a large website and the servers are getting overloaded. You are pretty sure there's an increase in the number of queries per second, probably because someone is crawling you like there is no tomorrow. To be really sure you need to help the sysadmin in setting up a monitoring system which will alert everyone when the num. of queries per second reach a certain threshold. All he needs to get this going is a file that has one number corresponding to the number of queries in the past x seconds. The file needs to be updated every x seconds automatically so he can integrate that in his monitoring system. You have a log file from the search engine which has one query per line and is constantly being written to. Now all you need to do is to come up with a little program that runs in the background with a very small footprint and is very efficient in counting the query

throughput every x seconds. This count is then written to a file. It has to be very precise, so if the program is set to count the last 3 seconds it really needs to be 3.
If there are no entries in the past x seconds then obviously the file needs to show 0.
The output file and the interval should be options with default values.

Author: soundjack
Formal Inputs & Outputs
Input Description
The input is a growing log file. The script should read the input from stdin.
Output Description
The output should be a file that contains only one single number that represents the number of lines counted in the last X seconds. For the purpose of this challenge it's ok if the output is stdout.
Sample Inputs & Outputs
Sample Input
```
INFO : [query] [2012/12/10 19:19:51.819] 0c9250e0-3272-4e2c-bab4-0a4fd88e0d75
INFO : [query] [2012/12/10 19:19:52.108] 2e9cf755-7f39-4c96-b1c7-f7ccd0a573aa
INFO : [query] [2012/12/11 19:19:52.120] 336974ad-d2b6-48e6-93f7-76a92aca0f64
INFO : [query] [2012/12/11 19:19:52.181] 71b5f768-d177-47f8-b280-b76eb1e85138
INFO : [query] [2012/12/11 19:19:52.183] d44df904-9bc4-46c6-a0c0-e23992345tfd
INFO : [query] [2012/12/12 19:19:52.377] 25473f3a-5043-4322-a759-6930abe30c50
```

Sample Output
23
Challenge Input
None needed
Challenge Input Solution
None needed
Note
None

Title: [03/04/13] Challenge #121 [Easy] Bytelandian Exchange 1
Text: # [](#EasyIcon) *(Easy)*: Bytelandian Exchange 1
Bytelandian Currency is made of coins with integers on them. There is a coin for each non-negative integer (including 0). You have access to a peculiar money changing machine. If you insert a N-valued coin, with N positive, It pays back 3 coins of the value N/2,N/3 and N/4, rounded down. For example, if you insert a 19-valued coin, you get three coins worth 9, 6, and 4. If you insert a 2-valued coin, you get three coins worth 1, 0, and 0. _0-valued coins cannot be used in this machine._

One day you're bored so you insert a 7-valued coin. You get three coins back, and you then insert each of these back into the machine. You continue to do this with every positive-valued coin you get back, until finally you're left with nothing but 0-valued coins. You count them up and see you have 15 coins.

How many 0-valued coins could you get starting with a single 1000-valued coin?

Author: Thomas1122
Formal Inputs & Outputs
Input Description
The value N of the coin you start with
Output Description
The number of 0-valued coins you wind up with after putting every positive-valued coin you have through the machine.
Sample Inputs & Outputs
Sample Input
7
Sample Output
15
Challenge Input
1000

Challenge Input Solution
???
Note
Hint: use recursion!

Title: [03/18/13] Challenge #122 [Easy] Words With Ordered Vowels
Text: # [](#EasyIcon) *(Easy)*: Words With Ordered Vowels
Find words in a word list that contain all the vowels in alphabetical
order, non-repeated, where vowels are defined as A E I O U Y.

Formal Inputs & Outputs
Input Description
A text file with one word on each line.
Output Description
All words in the list that contains all the vowels A E I O U Y in alphabetical order.
Sample Inputs & Outputs
Sample Input
Use [this word list](http://code.google.com/p/dotnetperls-controls/downloads/detail?name=enable1.txt)
Sample Output
abstemiously
...
Challenge Input
Nothing special, see sample input
Challenge Input Solution
Nothing special, see sample output
Note

Title: [04/01/13] Challenge #122 [Easy] Sum Them Digits
Text: # [](#EasyIcon) *(Easy)*: Sum Them Digits
As a crude form of hashing function, Lars wants to sum the digits of a number. Then he wants to sum the digits of the result, and repeat until he have only one digit left. He learnt that this is called the [digital root](http://en.wikipedia.org/wiki/Digital_root) of a number, but the Wikipedia article is just confusing him.

Can you help him implement this problem in your favourite programming language?

It is possible to treat the number as a string and work with each character at a time. This is pretty slow on big numbers, though, so Lars wants you to at least try solving it with only integer calculations (the [modulo operator](http://en.wikipedia.org/wiki/Modulo_operation) may prove to be useful!).

Author: TinyLebowski
Formal Inputs & Outputs
Input Description
A positive integer, possibly 0.
Output Description
An integer between 0 and 9, the digital root of the input number.
Sample Inputs & Outputs
Sample Input
31337
Sample Output
8, because 3+1+3+3+7=17 and 1+7=8
Challenge Input
1073741824
Challenge Input Solution
?
Note
None

Title: [04/15/13] Challenge #122 [Easy] Sum Them Digits
Text:
[](#EasyIcon) *(Easy)*: Sum Them Digits
As a crude form of hashing function, Lars wants to sum the digits of a number. Then he wants to sum the digits of the result, and repeat until he have only one digit left. He learnt that this is called the [digital root](http://en.wikipedia.org/wiki/Digital_root) of a number, but the Wikipedia article is just confusing him.

Can you help him implement this problem in your favourite programming language?

It is possible to treat the number as a string and work with each character at a time. This is pretty slow on big numbers, though, so Lars wants you to at least try solving it with only integer calculations (the [modulo operator](http://en.wikipedia.org/wiki/Modulo_operation) may prove to be useful!).

Author: TinyLebowski
Formal Inputs & Outputs
Input Description
A positive integer, possibly 0.
Output Description
An integer between 0 and 9, the digital root of the input number.
Sample Inputs & Outputs
Sample Input
31337
Sample Output
8, because 3+1+3+3+7=17 and 1+7=8
Challenge Input
1073741824
Challenge Input Solution
?
Note
None

Title: [04/22/13] Challenge #123 [Easy] Sum Them Digits
Text: # [](#EasyIcon) *(Easy)*: Sum Them Digits
As a crude form of hashing function, Lars wants to sum the digits of a number. Then he wants to sum the digits of the result, and repeat until he have only one digit left. He learnt that this is called the [digital root](http://en.wikipedia.org/wiki/Digital_root) of a number, but the Wikipedia article is just confusing him.

Can you help him implement this problem in your favourite programming language?

It is possible to treat the number as a string and work with each character at a time. This is pretty slow on big numbers, though, so Lars wants you to at least try solving it with only integer calculations (the [modulo operator](http://en.wikipedia.org/wiki/Modulo_operation) may prove to be useful!).

Author: TinyLebowski
Formal Inputs & Outputs
Input Description
A positive integer, possibly 0.
Output Description
An integer between 0 and 9, the digital root of the input number.
Sample Inputs & Outputs
Sample Input
31337
Sample Output
8, because 3+1+3+3+7=17 and 1+7=8
Challenge Input
1073741824

Challenge Input Solution
?
Note
None

Title: [04/29/13] Challenge #123 [Easy] New-Line Troubles
Text:
[](#EasyIcon) *(Easy)*: New-Line Troubles
A [newline character](http://en.wikipedia.org/wiki/Newline) is a special character in text for computers: though it is not a visual (e.g. renderable) character, it is a control character, informing the reader (whatever program that is) that the following text should be on a new line (hence "newline character").

As is the case with many computer standards, newline characters (and their rendering behavior) were not uniform across systems until much later. Some character-encoding standards (such as [ASCII](http://en.wikipedia.org/wiki/ASCII)) would encode the character as hex 0x0A (dec. 10), while [Unicode](http://en.wikipedia.org/wiki/Unicode) has a handful of subtly-different newline characters. Some systems even define newline characters as a set of characters: Windows-style new-line is done through two bytes: CR+LF (carriage-return and then the ASCII newline character).

Your goal is to read ASCII-encoding text files and "fix" them for the encoding you want. You may be given a Windows-style text file that you want to convert to UNIX-style, or vice-versa.

Author: nint22
Formal Inputs & Outputs
Input Description
On standard input, you will be given two strings in quotes: the first will be the text file location, with the second being which format you want it output to. Note that this second string will always either be "Windows" or "Unix".

Windows line endings will always be CR+LF (carriage-return and then newline), while Unix endings will always be just the LF (newline character).
Output Description
Simply echo the text file read back off onto standard output, with all line endings corrected.
Sample Inputs & Outputs
Sample Input
The following runs your program with the two arguments in the required quoted-strings.

```
./your_program.exe "/Users/nint22/WindowsFile.txt" "Unix"
```
Sample Output
The example output should be the contents of the WindowsFile.txt file, sans CR+LF characters, but just LF.
Challenge Input
None required.
Challenge Input Solution
None required.
Note
None

Title: [05/06/13] Challenge #124 [Easy] New-Line Troubles
Text:
[](#EasyIcon) *(Easy)*: New-Line Troubles
A [newline character](http://en.wikipedia.org/wiki/Newline) is a special character in text for computers: though it is not a visual (e.g. renderable) character, it is a control character, informing the reader (whatever program that is) that the following text should be on a new line (hence "newline character").

As is the case with many computer standards, newline characters (and their rendering behavior) were not uniform across systems until much later. Some character-encoding standards (such as [ASCII](http://en.wikipedia.org/wiki/ASCII)) would encode the character as hex 0x0A (dec. 10), while [Unicode](http://en.wikipedia.org/wiki/Unicode) has a handful of subtly-different newline

characters. Some systems even define newline characters as a set of characters: Windows-style new-line is done through two bytes: CR+LF (carriage-return and then the ASCII newline character).

Your goal is to read ASCII-encoding text files and "fix" them for the encoding you want. You may be given a Windows-style text file that you want to convert to UNIX-style, or vice-versa.

Author: nint22
Formal Inputs & Outputs
Input Description
On standard input, you will be given two strings in quotes: the first will be the text file location, with the second being which format you want it output to. Note that this second string will always either be "Windows" or "Unix".

Windows line endings will always be CR+LF (carriage-return and then newline), while Unix endings will always be just the LF (newline character).
Output Description
Simply echo the text file read back off onto standard output, with all line endings corrected.
Sample Inputs & Outputs
Sample Input
The following runs your program with the two arguments in the required quoted-strings.

```
./your_program.exe "/Users/nint22/WindowsFile.txt" "Unix"
```
Sample Output
The example output should be the contents of the WindowsFile.txt file, sans CR+LF characters, but just LF.
Challenge Input
None required.
Challenge Input Solution
None required.
Note
None

Title: [05/13/13] Challenge #125 [Easy] Word Analytics
Text: # [](#EasyIcon) *(Easy)*: Word Analytics

You're a newly hired engineer for a brand-new company that's building a "killer Word-like application". You've been specifically assigned to implement a tool that gives the user some details on common word usage, letter usage, and some other analytics for a given document! More specifically, you must read a given text file (no special formatting, just a plain ASCII text file) and print off the following details:

1. Number of words
2. Number of letters
3. Number of symbols (any non-letter and non-digit character, excluding white spaces)
4. Top three most common words (you may count "small words", such as "it" or "the")
5. Top three most common letters
6. Most common first word of a paragraph (paragraph being defined as a block of text with an empty line above it) (*Optional bonus*)
7. Number of words only used once (*Optional bonus*)
8. All letters not used in the document (*Optional bonus*)

Please note that your tool does *not* have to be case sensitive, meaning the word "Hello" is the same as "hello" and "HELLO".

Author: nint22

Formal Inputs & Outputs
Input Description

As an argument to your program on the command line, you will be given a text file location (such as "C:\Users\nint22\Document.txt" on Windows or "/Users/nint22/Document.txt" on any other sane file system). This file may be empty, but will be guaranteed well-formed (all valid ASCII characters). You can assume that line endings will follow the UNIX-style new-line ending ([unlike the Windows carriage-return & new-line format](http://www.cs.toronto.edu/~krueger/csc209h/tut/line-endings.html)).

Output Description

For each analytic feature, you must print the results in a special string format. Simply you will print off 6 to 8 sentences with the following format:

"A words", where A is the number of words in the given document
"B letters", where B is the number of letters in the given document
"C symbols", where C is the number of non-letter and non-digit character, excluding white spaces, in the document
"Top three most common words: D, E, F", where D, E, and F are the top three most common words
"Top three most common letters: G, H, I", where G, H, and I are the top three most common letters
"J is the most common first word of all paragraphs", where J is the most common word at the start of all paragraphs in the document (paragraph being defined as a block of text with an empty line above it) (*Optional bonus*)
"Words only used once: K", where K is a comma-delimited list of all words only used once (*Optional bonus*)
"Letters not used in the document: L", where L is a comma-delimited list of all alphabetic characters not in the document (*Optional bonus*)

If there are certain lines that have no answers (such as the situation in which a given document has no paragraph structures), simply do not print that line of text. In this example, I've just generated some random Lorem Ipsum text.

Sample Inputs & Outputs
Sample Input

*Note that "MyDocument.txt" is just a [Lorem Ipsum](http://en.wikipedia.org/wiki/Lorem_ipsum) text file that conforms to this challenge's well-formed text-file definition.

```
./MyApplication /Users/nint22/MyDocument.txt
```

Sample Output

Note that we do not print the "most common first word in paragraphs" in this example, nor do we print the last two bonus features:

```
265 words
1812 letters
59 symbols
Top three most common words: "Eu", "In", "Dolor"
Top three most common letters: 'I', 'E', 'S'
```

Title: [05/20/13] Challenge #126 [Easy] Real-World Merge Sort
Text: # [](#EasyIcon) *(Easy)*: Real-World Merge Sort

Imagine you are an engineer working on some legacy code that has some odd constraints: you're being asked to implement a new function, which basically merges and sorts one list of integers into another list of integers, where you cannot allocate any other structures apart from simple temporary variables (such as an index or counter variable).

You will be given two lists, list A and B, where the elements are positive integers from 1 to 2147483647; the integer '0' is reserved as "buffer space". List A will *not* be pre-sorted, though list B *will* be pre-sorted *and* will start with a series of '0' values. These '0' values are simply reserved space in list B which is the same number of elements that list A has. You cannot modify list A in *any* way, though list B is fair game. **Your goal is to** return a merged and sorted list of elements of list A into list B, where you cannot allocate any extra space other than simple / trivial local variables for your function.

Author: nint22

Formal Inputs & Outputs
Input Description

You will be given two lists, list A and B, of integers from 1 to 2147483647. List A will be unsorted, while list B will be sorted. List B has extra elements in the start of the list that are all '0' value: this is buffered space for you to work with when merging and sorting list A into B. These two lists will be space-delimited and on separate lines.

Output Description

Simply print the contents of list B, after it has had the contents of A merged & sorted within it.

Sample Inputs & Outputs
Sample Input

```
692 1 32
0 0 0 14 15 123 2431
```

Sample Output

```
1 14 15 32 123 692 2431
```

Note

Please note that the real challenge here is respecting the premise of the challenge: you **must** implement your sort / merge function inline into list B! If you do not understand the premise, please do ask questions and we will gladly answer. Good luck, and have fun!

Title: [05/28/13] Challenge #127 [Easy] McCarthy 91 Function
Text: # [](#EasyIcon) *(Easy)*: McCarthy 91 Function

The [McCarthy 91 Function](http://en.wikipedia.org/wiki/McCarthy_91_function) is a recursive function which, given an integer N, returns the integer 91 if N is equal to or smaller than 100, or simply N-10 if N is greater than 100. Sounds simple, but look at the function definition in the linked Wikipedia article! How could such a function work to always return a constant (for N <= 100) that isn't in the function body? Well, that's your task: write out each step that McCarthy's function does for a given integer N.

Author: nint22

Formal Inputs & Outputs
Input Description

You will be given a single integer N on standard console input. This integer will range between 0 and 2,147,483,647 (without commas).

Output Description

You must output what the function does on each recursion: first you must print the function the expression that is being computed, and then print which condition it took. Simply put, you must print each recursion event in the following string format: "<Expression being executed> since <is greater than | is equal to or less than> 100". Note that for the first line you do not need to print the "since" string (see example below). You should also print the final expression, which is the result (which should always be 91).

Sample Inputs & Outputs
Sample Input

Note: Take from Wikipedia for the sake of keeping things as simple and clear as possible.

```
99
```

Sample Output

```
M(99)
M(M(110)) since 99 is equal to or less than 100
M(100) since 110 is greater than 100
M(M(111)) since 100 is equal to or less than 100
M(101) since 111 is greater than 100
91 since 101 is greater than 100
91
```

Title: [06/4/13] Challenge #128 [Easy] Sum-the-Digits, Part II
Text: # [](#EasyIcon) *(Easy)*: Sum-the-Digits, Part II

Given a well-formed (non-empty, fully valid) string of digits, let the integer N be the sum of digits. Then, given this integer N, turn it into a string of digits. Repeat this process until you only have one digit left. Simple, clean, and easy: focus on writing this as cleanly as possible in your preferred programming language.

Author: nint22. This challenge is particularly easy, so don't worry about looking for crazy corner-cases or weird exceptions. This challenge is as up-front as it gets :-) Good luck, have fun!

Formal Inputs & Outputs
Input Description

On standard console input, you will be given a string of digits. This string will not be of zero-length and will be guaranteed well-formed (will always have digits, and nothing else, in the string).

Output Description

You must take the given string, sum the digits, and then convert this sum to a string and print it out onto standard console. Then, you must repeat this process again and again until you only have one digit left.

Sample Inputs & Outputs
Sample Input

Note: Take from Wikipedia for the sake of keeping things as simple and clear as possible.

```
12345
```

Sample Output

```
12345
15
6
```

Title: [Easy] Longest Two-Character Sub-String
Text: # [](#EasyIcon) *(Easy)*: Longest Two-Character Sub-String

This programming challenge is a [classic](http://en.wikipedia.org/wiki/Longest_common_substring_problem) interview question for software engineers: given a string, find the longest sub-string that contains, at most, two characters.

Author: /u/Regul

Formal Inputs & Outputs
Input Description

Through standard console input, you will be given a string to search, which only contains lower-case alphabet letters.

Output Description

Simply print the longest sub-string of the given string that contains, at most, two unique characters. If you find multiple sub-strings that match the description, print the last sub-string (furthest to the right).

Sample Inputs & Outputs
Sample Inputs

```
abbccc
abcabcabcabccc
qwertyytrewq
```

Sample Outputs

```
bbccc
bccc
tyyt
```

Title: [06/17/13] Challenge #130 [Easy] Roll the Dies
Text: # [](#EasyIcon) *(Easy)*: Roll the Dies

In many board games, you have to roll [multiple multi-faces dies](http://en.wikipedia.org/wiki/File:Dice_(typical_role_playing_game_dice).jpg) to generate random numbers as part of the game mechanics. A classic die used is the d20 (die of 20 faces) in the game Dungeons & Dragons. This notation, often called the [Dice Notation](http://en.wikipedia.org/wiki/Dice_notation), is where you write NdM, where N is a positive integer representing the number of dies to roll, while M is a positive integer equal to or grater than two (2), representing the number of faces on the die. Thus, the string "2d20" simply means to roll the 20-faced die twice. On the other hand "20d2" means to roll a two-sided die 20 times.

Your goal is to write a program that takes in one of these Dice Notation commands and correctly generates the appropriate random numbers. Note that it does not matter how you [seed your random number generation](http://en.wikipedia.org/wiki/Random_seed), but you should try to as good programming practice.

Author: nint22

Formal Inputs & Outputs
Input Description

You will be given a string of the for NdM, where N and M are describe above in the challenge description. Essentially N is the number of times to roll the die, while M is the number of faces of this die. N will range from 1 to 100, while M will range from 2 to 100, both inclusively. This string will be given through standard console input.

Output Description

You must simulate the die rolls N times, where if there is more than one roll you must space-delimit (not print each result on a separate line). Note that the range of the random numbers must be inclusive of 1 to M, meaning that a die with 6 faces could possibly choose face 1, 2, 3, 4, 5, or 6.

Sample Inputs & Outputs
Sample Input

```
2d20
4d6
```

Sample Output

```
19 7
5 3 4 6
```

Title: [07/01/13] Challenge #131 [Easy] Who tests the tests?
Text: # [](#EasyIcon) *(Easy)*: Who tests the tests?

[Unit Testing](http://en.wikipedia.org/wiki/Unit_testing) is one of the more basic, but effective, tools for [software testing](http://en.wikipedia.org/wiki/Software_testing) / quality assurance. Your job, as an expert test-engineer, is to double-check someone else's test data, and make sure that the expected output is indeed correct. The two functions you are testing is string-reversal and string-to-upper functions.

For each line of input, there are three space-delimited values: the first being the test index (as either 0 or 1), then the test input string, and finally the "expected" output. You must take the test input string, run it through your implementation of the appropriate function based on the test index, and then finally compare it to the "expected" output. If you are confident your code is correct and that the strings match, then the "expected" output is indeed good! Otherwise, the "expected" output is bad (and thus invalid for unit-testing).

Author: nint22

Formal Inputs & Outputs
Input Description

You will be given an integer N on the first line of input: it represents the following N lines of input. For each line, you will be given an integer T and then two strings A and B. If the integer T is zero (0), then you must reverse the string A. If the integer T is one (1), then you must upper-case (capitalize) the entire string A. At the end of either of these operations, you must test if the expected result (string B) and the result of the function (string A) match.

Output Description

If string A, after the above described functions are executed, and B match, then print "Good test data". Else, print "Mismatch! Bad test data". "Matching" is defined as two strings being letter-for-letter, equivalent case, and of the same length.

Sample Inputs & Outputs
Sample Input

```
6
0 Car raC
0 Alpha AhplA
0 Discuss noissucsiD
1 Batman BATMAN
1 Graph GRAPH
1 One one
```

Sample Output

Good test data
Mismatch! Bad test data
Mismatch! Bad test data
Good test data
Good test data
Mismatch! Bad test data

Title: [07/08/13] Challenge #132 [Easy] Greatest Common Divisor
Text: # [](#EasyIcon) *(Easy)*: Greatest Common Divisor

The [Greatest Common Divisor](https://en.wikipedia.org/wiki/Greatest_common_divisor) of a given set of integers is the greatest integer that can divide these integers without any remainder. From Wikipedia, take a look at this example: for the integers 8 and 12, the highest integer that divides them without remainder is 4.

Your goal is to write a program that takes two integers, and returns the greatest common divisor. You may pick any algorithm or approach you prefer, though a good starting point is [Euclid's algorithm](https://en.wikipedia.org/wiki/Greatest_common_divisor#Using_Euclid.27s_algorithm).

Author: nint22

Formal Inputs & Outputs
Input Description

You will be given two space-delimited integers on the standard console input.

Output Description

Simply print the GCD value for the two given integers. If no GCD exists, print one ('1').

Sample Inputs & Outputs
Sample Input

32 12
142341 512345
65535 4294967295

Sample Output

4
1
65535

Title: [07/15/13] Challenge #133 [Easy] Foot-Traffic Analysis
Text: # [](#EasyIcon) *(Easy)*: Foot-Traffic Analysis

The world's most prestigious art gallery in the world needs *your* help! Management wants to figure out how many people visit each room in the gallery, and for how long: this is to help improve the quality of the overall gallery in the future.

Your goal is to write a program that takes a formatted log file that describes the overall gallery's foot-traffic on a minute-to-minute basis. From this data you must compute the average time spent in each room, and how many visitors there were in each room.

Author: nint22

Formal Inputs & Outputs
Input Description

You will be first given an integer N which represents the following N-number of lines of text. Each line represents either a visitor entering or leaving a room: it starts with an integer, representing a visitor's unique identifier. Next on this line is another integer, representing the room index. Note that there are at most 100 rooms, starting at index 0, and at most 1,024 visitors, starting at index 0. Next is a single character, either 'I' (for "In") for this visitor entering the room, or 'O' (for "out") for the visitor leaving the room. Finally, at the end of this line, there is a time-stamp integer: it is an integer representing the minute the event occurred during the day. This integer will range from 0 to 1439 (inclusive). All of these elements are space-delimited.

You may assume that all input is logically well-formed: for each person entering a room, he or she will always leave it at some point in the future. A visitor will only be in one room at a time.

Note that the order of events in the log **are not sorted in any way**; it shouldn't matter, as you can solve this problem without sorting given data. Your output (see details below) **must** be sorted by room index, ascending.

Output Description

For each room that had log data associated with it, print the room index (starting at 0), then print the average length of time visitors have stayed as an integer (round down), and then finally print the total number of visitors in the room. All of this should be on the same line and be space delimited; you may optionally include labels on this text, like in our sample output 1.

Sample Inputs & Outputs
Sample Input 1

```
4
0 0 I 540
1 0 I 540
0 0 O 560
1 0 O 560
```

Sample Output 1

```
Room 0, 20 minute average visit, 2 visitor(s) total
```

Sample Input 2

```
36
0 11 I 347
1 13 I 307
2 15 I 334
3 6 I 334
4 9 I 334
5 2 I 334
6 2 I 334
7 11 I 334
8 1 I 334
0 11 O 376
1 13 O 321
2 15 O 389
3 6 O 412
4 9 O 418
5 2 O 414
6 2 O 349
7 11 O 418
8 1 O 418
0 12 I 437
```

1 28 I 343
2 32 I 408
3 15 I 458
4 18 I 424
5 26 I 442
6 7 I 435
7 19 I 456
8 19 I 450
0 12 O 455
1 28 O 374
2 32 O 495
3 15 O 462
4 18 O 500
5 26 O 479
6 7 O 493
7 19 O 471
8 19 O 458

Sample Output 2

Room 1, 85 minute average visit, 1 visitor total
Room 2, 48 minute average visit, 2 visitors total
Room 6, 79 minute average visit, 1 visitor total
Room 7, 59 minute average visit, 1 visitor total
Room 9, 85 minute average visit, 1 visitor total
Room 11, 57 minute average visit, 2 visitors total
Room 12, 19 minute average visit, 1 visitor total
Room 13, 15 minute average visit, 1 visitor total
Room 15, 30 minute average visit, 2 visitors total
Room 18, 77 minute average visit, 1 visitor total
Room 19, 12 minute average visit, 2 visitors total
Room 26, 38 minute average visit, 1 visitor total
Room 28, 32 minute average visit, 1 visitor total
Room 32, 88 minute average visit, 1 visitor total

Title: [08/06/13] Challenge #134 [Easy] N-Divisible Digits
Text: # [](#EasyIcon) *(Easy)*: N-Divisible Digits

Write a program that takes two integers, N and M, and find the largest integer composed of N-digits that is evenly divisible by M. N will always be 1 or greater, with M being 2 or greater. Note that some combinations of N and M will not have a solution.

Example: if you are given an N of 3 and M of 2, the largest integer with 3-digits is 999, but the largest 3-digit number that is evenly divisible by 2 is 998, since 998 [Modulo](https://en.wikipedia.org/wiki/Modular_arithmetic) 2 is 0. Another example is where N is 2 and M is 101. Since the largest 2-digit integer is 99, and no integers between 1 and 99 are divisible by 101, there is no solution.

Author: nint22. *Note:* Sorry for the absence of challenges; I've been away for the last two weeks, and am getting back into the grove of things.

Formal Inputs & Outputs
Input Description

You will be given two integers, N and M, on standard console input. They will be space delimited values where N will range from 1 to 9, and M will range from 2 to 999,999,999.

Output Description

Print the largest integer within the range of 1 to the largest integer formed by N-digits, that is evenly-divisible by the integer M. You only need to print the largest integer, not the set of evenly-divisible integers. If there is no solution, print "No solution found".

Sample Inputs & Outputs
Sample Input 1

    3 2

Sample Output 1

    998

Sample Input 2

    7 4241275

Sample Output 2

    8482550

Title: [08/13/13] Challenge #135 [Easy] Arithmetic Equations
Text: # [](#EasyIcon) *(Easy)*: Arithmetic Equations

[Unix](http://en.wikipedia.org/wiki/Unix), the famous multitasking and multi-user operating system, has several standards that defines Unix commands, system calls, subroutines, files, etc. Specifically within [Version 7](http://en.wikipedia.org/wiki/Version_7_Unix) (though this is included in many other Unix standards), there is a game called "arithmetic". To quote the [Man Page](http://en.wikipedia.org/wiki/Man_page):

> Arithmetic types out simple arithmetic problems, and waits for an answer to be typed in. If the answer is correct, it types back "Right!", and a new problem. If the answer is wrong, it replies "What?", and waits for another answer. Every twenty problems, it publishes statistics on correctness and the time required to answer.

Your goal is to implement this game, with some slight changes, to make this an [Easy]-level challenge. You will only have to use three arithmetic operators (addition, subtraction, multiplication) with four integers. An example equation you are to generate is "2 x 4 + 2 - 5".

Author: nint22

Formal Inputs & Outputs
Input Description

The first line of input will always be two integers representing an inclusive range of integers you are to pick from when filling out the constants of your equation. After that, you are to print off a single equation and wait for the user to respond. The user may either try to solve the equation by writing the integer result into the console, or the user may type the letters 'q' or 'Q' to quit the application.

Output Description

If the user's answer is correct, print "Correct!" and randomly generate another equation to show to the user. Otherwise print "Try Again" and ask the same equation again. Note that all equations must randomly pick and place the operators, as well as randomly pick the equation's constants (integers) from the given range. You are allowed to repeat constants and operators. You may use either the star '*' or the letter 'x' characters to represent multiplication.

Sample Inputs & Outputs
Sample Input / Output

Since this is an interactive application, lines that start with '>' are there to signify a statement from the console to the user, while any other lines are from the user to the console.

```
0 10
> 3 * 2 + 5 * 2
16
> Correct!
> 0 - 10 + 9 + 2
2
> Incorrect...
> 0 - 10 + 9 + 2
3
> Incorrect...
> 0 - 10 + 9 + 2
1
> Correct!
> 2 * 0 * 4 * 2
0
> Correct!
q
```

Title: [08/13/13] Challenge #136 [Easy] Student Management
Text: # [](#EasyIcon) *(Easy)*: Student Management

You are a computer science professor at South Harmon Institute of Technology, and are in dire need of automatic grading! The good news is you have all of your student's assignments in an easy-to-read format, making automation easy!

You will be given a list of unique student names, and then a list of their assignment grades. All assignments are based on 20 points and are scored in whole-numbers (integers). All students have received the same number of assignments, so you don't have to worry about managing [jagged arrays](http://en.wikipedia.org/wiki/Iliffe_vector).

Author: nint22

Formal Inputs & Outputs
Input Description

On standard console input, you will be given two space-delimited integers N and M: N is the number of students (which ranges from 1 to 60, inclusive), and M is the number of assignments (which ranges from 4 to 100, inclusive). This will be followed by N lines of text, each starting with an upper-case unique string being is your students name. This is then followed by M integers, which are the grades ranging from 0 to 20, inclusively.

Output Description

On the first line of output, print the class' average grade. Then, for each student, print their name and average grade (up to two decimal points precision).

Sample Inputs & Outputs
Sample Input 1

```
3 5
JON 19 14 15 15 16
JEREMY 15 11 10 15 16
JESSE 19 17 20 19 18
```

Sample Output 1

```
15.93
JON 15.80
JEREMY 13.40
JESSE 18.60
```

Sample Input 2

```
10 10
ABIGAIL 11 3 5 20 4 2 8 17 4 5
ALEXANDER 2 12 20 0 6 10 3 4 9 7
AVA 11 15 2 19 14 5 16 18 15 19
ETHAN 6 12 0 0 5 11 0 11 12 15
ISABELLA 16 0 10 7 20 20 7 2 0 1
JACOB 2 14 17 7 1 11 16 14 14 7
JAYDEN 10 10 3 16 15 16 8 17 15 3
MADISON 10 11 19 4 12 15 7 4 18 13
SOPHIA 5 17 14 7 1 17 18 8 1 2
WILLIAM 12 12 19 9 4 3 0 4 13 14
```

Sample Output 2

```
9.50
ABIGAIL          7.90
ALEXANDER 7.30
AVA    13.40
ETHAN 7.20
ISABELLA 8.30
JACOB 10.30
JAYDEN 11.30
MADISON 11.30
SOPHIA 9.00
WILLIAM 9.00
```

Title: [08/13/13] Challenge #137 [Easy] String Transposition
Text: # [](#EasyIcon) *(Easy)*: String Transposition

It can be helpful sometimes to rotate a string 90-degrees, like a big vertical "SALES" poster or your business name on vertical neon lights, like [this image from Las Vegas](http://imgur.com/766x8uM). Your goal is to write a program that does this, but for multiples lines of text. This is very similar to a [Matrix Transposition](http://en.wikipedia.org/wiki/Transpose), since the order we want returned is not a true 90-degree rotation of text.

Author: nint22

Formal Inputs & Outputs
Input Description

You will first be given an integer N which is the number of strings that follows. N will range inclusively from 1 to 16. Each line of text will have at most 256 characters, including the new-line (so at most 255 printable-characters, with the last being the new-line or carriage-return).

Output Description

Simply print the given lines top-to-bottom. The first given line should be the left-most vertical line.

Sample Inputs & Outputs
Sample Input 1

```
1
Hello, World!
```

Sample Output 1

```
H
e
l
l
o
,

W
o
r
l
d
!
```

Sample Input 2

```
5
Kernel
Microcontroller
Register
Memory
Operator
```

Sample Output 2

```
KMRMO
eieep
rcgme
nrior
eosra
lctyt
 oe o
 nr r
 t
 r
 o
 l
 l
 e
 r
```

Title: [09/17/13] Challenge #138 [Easy] Repulsion-Force
Text: # [](#EasyIcon) *(Easy)*: Repulsion-Force

[Colomb's Law](http://en.wikipedia.org/wiki/Coulomb%27s_law) describes the repulsion force for two electrically charged particles. In *very* general terms, it describes the rate at which particles move away from each-other based on each particle's mass and distance from one another.

Your goal is to compute the repulsion force for two electrons in 2D space. Assume that the two particles have the same mass and charge. The function that computes force is as follows:

    Force = (Particle 1's mass x Particle 2's mass) / Distance^2

Note that Colomb's Law uses a constant, but we choose to omit that for the sake of simplicity. For those not familiar with vector math, you can compute the distance between two points in 2D space using the following formula:

    deltaX = (Particle 1's x-position - Particle 2's x-position)
    deltaY = (Particle 1's y-position - Particle 2's y-position)
    Distance = Square-root(deltaX * deltaX + deltaY * deltaY)

Author: nint22

Formal Inputs & Outputs
Input Description

On standard console input, you will be given two rows of numbers: first row represents the first particle, with the second row representing the second particle. Each row will have three space-delimited real-numbers (floats), representing mass, x-position, and y-position. The mass will range, inclusively, from 0.001 to 100.0. The x and y positions will range inclusively from -100.0 to 100.0.

Output Description

Print the force as a float at a minimum three decimal places precision.

Sample Inputs & Outputs
Sample Input 1

    1 -5.2 3.8
    1 8.7 -4.1

Sample Output 1

    0.0039

Sample Input 2

    4 0.04 -0.02
    4 -0.02 -0.03

Sample Output 2

    4324.3279

Title: [11/4/13] Challenge #139 [Easy] Pangrams
Text: # [](#EasyIcon) *(Easy)*: Pangrams

[Wikipedia](http://en.wikipedia.org/wiki/Pangram) has a great definition for Pangrams: "*A pangram or holoalphabetic sentence for a given alphabet is a sentence using every letter of the alphabet at least once.*" A good example is the English-language sentence

"[The quick brown fox jumps over the lazy dog](http://en.wikipedia.org/wiki/The_quick_brown_fox_jumps_over_the_lazy_dog)"; note how all 26 English-language letters are used in the sentence.

Your goal is to implement a program that takes a series of strings (one per line) and prints either True (the given string is a pangram), or False (it is not).

Bonus: On the same line as the "True" or "False" result, print the number of letters used, starting from 'A' to 'Z'. The format should match the following example based on the above sentence:

    a: 1, b: 1, c: 1, d: 1, e: 3, f: 1, g: 1, h: 2, i: 1, j: 1, k: 1, l: 1, m: 1, n: 1, o: 4, p: 1, q: 1, r: 2, s: 1, t: 2, u: 2, v: 1, w: 1, x: 1, y: 1, z: 1

Formal Inputs & Outputs
Input Description

On standard console input, you will be given a single integer on the first line of input. This integer represents the number of lines you will then receive, each being a string of alpha-numeric characters ('a'-'z', 'A'-'Z', '0'-'9') as well as spaces and [period](http://en.wikipedia.org/wiki/Period_(punctuation\)).

Output Description

For each line of input, print either "True" if the given line was a pangram, or "False" if not.

Sample Inputs & Outputs
Sample Input

    3
    The quick brown fox jumps over the lazy dog.
    Pack my box with five dozen liquor jugs
    Saxophones quickly blew over my jazzy hair

Sample Output

    True
    True
    False

Title: [11/4/13] Challenge #140 [Easy] Variable Notation
Text: # [](#EasyIcon) *(Easy)*: Variable Notation

When writing code, it can be helpful to have a standard ([Identifier naming convention](http://en.wikipedia.org/wiki/Identifier_naming_convention)) that describes how to define all your variables and object names. This is to keep code easy to read and maintain. Sometimes the standard can help describe the type (such as in [Hungarian notation](http://en.wikipedia.org/wiki/Hungarian_notation)) or make the variables visually easy to read ([CamcelCase notation](http://en.wikipedia.org/wiki/CamelCase) or [snake_case](http://en.wikipedia.org/wiki/Snake_case)).

Your goal is to implement a program that takes an english-language series of words and converts them to a specific variable notation format. Your code must support CamcelCase, snake_case, and capitalized snake_case.

Formal Inputs & Outputs
Input Description

On standard console input, you will be given an integer one the first line of input, which describes the notation you want to convert to. If this integer is zero ('0'), then use CamcelCase. If it is one ('1'), use snake_case. If it is two ('2'), use capitalized snake_case. The line after this will be a space-delimited series of words, which will only be lower-case alpha-numeric characters (letters and digits).

Output Description

Simply print the given string in the appropriate notation.

Sample Inputs & Outputs
Sample Input
  0
  hello world

  1
  user id

  2
  map controller delegate manager

Sample Output
  0
  helloWorld

  1
  user_id

  2
  MAP_CONTROLLER_DELEGATE_MANAGER

Difficulty++
For an extra challenge, try to convert from one notation to another. Expect the first line to be two integers, the first one being the notation already used, and the second integer being the one you are to convert to. An example of this is:

Input:
  1 0
  user_id

Output:
  userId

Title: [11/11/13] Challenge #141 [Easy] Monty Hall Simulation
Text: # [](#EasyIcon) *(Easy)*: Monty Hall Simulation

The [Monty Hall Problem](http://en.wikipedia.org/wiki/Monty_Hall_problem) is a probability puzzle that has a very non-intuitive answer for the average person. Here's the problem description taken from Wikipedia:

"Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?"

AsapScience has a great [YouTube video describing this game](http://www.youtube.com/watch?v=9vRUxbzJZ9Y). If you don't understand *why* switching doors is the best tactic, feel free to discuss it here or on other great subreddits, like /r/Math, /r/ComputerScience, or even /r/AskScience!

Your goal is to simulate two tactics to this puzzle, and return the percentage of successful results. The first tactic is where you stick with your initial choice. The second tactic is where you always switch doors.

Edit: Make sure to actually simulate *both* techniques. Write that code out in its entirety, don't compute the second result being '100% - first techniques percentage', though that's certainly true mathematically.

Formal Inputs & Outputs
Input Description

On standard console input, you will be given a single integer ranging inclusively from 1 to 4,294,967,295 (unsigned 32-bit integer). This integer is the number of times you should simulate the game for both tactics. Remember that a single "game simulation" is your program randomly placing a car behind one door and two goats behind the two remaining doors. You must then randomly pick a door, have one of the two remaining doors open, but only open if it's a goat behind said door! After that, if using the first tactic, you may open the picked door, or if using the second tactic, you may open the other remaining door. Keep track if your success rates in both simulations.

Output Description

On two seperate lines, print "Tactic 1: X% winning chance" and "Tactic 2: Y% winning chance", where X and Y are the percentages of success for the respective tactics

Sample Inputs & Outputs
Sample Input

```
1000000
```

Sample Output

```
Tactic 1: 33.3% winning chance
Tactic 2: 66.6% winning chance
```

Difficulty++

For an extra challenge, visualize the simulation! Using whatever tools and platform you want, let the simulation visually show you the doors it's picking over time. Try to aim for one simulation a second, keeping it fast-paced.
Text: # [](#EasyIcon) *(Easy)*: Monty Hall Simulation

The [Monty Hall Problem](http://en.wikipedia.org/wiki/Monty_Hall_problem) is a probability puzzle that has a very non-intuitive answer for the average person. Here's the problem description taken from Wikipedia:

"Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?"

AsapScience has a great [YouTube video describing this game](http://www.youtube.com/watch?v=9vRUxbzJZ9Y). If you don't understand *why* switching doors is the best tactic, feel free to discuss it here or on other great subreddits, like /r/Math, /r/ComputerScience, or even /r/AskScience!

Your goal is to simulate two tactics to this puzzle, and return the percentage of successful results. The first tactic is where you stick with your initial choice. The second tactic is where you always switch doors.

Edit: Make sure to actually simulate *both* techniques. Write that code out in its entirety, don't compute the second result being '100% - first techniques percentage', though that's certainly true mathematically.

Formal Inputs & Outputs
Input Description

On standard console input, you will be given a single integer ranging inclusively from 1 to 4,294,967,295 (unsigned 32-bit integer). This integer is the number of times you should simulate the game for both tactics. Remember that a single "game simulation" is your program randomly placing a car behind one door and two goats behind the two remaining doors. You must then randomly pick a door, have one of the two remaining doors open, but only open if it's a goat behind said door! After that, if using the first tactic, you may open the picked door, or if using the second tactic, you may open the other remaining door. Keep track if your success rates in both simulations.

Output Description

On two seperate lines, print "Tactic 1: X% winning chance" and "Tactic 2: Y% winning chance", where X and Y are the percentages of success for the respective tactics

Sample Inputs & Outputs
Sample Input

    1000000

Sample Output

    Tactic 1: 33.3% winning chance
    Tactic 2: 66.6% winning chance

Difficulty++

For an extra challenge, visualize the simulation! Using whatever tools and platform you want, let the simulation visually show you the doors it's picking over time. Try to aim for one simulation a second, keeping it fast-paced.

Title: [11/11/13] Challenge #141 [Easy] Checksums
Text: # [](#EasyIcon) *(Easy)*: Checksums

[Checksums](http://en.wikipedia.org/wiki/Checksum) are a tool that allow you to verify the integrity of data (useful for networking, security, error-correction, etc.). Though there are *many* different Checksum algorithms, the general usage is that you give raw-data to your algorithm of choice, and a block of data (usually smaller than the given data) is generated and can later be used by re-computing the checksum and comparing the original and recent values.

A classic example for how helpful Checksums are is with data-networking: imagine you have a packet of information that must be guaranteed the same after receiving it. Before sending the data, you can compute its checksum, and send both blocks together. When received, the data can be used to re-compute a checksum, and validate that the given checksum and your own checksum are the same. The subject is much more complex, since there are issues of [data-entropy](http://en.wikipedia.org/wiki/Entropy_(information_theory\)) and the importance of the checksum's size compared to the raw data size.

This example is so common in network programming, one of the [basic Internet networking protocols (TCP)](http://en.wikipedia.org/wiki/Transmission_Control_Protocol#Checksum_computation) has it built-in!

Your goal will be more modest: you must implement a specific checksum algorithm ([Fletcher's 16-bit Checksum](http://en.wikipedia.org/wiki/Fletcher%27s_checksum)) for given lines of text input. The [C-like language pseudo-code found on Wikipedia](http://en.wikipedia.org/wiki/Fletcher%27s_checksum#Straightforward) is a great starting point!

Note: Make sure to explicitly implement this algorithm, and not call into other code (libraries). The challenge here is focused on your implementation of the algorithm.

Formal Inputs & Outputs
Input Description

On standard console input, you will first be given an integer N which ranges inclusively from 1 to 256. After this line, you will receive N-lines of ASCII text. This text will only contain regular printable characters, and will all be on a single line of input.

Output Description

For each line of input, print the index (starting from 1) and the 16-bit Fletcher's checksum as a 4-digit hexadecimal number.

Sample Inputs & Outputs

Sample Input

```
3
Fletcher
Sally sells seashells by the seashore.
Les chaussettes de l'archi-duchesse, sont-elles seches ou archi-seches ?
```

Sample Output

```
1 D330
2 D23E
3 404D
```

Title: [11/11/13] Challenge #142 [Easy] Falling Sand
Text: # [](#EasyIcon) *(Easy)*: Falling Sand

[Falling-sand Games](http://en.wikipedia.org/wiki/Falling-sand_game) are particle-simulation games that focus on the interaction between particles in a 2D-world. Sand, as an example, might fall to the ground forming a pile. Other particles might be much more complex, like fire, that might spread depending on adjacent particle types.

Your goal is to implement a mini falling-sand simulation for just sand and stone. The simulation is in 2D-space on a uniform grid, where we are viewing this grid from the side. Each type's simulation properties are as follows:

* Stone always stays where it was originally placed. It never moves.
* Sand keeps moving down through air, one step at a time, until it either hits the bottom of the grid, other sand, or stone.

Formal Inputs & Outputs
Input Description

On standard console input, you will be given an integer N which represents the N x N grid of ASCII characters. This means there will be N-lines of N-characters long. This is the starting grid of your simulated world: the character ' ' (space) means an empty space, while '.' (dot) means sand, and '#' (hash or pound) means stone. Once you parse this input, simulate the world until all particles are settled (e.g. the sand has fallen and either settled on the ground or on stone). "Ground" is defined as the solid surface right below the last row.

Output Description

Print the end result of all particle positions using the input format for particles.

Sample Inputs & Outputs
Sample Input

```
5
.....
  #
#

   .
```

Sample Output

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

Title: [12/03/13] Challenge #143 [Easy] Braille
Text: # [](#EasyIcon) *(Easy)*: Braille

[Braille](http://en.wikipedia.org/wiki/Braille) is a writing system based on a series of raised / lowered bumps on a material, for the purpose of being read through touch rather than sight. It's an incredibly powerful reading & writing system for those who are blind / visually impaired. Though the letter system has up to 64 unique glyph, 26 are used in [English Braille](http://en.wikipedia.org/wiki/English_Braille) for letters. The rest are used for numbers, words, accents, ligatures, etc.

Your goal is to read in a string of Braille characters ([using standard English Braille defined here](http://en.wikipedia.org/wiki/English_Braille#Alphabet)) and print off the word in standard English letters. You only have to support the 26 English letters.

Formal Inputs & Outputs
Input Description

Input will consistent of an array of 2x6 space-delimited Braille characters. This array is always on the same line, so regardless of how long the text is, it will always be on 3-rows of text. A lowered bump is a dot character '.', while a raised bump is an upper-case 'O' character.

Output Description

Print the transcribed Braille.

Sample Inputs & Outputs
Sample Input

```
O. O. O. O. O. .O O. O. O. OO
OO .O O. O. .O OO .O OO O. .O
.. .. O. O. O. .O O. O. O. ..
```

Sample Output

```
Helloworld
```

Title: [12/11/13] Challenge #144 [Easy] Nuts & Bolts
Text: # [](#EasyIcon) *(Easy)*: Nuts & Bolts

You have just been hired at a local home improvement store to help compute the proper costs of inventory. The current prices are out of date and wrong; you have to figure out which items need to be re-labeled with the correct price.

You will be first given a list of item-names and their current price. You will then be given another list of the same item-names but with the correct price. You must then print a list of items that have changed, and by how much.

Formal Inputs & Outputs
Input Description

The first line of input will be an integer N, which is for the number of rows in each list. Each list has N-lines of two space-delimited strings: the first string will be the unique item name (without spaces), the second string will be the price (in whole-integer cents). The second list, following the same format, will have the same unique item-names, but with the correct price. Note that the lists may not be in the same order!

Output Description

For each item that has had its price changed, print a row with the item name and the price difference (in cents). Print the sign of the change (e.g. '+' for a growth in price, or '-' for a loss in price). Order does not matter for output.

Sample Inputs & Outputs

Sample Input 1

```
4
CarriageBolt 45
Eyebolt 50
Washer 120
Rivet 10
CarriageBolt 45
Eyebolt 45
Washer 140
Rivet 10
```

Sample Output 1

```
Eyebolt -5
Washer +20
```

Sample Input 2

```
3
2DNail 3
4DNail 5
8DNail 10
8DNail 11
4DNail 5
2DNail 2
```

Sample Output 2

```
2DNail -1
8DNail +1
```

Title: [12/16/13] Challenge #145 [Easy] Tree Generation
Text: # [](#EasyIcon) *(Easy)*: Tree Generation

Your goal is to draw a tree given the base-width of the tree (the number of characters on the bottom-most row of the triangle section). This "tree" must be drawn through [ASCII art-style graphics](http://en.wikipedia.org/wiki/ASCII_art) on standard console output. It will consist of a 1x3 trunk on the bottom, and a triangle shape on the top. The tree must be centered, with the leaves growing from a base of N-characters, up to a top-layer of 1 character. Each layer reduces by 2 character, so the bottom might be 7, while shrinks to 5, 3, and 1 on top layers. See example output.

_[Originally submitted by u/Onkel_Wackelflugel](http://www.reddit.com/r/dailyprogrammer_ideas/comments/1srsde/easy_build_a_christmas_treemenorah_with/)_

Formal Inputs & Outputs
Input Description

You will be given one line of text on standard-console input: an integer and two characters, all space-delimited. The integer, N, will range inclusively from 3 to 21 and *always* be odd. The next character will be your trunk character. The next character will be your leaves character. Draw the trunk and leaves components with these characters, respectively.

Output Description

Given the three input arguments, draw a centered-tree. It should follow this pattern: (this is the smallest tree possible, with a base of 3)

```
  *
```

```
  ***
  ###
```

Here's a much larger tree, of base 7:

```
   *
  ***
 *****
*******
  ###
```

Sample Inputs & Outputs
Sample Input 1

```
3 # *
```

Sample Output 1

```
  *
 ***
 ###
```

Sample Input 2

```
13 = +
```

Sample Output 2

```
     +
    +++
   +++++
  +++++++
 +++++++++
+++++++++++
     ===
```

Challenge++

Draw something special! Experiment with your creativity and engineering, try to render this tree in whatever cool way you can think of. [Here's an example](http://www.youtube.com/watch?v=QMYfkOtYYlg#t=15) of how far you can push a simple console for rendering neat graphics!

Title: [12/23/13] Challenge #146 [Easy] Polygon Perimeter
Text: # [](#EasyIcon) *(Easy)*: Polygon Perimeter

[A Polygon](http://en.wikipedia.org/wiki/Polygon) is a geometric two-dimensional figure that has n-sides (line segments) that closes to form a loop. Polygons can be in many different shapes and have many different neat properties, though this challenge is about [Regular Polygons](http://en.wikipedia.org/wiki/Regular_polygon). Our goal is to compute the permitter of an n-sided polygon that has equal-length sides given the [circumradius](http://en.wikipedia.org/wiki/Regular_polygon#Circumradius). This is the distance between the center of the Polygon to any of its vertices; not to be confused with the [apothem](http://en.wikipedia.org/wiki/Apothem)!

Formal Inputs & Outputs
Input Description

Input will consist of one line on standard console input. This line will contain first an integer N, then a floating-point number R. They will be space-delimited. The integer N is for the number of sides of the Polygon, which is between 3 to 100, inclusive. R will be the circumradius, which ranges from 0.01 to 100.0, inclusive.

Output Description

Print the permitter of the given N-sided polygon that has a circumradius of R. Print up to three digits precision.

Sample Inputs & Outputs
Sample Input 1

```
5 3.7
```

Sample Output 1

```
21.748
```

Sample Input 2

```
100 1.0
```

Sample Output 2

```
6.282
```

Title: [01/07/14] Challenge #147 [Easy] Sport Points
Text: # [](#EasyIcon) *(Easy)*: Sport Points

You must write code that verifies the awarded points for a fictional sport are valid. This sport is a simplification of [American Football](http://en.wikipedia.org/wiki/American_football#Scoring) scoring rules. This means that the score values must be any logical combination of the following four rewards:

+ 6 points for a "touch-down"
+ 3 points for a "field-goal"
+ 1 point for an "extra-point"; can only be rewarded *after* a touch-down. Mutually-exclusive with "two-point conversion"
+ 2 points for a "two-point conversion"; can only be rewarded *after* a touch-down. Mutually-exclusive with "extra-point"

A valid score could be 7, which can come from a single "touch-down" and then an "extra-point". Another example could be 6, from either a single "touch-down" or two "field-goals". 4 is not a valid score, since it cannot be formed by any well-combined rewards.

Formal Inputs & Outputs
Input Description

Input will consist of a single positive integer given on standard console input.

Output Description

Print "Valid Score" or "Invalid Score" based on the respective validity of the given score.

Sample Inputs & Outputs
Sample Input 1

```
35
```

Sample Output 1

```
Valid Score
```

Sample Input 2

2

Sample Output 2

Invalid Score

Title: [01/13/14] Challenge #148 [Easy] Combination Lock
Text: # [](#EasyIcon) *(Easy)*: Combination Lock

[Combination locks](http://en.wikipedia.org/wiki/Combination_lock) are mechanisms that are locked until a specific number combination is input. Either the input is a single dial that must rotate around in a special procedure, or have three disks set in specific positions. This challenge will ask you to compute how much you have to spin a single-face lock to open it with a given three-digit code.

The procedure for our lock is as follows: (lock-face starts at number 0 and has up to N numbers)

+ Spin the lock a full 2 times clockwise, and continue rotating it to the code's first digit.
+ Spin the lock a single time counter-clockwise, and continue rotating to the code's second digit.
+ Spin the lock clockwise directly to the code's last digit.

Formal Inputs & Outputs
Input Description

Input will consist of four space-delimited integers on a single line through console standard input. This integers will range inclusively from 1 to 255. The first integer is N: the number of digits on the lock, starting from 0. A lock where N is 5 means the printed numbers on the dial are 0, 1, 2, 3, and 5, listed counter-clockwise. The next three numbers are the three digits for the opening code. They will always range inclusively between 0 and N-1.

Output Description

Print the total rotation increments you've had to rotate to open the lock with the given code. See example explanation for details.

Sample Inputs & Outputs
Sample Input

5 1 2 3

Sample Output

21

Here's how we got that number:

+ Spin lock 2 times clockwise: +10, at position 0
+ Spin lock to first number clockwise: +1, at position 1
+ Spin lock 1 time counter-clockwise: +5, at position 1
+ Spin lock to second number counter-clockwise: +4, at position 2
+ Spin lock to third number clockwise: +1, at position 3

Title: [02/24/14] Challenge #149 [Easy] Disemvoweler
Text: # _(Easy)_: Disemvoweler

Disemvoweling means removing the vowels from text. (For this challenge, the letters a, e, i, o, and u are considered vowels, and the letter y is not.) The idea is to make text difficult but not impossible to read, for when somebody posts something so idiotic you want people who are reading it to get extra frustrated.

To make things even harder to read, we'll remove spaces too. For example, this string:

    two drums and a cymbal fall off a cliff

can be disemvoweled to get:

    twdrmsndcymblfllffclff

We also want to keep the vowels we removed around (in their original order), which in this case is:

    ouaaaaoai

Formal Inputs & Outputs

Input description

A string consisting of a series of words to disemvowel. It will be all lowercase (letters a-z) and without punctuation. The only special character you need to handle is spaces.

Output description

Two strings, one of the disemvoweled text (spaces removed), and one of all the removed vowels.

Sample Inputs & Outputs

Sample Input 1

    all those who believe in psychokinesis raise my hand

Sample Output 1

    llthswhblvnpsychknssrsmyhnd
    aoeoeieeioieiaiea

Sample Input 2

    did you hear about the excellent farmer who was outstanding in his field

Sample Output 2

    ddyhrbtthxcllntfrmrwhwststndngnhsfld
    ioueaaoueeeeaeoaouaiiiie

Title: [17/04/2014] Challenge #153 [Easy] Pascal's Pyramid
Text: # [](#EasyIcon) _(Easy)_: Pascal's Pyramid

You may have seen Pascal's Triangle before. It has been known about for a long time now and is a very simple concept - it makes several appearances in mathematics, one of which is when you calculate the [binomial expansion](http://en.wikipedia.org/wiki/Binomial_theorem).
If you've not seen it before, you can calculate it by first putting **1** on the outermost numbers:

```
    1
   1 1
  1 _ 1
 1 _ _ 1
1 _ _ _ 1
```

And then each number within the triangle can be calculated by adding the two numbers above it, to form this:

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

It has several interesting properties, however what we're interested in is the **3-dimensional** version of this triangle - **Pascal's Pyramid.**
It works in much the same way - the corner numbers are all **1**s. However the edges are all layers of Pascal's triangle, and each inner number is the sum of the 3 numbers above it. Besides that there is nothing else to it.

Here are the first 5 cross-sectional 'layers', top to bottom:

```
1

 1
1 1

 1
2 2
1 2 1

  1
 3  3
3 6 3
1 3 3 1

    1
   4 4
  6 12 6
 4 12 12 4
1 4 6 4 1
```

See how the outermost 'rows' or 'edges' of numbers on all of the above are layers of Pascal's Triangle, as we saw above. Therefore, The faces of Pascal's Pyramid, were it a 3D object, would have Pascal's Triangle on them!

Your challenge is, given a number **N**, to calculate and display the **N**th layer of Pascal's Pyramid.

Formal Inputs and Outputs

Input Description

On the console, you will be given a number **N** where `N > 0`.

Output Description

You must print out layer **N** of Pascal's Pyramid. The triangle cross-section must be presented so the point is at the top. Each row shall be separated by newlines, and each number shall be separated by spaces. Spacing is not important but your submission would be even cooler if it were displayed properly. For example, for the 3rd layer, a valid output would be as so:

```
  1
 2 2
1 2 1
```

Or, better:

```
   1
  2 2
 1 2 1
```

Or even:

```
    1
   2   2
 1   2 1
```

But why you'd do the latter is beyond me.

Sample Inputs & Outputs

Sample Input

```
6
```

Sample Output

```
1
5 5
10 20 10
10 30 30 10
5 20 30 20 5
1 5 10 10 5 1
```

Challenge

Challenge Input

```
14
```

Notes

There are ways to quickly do this that use the [Factorial function](http://en.wikipedia.org/wiki/Factorial). Also, look at the pattern the 'rows' make in relation to the leftmost and rightmost number and Pascal's triangle.
Reading material on Pascal's Pyramid [can be found here](http://en.wikipedia.org/wiki/Pascal%27s_pyramid).

Jagged multidimensional arrays will come in handy here.

Title: [4/24/2014] Challenge #154 [Easy] March Madness Brackets

Text: # **Description:**

It is that time of year again when across some of the lands you hear about March Madness and NCAA Basketball. People ask about your brackets and how you are doing in your predictions. Of course to those of us who perform the art of coding we always get a bit confused by this.

You see we think of brackets like [] or {} or () to use in our many wonderful languages. As it turns out in a bit of madness some messages got the rough bracket treatment. I am asking you to decode these messages by removing the brackets and decoding the message. The order of the message should be ordered for the deepest bracket strings to be displayed first then the next deepest and so forth.

#**Input:**

(String of words with matching bracket sets in an order that can only be called mad)

##Example 1:

((your[drink {remember to}]) ovaltine)

##Example 2:

[racket for {brackets (matching) is a} computers]

##Example 3:

[can {and it(it (mix) up) } look silly]

#**Output:**

The words separated by a single space in order from deepest to shallow on the ordering of matched brackets.

##Example 1:

remember to drink your ovaltine

##Example 2:

matching brackets is a racket for computers

##Example 3:

mix it up and it can look silly

#**Notes:**

Assume your input is error checked.

Bracket groups can be either [] or () or {} and there will be no mismatches.

The pattern of when and what brackets are used is random. You could see all () or () then a [] then a () again. Etc.

Every closing bracket will have an opening one that matches. So] matches to a [and) matches to a (and } matches to a {.

Whitespace can be random and you need to clean it up. Sometimes there are spaces between bracket symbols and sometimes not. Words will be separated clearly with at least 1 whitespace.

Bracket levels will not be broken up between words. For example you would not see it like this.

{years [four score] ago (and seven) our fathers}

The [four score] (and seven) are on the same level but broken up between words. You would see this as

{years(and seven (four score)) ago our fathers}

Have fun! And good luck with those brackets!

#**Extra Challenge:**
Prior to handling the problem you will proof read your string and look for 2 errors.

1) Mismatch bracket -- ending a (with a] or a } for an example causes an error to be detected and reported.

2) Missing bracket having 3 starting brackets but only 2 closing brackets causes an error to be detected and reported.

##example:
((your[drink {remember to))) ovaltine)

Generates an error of "Mismatched bracket) instead of } found"

##example:

[can {and it(it (mix) up) look silly]

Generates an error "Missing closing bracket"

##example:

[racket for brackets (matching) is a} computers]

Generates an error "Missing opening bracket"

***
Also you can handle multiple sets on the same level broken up by words.

##example:
{years [four score] ago (and seven) our fathers}

Generates the output:

four score and seven years ago our fathers

You would use left to right to give priority to which equal sets to output.

Title: [4/1/2014] Challenge #156 [Easy] Simple Decoder
Text: #Oops:

By now you all have noticed /r/dailyprogrammer has added 3 new moderators. All of us including the existing moderators have been working hard to bring back 3 challenges a week.

We have had some minor issues with dates and challenge numbers. Many of the dates posted were said to be in "4" which is April and really should have been "3" for March. Also our numbering of challenges have been weird.

So going forward this week we will start with 156. Each challenge this week will be 156 (easy, intermediate and hard). Next week all 3 challenges will be 157. Etc. Also we will strive to update the 3 links at the top of the subreddit with the latest challenges and try to get dates correct on our postings. Thanks for your patience and your support!

#Description:
To honor our mistake this week's easy challenge is to decode a message. I have encoded a message by adding a "4" to each character's ASCII value. It will be your job to decode this message by reversing the process and making a decoder.

#Input:

Decode this message:

```
Etvmp$Jsspw%%%%
[e}$xs$ks%$]sy$lezi$wspzih$xli$lmhhir$qiwweki2$Rs{$mx$mw$}syv$xyvr$xs$nsmr
mr$sr$xlmw$tvero2$Hs$rsx$tswx$er}xlmrk$xlex${mpp$kmzi$e{e}$xlmw$qiwweki2$Pix
tistpi$higshi$xli$qiwweki$sr$xlimv$s{r$erh$vieh$xlmw$qiwweki2$]sy$ger$tpe}$epsrk
f}$RSX$tswxmrk$ls{$}sy$higshih$xlmw$qiwweki2$Mrwxieh$tswx$}syv$wspyxmsr$xs$fi$}syv
jezsvmxi$Lipps${svph$tvskveq$mr$sri$perkyeki$sj$}syv$glsmgi2$
Qeoi$wyvi$}syv$tvskveq$we}w$&Lipps$[svph%%%&${mxl$7$%$ex$xli$irh2$Xlmw${e}
tistpi$fvs{wmrk$xli$gleppirki${mpp$xlmro${i$lezi$epp$pswx$syv$qmrhw2$Xlswi${ls$tswx$lipps
{svph$wspyxmsrw${mxlsyx$xli$xlvii$%%%${mpp$lezi$rsx$higshih$xli$qiwweki$erh$ws$}sy$ger$
tspmxip}$tsmrx$syx$xlimv$wspyxmsr$mw$mr$ivvsv$,xli}$evi$nywx$jspps{mrk$epsrk${mxlsyx$ors{mrk-
Irns}$xlmw$jyr2$Xli$xvyxl${mpp$fi$liph$f}$xlswi${ls$ger$higshi$xli$qiwweki2$>-
```

#Output:
As part of the challenge we leave it to the programmer to discover the correct output.

Title: [4/7/2014] Challenge #157 [Easy] The Winning Move X-Games edition
Text: #Description:

The world championship in Tic Tac Toe, The X-Games is underway. We have been asked to write a program to see if there is a winning move possible. This tool will be used by live commentators to use in their play by play.

#input
(Next player's Move either an X or an O)

(3 x 3 grid showing the current game)

The grid can have 3 characters

* X for spot held by the X player
* O for spot held by the O player
* - for an empty spot

##Example Input 1:

```
X
```

```
XX-
-XO
OO-
```

##Example Input 2:

```
O
O-X
-OX
---
```

##Example Input 3:

```
X
--O
OXX
---
```

#Output:
Shows the board with the winning move in place. If there is no winning move print out "No winning move"

##Example Output 1:

```
XXX
-XO
OO-
```

##Example Output 2:

```
O-X
-OX
--O
```

#Example Output 3:

```
No Winning Move!
```

#Extra Challenge:

* Boards where several moves can "win" display all boards showing the winning moves with a message saying how many wins are possible
* Boards with no winning move -- suggest a "block" move the player should make instead with a message saying "No winning move: Block here"

Title: [4/14/2014] Challenge #158 [Easy] The Torn Number
Text: #Description:

I had the other day in my possession a label bearing the number 3 0 2 5 in large figures. This got accidentally torn in half, so that 3 0 was on one piece and 2 5 on the other. On looking at these pieces I began to make a calculation, when I discovered this little peculiarity. If we add the 3 0 and the 2 5 together and square the sum we get as the result, the complete original number on the label! Thus, 30 added to 25 is 55, and 55 multiplied by 55 is 3025. Curious, is it not?

Now, the challenge is to find another number, composed of four figures, all different, which may be divided in the middle and produce the same result.

#Bonus
Create a program that verifies if a number is a valid torn number.

Title: [4/21/2014] Challenge #159 [Easy] Rock Paper Scissors Lizard Spock - Part 1 The Basic Game
Text: #Theme Week:

Welcome to my first attempt at a theme week. All week long the challenges will be related to this fascinating advanced version of the game Rock Paper Scissors. We will explore the depths of this game like none have before.

#Description:

The best way to see this game and understand the rules is to do some basic research.

* [Wikipedia Entry Link] (http://en.wikipedia.org/wiki/Rock-paper-scissors-lizard-Spock)
* [Instructional Video] (https://www.youtube.com/watch?v=iapcKVn7DdY)

The challenge is to implement a basic game of Rock Paper Scissors Lizard Spock (to be called RPSLP for short). Your game will get the human choice. The computer AI will randomly pick a move. It will compare the results and display the moves and the out come (who wins or if a tie)

#Input:

Get from the user their move being Rock, Paper Scissors, Lizard, Spock. Design and how you do it is up to you all.

#Output:

Once the human move is obtained have the computer randomly pick their move. Display the moves back to the user and then give the results.

Again the exact design is up to you as long as the output shows the moves again and the result of the game (who wins or if a tie).

##Example Output:

```
Player Picks: Rock.
Computer Picks: Spock.

Spock Vaporizes Rock. Computer Wins!
```

#For Weds:

As this is a theme challenge. Weds we continue with a more intermediate approach to the game. To plan ahead please consider in your design some ability to have a human choice be compared to a computer choice or a computer to play itself as well.

#Extra Challenge:

The game loops and continues to play matches until the user quits or a fixed number of games is played. At the end it records some basic stats.

* Total Games played
* Computer Wins (Number and percentage)
* Human Wins (Number and percentage)

* Ties (Number and Percentage)

Title: [4/28/2014] Challenge #160 [Easy] Trigonometric Triangle Trouble, pt. 1
Text: # [](#EasyIcon) _(Easy)_: Trigonometric Triangle Trouble, pt. 1

A triangle on a flat plane is described by its angles and side lengths, and you don't need to be given all of the angles and side lengths to work out the rest. In this challenge, you'll be working with **right-angled triangles only**.

[Here's a representation of how this challenge will describe a triangle](http://i.imgur.com/qtHozmc.png). Each side-length is a **lower-case** letter, and the angle opposite each side is an **upper-case** letter. For the purposes of this challenge, the angle C will always be the right-angle. Your challenge is, using basic trigonometry and given an appropriate number of values for the angles or side lengths, to find the rest of the values.

Formal Inputs and Outputs

Input Description

On the console, you will be given a number **N**. You will then be given **N** lines, expressing **some** details of a triangle in the format below, where all angles are in degrees; the input data will always give enough information and will describe a valid triangle. Note that, depending on your language of choice, a conversion from degrees to radians may be needed to use trigonometric functions such as *sin*, *cos* and *tan*.

Output Description

You must print out **all** of the details of the triangle in the same format as above.

Sample Inputs & Outputs

Sample Input

```
3
a=3
b=4
C=90
```

Sample Output

```
a=3
b=4
c=5
A=36.87
B=53.13
C=90
```

Tips & Notes

There are 4 useful trigonometric identities you may find very useful.

* [Pythagoreas' Theorem](http://latex.codecogs.com/gif.latex?r%5E2+s%5E2%3Dh%5E2), where *h* is the side length opposite the right-angle and *r* and *s* are any 2 other sides.

* [3 Trigonometric Ratios](http://simple.wikipedia.org/wiki/Trigonometry#Trigonometric_Ratios)

Part 2 will be submitted on the 2nd of May. To make it easier to complete Part 2, write your code in such a way that it can be extended later on. Use good programming practices (as always!).

Title: [5/5/2014] #161 [Easy] Blackjack!
Text: #Description:

So went to a Casino recently. I noticed at the Blackjack tables the house tends to use several decks and not 1. My mind began to wonder about how likely natural blackjacks (getting an ace and a card worth 10 points on the deal) can occur.

So for this monday challenge lets look into this. We need to be able to shuffle deck of playing cards. (52 cards) and be able to deal out virtual 2 card hands and see if it totals 21 or not.

* Develop a way to shuffle 1 to 10 decks of 52 playing cards.
* Using this shuffle deck(s) deal out hands of 2s
* count how many hands you deal out and how many total 21 and output the percentage.

#Input:

n: being 1 to 10 which represents how many deck of playing cards to shuffle together.

#Output:

After x hands there was y blackjacks at z%.

##Example Output:

After 26 hands there was 2 blackjacks at %7.

##Optional Output:

Show the hands of 2 cards. So the card must have suit and the card.

* D for diamonds, C for clubs, H for hearts, S for spades or use unicode characters.
* Card from Ace, 2, 3, 4, 5, 6, 8, 9, 10, J for jack, Q for Queen, K for king

#Make Challenge Easier:

Just shuffle 1 deck of 52 cards and output how many natural 21s (blackjack) hands if any you get when dealing 2 card hands.

#Make Challenge Harder:

When people hit in blackjack it can effect the game. If your 2 card hand is 11 or less always get a hit on it. See if this improves or decays your rate of blackjacks with cards being used for hits.

#Card Values:

Face value should match up. 2 for 2, 3 for 3, etc. Jacks, Queens and Kings are 10. Aces are 11 unless you get 2 Aces then 1 will have to count as 1.

Title: [5/12/2014] Challenge #162 [Easy] Novel Compression, pt. 1: Unpacking the Data
Text: # [](#EasyIcon) _(Easy)_: Novel Compression, pt. 1: Unpacking the Data

Welcome to this week's Theme Week. We're going to be creating our very own basic compression format for short novels or writing. This format will probably not be practical for actual use, but may serve as a rudimentary introduction to how data compression works. As a side task, it is advised to use structured programming techniques, so your program is easy to extend, modify and maintain later on (ie. later this week.) To keep in line with our Easy-Intermediate-Hard trend, our first step will be to write the **decompresser**.

The basic idea of this compression scheme will be the dictionary system. Words used in the data will be put into a dictionary, so instead of repeating phrases over and over again, you can just repeat a number instead, thus saving space. Also, because this system is based around written text, it will be specifically designed to handle sentences and punctuation, and will not be geared to handle binary data.

Formal Inputs and Outputs

Input Description

The first section of the input we are going to receive will be the dictionary. This dictionary is just a list of words present in the text we're decompressing. The first line of input will be a number **N** representing how many entries the dictionary has. Following from that will be a list of **N** words, on separate lines. This will be our simple dictionary. After this will be the data.

Data Format

The pre-compressed data will be split up into human-readable 'chunks', representing one little segment of the output. In a practical compression system, they will be represented by a few bytes rather than text, but doing it this way makes it easier to write. Our chunks will follow 7 simple rules:

* If the chunk is just a number (eg. `37`), word number 37 from the dictionary (zero-indexed, so 0 is the 1st word) is printed **lower-case**.

* If the chunk is a number followed by a caret (eg. `37^`), then word 37 from the dictionary will be printed lower-case, **with the first letter capitalised**.

* If the chunk is a number followed by an exclamation point (eg. `37!`), then word 37 from the dictionary will be printed **upper-case**.

* If it's a hyphen (`-`), then instead of putting a space in-between the previous and next words, put a hyphen instead.

* If it's any of the following symbols: `. , ? ! ; :`, then put that symbol at the end of the previous outputted word.

* If it's a letter `R` (upper or lower), print a new line.

* If it's a letter `E` (upper or lower), the end of input has been reached.

Remember, this is just for basic text, so not all possible inputs can be compressed. Ignore any other whitespace, like tabs or newlines, in the input.

Note: All words will be in the Latin alphabet.

Example Data

Therefore, if our dictionary consists of the following:

```
is
my
hello
name
```

stan

And we are given the following chunks:

2! ! R 1^ 3 0 4^ . E

Then the output text is:

HELLO!
My name is Stan.

Words are always separated by spaces unless they're hyphenated.

Output Description

Print the resultant decompressed data from your decompression algorithm, using the rules described above.

Challenge

Challenge Input

20
i
do
house
with
mouse
in
not
like
them
ham
a
anywhere
green
eggs
and
here
or
there
sam
am
0^ 1 6 7 8 5 10 2 . R
0^ 1 6 7 8 3 10 4 . R
0^ 1 6 7 8 15 16 17 . R
0^ 1 6 7 8 11 . R
0^ 1 6 7 12 13 14 9 . R
0^ 1 6 7 8 , 18^ - 0^ - 19 . R E

(Line breaks added in data for clarity and ease of testing.)
Expected Challenge Output

I do not like them in a house.
I do not like them with a mouse.
I do not like them here or there.
I do not like them anywhere.
I do not like green eggs and ham.
I do not like them, Sam-I-am.

Title: [5/19/2014] Challenge #163 [Easy] Probability Distribution of a 6 Sided Di
Text: #Description:

Today's challenge we explore some curiosity in rolling a 6 sided di. I often wonder about the outcomes of a rolling a simple 6 side di in a game or even simulating the roll on a computer.

I could roll a 6 side di and record the results. This can be time consuming, tedious and I think it is something a computer can do very well.

So what I want to do is simulate rolling a 6 sided di in 6 groups and record how often each number 1-6 comes up. Then print out a fancy chart comparing the data. What I want to see is if I roll the 6 sided di more often does the results flatten out in distribution of the results or is it very chaotic and have spikes in what numbers can come up.

So roll a D6 10, 100, 1000, 10000, 100000, 1000000 times and each time record how often a 1-6 comes up and produce a chart of % of each outcome.

Run the program one time or several times and decide for yourself. Does the results flatten out over time? Is it always flat? Spikes can occur?

#Input:

None.

#Output:

Show a nicely formatted chart showing the groups of rolls and the percentages of results coming up for human analysis.

##example:

```
  # of Rolls 1s     2s     3s     4s     5s     6s
  ==================================================
  10         18.10% 19.20% 18.23% 20.21% 22.98% 23.20%
  100        18.10% 19.20% 18.23% 20.21% 22.98% 23.20%
  1000       18.10% 19.20% 18.23% 20.21% 22.98% 23.20%
  10000      18.10% 19.20% 18.23% 20.21% 22.98% 23.20%
  100000     18.10% 19.20% 18.23% 20.21% 22.98% 23.20%
  1000000    18.10% 19.20% 18.23% 20.21% 22.98% 23.20%
```

notes on example output:

* Yes in the example the percentages don't add up to 100% but your results should
* Yes I used the same percentages as examples for each outcome. Results will vary.
* Your choice on how many places past the decimal you wish to show. I picked 2. if you want to show less/more go for it.

#Code Submission + Conclusion:

Do not just post your code. Also post your conclusion based on the simulation output. Have fun and enjoy not having to tally 1 million rolls by hand.

Title: [5/26/2014] Challenge #164 [Easy] Assemble this Scheme into Python
Text: ##Description

You have just been hired by the company 'Super-Corp 5000' and they require you to be up to speed on a new programming language you haven't yet tried.

It is your task to familiarise yourself with this language following this criteria:

* The language must be one you've shown interest for in the past
* You must not have had past experience with the language

In order to Impress HR and convince the manager to hire you, you must complete 5 small tasks. You will definitely be hired if you complete the bonus task.

##Input & Output

These 5 tasks are:

* Output 'Hello World' to the console.

* Return an array of the first 100 numbers that are divisible by 3 and 5.

* Create a program that verifies if a word is an anagram of another word.

* Create a program that removes a specificed letter from a word.

* Sum all the elements of an array

All output will be the expected output of these processes which can be verified in your normal programming language.

##Bonus

Implement a bubble-sort.

##Note

Don't use a language you've had contact with before, otherwise this will be very easy. The idea is to learn a new language that you've been curious about.

Title: [6/2/2014] Challenge #165 [Easy] ASCII Game of Life
Text: # [](#EasyIcon) _(Easy)_: ASCII Game of Life

Hello people. Sorry for submitting this early, but I have exams this week and the next so I'll have to submit these challenges a little bit early - I'm sure that's not an issue though! Welcome to June, and it's time for a run of similarly themed challenges - all of them will be based on ASCII data. Not too dissimilar to [this challenge](http://www.reddit.com/r/dailyprogrammer/comments/236va2/4162014_challenge_158_intermediate_part_1_the/) from a while ago.

This first challenge is based on a game (the mathematical variety - not quite as fun!) called [Conway's Game of Life](http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life). This is called a cellular automaton. This means it is based on a 'playing field' of sorts, made up of lots of little cells or spaces. For Conway's game of life, the grid is square - but other shapes like hexagonal ones could potentially exist too. Each cell can have a value - in this case, on or off - and for each 'iteration' or loop of the game, the value of each cell will change depending on the other cells around it. This might sound confusing at first, but looks easier when you break it down a bit.

* A cell's "neighbours" are the 8 cells around it.

* If a cell is 'off' but exactly 3 of its neighbours are on, that cell will also turn on - like reproduction.

* If a cell is 'on' but less than two of its neighbours are on, it will die out - like underpopulation.

* If a cell is 'on' but more than three of its neighbours are on, it will die out - like overcrowding.

Fairly simple, right? This might sound boring, but it can generate fairly complex patterns - [this one, for example](http://upload.wikimedia.org/wikipedia/commons/e/e5/Gospers_glider_gun.gif), is called the Gosper Glider Gun and is designed in such a way that it generates little patterns that fly away from it. There are other examples of such patterns, like ones which grow indefinitely.

Your challenge is, given an initial 'state' of 'on' and 'off' cells, and a number, simulate that many steps of the Game of Life.

Formal Inputs and Outputs

Input Description

You will be given a number **N**, and then two more numbers **X** and **Y**. After that you will be given a textual ASCII grid of 'on' and 'off' states that is **X** cells wide and **Y** cells tall. On the grid, a period or full-stop `.` will represent 'off', and a hash sign `#` will represent 'on'.

The grid that you are using must 'wrap around'. That means, if something goes off the bottom of the playing field, then it will wrap around to the top, like this: http://upload.wikimedia.org/wikipedia/en/d/d1/Long_gun.gif See how those cells act like the top and bottom, and the left and right of the field are joined up? In other words, the neighbours of a cell can look like this - where the lines coming out are the neighbours:

```
#-...-  ......  ../|\.
|\.../  ......  ......
......  |/...\  ......
......  #-...-  ......
......  |\.../  ..\|/.
|/...\  ......  ..-#-.
```

Output Description

Using that starting state, simulate **N** iterations of Conway's Game of Life. Print the final state in the same format as above - `.` is off and `#` is on.

Sample Inputs & Output

Sample Input

```
7 10 10
..........
..........
..#.......
...#......
.###......
..........
..........
..........
..........
..........
```

Sample Output

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

Challenge

Challenge Input

```
32 17 17
.................
.................
....###...###....
.................
..#....#.#....#..
..#....#.#....#..
..#....#.#....#..
....###...###....
.................
....###...###....
..#....#.#....#..
..#....#.#....#..
..#....#.#....#..
.................
....###...###....
.................
.................
```

(just for laughs, see how the output changes when you change **N**. Cool, eh?)

Notes

To test your program, use one of the many online simulation programs. There are plenty written in JavaScript you can get at with a Google search (or Bing, if you'd prefer. I wouldn't.)

Title: [6/9/2014] Challenge #166 [Easy] ASCII Fractal Curves
Text: # [](#EasyIcon) _(Easy)_: ASCII Fractal Curves

Today we're going to set a more open-ended challenge. First, let's look at what a space-filling curve is.

A [space-filling curve](http://en.wikipedia.org/wiki/Space-filling_curve) is a specific type of line/curve that, as you recreate it in more and more detail, fills more and more of the space that it's in, without (usually) ever intersecting itself. There are several types of space-filling curve, and all behave slightly differently. Some get more and more complex over time whereas some are the same pattern repeated over and over again.

Your challenge will be to take any space-fulling curve you want, and write a program that displays it to a given degree of complexity.

Formal Inputs and Outputs

Input Description

The input to this challenge is extremely simple. You will take a number **N** which will be the degree of complexity to which you will display your fractal curve. For example, [this image](http://upload.wikimedia.org/wikipedia/en/a/a5/Hilbert_curve.svg) shows the *Hilbert curve* shown to 1 through 6 degrees of complexity.

Output Description

You must print out your own curve to the given degree of complexity. The method of display is up to you, but try and stick with the ASCII theme - for example, see below.

Sample Inputs & Output

Sample Input

(Hilbert curve program)

```
  3
```

Sample Output

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

Notes

Recursive algorithms will come in very handy here. You'll need to do some of your own research into the curve of your choice.

Title: [6/14/2014] Challenge #166b [Easy] Planetary Gravity Calculator
Text: # [](#EasyIcon) _(Easy)_: Planetary Gravity Calculator

Welcome to this week's rebooted challenges. While this challenge is very simple at its core (which I think gives it an Easy rating), it gives me a chance to teach a bit of physics while I'm at it, so I may as well!

Newton's Law of Universal Gravitation says that:

* Any two objects in the universe attract **each other** gravitationally...

* With a force that's proportional to the product of their masses, and...

* Inversely proportional to the *square* of the distance between them. (distance is measured from the center of the object - so if you're standing on Earth, you are about 6353 km away from it.

* Because this is only a proportionality (not an equality), you will need a constant multiplier - this is called G, the gravitational constant.

This gives us the remarkably simple formula:

$$\text{force} = G \times \frac{\text{mass of first object} \times \text{mass of second object}}{(\text{distance between objects})^2}$$

This force is applied on *both* objects equally and in opposite directions, toward each other. The value of G is currently known to be about **6.67e-11** which is why gravity is so weak - you can overcome the force of the entire planet just by jumping!

These 4 simple rules were used to describe gravity in nearly its entirety before Albert Einstein found out it was incomplete and discovered Special and General relativity - which you won't need today! Anyway, this is the only bit of physics you'll need for today's challenge - the rest is basic maths.

We're going to assume all planets are perfect spheres. This means you can find the volume of a planet, given its radius, with the fomula `V = 4/3 × π × radius³` like a normal sphere. We'll also assume they are made of a material which has the exact same density everywhere - so a handful of material from one bit of the planet weighs the same as any other. This means, given a density (in kilograms per cubic metre), and using the volume you worked out, you can compute the mass of the planet with the formula `mass = volume × density`. Assume the units you are using are kilograms and metres. Sorry, imperial folk!

Now, in case you are new to physics, you may need to know a little bit about forces. Forces are measured in Newtons (N) and measure, essentially, how hard an object is pushing another object. The object could be pushing physically - eg. pushing a lawn mower - or via an elementary force, such as Earth's gravity pushing you toward it. They can all be measured in Newtons. The force of a planet on something due to gravity is called *weight* - which is not to be confused with [*mass*](http://en.wikipedia.org/wiki/Mass), which is measured in kilograms and is a measure of how much matter something contains. As we saw before, the more mass two objects have, the greater the force they exert on each other. As gravitational force is dependent on the product of the masses of both objects, an object will weigh more if either the object itself, or the planet, is heavier - which is why you weigh less on the Moon!

Anyway, after that lengthy backstory, the challenge for you today is, given the dimensions of several planets and an object's mass, calculate how much force is applied on the object at the surface of the planet. Pretend the object is quite small for simplicity of your caluclations.

This is certainly a lot of physics to get your teeth into, so if you need any help, leave a comment and either I or someone else should be happy to help you out.

Formal Inputs and Outputs

Input Description

You will be given a number **M** which is the mass of an object in kilograms, on its own line, for example:

```
100
```

Followed by a number **N**:

```
4
```

You will then, on separate lines, be given a list of **N** planets. This will be given as its name, its radius (in metres), and its average density (in kilograms per cubic metre), like so:

```
Mercury, 2439700, 5427
```

Output Description

Print the weight (in Newtons) of the object if it were at the surface of each planet, like so:

Mercury: 314.623

Example Inputs and Outputs

Example Input

```
100
4
Tantalus, 3104500, 5009
Reach, 7636500, 4966
Circumstance, 4127000, 4132
Tribute, 2818000, 4358
```

Example Output

```
Tantalus: 434.467
Reach: 1059.536
Circumstance: 476.441
Tribute: 343.117
```

Challenge

Challenge Input

```
75
9
Mercury, 2439700, 5427
Venus, 6051900, 5243
Earth, 6367445, 5515
Mars, 3386000, 3934
Jupiter, 69173000, 1326
Saturn, 57316000, 687
Uranus, 25266000, 1270
Neptune, 24553000, 1638
Pluto, 1173000, 2050
```

Expected Challenge Output

```
Mercury: 277.442
Venus: 664.886
Earth: 735.845
Mars: 279.124
Jupiter: 1922.011
Saturn: 825.103
Uranus: 672.382
Neptune: 842.741
Pluto: 50.388
```

(These values are all very nearly exact!)

Notes

You have a chance to utilise some OOP here. If your programming language supports it, you may want to create a `Planet` object.

Title: [6/16/2014] Challenge #167 [Easy] HTML markup generator
Text: #Description
You're a well known web-dev with far too much confidence, you mistakingly agreed to complete too many projects in too little a timeframe. In order to fix this, you devise a program that will automatically write all of the HTML for you!

But first, you'll need to program it.

#Formal Inputs & Outputs

##Input description

On standard console input you should be prompted to enter a paragraph.

##Output description

Once your paragraph has been entered, it should be saved as a valid HTML file and opened in your default brower to display the results

#Sample Inputs & Outputs

##Input

```
"Enter your paragraph:"
"This is my paragraph entry"
```

##Output
(this is the expected content of the .html file)

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>

  <body>
    <p>This is my paragraph entry</p>
  </body>
</html>
```

#Bonus

Implement a good looking default CSS style-sheet that also gets automatically generated.

Title: [6/23/2014] Challenge #168 [Easy] Final Grades - Test Data
Text: #Description:

Last week we had [[6/18/2014] Challenge #167 [Intermediate] Final Grades]
(http://www.reddit.com/r/dailyprogrammer/comments/28gq9b/6182014_challenge_167_intermediate_final_grades/)

For this challenge I generated data using excel. It was a bit time consuming but for the limited data set it was not too bad. But what if we needed 100 students? Or 1000? Or even 10000?

Sometimes it is useful to use programs to generate test data to test programs. For today's challenge your task is to generate the test data for that challenge.

#Input:

* n -- representing how many random student records to generate

Let us assume N will always be a positive number and I will let you decide what upper bound if any you want to set.

I would recommend running your solution on 1, 10, 100, 1000, 10000. Maybe post a sampling of 10 to show what you can generate.

#Output:

Should be a listing either via console out or a text file or other of your choice that is the test data. To remind you what a record looks like:

* (first name) , (last name) (score 1) (score 2) (score 3) (score 4) (score 5)

For example of a student roster see the [Intermediate] (http://www.reddit.com/r/dailyprogrammer/comments/28gq9b/6182014_challenge_167_intermediate_final_grades/) challenge's input
#Data:

To generate this data you will need to find a way to generate random first and last names and 5 scores (between 0 to 100)

#Optional:

Check your output and look for duplicate first and last names generated and remove the duplicates. It is up to you to decide how to do this.

Example would be if you generated "John , Smith" two times you would want to take action.

Also keep in mind the larger N values could more likely create duplicates. Consider a "give up" logic where you attempt to be unique but at some point have to accept that there will be some duplicates.

Title: [6/27/2014] Challenge #168 [Easy] String Index
Text: #What no hard?:

So my originally planned [Hard] has issues. So it is not ready for posting. I don't have another [Hard] so we are gonna do a nice [Easy] one for Friday for all of us to enjoy.

#Description:

We know arrays. We index into them to get a value. What if we could apply this to a string? But the index finds a "word". Imagine being able to parse the words in a string by giving an index. This can be useful for many reasons.

Example:

Say you have the String "The lazy cat slept in the sunlight."

If you asked for the Word at index 3 you would get "cat" back. If you asked for the Word at index 0 you get back an empty string "". Why an empty string at 0? Because we will not use a 0 index but our index begins at 1. If you ask for word at index 8 you will get back an empty string as the string only has 7 words. Any negative index makes no sense and return an empty string "".

#Rules to parse:

* Words is defined as [a-zA-Z0-9]+ so at least one of these and many more in a row defines a word.

* Any other character is just a buffer between words."
* Index can be any integer (this oddly enough includes negative value).
* If the index into the string does not make sense because the word does not exist then return an empty string.

#Challenge Input:

Your string:
"...You...!!!@!3124131212 Hello have this is a --- string Solved !!...? to test @\n\n\n#!#@#@%$^**#$@ Congratz this!!!!!!!!!!!!!!!one ---Problem\n\n"

Find the words at these indexes and display them with a " " between them: 12 -1 1 -100 4 1000 9 -1000 16 13 17 15

Title: [6/30/2014] Challenge #169 [Easy] 90 Degree 2D Array Rotate
Text: #Description:

Given a NxN size 2D array of numbers. Develop a way to rotate the data as if you rotated the data by 90 degrees clockwise.

#Example:

N = 3

Data:

```
1 2 3
4 5 6
7 8 9
```

Rotate 90 degrees:

```
7 4 1
8 5 2
9 6 3
```

Rotate it again 90 degrees:

```
9 8 7
6 5 4
3 2 1
```

#Challenge Input:

N = 10

```
1 2 3 4 5 6 7 8 9 0
0 9 8 7 6 5 4 3 2 1
1 3 5 7 9 2 4 6 8 0
0 8 6 4 2 9 7 5 3 1
0 1 2 3 4 5 4 3 2 1
9 8 7 6 5 6 7 8 9 0
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2
9 8 7 6 7 8 9 8 7 6
0 0 0 0 0 0 0 0 0 0
```

#Optional:
Show the 2D array at 90, 180, 270 degree clockwise from the original position.

Title: [7/7/2014] Challenge #170 [Easy] Blackjack Checker
Text: # [](#EasyIcon) _(Easy)_: Blackjack Checker

[Blackjack](https://en.wikipedia.org/wiki/Blackjack) is a very common card game, where the primary aim is to pick up cards until your hand has a higher value than everyone else but is less than or equal to 21. This challenge will look at the outcome of the game, rather than playing the game itself.

The value of a hand is determined by the cards in it.

* Numbered cards are worth their number - eg. a 6 of Hearts is worth 6.

* Face cards (JQK) are worth 10.

* Ace can be worth 1 or 11.

The person with the highest valued hand wins, with one exception - if a person has 5 cards in their hand and it has any value 21 or less, then they win automatically. This is called a 5 card trick.

If the value of your hand is worth over 21, you are 'bust', and automatically lose.

Your challenge is, given a set of players and their hands, print who wins (or if it is a tie game.)

Input Description

First you will be given a number, **N**. This is the number of players in the game.

Next, you will be given a further **N** lines of input. Each line contains the name of the player and the cards in their hand, like so:

    Bill: Ace of Diamonds, Four of Hearts, Six of Clubs

Would have a value of 21 (or 11 if you wanted, as the Ace could be 1 or 11.)

Output Description

Print the winning player. If two or more players won, print "Tie".

Example Inputs and Outputs

Example Input 1

    3
    Alice: Ace of Diamonds, Ten of Clubs
    Bob: Three of Hearts, Six of Spades, Seven of Spades
    Chris: Ten of Hearts, Three of Diamonds, Jack of Clubs

Example Output 1

    Alice has won!

Example Input 2
    4
    Alice: Ace of Diamonds, Ten of Clubs
    Bob: Three of Hearts, Six of Spades, Seven of Spades
    Chris: Ten of Hearts, Three of Diamonds, Jack of Clubs
    David: Two of Hearts, Three of Clubs, Three of Hearts, Five of Hearts, Six of Hearts

Example Output 2
    David has won with a 5-card trick!

Notes
Here's a tip to simplify things. If your programming language supports it, create enumerations (`enum`) for card ranks and card suits, and create structures/classes (`struct`/`class`) for the cards themselves - see [this example C# code](https://github.com/DropTableSpoon/Challenge170Easy/blob/master/Challenge170Easy/Cards/Card.cs).

For resources on using structs and enums if you haven't used them before (in C#): [structs](https://duckduckgo.com/l/?kh=-1&uddg=http%3A%2F%2Fmsdn.microsoft.com%2Fen-us%2Flibrary%2Fsaxz13w4.aspx), [enums](http://msdn.microsoft.com/en-us/library/sbbt4032.aspx).

You may want to re-use some code from your solution to [this challenge](http://www.reddit.com/r/dailyprogrammer/comments/24r50l/) where appropriate.

Title: [7/14/2014] Challenge #171 [Easy] Hex to 8x8 Bitmap
Text: #Description:

Today we will be making some simple 8x8 bitmap pictures. You will be given 8 hex values that can be 0-255 in decimal value (so 1 byte). Each value represents a row. So 8 rows of 8 bits so a 8x8 bitmap picture.

#Input:

8 Hex values.

##example:

18 3C 7E 7E 18 18 18 18

#Output:

A 8x8 picture that represents the values you read in.

For example say you got the hex value FF. This is 1111 1111 . "1" means the bitmap at that location is on and print something. "0" means nothing is printed so put a space. 1111 1111 would output this row:

```
xxxxxxxx
```

if the next hex value is 81 it would be 1000 0001 in binary and so the 2nd row would output (with the first row)

```
xxxxxxxx
x      x
```

##Example output based on example input:

```
   xx
  xxxx
 xxxxxx
 xxxxxx
  xx
  xx
  xx
  xx
```

#Challenge input:

Here are 4 pictures to process and display:

FF 81 BD A5 A5 BD 81 FF
AA 55 AA 55 AA 55 AA 55
3E 7F FC F8 F8 FC 7F 3E
93 93 93 F3 F3 93 93 93

#Output Character:

I used "x" but feel free to use any ASCII value you want. Heck if you want to display it using graphics, feel free to be creative here.

Title: [7/21/2014] Challenge #172 [Easy] ■■□□□■■□
Text: #Description

A portable bitmap is one of the oldest image formats around and grants access to very simple image creation and sharing. Today, you will be creating an image of this format.

A simple PBM program can be seen [here] (http://en.wikipedia.org/wiki/Netpbm_format)
(Note that we'll be creating the simplest version, a PBM, not PPM or PGM.)

But basically the program consists of the following:

* A 2byte string (usually 'P1') denoting the file format for that PBM

* 2 integers denoting the Width and Height of our image file respectively

* And finally, our pixel data - Whether a pixel is 1 - Black or 0 - White.

#Formal Inputs & Outputs

##Input description

On standard console input you should be prompted to enter a small piece of text ("programming", "proggit", "hello world" etc...)

##Output description

The output will be a .PBM file consiting of an image which contains the text you have entered

#Notes

/u/chunes has kindly mapped all alpha characters to their 0 1 equivalents, saving you a lot of time.

https://gist.github.com/anonymous/0ce707518d9e581499f5

Here is a worthwhile tutorial on the PBM format and programming for it

http://blog.plover.com/prog/perl/lines.html

The .PBM (you may also see it called NetPBM) is not very well supported any more, this makes actually viewing the PBM difficult as not many programs support it.

Feel free to download software which would render your .PBM to the screen but for all intents and purposes, the format is more important than the output cosidering the difficulty of viewing the image.

Title: [7/28/2014] Challenge #173 [Easy] Unit Calculator
Text: # [](#EasyIcon) _(Easy): Unit Calculator

You have a 30-centimetre ruler. Or is it a 11.8-inch ruler? Or is it even a 9.7-attoparsec ruler? It means the same thing, of course, but no-one can quite decide which one is the standard. To help people with this often-frustrating situation you've been tasked with creating a calculator to do the nasty conversion work for you.

Your calculator must be able to convert between metres, inches, miles and [attoparsecs](https://www.google.com/search?q=attoparsec). It must also be able to convert between kilograms, pounds, ounces and [hogsheads of Beryllium](http://www.wolframalpha.com/input/?i=mass+of+1+hogshead+of+berylliumm).

Input Description

You will be given a request in the format: **N** oldUnits to newUnits

For example:

    3 metres to inches

Output Description

If it's possible to convert between the units, print the output as follows:

    3 metres is 118.1 inches

If it's not possible to convert between the units, print as follows:

    3 metres can't be converted to pounds

Notes

Rather than creating a method to do each separate type of conversion, it's worth storing the ratios between all of the units in a 2-D array or something similar to that.

Title: [8/04/2014] Challenge #174 [Easy] Thue-Morse Sequences
Text: #Description:

The Thue-Morse sequence is a binary sequence (of 0s and 1s) that never repeats.
It is obtained by starting with 0 and successively calculating the Boolean complement
of the sequence so far. It turns out that doing this yields an infinite,
non-repeating sequence. This procedure yields 0 then 01, 0110, 01101001,
0110100110010110, and so on.

[Thue-Morse Wikipedia Article] (http://en.wikipedia.org/wiki/Thue%E2%80%93Morse_sequence) for more information.

#Input:
Nothing.

#Output:
Output the 0 to 6th order Thue-Morse Sequences.

##Example:

    nth   Sequence

```
==========================================================================
0               0
1               01
2               0110
3               01101001
4               0110100110010110
5               01101001100101101001011001101001
6               0110100110010110100101100110100110010110011010010110100110010110
```

#Extra Challenge:

Be able to output any nth order sequence. Display the Thue-Morse Sequences for 100.

Note: Due to the size of the sequence it seems people are crashing beyond 25th order or the time it takes is very long. So how long until you crash. Experiment with it.

Title: [8/11/2014] Challenge #175 [Easy] Bogo!
Text: #Description

A bogo sort is a purposefully inefficient algorithm for sorting a sequence. Today we will be using this for strings to test for equality.

Here is wikipedias entry for a [Bogo-Sort](http://en.wikipedia.org/wiki/Bogo-sort)

#Inputs & Outputs

Given a scrambled string N and another string M. You must sort N so that it matches M. After it has been sorted, it must output how many iterations it took to complete the sorting.

##Sample Inputs & Outputs

Input:

```
Bogo("lolhe","Hello")
```

Output:

```
1456 iterations
```

#Bonus

For a bit of fun, the LEAST efficient algorithm wins. Check out the bogo-bogo sort, an algorithm that's designed not to succeed before the heat death of the universe

http://www.dangermouse.net/esoteric/bogobogosort.html

If you have designed an algorithm but it still hasn't finished sorting, if you can prove it WILL sort, you may post your proof.

Title: [8/18/2014] Challenge #176 [Easy] Spreadsheet Developer pt. 1: Cell Selection
Text: # [](#EasyIcon) _(Easy)_: Spreadsheet Developer pt. 1: Cell Selection

Today and on Wednesday we will be developing a terminal-based spreadsheet package somewhat like [ed](http://en.wikipedia.org/wiki/Ed_%28text_editor%29) used to be. Today we'll be taking a look at the mechanism for selecting ranges of cells from textual data.

In the spreadsheet, each cell may be represented by one of two systems:

* Co-ordinate in memory. This looks like `[X, Y]` and represents the cell's position in the internal array or memory structure. X and Y begin at 0.

* Column-row syntax. This looks like `A3`, `B9` or `AF140` and is created from the row's alphabetical header and the column number, starting from 1. You may be more familiar with this syntax in programs such as Excel, Lotus 1-2-3 (lol as if) or LibreOffice Calc. Pay close attention to the naming of the columns - it's not a simple Base-26 system as you may expect. It's called [bijective Base-26](http://en.wikipedia.org/wiki/Hexavigesimal#Bijective_base-26).

Now to select a range, we need another syntax. The following symbols apply in order of precedence, top-to-bottom:

* A formula may have one or more `:`s (colons) in it. If so, a rectangle of cells is selected. This behaves the same way in Excel. Such a selection is called a **range**. For example, [`A3:C7` looks like this](http://i.imgur.com/yfdT43W.png).

* A formula may have one or more `&`s (ampersands) in it. If so, both the cell/range specified to the left and right are selected. This is just a concatenation. For example, [`A1:B2&C3:D4` looks like this](http://i.imgur.com/rnYmCtG.png).

* A formula may have one `~` (tilde) symbol in it. If so, any cells specified *before* the tilde are added to the final selection, and any cells *after* the tilde are removed from the final selection of cells. For example, if I enter `A1:C3~B2` then all cells from A1 to C3 *except* B2 are selected, [which looks like this](http://i.imgur.com/7fop7wJ.png). (This acts like a [relative complement](http://en.wikipedia.org/wiki/Complement_%28set_theory%29#Relative_complement) of the right hand side in the left hand side.)

Your challenge today will be, given a selection string like `A3:C6&D1~B4&B5`, print the co-ordinates of all of the selected cells, along with the count of selected cells.

Formal Inputs and Outputs

Input Description
You will be given a selection string like `A3:C6&D1~B4&B5` on one line.

Output Description
First, print the number of cells selected (eg. if 50 cells are selected, print `50`.)

Then, on separate lines, print the co-ordinates of each selected cell.

Example Inputs and Outputs

Example Input
```
B1:B3&B4:E10&F1:G1&F4~C5:C8&B2
```

Example Output

```
29
1, 0
1, 2
1, 3
1, 4
1, 5
1, 6
1, 7
1, 8
1, 9
2, 3
2, 8
2, 9
3, 3
```

3, 4
3, 5
3, 6
3, 7
3, 8
3, 9
4, 3
4, 4
4, 5
4, 6
4, 7
4, 8
4, 9
5, 0
6, 0
5, 3

Title: [8/22/2014] Challenge #176 [Easy] Pivot Table
Text: #Description:

An interesting way to represent data is a pivot table. If you use spreadsheet programs like Excel you might have seen these before. If not then you are about to enjoy it.

Say you have data that is related in three parts. We can field this in a table with column and rows and the middle intersection is a related field. For this challenge you will need to make a pivot table for a wind energy farm. These farms of wind mills run several windmills with tower numbers. They generate energy measured in kilowatt hours (kWh).

You will need to read in raw data from the field computers that collect readings throughout the week. The data is not sorted very well. You will need to display it all in a nice pivot table.

Top Columns should be the days of the week.
Side Rows should be the tower numbers and the data in the middle the total kWh hours produced for that tower on that day of the week.

#input:

The challenge input is 1000 lines of the computer logs. You will find it [HERE - gist of it](https://gist.github.com/coderd00d/ca718df8e633285885fa)

The log data is in the format:

(tower #) (day of the week) (kWh)

#output:
A nicely formatted pivot table to report to management of the weekly kilowatt hours of the wind farm by day of the week.

#Code Solutions:
I am sure a clever user will simply put the data in Excel and make a pivot table. We are looking for a coded solution. :)

Title: [8/25/2014] Challenge #177 [Easy] Quicksort
Text: # [](#EasyIcon) *(Easy)*: Quicksort

On a daily basis we take advantage of the power of a language's standard library. One of the common functions within such libraries is for sorting sets of data. This saves you some time so you don't have to write it yourself. But what about the occasions when you don't have a standard library?

You might be tempted to implement a sorting algorithm such as [insertion sort](http://en.wikipedia.org/wiki/Insertion_sort) or [bubble sort](http://en.wikipedia.org/wiki/Bubble_sort). However, while being simple, they are slow and inefficient, with O(n^(2)) running time - meaning on average, doubling the length of the list to be sorted increases the running time by a factor of four. Now this can be useful sometimes, eg. if you need to write a tiny program, like on an Arduino. However, in the vast majority of cases this is bad.

Luckily there are alternate methods of sorting. Today, we will be looking at a method known as [quicksort](http://en.wikipedia.org/wiki/Quicksort). This involves:

1. Start with the whole list.

2. Pick a random value from the list - it does not have to be from the middle. This will be our **pivot**.

3. Reorder the list, moving (in no particular order) everything that is smaller than the pivot to the left of it and everything that's greater than the pivot to the right of it.
Now, the list to the left of, not including, the pivot is known as list S, and the list to the right of, not including, the pivot is known as list G.
S and G don't have to be created in order, just so long as they are all smaller or greater than the pivot respectively.
4. Now repeat step 2 onward for lists S and G. If the list only contains zero or one items, you can stop, as it's by default sorted. If either only contains 2 items, it might make it quicker to just compare and swap if necessary instead of doing the whole sorting procedure.

You challenge today is, given an arbitrarily long list of real numbers, sort them using your own, *non-library* version of quicksort.

Formal Inputs & Outputs

Input

You will take an integer **N**. This will be the size of our list. You will then take a further **N** real (ie. floating/decimal) numbers on separate lines. This is the content of our list.

Output

Output the list after sorting with your version of quicksort. This should also be on separate line.

Notes

If you have not already learned it, this is a golden opportunity to use and learn [recursion](http://en.wikipedia.org/wiki/Recursion_%28computer_science%29). Remember, using your language's built-in sorting implementation defeats the purpose of this exercise, and if you do post such a solution, prepare for a sarcastic response.

Title: [9/01/2014] Challenge #178 [Easy] Transformers: Matrices in Disguise, pt. 1
Text: # [](#EasyIcon) *(Easy)*: Transformers: Matrices in Disguise, pt. 1

Or, rather, transform*ation*s. Today we'll be doing a bit of basic geometry. We'll be writing a program which will take a point in 2-dimensional space, represented as `(X, Y)` (where X and Y can be decimal and negative), transform them a number of times in different ways and then find the final position of the point.

Your program must be able to do the following:

* Translations - ie. offsetting the X and Y co-ordinates by a given amount http://i.imgur.com/3jI4sGI.png

* Rotations by an arbitrary angle around a given point http://i.imgur.com/9c0ji7c.png

* Scale relative to a point http://i.imgur.com/vHUfXv2.png

* Reflection over the X or Y axis http://i.imgur.com/X6JH6pT.png

Formal Inputs & Outputs

Input

You will take an starting point `(X, Y)`, such as:

```
(3, 4)
```

On new lines, you will then take commands in the format:

```
translate(A, B)     - translate by (A, B)
rotate(A, B, C)     - rotate around (A, B) by angle C (in radians) clockwise
scale(A, B, C)      - scale relative to (A, B) with scale-factor C
reflect(axis)       - reflect over the given axis
finish()            - end input and print the modified location
```

Where `axis` is one of `X` or `Y`.

Output

Print the final value of `(X, Y)` in the format:

```
(2.5, -0.666666)
```

Test Case

Test Case Input

```
(0, 5)
translate(3, 2)
scale(1,3,0.5)
rotate(3,2,1.57079632679)
reflect(X)
translate(2,-1)
scale(0,0,-0.25)
rotate(1,-3,3.14159265359)
reflect(Y)
```

Test Case Output

```
(-4, -7)
```

Notes

I want to say two things. First, this may be a good opportunity to learn your language's 2-D drawing capabilities - every time a command is given, represent it on an image like I have done with the examples, so you can see the path the co-ordinate has taken. Secondly, this is a multi-part challenge. I'm not sure how many parts there will be, however it may be a good idea to prepare for more possible commands (or, if you're crazy enough to use Prolog - you know who you are - write an EBNF parser like last time, lol.) If you know how, it would be clever to start using matrices for transformations now rather than later.

Title: [9/08/2014] Challenge #179 [Easy] You make me happy when clouds are gray...scale
Text: #Description

The 'Daily Business' newspaper are a distributor of the most recent news concerning business. They have a problem though, there is a new newspaper brought out every single day and up to this point, all of the images and advertisements featured have been in full colour and this is costing the company.

If you can convert these images before they reach the publisher, then you will surely get a promotion, or at least a raise!

#Formal Inputs & Outputs

##Input description

On console input you should enter a filepath to the image you wish to convert to grayscale.

##Output description

The program should save an image in the current directory of the image passed as input, the only difference being that it is now in black and white.

#Notes/Hints

There are several methods to convert an image to grayscale, the easiest is to sum up all of the RGB values and divide it by 3 (The length of the array) and fill each R,G and B value with that number.

For example

```
RED = (255,0,0)
```

Would turn to

```
(85,85,85)      //Because 255/3 == 85.
```

There is a problem with this method though,

```
GREEN = (0,255,0)
```

brings back the exact same value!

There is a formula to solve this, see if you can find it.

Title: [9/15/2014] Challenge#180 [Easy] Look'n'Say
Text: #Description

The Look and Say sequence is an interesting sequence of numbers where each term is given by describing the makeup of the previous term.

The 1st term is given as 1. The 2nd term is 11 ('one one') because the first term (1) consisted of a single 1. The 3rd term is then 21 ('two one') because the second term consisted of two 1s. The first 6 terms are:

```
1
11
21
1211
```

111221
312211

#Formal Inputs & Outputs

##Input
On console input you should enter a number N

##Output
The Nth Look and Say number.

#Bonus

Allow any 'seed' number, not just 1. Can you find any interesting cases?

Title: [09/22/2014] Challenge #181 [Easy] Basic Equations
Text: # [](#EasyIcon) _(Easy)_: Basic Equations

Today, we'll be creating a simple calculator, that we may extend in later challenges. Assuming you have done basic algebra, you may have seen equations in the form [`y=ax+b`](http://latex.codecogs.com/gif.latex?y%3Dax+b), where `a` and `b` are constants. This forms a graph of a straight line, when you plot `y` in respect to `x`. If you have not explored this concept yet, you can visualise a linear equation such as this using [this online tool](http://www.mathopenref.com/linearexplorer.html), which will plot it for you.

The question is, how can you find out where two such 'lines' intersect when plotted - ie. when the lines cross? Using algebra, you can solve this problem easily. For example, given [`y=2x+2`](http://latex.codecogs.com/gif.latex?y%3D2x+2) and [`y=5x-4`](http://latex.codecogs.com/gif.latex?y%3D5x-4), how would you find out where they intersect? This situation would look like [this](http://i.imgur.com/wLr5Aei.png). Where do the red and blue lines meet? You would substitute `y`, forming one equation, [`2x+2=5x-4`](http://latex.codecogs.com/gif.latex?2x+2%3D5x-4), as they both refer to the same variable `y`. Then, subtract one of the sides of the equation from the other side - like [`2x+2-(2x+2)=5x-4-(2x+2)`](http://latex.codecogs.com/gif.latex?2x+2-%282x+2%29%3D5x-4-%282x+2%29) which is the same as [`3x-6=0`](http://latex.codecogs.com/gif.latex?3x-6%3D0) - to solve, move the -6 to the other side of the `=` sign by adding 6 to both sides, and divide both sides by 3: [`x=2`](http://latex.codecogs.com/gif.latex?x%3D2). You now have the `x` value of the co-ordinate at where they meet, and as `y` is the same for both equations at this point (hence why they intersect) you can use either equation to find the `y` value, [like so](http://latex.codecogs.com/gif.latex?%282x+2%5C%3B%20%5Ctextup%7Bwhere%7D%5C%3B%20x%3D2%29%3D2%282%29+2%3D4+2%3D6). So the co-ordinate where they insersect is `(2, 6)`. Fairly simple.

Your task is, given two such linear-style equations, find out the point at which they intersect.

Formal Inputs and Outputs

Input Description

You will be given 2 equations, in the form `y=ax+b`, on 2 separate lines, where `a` and `b` are constants and `y` and `x` are variables.

Output Description

You will print a point in the format `(x, y)`, which is the point at which the two lines intersect.

Sample Inputs and Outputs

Sample Input

```
y=2x+2
y=5x-4
```

Sample Output

(2, 6)

Sample Input

y=-5x
y=-4x+1

Sample Output

(-1, 5)

Sample Input

y=0.5x+1.3
y=-1.4x-0.2

Sample Output

(-0.7895, 0.9053)

Notes

If you are new to the concept, this might be a good time to learn [regular expressions](http://www.regular-expressions.info/tutorial.html). If you're feeling more adventurous, write a little parser.

Extension

Draw a graph with 2 lines to represent the inputted equations - preferably with 2 different colours. Draw a point or dot representing the point of intersection.

Title: [29/09/2014] Challenge #182 [Easy] The Column Conundrum
Text: # [](#EasyIcon) **(Easy)**: The Column Conundrum
Text formatting is big business. Every day we read information in one of several formats. Scientific publications often have their text split into [two columns, like this](https://isotropic.org/papers/chicken.pdf). Websites are often bearing one major column and a sidebar column, such as Reddit itself. Newspapers very often have three to five columns. You've been commisioned by some bloke you met in Asda to write a program which, given some input text and some numbers, will split the data into the appropriate number of columns.

Formal Inputs and Outputs

Input Description

To start, you will be given 3 numbers on one line:

<number of columns> <column width> <space width>

* *number of columns*: The number of columns to collect the text into.
* *column width*: The width, in characters, of each column.
* *space width*: The width, in spaces, of the space between each column.

After that first line, the rest of the input will be the text to format.

Output Description

You will print the text formatted into the appropriate style.

You do not need to account for words and spaces. If you wish, cut a word into two, so as to keep the column width constant.

Sample Inputs and Outputs

Sample Input

Input file [is available here](https://gist.githubusercontent.com/Quackmatic/b19f592be2c0ee9e22d7/raw/45457a757d1f126d94a4736354c78906eeb819a3/c182e-input.txt). *(NB: I promise this input actually works this time, haha.)*

Sample Output

Outout, according to my solution, [is available here](https://gist.githubusercontent.com/Quackmatic/1ef9af9f3989e48ee1c4/raw/4cbcd546b7bc1dd415b9a804eb93e671d927cb43/c182e-output.txt). I completed the Extension challenge too - you do not have to account for longer words if you don't want to, or don't know how.

Extension

Split words correctly, like in my sample output.

Title: [10/06/2014] Challenge #183 [Easy] Semantic Version Sort
Text: # [](#EasyIcon) __(Easy)__: Semantic Version Sort

Semantic Versioning, or *Semver* as it's known on the streets, is an attempt to standardise the way that software versions are incrementally changed. In the world there are many different pieces of software whose developers have conflicting ideas about how software should be developed. For example, [Dwarf Fortress](http://www.bay12games.com/dwarves/) is currently at version 0.40.13, whereas [Google Chrome](https://en.wikipedia.org/wiki/Google_Chrome) (which has been around for 2 years *less* than Dwarf Fortress) is currently at version 37.0.2062.124. How can those version numbers even be compared? They both represent around the same progress of development but in totally different ways. Semantic versioning aims to solve this problem by splitting the version string into 3, 4 or 5 parts:

    <major>.<minor>.<patch>-<label>+<metadata>

* **major**: Increased when your program changes in a way that makes it incompatible with older versions (major changes) - like the Python 2 to Python 3 change which, in order to make progress, broke a lot of existing programs.
* **minor**: Increased when you add functionality but keep compatibility and don't change existing bits of the API (minor changes) - for example, adding a new section of a standard library to a programming language.
* **patch**: Increased when you make minor functionality changes or bug fixes, like adding a new keyboard shortcut.
* **label**: Used to indicate pre-release program status, such as *beta*, *alpha* or *rc2* (release candidate 2.)
* **metadata**: Used to describe build metadata when a version is in the early development stages - this might include the build date of the program.

For the purpose of this challenge, you will be sorting a list of Semantic Versions into chronological order, and you will do it like so:

1. First, compare the major version.

2. If they are the same, compare the minor version.

3. If they are the same, compare the patch version.

4. If those are all the same, check if the version has a label - ignore the content of the label. A version with a label (prerelease) comes before one without a label (final release.)

5. Ignore the build metadata.

If the semantic versions are still the same at this point, they can be considered the same version. For the purpose of this challenge we won't attempt to parse the label - but if you're feeling up to you can give it a try!

The full specification for Semantic Versioning [can be found here](http://semver.org/).

Formal Inputs and Outputs

Input Description

You will first be given a number **N**. You will then be given **N** more lines, each one with a semantic version.

Output Description

You will print the versions in chronological order, as described by the rules above.

Sample Inputs and Outputs

Sample Input

```
7
2.0.11-alpha
0.1.7+amd64
0.10.7+20141005
2.0.12+i386
1.2.34
2.0.11+i386
20.1.1+i386
```

Sample Output

```
0.1.7+amd64
0.10.7+20141005
1.2.34
2.0.11-alpha
2.0.11+i386
2.0.12+i386
20.1.1+i386
```

Tip

If your chosen language supports it, create a `SemanticVersion` record/structure with the appropriate fields. If your language supports it, overload the comparison (`<`, `>`) operators to compare for you.

If your language does not support sorting of data structures by default, you could adjust your solution to [the Quicksort](/r/dailyprogrammer/comments/2ejl4x/) challenge to work with this one.

Title: [10/13/2014] Challenge #184 [Easy] Smart Stack List
Text: #Description:

We all know the famous link list. We can use these to hold data in a linear fashion. The link list can be used to implement a stack as well for example.

For this challenge you will need to develop a smart stack list. So what makes this link list so smart? This link list will behave like a stack but also maintain an ascending sorted order of the data as well. So our link list maintains 2 orders (stack and sorted)

In today's world link list data structures are part of many development platforms. For this challenge you will be implementing this and not using already pre-made frameworks/standard link lists code that you call. You have to implement it and all the features.

The Challenge will have 2 steps.

* Implement your smart link list
* Test your smart link list

#Data:

The smart link list will hold integer data.

#Methods:

The smart link list must support these methods or operations.

* Push - Push a number on top of the stack (our link list is a stack)
* Pop - Pop the number off the top of the stack
* Size - how many numbers are on your stack
* Remove Greater - remove all integers off the stack greater in value then the given number
* Display Stack - shows the stack order of the list (the order they were pushed from recent to oldest)
* Display Ordered - shows the integers sorted from lowest to highest.

#Smart list:

One could make a stack and when you do the display ordered do a sort. But our smart list must have a way so that it takes O(n) to display the link list in stack order or ascending order. In other words our link list is always in stack and sorted order. How do you make this work? That is the real challenge.

#Test your list:

Develop a quick program that uses your smart stack list.

* Generate a random number between 1-40. Call this number n.
* Generate n random numbers between -1000 to 1000 and push them on your list
* Display stack and sorted order
* Generate a random number between -1000 and 1000 can call it x
* Remove greater X from your list. (all integers greater than x)
* Display stack and sorted order
* pop your list (size of list / 2) times (50% of your list is removed)
* Display stack and sorted order

Title: [10/20/2014] Challenge #185 [Easy] Generated twitter handles
Text: #Description

For those that don't tweet or know the workings of Twitter, you can reply to 'tweets' by replying to that user with an @ symbol and their username.

Here's an example from [John Carmack's](https://twitter.com/ID_AA_Carmack) twitter.

His **initial** **tweet**

@ID_AA_Carmack : "Even more than most things, the challenges in computer vision seem to be the gulf between theory and practice."
And **a** **reply**

@professorlamp : @ID_AA_Carmack Couldn't say I have too much experience with that

You can see, the '@' symbol is more or less an integral part of the tweet and the reply. Wouldn't it be neat if we could think of names that incorporate the @ symbol and also form a word?

e.g.

@tack -> (attack)

@trocious ->(atrocious)

#Formal Inputs & Outputs

##Input description

As input, you should give a word list for your program to scout through to find viable matches. The most popular word list is good ol' [enable1.txt](https://code.google.com/p/dotnetperls-controls/downloads/detail?name=enable1.txt)

/u/G33kDude has supplied an even bigger text file. I've hosted it on my site over [here](http://www.joereynoldsaudio.com/WordList.txt) , I recommend 'saving as' to download the file.

##Output description

Both outputs should contain the 'truncated' version of the word and the original word. For example.

    @tack : attack

There are two outputs that we are interested in:

* The 10 longest twitter handles sorted by length in descending order.
* The 10 shortest twitter handles sorted by length in ascending order.

#Bonus

I think it would be even better if we could find words that have 'at' in them at any point of the word and replace it with the @ symbol. Most of these wouldn't be valid in Twitter but that's not the point here.

For example

r@@a -> (ratata)

r@ic@e ->(raticate)

dr@ ->(drat)

Title: [10/27/2014] Challenge #186 [Easy] Admin Schmadmin
Text: #Description

"I'm sorry we had to call you in at such small notice but our last admin royally screwed us over. I don't suppose you can have a scout through the files and see if there's any remnants of that slimeball left in our system can you? Any leftover documents, programs, CV's, ANYTHING you can find about him, I need it so I can finish him."

A few weeks pass

...Congratulations!

You've been hired as a temp to do some administrative duties that involve digging through the records of the filesystem in search for any hints as to where the ex-employee may have fled to. But first, you'll need some training. I've assigned you a few simple tasks that should build your command line skills to that of an adequate admin.

#Formal Inputs & Outputs

For this task, you are given a tasklist of tasks to perform. Each task has a bulleted point and a summary. The bulleted point contains the dialogue of what the manager wants you to perform, the summary can be seen as a sort of 'technical overview' of what needs to be done.

##Input description

As input, you are expected to execute commands into your terminal that correspond to the given task on the tasklist.

##Output description

The program should output the expected output of your command.

#Tasklist

"Okay employee, I've hired you now get to work! Listen carefully to what I have to say, I'm not going to say it twice!..."

* "Bring up that list of his most used files, let's see what that scumbag's been up to!"

Summary : Get the 20 last used documents from the system and sort by the date they were modified.

---

* "Great, can you email that to me?"

Summary : Output the above command to a .txt file.

---

* "Hmm, still nothing. Maybe the answer is right in front of us? Get the last commands he used on the console!"

Summary : Retrieve the last 10 commands used on the console.

---

* "AHA, this looks good I'll just email it to my...what the? What's going on!..." *10 minutes later* "He crashed our machine! I knew he had some software throttling our machines, find out what's causing it, and fix it!"

Summary : Get the 10 most CPU-heavy processes in descending order.

---

* "wait, wait, WAIT! Before you go any further. Let's look through the error logs! I won't be able to understand them and you don't have access to most of what's needed but if you could link them to my tech team, I'm sure they could figure it out!

Summary : Retrieve the last 20 error logs/messages and output these as a formatted HTML table

---

* "Okay, now we're getting somewhere. Let's put the nail in the coffin. Bruteforce it. Search every file, every directory, every nook and cranny for any .txt files, any .pdf and any .exe files"

Summary : Retrieve all txt/pdf/exe files on the machine (You do not need to do the whole machine, just 1 drive is enough, or less if your machine is struggling).

"Thanks kid, you saved our bacon! Now get out."

#Notes/Hints

Beginners, consider using a shell environent for this. For windows I recommend Powershell. I'm not a Unix man but I hear the default shell is more than up to this task. Doing this in a programming language will prove to be a lot of work, choose a shell if you want your sanity.

Title: [11/03/2014] Challenge #187 [Easy] A Flagon of Flags
Text: # [](#EasyIcon) **(Easy)**: A Flagon of Flags

In the command-line world, programs are operated not with graphical user interfaces but with command line flags. These flags are what the operator uses to pass parameters to the program. The standard form of flag starts with a double hyphen `--` and consists of a word in `lower-case-separated-by-hyphens`. For example, to forcefully remove a directory recursively on Unix based systems, the command used may be:

```
rm --recursive --force dir/
```

Here, the `recursive` and `force` flags have been enabled, which the program detects and changes its behaviour accordingly. Alternatively, many programs allow a *short-form* of command-line flag. These flags are one letter long andn start with a single hyphen `-`. For example, the above command can be reduced to:

```
rm -r -f dir/
```

This is much shorter, so commonly used flags are often abbreviated as such. An even shorter form merges several of these flags into one flag. This is still separated by a hyphen but consists of multiple letters. For example, in the `tar` command on Unix based systems, the `-x -z -v` flags can be merged into `-xzv` with the exact same meaning. The above `rm` command looks like this:

```
rm -rf dir/
```

This is even more convenient for a terminal operator to enter. Today, you will write a program which will accept a string of flags in the above formats and output which flags were activated.

Formal Inputs and Outputs

Input Description

You will first input a number **N**. You will then accept **N** lines of input in the format:

```
f:force
```

This is a *short-form definition*; this particular example denotes that the flag `-f` is equivalent to the flag `--force`. Lastly you are to accept one further line of input containing the flags and other parameters passed to the program. Remember that programs can accept parameters that are *not* flags. These don't start with a hyphen and there may be several of them. For example,

```
-Q -rf --no-preserve-root directory1/ directory2/
```

In which the flags given are `-Q` `-rf` (same as `-r -f`) and `--no-preserve-root`, and the parameters are `directory1/` and `directory2/`. Remember the `Q`, `r` and `f` flags are defined in the short-form definition format above.

Output Description

You are to output a list of the *full names* of all of the flags entered (eg. `force` rather than `f`), as well as all of the parameters entered. Alternatively, if a short-form flag is entered that doesn't have a difinition, print an error.

Sample Inputs and Outputs

Sample Input

```
4
a:all
f:force
n:networking
N:numerical-list
-aN 12 --verbose 192.168.0.44
```

(not all commands need a short-form expression; here, `verbose` only exists as the long-form.)

Sample Output

```
flag: all
flag: numerical-list
parameter: 12
flag: verbose
parameter: 192.168.0.44
```

Extension (Intermediate)

Some flags may have a parameter. For example, a flag `output` may take a filename parameter. The long form of this would be:

```
--output=log.txt
```

The short form of this would be:

```
-o log.txt
```

The short form has no equals sign, but the long form does. The short form can still be used as a combination, like

```
-rxo log.txt
```

Would activate the `r` and `x` flags, along with setting the value of `o` to `log.txt`. In this case, print the output like so:

```
flag: output (value: log.txt)
```

To denote that a flag can take a parameter, its input short-form definition is prefixed with a star `*`, like so:

```
*o:output
```

Sample Extension Input

```
6
a:all
*A:address
f:force
n:networking
N:numerical-list
*o:output
-aNo output-dir/file.txt 12 --verbose --address=192.168.0.44
```

Sample Extension Output

```
flag: all
flag: numerical-list
flag: output (value: output-dir/file.txt)
parameter: 12
flag: verbose
flag: address (value: 192.168.0.44)
```

Notes and Further Reading

Here is a [StackOverflow post](http://stackoverflow.com/questions/2160083/what-is-the-general-syntax-of-a-unix-shell-command/2160165#2160165) describing the standard in greater detail for command line flags.

Title: [2014-11-10] Challenge #188 [Easy] yyyy-mm-dd
Text: #Description:

iso 8601 standard for dates tells us the proper way to do an extended day is yyyy-mm-dd

* yyyy = year
* mm = month
* dd = day

A company's database has become polluted with mixed date formats. They could be one of 6 different formats

* yyyy-mm-dd
* mm/dd/yy
* mm#yy#dd
* dd*mm*yyyy
* (month word) dd, yy
* (month word) dd, yyyy

(month word) can be: Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Note if is yyyy it is a full 4 digit year. If it is yy then it is only the last 2 digits of the year. Years only go between 1950-2049.

#Input:

You will be given 1000 dates to correct.

#Output:

You must output the dates to the proper iso 8601 standard of yyyy-mm-dd

#Challenge Input:

https://gist.github.com/coderd00d/a88d4d2da014203898af

#Posting Solutions:

Please do not post your 1000 dates converted. If you must use a gist or link to another site. Or just show a sampling

#Challenge Idea:
Thanks to all the people pointing out the iso standard for dates in last week's intermediate challenge. Not only did it inspire today's easy challenge but help give us a weekly topic. You all are awesome :)

Title: [2014-11-17] Challenge #189 [Easy] Hangman!

Text: We all know the classic game hangman, today we'll be making it. With the wonderful bonus that we are programmers and we can make it as hard or as easy as we want. [here](http://www.joereynoldsaudio.com/wordlist.txt) is a wordlist to use if you don't already have one. That wordlist comprises of words spanning 3 - 15+ letter words in length so there is plenty of scope to make this interesting!

#Rules

For those that don't know the rules of hangman, it's quite simple.

There is 1 player and another person (in this case a computer) that randomly chooses a word and marks correct/incorrect guesses.

The steps of a game go as follows:

* Computer chooses a word from a predefined list of words
* The word is then populated with underscores in place of where the letters should.
('hello' would be '_ _ _ _ _')
* Player then guesses if a word from the alphabet [a-z] is in that word
* If that letter is in the word, the computer replaces all occurences of '_' with the correct letter
* If that letter is NOT in the word, the computer draws part of the gallow and eventually all of the hangman until he is hung (see [here](http://en.wikipedia.org/wiki/Hangman_%28game%29) for additional clarification)

This carries on until either

* The player has correctly guessed the word without getting hung

or

* The player has been hung

#Formal inputs and outputs

##input description

Apart from providing a wordlist, we should be able to choose a difficulty to filter our words down further. For example, hard could provide 3-5 letter words, medium 5-7, and easy could be anything above and beyond!

On input, you should enter a difficulty you wish to play in.

##output description

The output will occur in steps as it is a turn based game. The final condition is either win, or lose.

#Clarifications

* Punctuation should be stripped before the word is inserted into the game ("administrator's" would be "administrators")

Title: [2014-11-24] Challenge #190 [Easy] Webscraping sentiments
Text: #Description

Webscraping is the delicate process of gathering information from a website (usually) without the assistance of an API. Without an API, it often involves finding what ID or CLASS a certain HTML element has and then targeting it. In our latest challenge, we'll need to do this (you're free to use an API, but, where's the fun in that!?) to find out the overall sentiment of a sample size of people.

We will be performing very basic sentiment analysis on a YouTube video of your choosing.

#Task

Your task is to scrape N (You decide but generally, the higher the sample, the more accurate) number of comments from a YouTube video of your choice and then analyse their sentiments based on a short list of happy/sad keywords

Analysis will be done by seeing how many Happy/Sad keywords are in each comment. If a comment contains more sad keywords than happy, then it can be deemed sad.

Here's a basic list of keywords for you to test against. I've ommited expletives to please all readers...

happy = ['love','loved','like','liked','awesome','amazing','good','great','excellent']

sad = ['hate','hated','dislike','disliked','awful','terrible','bad','painful','worst']

Feel free to share a bigger list of keywords if you find one. A larger one would be much appreciated if you can find one.

#Formal inputs and outputs
##Input description

On console input, you should pass the URL of your video to be analysed.

##Output description
The output should consist of a statement stating something along the lines of -

"From a sample size of" N "Persons. This sentence is mostly" [Happy|Sad] "It contained" X "amount of Happy keywords and" X "amount of sad keywords. The general feelings towards this video were" [Happy|Sad]

#Notes

As pointed out by /u/pshatmsft , YouTube loads the comments via AJAX so there's a slight workaround that's been posted by /u/threeifbywhiskey .

Given the URL below, all you need to do is replace FullYoutubePathHere with your URL
https://plus.googleapis.com/u/0/_/widget/render/comments?first_party_property=YOUTUBE&href=FullYoutubePathHere

Remember to append your url in full (https://www.youtube.com/watch?v=dQw4w9WgXcQ as an example)

#Hints

The string for a Youtube comment is the following

```
<div class="CT">Youtube comment here</div>
```

Title: [2014-12-1] Challenge #191 [Easy] Word Counting
Text: You've recently taken an internship at an up and coming lingustic and natural language centre. Unfortunately, as with real life, the professors have allocated you the mundane task of counting every single word in a book and finding out how many occurences of each word there are.

To them, this task would take hours but they are unaware of your programming background (They really didn't assess the candidates much). Impress them with that word count by the end of the day and you're surely in for more smooth sailing.

#Description

Given a text file, count how many occurences of each word are present in that text file. To make it more interesting we'll be analyzing the free books offered by [Project Gutenberg](https://www.gutenberg.org)

The book I'm giving to you in this challenge is an [illustrated monthly on birds](https://www.gutenberg.org/cache/epub/47498/pg47498.txt). You're free to choose other books if you wish.

#Inputs and Outputs

##Input

Pass your book through for processing

##Output

Output should consist of a key-value pair of the word and its word count.

##Example

```
{'the' : 56,
'example' : 16,
'blue-tit' : 4,
'wings' : 75}
```

#Clarifications

For the sake of ease, you don't have to begin the word count when the book starts, you can just count all the words in that text file (including the boilerplate legal stuff put in by Gutenberg).

#Bonus

As a bonus, only extract the book's contents and nothing else.

Title: [2014-12-8] Challenge #192 [Easy] Carry Adding
Text: # [](#EasyIcon) **(Easy)**: Carry Adding

When you were first learning arithmetic, the way most people were tought to set out addition problems was like follows:

```
23+456=

 23
+456
 ---
 479
 ---
```

Look familiar? And remember how, if the number went above 10, you put the number below the line:

```
 559
+447
 ---
1006
 ---
 11
```

Those `1`s under the line are called the *carry* values - they are 'carried' from one column into the next. In today's challenge, you will take some numbers, add them up and (most importantly) display the output like it is shown above.

Formal Inputs and Outputs

Input Description

You will accept a list of non-negative integers in the format:

```
559+447+13+102
```

Such that the carry value will never be greater than 9.

Output Description

You are to output the result of the addition of the input numbers, in the format shown above.

Sample Inputs and Outputs

Sample Input

```
23+9+66
```

Sample Output

```
 23
  9
 66
 --
 98
 --
  1
```

Sample Input

```
8765+305
```

Sample Output

```
 8765
  305
 ----
 9070
  ---
 1 1
```

Sample Input

```
12+34+56+78+90
```

Sample Output

```
 12
 34
 56
 78
 90
---
270
---
22
```

Sample Input

```
999999+1
```

Sample Output

```
 999999
      1
-------
1000000
-------
111111
```

Extension

Extend your program to handle non-integer (ie. decimal) numbers.

Title: [2014-12-15] Challenge #193 [Easy] A Cube, Ball, Cylinder, Cone walk into a warehouse
Text: #Description:

An international shipping company is trying to figure out how to manufacture various types of containers. Given a volume they want to figure out the dimensions of various shapes that would all hold the same volume.

#Input:

A volume in cubic meters.

#Output:

Dimensions of containers of various types that would hold the volume.
The following containers are possible.

* Cube
* Ball (Sphere)
* Cylinder
* Cone

#Example Input:

27

#Example Output:

```
Cube: 3.00m width, 3.00m, high, 3.00m tall
```

Cylinder: 3.00m tall, Diameter of 3.38m
Sphere: 1.86m Radius
Cone: 9.00m tall, 1.69m Radius

#Some Inputs to test.

27, 42, 1000, 2197

Title: [2014-12-19] Challenge #193 [Easy] Acronym Expander
Text: #Description

During online gaming (or any video game that requires teamwork) , there is often times that you need to speak to your teammates. Given the nature of the game, it may be inconvenient to say full sentences and it's for this reason that a lot of games have acronyms in place of sentences that are regularly said.

Example

gg : expands to 'Good Game'
brb : expands to 'be right back'

and so on...

This is even evident on IRC's and other chat systems.

However, all this abbreviated text can be confusing and intimidating for someone new to a game. They're not going to instantly know what 'gl hf all'(good luck have fun all) means. It is with this problem that you come in.

You are tasked with converting an abbreviated sentence into its full version.
#Inputs & Outputs

##Input
On console input you will be given a string that represents the abbreviated chat message.

##Output
Output should consist of the expanded sentence

#Wordlist

Below is a short list of acronyms paired with their meaning to use for this challenge.

* lol - laugh out loud
* dw - don't worry
* hf - have fun
* gg - good game
* brb - be right back
* g2g - got to go
* wtf - what the fuck
* wp - well played
* gl - good luck
* imo - in my opinion

#Sample cases

##input

wtf that was unfair

##output

'what the fuck that was unfair'

##input

gl all hf

##output

'good luck all have fun'

#Test case
input

imo that was wp. Anyway I've g2g

##output

????

Title: [2014-12-22] Challenge #194 [Easy] Destringification
Text: # [](#EasyIcon) **(Easy)**: Destringification

Most programming languages understand the concept of escaping strings. For example, if you wanted to put a double-quote `"` into a string that is delimited by double quotes, you can't just do this:

"this string contains " a quote."

That would end the string after the word *contains*, causing a syntax error. To remedy this, you can prefix the quote with a backslash `\` to *escape* the character.

"this string really does \" contain a quote."

However, what if you wanted to type a backslash instead? For example:

"the end of this string contains a backslash. \"

The parser would think the string never ends, as that last quote is escaped! The obvious fix is to also escape the back-slashes, like so.

"lorem ipsum dolor sit amet \\\\"

The same goes for putting newlines in strings. To make a string that spans two lines, you cannot put a line break in the string literal:

"this string...
...spans two lines!"

The parser would reach the end of the first line and panic! This is fixed by replacing the newline with a special escape code, such as `\n`:

"a new line \n hath begun."

Your task is, given an escaped string, un-escape it to produce what the parser would understand.

Input Description

You will accept a string literal, surrounded by quotes, like the following:

```
"A random\nstring\\\""
```

If the string is valid, un-escape it. If it's not (like if the string doesn't end), throw an error!

Output Description

Expand it into its true form, for example:

```
A random
string\"
```

Sample Inputs and Outputs

Sample Input

```
"hello,\nworld!"
```

Sample Output

```
hello,
world!
```

Sample Input

```
"\"\\\""
```

Sample Output

```
"\"
```

Sample Input

```
"an invalid\nstring\"
```

Sample Output

Invalid string! (Doesn't end)

Sample Input

```
"another invalid string \q"
```

Sample Output

Invalid string! (Bad escape code, `\q`)

Extension

Extend your program to support entering multiple string literals:

```
"hello\nhello again" "\\\"world!\\\""
```

The gap between string literals can only be whitespace (ie. new lines, spaces, tabs.) Anything else, throw an error. Output like the following for the above:

String 1:
hello
hello again

String 2:
\"world!\"

Title: [2014-12-28] Challenge #195 [Easy] Symbolic Link Resolution
Text: # [](#EasyIcon) **(Easy)**: Symbolic Link Resolution

Many Unix-based systems support the concept of a **symbolic link**. This is where one directory name is transparently mapped to another. Before we look further at symbolic links, here's a brief primer on Unix paths.

* The root directory on a file-system is `/`. Everything is contained with in `/`. This is like `C:\` on Windows, but contains everything rather than just the system drive. Thus, all absolute paths begin with a `/` - if it doesn't, the path is assumed to be relative to the current location.

* Successive nested directorys are joined with slashes, so a directory `a` in a directory `b` in a directory `c` in root is denoted `/c/b/a`.

* To distinguish a directory from a file, a trailing slash can be added, so `/c/b/a` and `/c/b/a/` are equivalent assuming `a` is a directory and not a file.

* Path names are case sensitive. `/bin/thing` is different from `/bin/Thing`.

Now, symbolic links are the more general equivalent of Windows shortcuts. They can be used to 'redirect' one directory to another. For example, if I have a version of a program `thing` located at `/bin/thing-2`, then when thing upgrades to thing 3 then any programs referring to `/bin/thing-2` will break once it changes to `/bin/thing-3`. Thus, I might make a symbolic link `/bin/thing` which refers to `/bin/thing-2`. This means any attempt to visit a path beginning with `/bin/thing` will be silently redirected to `/bin/thing-2`. Hence, once the program updates, just change the symbolic link and everything is working still.

Symbolic links can have more to them, and you can in fact make them on Windows with some NTFS trickery, but this challenge focuses just on Unix style directories.

Our challenge is to resolve a given path name into its actual location given a number of symbolic links. Assume that symbolic links can point to other links.

Input Description

You will accept a number **N**. You will then accept **N** lines in the format:

/path/of/link:/path/of/destination

Then you will accept a path of a directory to be fully expanded.

For example:

4
/bin/thing:/bin/thing-3
/bin/thing-3:/bin/thing-3.2
/bin/thing-3.2/include:/usr/include
/usr/include/SDL:/usr/local/include/SDL
/bin/thing/include/SDL/stan

Output Description

Expand it into its true form, for example:

```
/usr/local/include/SDL/stan
```

Sample Inputs and Outputs

Sample Input

```
1
/home/elite/documents:/media/mmcstick/docs
/home/elite/documents/office
```

Sample Output

```
/media/mmcstick/docs/office
```

Sample Input

```
3
/bin:/usr/bin
/usr/bin:/usr/local/bin/
/usr/local/bin/log:/var/log-2014
/bin/log/rc
```

Sample Output

```
/var/log-2014/rc
```

Sample Input

```
2
/etc:/tmp/etc
/tmp/etc/:/etc/
/etc/modprobe.d/config/
```

Sample Output

Program should hang - recursive loop.

(I know nested symlinks are restricted in practice, but we're livin' life on the edge in this subreddit.)

Extension

Extend your solution to resolve existing symlinks in the definition of successive symlinks. For example:

```
4
/bin/thing:/bin/thing-3
/bin/thing-3:/bin/thing-3.2
/bin/thing/include:/usr/include
/bin/thing-3.2/include/SDL:/usr/local/include/SDL
/bin/thing/include/SDL/stan
```

Notice how the 3rd link relies on the first and second symlinks, and the 4th link relies on the 3rd link working.

This should resolve correctly into `/usr/local/include/SDL/stan`.

Title: [2015-01-12] Challenge #197 [Easy] ISBN Validator
Text: #Description

ISBN's (International Standard Book Numbers) are identifiers for books. Given the correct sequence of digits, one book can be identified out of millions of others thanks to this ISBN. But when is an ISBN not just a random slurry of digits? That's for you to find out.

#Rules

Given the following constraints of the ISBN number, you should write a function that can return True if a number is a valid ISBN and False otherwise.

An ISBN is a ten digit code which identifies a book. The first nine digits represent the book and the last digit is used to make sure the ISBN is correct.

To verify an ISBN you :-

* obtain the sum of 10 times the first digit, 9 times the second digit, 8 times the third digit... all the way till you add 1 times the last digit. If the sum leaves no remainder when divided by 11 the code is a valid ISBN.

For example :

0-7475-3269-9 is Valid because

(10 * 0) + (9 * 7) + (8 * 4) + (7 * 7) + (6 * 5) + (5 * 3) + (4 * 2) + (3 * 6) + (2 * 9) + (1 * 9) = 242 which can be divided by 11 and have no remainder.

For the cases where the last digit has to equal to ten, the last digit is written as X. For example 156881111X.

#Bonus
Write an ISBN generator. That is, a programme that will output a valid ISBN number (bonus if you output an ISBN that is already in use :P)

Title: [2015-01-19] Challenge #198 [Easy] Words with Enemies
Text: #Description:

I had a dream a few weeks back that I thought would be a good challenge. I woke up early and quickly typed up a text description so I wouldn't forget (Seriously, it was about 5am and when I explained it to my wife she just laughed at me)

Okay so there is a valley. On each side you got cannons. They are firing words at each other. In the middle of the valley the words would make contact and explode. Similar letters from each word would cancel out. But the left over unique letters from each word would fall to the valley and slowly fill it up.

So your challenge is to come up with the code given two words you eliminate letters in common at a ratio of 1 for 1 and produce a set of letters that are left over from each word after colliding in mid air. Which ever side has the most letters left over "wins". If each side donates an equal amount of letters it is a "tie".

#Examples:

hat cat

both have an "a" and a "t". They will explode and cancel each other out so you get an "h" and a "c" left and so the answer will be "hc" that falls to the valley. Each side donates 1 letter so a "tie"

miss hiss

both have an "i" and "s" and a 2nd "s" so the "m" and "h" falls into the valley below. Again each side donates a letter so a "tie"

because cause

one word "cause" is in the bigger word "because" and so all those letters cancel out. "be" is donated from the left side. Left side "wins" 2 letters to 0 letters donated.

hello below

an "e" "l" "o" cancel out. The left side donates "hl" and the right side donates "bw". Again a tie. Notice that hello has two "l" and below only had the one "l" so only 1 "l" in hello is cancelled out and not both. It has to be a 1 letter for 1 letter. It is not a 1 letter for all letters relationship.

All words will be lower case. They will be in the set [a-z]

#Input:

Two words ordered from which side of the valley they come from:

<left side word> <right side word>

#Output:

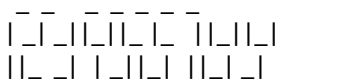
List the extra letters left over after they collide and explode in mid air and determine which side wins or if it was a tie. The design of the output I leave it for you to design and create.

#Challenge inputs:

because cause
hello below
hit miss
rekt pwn
combo jumbo
critical optical
isoenzyme apoenzyme
tribesman brainstem
blames nimble
yakuza wizard
longbow blowup

Title: [2015-1-26] Challenge #199 Bank Number Banners Pt 1
Text: # Description

You work for a bank, which has recently purchased an ingenious machine to assist in reading letters and faxes sent in by branch offices. The machine scans the paper documents, and produces a file with a number of entries which each look like this:



Each entry is 4 lines long, and each line has 27 characters. The first 3 lines of each entry contain an account number written using pipes and underscores, and the fourth line is blank. Each account number should have 9 digits, all of which should be in the range 0-9.

Right now you're working in the print shop and you have to take account numbers and produce those paper documents.

Input

You'll be given a series of numbers and you have to parse them into the previously mentioned banner format. This input...

```
000000000
111111111
490067715
```

#Output

...would reveal an output that looks like this

```
 _  _  _  _  _  _  _  _  _
| || || || || || || || || |
|_||_||_||_||_||_||_||_||_|


 |  |  |  |  |  |  |  |  |
 |  |  |  |  |  |  |  |  |


  _  _  _  _  _  _     _
|_||_|| || ||_  |  |  ||_
 |  _||_||_||_|  |  |  |  _|
```

Title: [2015-02-02] Challenge #200 [Easy] Flood-Fill
Text: # [](#EasyIcon) _(Easy)_: Flood-Fill

Flood-fill is a tool used in essentially any image editing program that's worth its salt. It allows you to fill in any contigious region of colour with another colour, like flooding a depression in a board with paint. For example, take [this beautiful image](http://i.imgur.com/NlCcrKj.png). If I was to flood-fill the colour orange into [this region of the image](http://i.imgur.com/yCavN08.png), then that region would be [turned completely orange](http://i.imgur.com/u6626BA.png).

Today, you're going to implement an algorithm to perform a flood-fill on a text ASCII-style image.

Input and Output Description

Challenge Input

You will accept two numbers, **w** and **h**, separated by a space. These are to be the width and height of the image in characters, with the top-left being (0, 0). You will then accept a grid of ASCII characters of size **w*****h**. Finally you will accept two more numbers, **x** and **y**, and a character **c**. **x** and **y** are the co-ordinates on the image where the flood fill should be done, and **c** is the character that will be filled.

Pixels are defined as contigious (touching) when they share at least one edge (pixels that only touch at corners aren't contigious).

For example:

```
37 22
```

```
.....................................
...########################.....
...#......................#.....
...#......................#.....
...#......................#.....
...#......................#.....
...#......................#.....
...#......................#######.....
...###..................##......#.....
...#..##...............##........#.....
...#....##.........##..........#.....
...#......##.....##.............#.....
...#........#####...............#.....
...#........#...................#.....
...#.......##...................#.....
...#.....##.....................#.....
...#...##.......................#.....
...################################.....
.....................................
.....................................
.....................................
.....................................
8 12 @
```

Challenge Output

Output the image given, after the specified flood-fill has taken place.

```
.....................................
...########################.....
...#......................#.....
...#......................#.....
...#......................#.....
...#......................#.....
...#......................#.....
...#......................#######.....
...###..................##......#.....
...#@@##...............##........#.....
...#@@@@##.............##..........#.....
...#@@@@@@##.....##.............#.....
...#@@@@@@@@@#####...............#.....
...#@@@@@@@@@#.....................#.....
...#@@@@@@@@##.....................#.....
...#@@@@@##.....................#.....
...#@@@##.....................#.....
...################################.....
.....................................
.....................................
.....................................
.....................................
```

Sample Inputs and Outputs

Input

```
16 15
----------------
```

```
-+++++++++++++-
-+-----------+-
-++++++++++++-+-
-+-----------+-
-+-++++++++++++-
-+-----------+-
-++++++++++++-+-
-+-----------+-
-+-++++++++++++-
-+-----------+-
-+++++++++++++-
-+-----------+-
-+++++++++++++-
---------------
2 2 @
```

Output

```
---------------
-+++++++++++++-
-+@@@@@@@@@@@@+-
-++++++++++++@+-
-+@@@@@@@@@@@@+-
-+@++++++++++++-
-+@@@@@@@@@@@@+-
-++++++++++++@+-
-+@@@@@@@@@@@@+-
-+@++++++++++++-
-+@@@@@@@@@@@@+-
-+++++++++++++-
-+-----------+-
-+++++++++++++-
---------------
```

Input

```
9 9
aaaaaaaaa
aaadefaaa
abcdafgha
abcdafgha
abcdafgha
abcdafgha
aacdafgaa
aaadafaaa
aaaaaaaaa
8 3 ,
```

Output

```
,,,,,,,,
,,,def,,,
,bcd,fgh,
,bcd,fgh,
,bcd,fgh,
,bcd,fgh,
,,cd,fg,,
,,,d,f,,,
```

,,,,,,,,

Extension (Easy/Intermediate)

Extend your program so that the image 'wraps' around from the bottom to the top, and from the left to the right (and vice versa). This makes it so that the top and bottom, and left and right edges of the image are touching (like the surface map of a torus).

Input

```
9 9
\/\/\/\.\
/./..././
\.\.\.\.\
/.../.../
\/\/\/\/\
/.../.../
\.\.\.\.\
/./..././
\/\/\/\.\
1 7 #
```

Output

```
\/\/\/\#\
/#/###/#/
\#\#\#\#\
/###/###/
\/\/\/\/\
/###/###/
\#\#\#\#\
/#/###/#/
\/\/\/\#\
```

Further Reading

If you need a starting point with recursion, here are some reading resources.

* [Recursive Algorithms](http://www2.its.strath.ac.uk/courses/c/subsection3_9_5.html)
* [Recursive function calls](http://www.cs.cmu.edu/~rwh/introsml/core/recfns.htm)

Consider using list-like data structures in your solution, too.

Title: [2015-02-09] Challenge #201 [Easy] Counting the Days until...
Text: #Description:

Sometimes you wonder. How many days I have left until.....Whatever date you are curious about. Maybe a holiday. Maybe a vacation. Maybe a special event like a birthday.

So today let us do some calendar math. Given a date that is in the future how many days until that date from the current date?

#Input:

The date you want to know about in 3 integers. I leave it to you to decide if you want to do yyyy mm dd or mm dd yyyy or whatever. For my examples I will be using yyyy mm dd. Your solution should have 1 comment saying what format you are using for people reading your code. (Note you will need to convert your inputs to your format from mine if not using yyyy mm dd)

#Output:

The number of days until that date from today's date (the time you run the program)

##Example Input: 2015 2 14
##Example Output: 5 days from 2015 2 9 to 2015 2 14

#Challenge Inputs:

    2015 7 4
    2015 10 31
    2015 12 24
    2016 1 1
    2016 2 9
    2020 1 1
    2020 2 9
    2020 3 1
    3015 2 9

#Challenge Outputs:

Vary from the date you will run the solution and I leave it to you all to compare results.

Title: [2015-02-16] Challenge #202 [Easy] I AM BENDER. PLEASE INSERT GIRDER.
Text: #Description

Poor Mr.Tinkles is having some troubles. Similar to [The Loneliest Whale In The World](http://en.wikipedia.org/wiki/52-hertz_whale), no one can hear his cries. Or in this case, understand them.

He talks in a sequence of on's and off's. 0's and 1's, it's binary. Obviously as a mere human you can't possibly translate what he's saying as he says it. Looks like you'll have to think of a way around this....

#Formal Inputs & Outputs

##Input description

On console input you will be given a variable number of 0's and 1's that correspond to letters in the alphabet [a-z] and whitespace ' '. These will be integers coming in, it's your job to cast them however you need.

##Output description

The program should output the english translation (or other languages if you feel so inclined!) of the binary phrase

#Samples

Input

    01001000011001010110110001101100011011110010 0
    00001010111011011110111001001101100011001 00

Output

    Hello World

#Test Input

##1
01110000011011000110010101100 0

0101110011011001010010000001 11

0100011000010110110001101011 00

1000000111010001101111001000 00

0110110101100101

##2
01101100011010010110011001 1001

0100100000011100100110100101 10

0111011010000111010000100000 01

1011100110111101110111001000 00

0110100101110011001000000 11011

0001101111011011100110010101 10

110001111001

Title: [2015-2-23] Challenge #203 [Easy] The Start of Something Big
Text: #Description
All great things start with something small. Sometimes people [don't even realise what goes into making a 'small' thing](http://rampantgames.com/blog/?p=7745).

A popular story is linked above about a group of graphics programmers who create a rendering engine in some amount of time. After some time HR came to see what the programmers had accomplished. They responded by showing a black triangle on a tv.

HR was less than impressed (understandle for a non techie) but it goes to show the natural evolution of a program. What they didn't realise is that the programmers have created their base engine and can now easily add and extend on top of it.

Maybe you can follow similar steps?

#Challenge

On your screen, display a square.

You may use any libraries available to you.

The square may be of any size and of any colour.

Title: [2015-03-02] Challenge #204 [Easy] Remembering your lines
Text: #Description

I didn't always want to be a computer programmer, you know. I used to have dreams, dreams of standing on the world stage, being one of the great actors of my generation!

Alas, my acting career was brief, lasting exactly as long as one high-school production of Macbeth. I played old King Duncan, who gets brutally murdered by Macbeth in the beginning of Act II. It was just as well, really, because I had a terribly hard time remembering all those lines!

For instance: I would remember that Act IV started with the three witches brewing up some sort of horrible potion, filled will all sorts nasty stuff, but except for "Eye of newt", I couldn't for the life of me remember what was in it! Today, with our modern computers and internet, such a question is easy to settle: you simply open up [the full text of the play](https://gist.githubusercontent.com/Quackmatic/f8deb2b64dd07ea0985d/raw/macbeth.txt) and press Ctrl-F (or Cmd-F, if you're on a Mac) and search for "Eye of newt".

And, indeed, here's the passage:

> Fillet of a fenny snake,
> In the caldron boil and bake;
> Eye of newt, and toe of frog,
> Wool of bat, and tongue of dog,
> Adder's fork, and blind-worm's sting,
> Lizard's leg, and howlet's wing,—
> For a charm of powerful trouble,
> Like a hell-broth boil and bubble.

Sounds delicious!

In today's challenge, we will automate this process. You will be given the full text of Shakespeare's Macbeth, and then a phrase that's used somewhere in it. You will then output the full passage of dialog where the phrase appears.

#Formal inputs & outputs

##Input description
First off all, you're going to need a full copy of the play, which you can find here: [macbeth.txt](https://gist.githubusercontent.com/Quackmatic/f8deb2b64dd07ea0985d/raw/macbeth.txt). Either right click and save it to your local computer, or open it and copy the contents into a local file.

This version of the play uses consistent formatting, and should be especially easy for computers to parse. I recommend perusing it briefly to get a feel for how it's formatted, but in particular you should notice that all lines of dialog are indented 4 spaces, and only dialog is indented that far.

(edit: thanks to /u/Elite6809 for spotting some formatting errors. I've replaced the link with the fixed version)

Second, you will be given a single line containing a phrase that appears exactly once somewhere in the text of the play. You can assume that the phrase in the input uses the same case as the phrase in the source material, and that the full input is contained in a single line.

##Output description

You will output the line containing the quote, as well all the lines directly above and below it which are also dialog lines. In other words, output the whole "passage".

All the dialog in the source material is indented 4 spaces, you can choose to keep that indent for your output, or you can remove, whichever you want.

#Examples

##Input 1

Eye of newt

##Output 1

Fillet of a fenny snake,
In the caldron boil and bake;
Eye of newt, and toe of frog,
Wool of bat, and tongue of dog,
Adder's fork, and blind-worm's sting,
Lizard's leg, and howlet's wing,—
For a charm of powerful trouble,
Like a hell-broth boil and bubble.

##Input 2

rugged Russian bear

##Output 2

What man dare, I dare:
Approach thou like the rugged Russian bear,
The arm'd rhinoceros, or the Hyrcan tiger;
Take any shape but that, and my firm nerves
Shall never tremble: or be alive again,
And dare me to the desert with thy sword;
If trembling I inhabit then, protest me
The baby of a girl. Hence, horrible shadow!
Unreal mockery, hence!

#Challenge inputs

#Input 1

break this enterprise

#Input 2

Yet who would have thought

#Bonus

If you're itching to do a little bit more work on this, output some more information in addition to the passage: which act and scene the quote appears, all characters with speaking parts in that scene, as well as who spoke the quote. For the second example input, it might look something like this:

ACT III
SCENE IV
Characters in scene: LORDS, ROSS, LADY MACBETH, MURDERER, MACBETH, LENNOX
Spoken by MACBETH:
What man dare, I dare:
Approach thou like the rugged Russian bear,
The arm'd rhinoceros, or the Hyrcan tiger;
Take any shape but that, and my firm nerves
Shall never tremble: or be alive again,

And dare me to the desert with thy sword;
If trembling I inhabit then, protest me
The baby of a girl. Hence, horrible shadow!
Unreal mockery, hence!

Title: [2015-03-09] Challenge #205 [Easy] Friendly Date Ranges
Text: # [](#EasyIcon) _(Easy)_: Friendly Date Ranges

The goal of this challenge is to implement a way of converting two dates into a more friendly date range that could be presented to a user. It must not show any redundant information in the date range. For example, if the year and month are the same in the start and end dates, then only the day range should be displayed. Secondly, if the starting year is the current year, and the ending year can be inferred by the reader, the year should be omitted also (see below for examples).

Formal Inputs and Outputs

Input Description

The input will be two dates in the `YYYY-MM-DD` format, such as:

1. `2015-07-01 2015-07-04`
2. `2015-12-01 2016-02-03`
3. `2015-12-01 2017-02-03`
4. `2016-03-01 2016-05-05`
5. `2017-01-01 2017-01-01`
6. `2022-09-05 2023-09-04`

Output Description

The program must turn this into a human readable date in the `Month Day, Year` format (omitting the year where possible). These outputs correspond to the above inputs:

1. `July 1st - 4th`
2. `December 1st - February 3rd`
3. `December 1st, 2015 - February 3rd, 2017`
4. `March 1st - May 5th, 2016`
5. `January 1st, 2017`
6. `September 5th, 2022 - September 4th, 2023`

Edge Case 1

If the starting year is the current year, **but the ending year isn't** and **the dates are at least a year apart**, then specify the year in both. For example, this input:

2015-04-01 2020-09-10

Must not omit the 2015, so it should output `April 1st, 2015 - September 10th, 2020`, and **NOT** `April 1st - September 10th, 2020`, which would otherwise be ambiguous.

Of course if the dates are less than a year apart, as in the case of `2015-12-01 2016-02-03`, then you can safely omit the years (`December 1st - February 3rd`), as that makes it clear that it's the February next year.

Edge Case 2

Similarly, if the starting year is the current year, **but the two dates are exactly one year apart**, also specify the year in both. For example, this input:

2015-12-11 2016-12-11

Must specify both years, i.e. `December 11th, 2015 - December 11th, 2016`.

Bonus (Intermediate)

Of course, not all users will want to read a `Month Day, Year` format. To fix this, allow your program to receive hints on how to format the dates, by accepting a date format as a third parameter, for example:

1. `2015-07-01 2015-07-04 DMY`
2. `2016-03-01 2016-05-05 YDM`
3. `2022-09-05 2023-09-04 YMD`

would produce:

1. `1st - 4th July`
2. `2016, 1st March - 5th May`
3. `2022, September 5th - 2023, September 4th`

You only need to handle date format strings `DMY`, `MDY`, `YMD` and `YDM`.

Title: [2015-03-16] Challenge #206 [Easy] Recurrence Relations, part 1
Text: # [](#EasyIcon) _(Easy)_: Recurrence Relations, part 1

A recurrence relation is a mathematical construct for defining a series of numbers. It works by first giving an initial term, and then recursively defining the rest of the series as functions of the first one. For example, let's say we have a series of numbers called **u**, which is defined by this recurrence relation:

```
u[0] = 1
u[n+1] = 2 * u[n]
```

The first relation tells us that *u*(0), the first term in the series, is 1. The second relation says that, given the *n*-th term *u*(n), the next term (*u*(n+1)) is the previous term multiplied by two. So, to get the second term in the series, you multiply the first term by two, to get 2. To get the third term in the series, you multiply the second term by two, to get 4.

Recurrence relations get their name in part due to their *recursive* nature, as successive terms are essentially defined as recursive application of a function, like this Python example:

```
def recurrence(n):
    return n * 2

first_term = 1
second_term = recurrence(first_term)
third_term = recurrence(recurrence(first_term))
fourth_term = recurrence(third_term) # or recurrence(recurrence(recurrence(first_term)))
```

Or, with the help of another function to apply the `recurrence` function for us:

```
def get_nth_term(recurrence_relation, first_term, n):
    if n == 0:
        return first_term
    else:
        return get_nth_term(recurrence_relation, recurrence_relation(first_term), n - 1)

sixteenth_term = get_nth_term(recurrence, first_term, 16) #65536
```

You get the idea. Today you will be writing a program to compute the *n*-th term of a given series defined by a recurrence relation.

Formal Inputs and Outputs

Input Description

You will first accept a line of input like this one:

```
*3 +2 *2
```

This string means that, to get the next term in the series, you multiply by three, add two, then multiply by two. The operation that takes place is the first character of every space-separated part of the line, and the possible characters are `+` for add, `-` for subtract, `*` for multiply, and `/` for divide; the number given can be any real number. Next, you will accept the starting value, like so:

```
0
```

Finally, you will accept the term number to print to (where the first term in the series is term **0**):

```
7
```

(this input can be viewed on [Wolfram|Alpha](http://www.wolframalpha.com/input/?i={+u%280%29%3D0%3B+u%28n%29%3D%28u%28n-1%29*3%2B2%29*2+}).)

Output Description

You are to print all terms in the series, up to the given term number, like so:

```
Term 0: 0
Term 1: 4
Term 2: 28
Term 3: 172
Term 4: 1036
Term 5: 6220
Term 6: 37324
Term 7: 223948
```

Sample Inputs and Outputs

Series 1

This series starts with 1, and for each successive member of the series, will multiply the last one by two and add one.

Input

```
*2 +1
1
10
```

Output

```
Term 0: 1
Term 1: 3
Term 2: 7
Term 3: 15
Term 4: 31
Term 5: 63
```

```
Term 6: 127
Term 7: 255
Term 8: 511
Term 9: 1023
Term 10: 2047
```

Series 2

This one is a bit different. This just multiplies each successive term by `-2`. Notice how the series oscillates between positive and negative numbers?

Input

```
*-2
1
8
```

Output

```
Term 0: 1
Term 1: -2
Term 2: 4
Term 3: -8
Term 4: 16
Term 5: -32
Term 6: 64
Term 7: -128
Term 8: 256
```

Series 3

Input

```
+2 *3 -5
0
10
```

Output

```
Term 0: 0
Term 1: 1
Term 2: 4
Term 3: 13
Term 4: 40
Term 5: 121
Term 6: 364
Term 7: 1093
Term 8: 3280
Term 9: 9841
Term 10: 29524
```

Notes

More on recurrence relations [on Wikipedia](http://en.wikipedia.org/wiki/Recurrence_relation). Have a cool idea for a challenge? Submit it to /r/DailyProgrammer_Ideas!

Title: [2015-03-23] Challenge #207 [Easy] Bioinformatics 1: DNA Replication
Text: For this week my theme is bioinformatics, I hope you enjoy the taste of the field through these challenges.

Description

DNA - deoxyribonucleic acid - is the building block of every organism. It contains information about hair color, skin tone, allergies, and more.
[It's usually visualized as a long double helix of base
pairs.](http://cdn.theatlantic.com/static/mt/assets/science/shutterstock_34693498%20copy.jpg) DNA is composed of four bases - adenine, thymine, cytosine, guanine - paired as follows: A-T and G-C.

Meaning: on one side of the strand there may be a series of bases

A T A A G C

And on the other strand there will have to be

T A T T C G

It is your job to generate one side of the DNA strand and output the two DNA strands. Your program should take a DNA sequence as input and return the complementary strand.

Input

A A T G C C T A T G G C

Output
A A T G C C T A T G G C
T T A C G G A T A C C G

Extra Challenge

Three base pairs make a codon. These all have different names based on what combination of the base pairs you have. A handy table can be found [here](http://en.wikipedia.org/wiki/DNA_codon_table).
The string of codons starts with an ATG (Met) codon ends when a STOP codon is hit.

For this part of the challenge, you should implement functionality for translating the DNA to a protein sequence based on the codons, recalling that every generated DNA strand starts with a Met codon and ends with a STOP codon. Your program should take a DNA sequence and emit the translated protein sequence, complete with a STOP at the terminus.

#Input

A T G T T T C G A G G C T A A

#Output

A T G T T T C G A G G C T A A
Met Phe Arg Gly STOP

Title: [2015-03-30] Challenge #208 [Easy] Culling Numbers
Text: #Description:

Numbers surround us. Almost too much sometimes. It would be good to just cut these numbers down and cull out the repeats.

Given some numbers let us do some number "culling".

#Input:

You will be given many unsigned integers.

#Output:

Find the repeats and remove them. Then display the numbers again.

#Example:

Say you were given:

* 1 1 2 2 3 3 4 4

Your output would simply be:

* 1 2 3 4

#Challenge Inputs:

##1:

3 1 3 4 4 1 4 5 2 1 4 4 4 4 1 4 3 2 5 5 2 2 2 4 2 4 4 4 4 1

##2:

65 36 23 27 42 43 3 40 3 40 23 32 23 26 23 67 13 99 65 1 3 65 13 27 36 4 65 57 13 7 89 58 23 74 23 50 65 8 99 86 23 78 89 54 89 61 19 85 65 19 31 52 3 95 89 81 13 46 89 59 36 14 42 41 19 81 13 26 36 18 65 46 99 75 89 21 19 67 65 16 31 8 89 63 42 47 13 31 23 10 42 63 42 1 13 51 65 31 23 28

Title: [2015-04-06] Challenge #209 [Easy] The Button can be pressed but once...
Text: # [](#EasyIcon) _(Easy)_: The Button can be pressed but once...

The 1st of April brought [the Button](/r/thebutton) to Reddit - if you've not heard of it, read the blog post on it [here](http://www.redditblog.com/2015/04/the-button.html). The value of the countdown at the instant that someone presses the button determines the flair that they obtain on the subreddit. For example, if the counter is at 53.04 seconds, then I would obtain a **53** flair, as that is the number of seconds (rounded down). After a person presses the button, the countdown resets from 60.00 seconds. Today's challenge is simple - you'll be given a list of users in no particular order, and told at which time each user pressed the button; you'll need to work out which flair each user gets.

You can assume that the countdown never runs to zero for this challenge, and that no two users will press the button at exactly the same moment.

Formal Inputs and Outputs

Input Description

At a time of 0.00 seconds, the countdown starts from 60.00 seconds, counting down. So at a time of 27.34 seconds, the countdown will read `32.66` assuming no-one has pressed the button; all times are given in this format, with a number of seconds and a number of hundredths of a second. The list of users will be given in this format:

```
7
UserA: 41.04
UserB: 7.06
UserC: 20.63
UserD: 54.28
UserE: 12.59
UserF: 31.17
UserG: 63.04
```

The number on the first line is the number of users in the input string; after that, the username of each user, followed by the number of seconds since the beginning of the countdown.

Output Description

Output the numerical flair that each user will receive, in the order in which the users click the buttons - for example:

```
UserB: 52
UserE: 54
UserC: 51
UserF: 49
UserA: 50
UserD: 46
UserG: 51
```

UserG clicked the button last, and so they are printed last - when they clicked the button, the countdown was at `51.24`, so they receive the **51** flair.

Sample Inputs and Outputs

Sample Input

```
8
Coder_d00d: 3.14
Cosmologicon: 22.15
Elite6809: 17.25
jnazario: 33.81
nint22: 10.13
rya11111: 36.29
professorlamp: 31.60
XenophonOfAthens: 28.74
```

Sample Output

```
Coder_d00d: 56
nint22: 53
Elite6809: 52
Cosmologicon: 55
XenophonOfAthens: 53
professorlamp: 57
jnazario: 57
rya11111: 57
```

Sample Input

7
bholzer: 101.09
Cosmologicon: 27.45
nint22: 13.76
nooodl: 7.29
nottoobadguy: 74.56
oskar_s: 39.90
Steve132: 61.82

Sample Output

nooodl: 52
nint22: 53
Cosmologicon: 46
oskar_s: 47
Steve132: 38
nottoobadguy: 47
bholzer: 33

Title: [2015-04-13] Challenge #210 [Easy] intHarmony.com
Text: #Description:

In this modern fast paced time of the internet it is a busy place for hardworking unsigned integers (lets just call them ints) Believe it or not these ints love to date and hook up. But they don't always get along.

Computer scientists have discovered 32 levels of compatibility. By comparing the binary value of ints we can develop a percentage of compatibility. (these unsigned integers need 32 bits to store in memory)

For today's challenge you will be given 2 unsigned ints who might be a match. You will compute a percentage of compatibility by comparing the binary value of each unsigned ints to each other. For every bit (1 or 0) in common they generate 1 match point. The max will be 32 the lowest will be 0. You then display the percentage of compatibility.

Also as a service to the unsigned ints you will list the avoid number. This is the number that is the pure opposite of that number (in binary)

#Finding Compatibility:

So for my example I am using 8 bit integers. You must do this for all 32 bits of an integer. But I am using 8 bit examples to keep it simple.

We are gonna compare 1 and 2

1 in binary is 0000 0001
2 in binary is 0000 0010

If you compare each bit place with each number you find that they have 6 bits in common. (From left to right -- the first 6 bits are all 0 and so the same bit and so that is 6 match points)

the last 2 bits are not the same. They are different.

Therefore 1 and 2 have 6 out of 8 match points. For a compatibility score of 75%

The most compatible numbers will be the same number as all the bits match perfectly. (We are all most compatible with ourselves the most)

So taking 1 in binary (0000 0001) the complete opposite number would have to be (1111 1110) or 254. 1 and 254 should not be in the same data structure together ever.

#Input:

2 unsigned Integers x and y

#Output

% of compatibility
x should avoid (x's opposite)
y should avoid (y's opposite)

#Example:

This is an 8 bit example - for your challenge you will be using 32 bit unsigned ints.

100 42

100 in binary is 0110 0100
42 in binary is 0010 1010

Comparing the bits we see that they have 4 match points. 50% compatible.

the opposite of 100 in binary is 1001 1011 or (155)
the opposite of 42 in binary is 1101 0101 or (213)

So our output should be

50% Compatibility
100 should avoid 155
42 should avoid 213

Okay so not a great match but at intHarmony.com but we have done our job.

#Challenge Inputs:

20 65515
32000 101
42000 42
13 12345
9999 9999
8008 37331
54311 2
31200 34335

Title: [2015-04-20] Challenge #211 [Easy] The Name Game
Text: #Description

If computer programmers had a "patron musician" (if such a thing even exists), it would surely be the great [Shirley Ellis](http://en.wikipedia.org/wiki/Shirley_Ellis). It is my opinion that in the history of music, not song has ever come closer to replicating the experience of programming as her 1964 novelty hit [The Name Game](https://www.youtube.com/watch?v=5MJLi5_dyn0). In the lyrics of that song she lays out quite an elegant and fun algorithm for making a rhyme out of anybody's name. The lyrics are almost like sung pseudo-code!

Your challenge today will be to implement a computer program that can play Ms. Ellis' Name Game. You will recieve a name for input, and output the rhyme for that name.

It should be noted that while the rhyming algorithm is very elegant and easy for humans to follow, Ms. Ellis description is not *quite* rigorous. For instance, there's an extra rule that she doesn't mention that only applies when names start with a vowel (such as "Arnold"), and it's not quite clear exactly what you should do when the names start with M, F or B. You will have to fill in the blanks as best you can on your own. If you're not sure how a specific rule goes, implement what sounds best to you.

You should primarily refer to the song for instructions, but I've includeded the relevant lyrics here:

> Come on everybody!
> I say now let's play a game
> I betcha I can make a rhyme out of anybody's name
>
> The first letter of the name, I treat it like it wasn't there
> But a "B" or an "F" or an "M" will appear
> And then I say "bo", add a "B", then I say the name
> and "Bonana fanna" and a "fo"
>
> And then I say the name again with an "F" very plain
> and a "fee fy" and a "mo"
> And then I say the name again with an "M" this time
> and there isn't any name that I can't rhyme
>
> But if the first two letters are ever the same,
> I drop them both and say the name like
>
> Bob, Bob drop the B's "Bo-ob"
> For Fred, Fred drop the F's "Fo-red"
> For Mary, Mary drop the M's Mo-ary
> That's the only rule that is contrary.

#Formal Inputs & Outputs

##Input description

Your input will be a single line with a single name on it. Note that in all the excitement, an exclamation point has been added to the end.

##Output description

The rhyme of the name!

#Example Inputs & Outputs

Examples helpfully provided by Ms. Ellis herself.

##Example 1

Lincoln!

##Output 1

Lincoln, Lincoln bo Bincoln,
Bonana fanna fo Fincoln,
Fee fy mo Mincoln,
Lincoln!

##Example 2

Nick!

##Output 2

Nick, Nick bo Bick,
Bonana fanna fo Fick,
Fee fy mo Mick,
Nick!

#Challenge input

##Input 1

Arnold!

##Input 2

Billy!

##Input 3

Your username! Or even, if you feel comfortable sharing it, your real name! Or even my name! Or whatever! I've listened to this music video, like, six times in a row while writing this challenge, and all I want to do is dance!

Title: [2015-04-27] Challenge #212 [Easy] Rövarspråket
Text: # Description

When we Swedes are young, we are taught a SUPER-SECRET language that only kids know, so we can hide secrets from our confused parents. This language is known as "Rövarspråket" (which means "Robber's language", more or less), and is surprisingly easy to become fluent in, at least when you're a kid. Recently, the cheeky residents of /r/Sweden decided to play a trick on the rest on reddit, and get a thread entirely in Rövarspråket to /r/all. [The results were hilarious](http://np.reddit.com/r/sweden/comments/301sqr/dodetot_%C3%A4ror_fof%C3%B6ror_lolitote/).

Rövarspråket is not very complicated: you take an ordinary word and replace the consonants with the consonant doubled and with an "o" in between. So the consonant "b" is replaced by "bob", "r" is replaced with "ror", "s" is replaced with "sos", and so on. Vowels are left intact. It's made for Swedish, but it works just as well in English.

Your task today is to write a program that can encode a string of text into Rövarspråket.

(note: this is a higly guarded Swedish state secret, so I trust that none of you will share this very privileged information with anyone! If you do, you will be extradited to Sweden and be forced to eat [surströmming](http://en.wikipedia.org/wiki/Surstr%C3%B6mming) as penance.)

(note 2: surströmming is actually not that bad, it's much tastier than its reputation would suggest! I'd go so far as to say that it's the tastiest half-rotten fish in the world!)

Formal inputs & outputs
Input

You will recieve one line of input, which is a text string that should be encoded into Rövarspråket.

Output

The output will be the encoded string.

A few notes: your program should be able to handle case properly, which means that "Hello" should be encoded to "Hohelollolo", and *not* as "HoHelollolo" (note the second capital "H").

Also, since Rövarspråket is a Swedish invention, your program should follow Swedish rules regarding what is a vowel and what is a consonant. The Swedish alphabet is the same as the English alphabet except that there are three extra characters at the end (Å, Ä and Ö) which are all vowels. In addition, Y is always a vowel in Swedish, so the full list of vowels in Swedish is A, E, I, O, U, Y, Å, Ä and Ö. The rest are consonants.

Lastly, any character that is not a vowel or a consonant (i.e. things like punctuation) should be left intact in the output.

Example inputs
Input 1

Jag talar Rövarspråket!

Output 1

Jojagog totalolaror Rorövovarorsospoproråkoketot!

Input 2

I'm speaking Robber's language!

Output 2

I'mom sospopeakokinongog Rorobobboberor'sos lolanongoguagoge!

Challenge inputs

Input 1

Tre Kronor är världens bästa ishockeylag.

Input 2

Vår kung är coolare än er kung.

Bonus

Make your program able to decode a Rövarspråket-encoded sentence as well as encode it.

Title: [2015-05-04] Challenge #213 [Easy] Pronouncing Hex
Text: # Description

The HBO network show "Silicon Valley" has introduced a way to pronounce hex.

> Kid: Here it is: Bit… soup. It's like alphabet soup, BUT… it's ones and zeros instead of letters.
> Bachman: {silence}
> Kid: 'Cause it's binary? You know, binary's just ones and zeroes.
> Bachman: Yeah, I know what binary is. Jesus Christ, I memorized the hexadecimal
> times tables when I was fourteen writing machine code. Okay? Ask me
> what nine times F is. It's fleventy-five. I don't need you to tell me what
> binary is.

Not "eff five", fleventy. `0xF0` is now fleventy. Awesome. Above a full byte you add "bitey" to the name. The hexidecimal pronunciation rules:

HEX PLACE VALUE	WORD
0xA0	"Atta"
0xB0	"Bibbity"
0xC0	"City"
0xD0	"Dickety"
0xE0	"Ebbity"
0xF0	"Fleventy"
0xA000	"Atta-bitey"
0xB000	"Bibbity-bitey"
0xC000	"City-bitey"
0xD000	"Dickety-bitey"
0xE000	"Ebbity-bitey"
0xF000	"Fleventy-bitey"

Combinations like `0xABCD` are then spelled out "atta-bee bitey city-dee".

For this challenge you'll be given some hex strings and asked to pronounce them.

Input Description

You'll be given a list of hex values, one per line. Examples:

> 0xF5
> 0xB3
> 0xE4
> 0xBBBB
> 0xA0C9

Output Description

Your program should emit the pronounced hex. Examples from above:

> 0xF5 "fleventy-five"
> 0xB3 "bibbity-three"
> 0xE4 "ebbity-four"
> 0xBBBB "bibbity-bee bitey bibbity-bee"
> 0xA0C9 "atta-bitey city-nine"

Title: [2015-05-11] Challenge #214 [Easy] Calculating the standard deviation
Text: # Description

Standard deviation is one of the most basic measurments in statistics. For some collection of values (known as a "population" in statistics), it measures how dispersed those values are. If the standard deviation is high, it means that the values in the population are very spread out; if it's low, it means that the values are tightly clustered around the mean value.

For today's challenge, you will get a list of numbers as input which will serve as your statistical population, and you are then going to calculate the standard deviation of that population. There are statistical packages for many programming languages that can do this for you, but you are highly encouraged not to use them: the spirit of today's challenge is to implement the standard deviation function yourself.

The following steps describe how to calculate standard deviation for a collection of numbers. For this example, we will use the following values:

    5 6 11 13 19 20 25 26 28 37

1. First, calculate the average (or **mean**) of all your values, which is defined as the sum of all the values divided by the total number of values in the population. For our example, the sum of the values is 190 and since there are 10 different values, the mean value is 190/10 = 19

2. Next, for each value in the population, calculate the difference between it and the mean value, and square that difference. So, in our example, the first value is 5 and the mean 19, so you calculate (5 - 19)^2 which is equal to 196. For the second value (which is 6), you calculate (6 - 19)^2 which is equal to 169, and so on.

3. Calculate the sum of all the values from the previous step. For our example, it will be equal to 196 + 169 + 64 + ... = 956.

3. Divide that sum by the number of values in your population. The result is known as the **variance** of the population, and is equal to the square of the standard deviation. For our example, the number of values in the population is 10, so the variance is equal to 956/10 = 95.6.

4. Finally, to get standard deviation, take the square root of the variance. For our example, sqrt(95.6) ≈ 9.7775.

#Formal inputs & outputs

#Input

The input will consist of a single line of numbers separated by spaces. The numbers will all be positive integers.

#Output

Your output should consist of a single line with the standard deviation rounded off to at most 4 digits after the decimal point.

#Sample inputs & outputs

##Input 1

    5 6 11 13 19 20 25 26 28 37

##Output 1

    9.7775

##Input 2

    37 81 86 91 97 108 109 112 112 114 115 117 121 123 141

##Output 2

23.2908

#Challenge inputs

##Challenge input 1

266 344 375 399 409 433 436 440 449 476 502 504 530 584 587

##Challenge input 2

809 816 833 849 851 961 976 1009 1069 1125 1161 1172 1178 1187 1208 1215 1229 1241 1260 1373

#Notes

For you statistics nerds out there, note that this is the *population* standard deviation, not the *sample* standard deviation. We are, after all, given the entire population and not just a sample.

Title: [2015-05-18] Challenge #215 [Easy] Sad Cycles
Text: # [](#EasyIcon) _(Easy)_: Sad Cycles

Take a number, and add up the square of each digit. You'll end up with another number. If you repeat this process over and over again, you'll see that one of two things happen:

* You'll reach one, and from that point you'll get one again and again.
* You'll reach a cycle of 4, 16, 37, 58, 89, 145, 42, 20, 4, 16, 37, ...

For example, starting with the number 12:

* 1^(2)+2^(2)=5
* 5^(2)=25
* 2^(2)+5^(2)=29
* 2^(2)+9^(2)=85
* 8^(2)+5^(2)=89
* 8^(2)+9^(2)=145
* From this point on, you'll join the cycle described above.

However, if we start with the number 13:

* 1^(2)+3^(2)=10
* 1^(2)+0^(2)=1
* 1^(2)=1
* 1^(2)=1
* We get the number 1 forever.

The sequence of numbers that we end up with is called a *sad cycle*, and it depends on the number you start with. If you start the process with a number **n**, the sad cycle for **n** is the cycle which ends up eventually repeating itself; this will either just be the cycle `1`, or the cycle `4, 16, 37, 58, 89, 145, 42, 20`.

But what if we cube the digits instead of squaring them? This gives us a different set of cycles all together. For example, starting with 82375 and repeatedly getting the sum of the *cube* of the digits will lead us to the cycle `352, 160, 217`. Other numbers gravitate toward certain end points. These cycles are called *3-sad cycles* (as the digits are raised to the power 3). This can be extended toward higher powers. For example, the 7-sad cycle for 1060925 is `5141159, 4955606, 5515475, 1152428, 2191919, 14349038, 6917264, 6182897, 10080881, 6291458, 7254695, 6059210`. Your challenge today, will be to find the *b*-sad cycle for a given *n*.

Formal Inputs and Outputs

Input Description

You will input the base ***b*** on the first line, and the starting number ***n*** on the second line, like so:

```
5
117649
```

Output Description

Output a comma-separated list containing the ***b***-sad cycle for ***n***. For example, the 5-sad cycle for 117649 is:

```
10933, 59536, 73318, 50062
```

The starting point of the cycle doesn't matter - you can give a circularly permuted version of the cycle, too; rotating the output around, wrapping from the start to the end, is also a correct output. The following outputs are equivalent to the above output:

```
59536, 73318, 50062, 10933
73318, 50062, 10933, 59536
50062, 10933, 59536, 73318
```

Sample Inputs and Outputs

Sample 1

Input

```
6
2
```

Output

```
383890, 1057187, 513069, 594452, 570947, 786460, 477201, 239459, 1083396, 841700
```

Sample 2

Input

```
7
7
```

Output

```
5345158, 2350099, 9646378, 8282107, 5018104, 2191663
```

Sample 3

Input

```
3
14
```

Output

```
371
```

Sample 4

Input

11
2

Output

5410213163, 416175830, 10983257969, 105122244539, 31487287760, 23479019969, 127868735735, 23572659062, 34181820005, 17233070810, 12544944422, 31450865399, 71817055715, 14668399199, 134844138593, 48622871273, 21501697322, 33770194826, 44292995390, 125581636412, 9417560504, 33827228267, 21497682212, 42315320498, 40028569325, 40435823054, 8700530096, 42360123272, 2344680590, 40391187185, 50591455115, 31629394541, 63182489351, 48977104622, 44296837448, 50918009003, 71401059083, 42001520522, 101858747, 21187545101, 10669113941, 63492084785, 50958448520, 48715803824, 27804526448, 19581408116, 48976748282, 61476706631

Comment Order

Some people have notified us that new solutions are getting buried if you're not one of the first to submit. This is valid concern, so today we're trialling a method of **setting the suggested sort order to** ***new*** (suggested sorts are a newly introduced feature on Reddit). We'll take feedback on this and see how it goes. This means newer solutions will appear at the top.

If you don't like this new sorting, you can still change the method back to **sort by** ***best***, which is the default.

Title: [2015-05-25] Challenge #216 [Easy] Texas Hold 'Em 1 of 3 - Let's deal.
Text: #Theme Week:

I got the whole week so I am merging all 3 challenges into a theme of Texas Hold 'em Poker. All 3 challenges will be related on this popular card game of poker.

#Description:

For those who want to know more about Texas Hold 'Em Poker or just need a refresher. Check [Wikipedia Article on Texas Hold 'Em Poker] (http://en.wikipedia.org/wiki/Texas_hold_%27em)

For the first challenge we will simulate the dealing part of the game.

#Input:

You will be asked how many players 2 to 8. You will always be one of the players and you are facing 1 to 7 other computer controlled players.

#Output:

Display the 2 cards each player is dealt and the display the 5 community cards.

Format is left up to you. (The exact method of the output a card. For my examples I am using verbal words but someone might use unicode symbols for the card suit or other. You design this as long as we can tell the cards apart it is all good)

#Example:

How many players (2-8) ? 3

Your hand: 2 of Clubs, 5 of Diamonds
CPU 1 Hand: Ace of Spades, Ace of Hearts
CPU 2 Hand: King of Clubs, Queen of Clubs

Flop: 2 of Hearts, 5 of Clubs, Ace of Clubs

Turn: King of Hearts
River: Jack of Hearts

#Dealing Cards:

To keep things close to the game you will be dealing from 1 deck of standard playing cards. Once you deal that card you cannot deal it again. The exact method is part of the challenge and for you to decide, design and implement.

In Texas Hold em you burn a card (draw and discard without looking at it) before you do the flop, turn and river. It removes these cards from the pool of possible cards that can be dealt. If you wish to show these cards (I did not in my example) then please for science go for it.

#Looking ahead for the Intermediate:

In the intermediate you will be asked to compare various hands of poker to find which hand is the winning hand.

Title: [2015-06-01] Challenge #217 [Easy] Lumberjack Pile Problem
Text: #Description:

The famous lumberjacks of /r/dailyprogrammer are well known to be weird and interesting. But we always enjoy solving their problems with some code.

For today's challenge the lumberjacks pile their logs from the forest in a grid n x n. Before using us to solve their inventory woes they randomly just put logs in random piles. Currently the pile sizes vary and they want to even them out. So let us help them out.

#Input:

You will be given the size of the storage area. The number of logs we have to put into storage and the log count in each pile currently in storage. You can either read it in from the user or hardcode this data.

##Input Example:

```
3
7
1 1 1
2 1 3
1 4 1
```

So the size is 3 x 3. We have 7 logs to place and we see the 3 x 3 grid of current size of the log piles.

#Log Placement:

We want to fill the smallest piles first and we want to evenly spread out the logs. So in the above example we have 7 logs. The lowest log count is 1. So starting with the first pile in the upper left
and going left-right on each row we place 1 log in each 1 pile until all the current 1 piles get a log. (or until we run out). After that if we have more logs we then have to add logs to piles with 2 (again moving left-right on each row.)

Keep in mind lumberjacks do not want to move logs already in a pile. To even out the storage they will do it over time by adding new logs to piles. But they are also doing this in an even distribution.

Once we have placed the logs we need to output the new log count for the lumberjacks to tack up on their cork board.

#Output:

Show the new n x n log piles after placing the logs evenly in the storage area.

Using the example input I would generate the following:

##example output:

```
3 2 2
2 2 3
2 4 2
```

Notice we had 6 piles of 1s. Each pile got a log. We still have 1 left. So then we had to place logs in piles of size 2. So the first pile gets the last log and becomes a 3 and we run out of logs and we are done.

#Challenge inputs:

Please solve the challenge using these inputs:

##Input 1:

```
 4
200
15 12 13 11
19 14  8 18
13 14 17 15
 7 14 20  7
```

##Input 2:

```
15
2048
 5 15 20 19 13 16  5  2 20  5  9 15  7 11 13
17 13  7 17  2 17 17 15  4 17  4 14  8  2  1
13  8  5  2  9  8  4  2  2 18  8 12  9 10 14
18  8 13 13  4  4 12 19  3  4 14 17 15 20  8
19  9 15 13  9  9  1 13 14  9 10 20 17 20  3
12  7 19 14 16  2  9  5 13  4  1 17  9 14 19
 6  3  1  7 14  3  8  6  4 18 13 16  1 10  3
16  3  4  6  7 17  7  1 10 10 15  8  9 14  6
16  2 10 18 19 11 16  6 17  7  9 13 10  5 11
12 19 12  6  6  9 13  6 13 12 10  1 13 15 14
19 18 17  1 10  3  1  6 14  9 10 17 18 18  7
 7  2 10 12 10 20 14 13 19 11  7 18 10 11 12
 5 16  6  8 20 17 19 17 14 10 10  1 14  8 12
19 10 15  5 11  6 20  1  5  2  5 10  5 14 14
12  7 15  4 18 11  4 10 20  1 16 18  7 13 15
```

Input 3:

```
1
41
1
```

Input 4:

```
12
10000
 9 15 16 18 16  2 20  2 10 12 15 13
20  6  4 15 20 16 13  6  7 12 12 18
11 11  7 12  5  7  2 14 17 18  7 19
 7 14  4 19  8  6  4 11 14 13  1  4
```

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

#Other Lumberjack Problems:

* [Hard - Simulated Ecology - The Forest](http://www.reddit.com/r/dailyprogrammer/comments/27h53e/662014_challenge_165_hard_simulated_ecology_the/)
* [Hard - Lumberjack Floating Log Problem] (http://www.reddit.com/r/dailyprogrammer/comments/2lljyq/11052014_challenge_187_hard_lumberjack_floating/)

Title: [2015-06-08] Challenge #218 [Easy] Making numbers palindromic

Text:

Description

To covert nearly any number into a palindromic number you operate by reversing the digits and adding and then repeating the steps until you get a palindromic number. Some require many steps.

e.g. 24 gets palindromic after 1 steps: 66 -> 24 + 42 = 66

while 28 gets palindromic after 2 steps: 121 -> 28 + 82 = 110, so 110 + 11 (110 reversed) = 121.

Note that, as an example, 196 never gets palindromic (at least according to researchers, at least never in reasonable time). Several numbers never appear to approach being palindromic.

Input Description

You will be given a number, one per line. Example:

```
11
68
```

Output Description

You will describe how many steps it took to get it to be palindromic, and what the resulting palindrome is. Example:

```
11 gets palindromic after 0 steps: 11
68 gets palindromic after 3 steps: 1111
```

Challenge Input

```
123
286
196196871
```

Challenge Output

```
123 gets palindromic after 1 steps: 444
286 gets palindromic after 23 steps: 8813200023188
196196871 gets palindromic after 45 steps: 4478555400006996000045558744
```

Note

Bonus: see which input numbers, through 1000, yield identical palindromes.

Bonus 2: See which numbers don't get palindromic in under 10000 steps. Numbers that never converge are called Lychrel numbers.

Title: [2015-06-15] Challenge #218 [Easy] To-do list (Part 1)
Text: #Description

Todays challenge will be something slightly different! Atleast I think the challenge is meant to be for today? Wait, am I meant to even be submitting today?

Okay maybe I need some help on organising my thoughts before I submit my challenge. A to-do list would be fine, just something so that I can organise my thoughts!

It should have the following basic functionality

* Add an item to the to-do list
* Delete a selected item from the to-do list
* And obviously, print out the list so I can see what to do!

#Formal Inputs & Outputs

##Output description

Any output that is created should be user-friendly. When I'm viewing my to-do list, I should be able to easily discern one list item from another.

#Examples
Input:

```
addItem('Take a shower');
addItem('Go to work');
viewList();
```

Output:

```
Take a shower
Go to work
```

Further Input:

```
addItem('Buy a new phone');
deleteItem('Go to work');
viewList();
```

Outputs:

```
Take a shower
Buy a new phone
```

Title: [2015-06-22] Challenge #220 [Easy] Mangling sentences
Text: #Description

In this challenge, we are going to take a sentence and mangle it up by sorting the letters in each word. So, for instance, if you take the word "hello" and sort the letters in it, you get "ehllo". If you take the two words "hello world", and sort the letters in each word, you get "ehllo dlorw".

#Inputs & outputs

##Input

The input will be a single line that is exactly one English sentence, starting with a capital letter and ending with a period

##Output

The output will be the same sentence with all the letters in each word sorted. Words that were capitalized in the input needs to be capitalized properly in the output, and any punctuation should remain at the same place as it started. So, for instance, "Dailyprogrammer" should become "Aadegilmmoprrry" (note the capital A), and "doesn't" should become "denos't".

To be clear, only spaces separate words, not any other kind of punctuation. So "time-worn" should be transformed into "eimn-ortw", not "eimt-norw", and "Mickey's" should be transformed into "Ceikms'y", not anything else.

Edit: It has been pointed out to me that this criterion might make the problem a bit too difficult for [easy] difficulty. If you find this version too challenging, you can consider every non-alphabetic character as splitting a word. So "time-worn" becomes "eimt-norw" and "Mickey's" becomes ""Ceikmy's". Consider the harder version as a Bonus.

#Sample inputs & outputs

##Input 1

> This challenge doesn't seem so hard.

##Output 1

> Hist aceeghlln denos't eems os adhr.

##Input 2

> There are more things between heaven and earth, Horatio, than are dreamt of in your philosophy.

##Output 2

> Eehrt aer emor ghinst beeentw aeehnv adn aehrt, Ahioort, ahnt aer ademrt fo in oruy hhilooppsy.

#Challenge inputs

##Input 1

> Eye of Newt, and Toe of Frog, Wool of Bat, and Tongue of Dog.

##Input 2

> Adder's fork, and Blind-worm's sting, Lizard's leg, and Howlet's wing.

##Input 3

> For a charm of powerful trouble, like a hell-broth boil and bubble.

Title: [2015-06-29] Challenge #221 [Easy] Word snake
Text: #Description

A word snake is (unsurprisingly) a snake made up of a sequence of words.

For instance, take this sequence of words:

```
SHENANIGANS SALTY YOUNGSTER ROUND DOUBLET TERABYTE ESSENCE
```

Notice that the last letter in each word is the same as the first letter in the next word. In order to make this into a word snake, you simply snake it across the screen

```
SHENANIGANS
      A
      L
      T
      YOUNGSTER
            O
            U
            N
       TELBUOD
       E
       R
       A
       B
       Y
       T
       ESSENCE
```

Your task today is to take an input word sequence and turn it into a word snake. Here are the rules for the snake:

* It has to start in the top left corner
* Each word has to turn 90 degrees left or right to the previous word
* The snake can't intersect itself

Other than that, you're free to decide how the snake should "snake around". If you want to make it easy for yourself and simply have it alternate between going right and going down, that's perfectly fine. If you want to make more elaborate shapes, that's fine too.

#Formal inputs & outputs

##Input

The input will be a single line of words (written in ALL CAPS). The last letter of each word will be the first letter in the next.

##Output

Your word snake! Make it look however you like, as long as it follows the rules.

#Sample inputs & outputs

There are of course many possible outputs for each inputs, these just show a sample that follows the rules

##Input 1

```
SHENANIGANS SALTY YOUNGSTER ROUND DOUBLET TERABYTE ESSENCE
```

##Output 1

```
  SHENANIGANS      DOUBLET
        A      N    E
        L      U    R
        T      O    A
        YOUNGSTER    B
                 Y
                 T
                 ESSENCE
```

##Input 2

DELOREAN NEUTER RAMSHACKLE EAR RUMP PALINDROME EXEMPLARY YARD

##Output 2

```
  D
  E
  L
  O
  R
  E          DRAY
  A            R
  NEUTER       A
     A         L
     M         P
     S         M
     H         E
     A         X
     C PALINDROME
     K M
     L U
     EAR
```

#Challenge inputs

##Input 1

CAN NINCOMPOOP PANTS SCRIMSHAW WASTELAND DIRK KOMBAT TEMP PLUNGE ESTER REGRET TOMBOY

##Input 2

NICKEL LEDERHOSEN NARCOTRAFFICANTE EAT TO OATS SOUP PAST TELEMARKETER RUST THINGAMAJIG GROSS SALTPETER REISSUE ELEPHANTITIS

Title: [2015-07-06] Challenge #222 [Easy] Balancing Words
Text: # Description

Today we're going to balance words on one of the letters in them. We'll use the position and letter itself to calculate the weight around the balance point. A word can be balanced if the weight on either side of the balance point is equal. Not all words can be balanced, but those that can are interesting for this challenge.

The formula to calculate the weight of the word is to look at the letter position in the English alphabet (so A=1, B=2, C=3 ... Z=26) as the letter weight, then multiply that by the distance from the balance point, so the first letter away is multiplied by 1, the second away by 2, etc.

As an example:

STEAD balances at T: 1 * S(19) = 1 * E(5) + 2 * A(1) + 3 * D(4))

Input Description

You'll be given a series of English words. Example:

    STEAD

Output Description

Your program or function should emit the words split by their balance point and the weight on either side of the balance point. Example:

    S T EAD - 19

This indicates that the T is the balance point and that the weight on either side is 19.

Challenge Input

    CONSUBSTANTIATION
    WRONGHEADED
    UNINTELLIGIBILITY
    SUPERGLUE

Challenge Output

Updated - the weights and answers I had originally were wrong. My apologies.

    CONSUBST A NTIATION - 456
    WRO N GHEADED - 120
    UNINTELL I GIBILITY - 521
    SUPERGLUE DOES NOT BALANCE

Title: [2015-07-13] Challenge #223 [Easy] Garland words
Text: # Description

A [_garland word_](http://blog.vivekhaldar.com/post/89763722591/garland-words) is one that starts and ends with the same N letters in the same order, for some N greater than 0, but less than the length of the word. I'll call the maximum N for which this works the garland word's _degree_. For instance, "onion" is a garland word of degree 2, because its first 2 letters "on" are the same as its last 2 letters. The name "garland word" comes from the fact that you can make chains of the word in this manner:

    onionionionionionionionionionion...

Today's challenge is to write a function `garland` that, given a lowercase word, returns the degree of the word if it's a garland word, and 0 otherwise.

Examples

    garland("programmer") -> 0

garland("ceramic") -> 1
garland("onion") -> 2
garland("alfalfa") -> 4

Optional challenges

1. Given a garland word, print out the chain using that word, as with "onion" above. You can make it as long or short as you like, even infinite.
1. Find the largest degree of any garland word in the [enable1 English word list](https://code.google.com/p/dotnetperls-controls/downloads/detail?name=enable1.txt).
1. Find a word list for some other language, and see if you can find a language with a garland word with a higher degree.

Title: [2015-07-20] Challenge #224 [Easy] Shuffling a List
Text: # Description

We've had our fair share of sorting algorithms, now let's do a *shuffling* challenge. In this challenge, your challenge is to take a list of inputs and change around the order in random ways. Think about shuffling cards - can your program shuffle cards?

EDIT 07-25-2014 In case this isn't obvious, the intention of this challenge is for you to implement this yourself and not rely on a standard library built in (e.g. Python's "random.shuffle()" or glibc's "strfry()").

Input Description

You'll be given a list of values - integers, letters, words - in one order. The input list will be space separated. Example:

    1 2 3 4 5 6 7 8

Output Description

Your program should emit the values in any non-sorted order; sequential runs of the program or function should yield different outputs. You should maximize the disorder if you can. From our example:

    7 5 4 3 1 8 2 6

Challenge Input

    apple blackberry cherry dragonfruit grapefruit kumquat mango nectarine persimmon raspberry raspberry
    a e i o u

Challenge Output

Examples only, this is all about shuffling

    raspberry blackberry nectarine kumquat grapefruit cherry raspberry apple mango persimmon dragonfruit
    e a i o u

Bonus

Check out the [Faro shuffle](https://en.wikipedia.org/wiki/Faro_shuffle) and the [Fisher-Yates shuffles](https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle), which are algorithms for specific shuffles. Shuffling has some interesting mathematical properties.

Title: [2015-07-27] Challenge #225 [Easy/Intermediate] De-columnizing
Text: # [](#EasyIcon) _(Easy/Intermediate)_: De-columnizing

Often, column-style writing will put images and features to the left or right of the body of text, for example:

```
  24
  This is an example piece of text. This is an exam-
  ple piece of text. This is an example piece of
  text. This is an example
  piece of text. This is a +-----------------------+
  sample for a challenge.  |                       |
  Lorum ipsum dolor sit a- |      top class        |
  met and other words. The |       feature         |
  proper word for a layout |                       |
  like this would be type- +-----------------------+
  setting, or so I would
  imagine, but for now let's carry on calling it an
  example piece of text. Hold up - the end of the
               paragraph is approaching - notice
  +--------------+ the double line break for a para-
  |              | graph.
  |              |
  |   feature    | And so begins the start of the
  |   bonanza    | second paragraph but as you can
  |              | see it's only marginally better
  |              | than the other one so you've not
  +--------------+ really gained much - sorry. I am
               certainly not a budding author
  as you can see from this example input. Perhaps I
  need to work on my writing skills.
```

In order to fit into the column format, some words are hyphenated. For the purpose of the challenge, you may assume that any hyphens at the end of a line join a single un-hyphenated word together (for example, the `exam-` and `ple` in the above input form the word `example` and not `exam-ple`). However, hyphenated words that do not span multiple lines should retain their hyphens. Side features will only appear at the far left or right of the input, and will always be bordered by the `+---+` style shown above. They will also never have 'holes' in them, like this:

```
  +--------------------+
  |                    |
  | Inside the feature |
  |                    |
  | +----------------+ |
  | |                | |
  | |    Outside     | |
  | |                | |
  | +----------------+ |
  |                    |
  +--------------------+
```

Paragraphs in the input are separated by double line breaks, like Reddit markdown. Your task today is to extract just the paragraph text from the input, removing the feature-boxes.

Formal Inputs and Outputs

Input Specification

You'll be given a number **N** on one line, followed by **N** further lines of input like the example in the description above.

Output Description

Output just the paragraph text, de-hyphenating words where appropriate (ie. a line of text ends with a hyphen).

Sample Inputs and Outputs

Example 1

This corresponds to the input given in the Description.

Output

This is an example piece of text. This is an example piece of text. This is an example piece of text. This is an example piece of text. This is a sample for a challenge. Lorum ipsum dolor sit amet and other words. The proper word for a layout like this would be typesetting, or so I would imagine, but for now let's carry on calling it an example piece of text. Hold up - the end of the paragraph is approaching - notice the double line break for a paragraph.

And so begins the start of the second paragraph but as you can see it's only marginally better than the other one so you've not really gained much - sorry. I am certainly not a budding author as you can see from this example input. Perhaps I need to work on my writing skills.

Example 2

Input

```
  22
  +-------------+ One hundred and fifty quadrillion,
  |             | seventy-two trillion, six hundred
  | 150 072 626 | and twenty-six billion, eight hun-
  | 840 312 999 | dred and fourty million, three
  |             | hundred and thirteen thousand sub-
  +-------------+ tract one is a rather large prime
                  number which equals one to five if
  calculated modulo two to six respectively.

  However, one other rather more in- +-------------+
  teresting number is two hundred    |             |
  and twenty-one quadrillion, eight  | 221 806 434 |
  hundred and six trillion, four     | 537 978 679 |
  hundred and thirty-four billion,   |             |
  five hundred and thirty-seven mil- +-------------+
  million, nine hundred and seven-
                            ty-eight thousand,
  +-----------------------------+ six hundred and
  |                             | seventy nine,
  | Subscribe for more Useless  | which isn't prime
  |     Number Facts(tm)!       | but is the 83rd
  +-----------------------------+ Lucas number.
```

Output

One hundred and fifty quadrillion, seventy-two trillion, six hundred and twenty-six billion, eight hundred and fourty million, three hundred and thirteen thousand subtract one is a rather large prime number which equals one to five if calculated modulo two to six respectively.

However, one other rather more interesting number is two hundred and twenty-one quadrillion, eight hundred and six trillion, four hundred and thirty-four billion, five hundred and thirty-seven milmillion, nine hundred and seventy-eight thousand, six hundred and seventy nine, which isn't prime but is the 83rd Lucas number.

Example 3

Input

```
  16
  +----------------+ Lorem ipsum dolor sit amet,
  |                | consectetur adipiscing elit,
  |  Aha, now you  | sed do eiusmod tempor incid-
  |  are stumped!! | idunt ut labore et dolore
  |                | magna aliqua. Ut enim ad mi-
  |      +---------+ nim veniam, quis nostrud ex-
  |  top  |          ercitation ullamco laboris
  |  kek  | nisi ut aliquip ex.
  |       |                    +--------------+
  +-------+ Duis aute irure dolor |              |
  in repre-henderit in voluptate  | Nothing to   |
  velit esse cillum dolore eu fu- |  see here.   |
  giat nulla pariatur. Excepteur  |              |
  sint occaecat cupidatat non     +--------------+
  proident, sunt in culpa qui of-
  ficia deserunt mollit anim id est laborum.
```

Output

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex.

Duis aute irure dolor in repre-henderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[](#IntermediateIcon) Extension (Intermediate)

At the start of each paragraph in your output, list the text of each feature associated with that paragraph. A feature is "associated" with a paragraph if the top of the feature box (the `+--------+`) starts on or below the line that the paragraph starts on. For example, the outputs for the above three examples would be:

Example 1 Output

(top class feature) (feature bonanza) This is an example piece of text. This is an example piece of text. This is an example piece of text. This is an example piece of text. This is a sample for a challenge. Lorum ipsum dolor sit amet and other words. The proper word for a layout like this would be typesetting, or so I would imagine, but for now let's carry on calling it an example piece of text. Hold up - the end of the paragraph is approaching - notice the double line break for a paragraph.

And so begins the start of the second paragraph but as you can see it's only marginally better than the other one so you've not really gained much - sorry. I am certainly not a budding author as you can see from this example input. Perhaps I need to work on my writing skills.

Example 2 Output

(**150 072 626 840 312 999**) One hundred and fifty quadrillion, seventy-two trillion, six hundred and twenty-six billion, eight hundred and fourty million, three hundred and thirteen thousand subtract one is a rather large prime number which equals one to five if calculated modulo two to six respectively.

(**221 806 434 537 978 679**) (**Subscribe for more Useless Number Facts(tm)!**) However, one other rather more interesting number is two hundred and twenty-one quadrillion, eight hundred and six trillion, four hundred and thirty-four billion, five hundred and thirty-seven milmillion, nine hundred and seventy-eight thousand, six hundred and seventy nine, which isn't prime but is the 83rd Lucas number.

Example 3 Output

(**Aha, now you are stumped! top kek**) (**Nothing to see here.**) Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex.

Duis aute irure dolor in repre-henderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Title: [2015-08-03] Challenge #226 [Easy] Adding fractions
Text: #Description

Fractions are the bane of existence for many elementary and middle-schoolers. They're sort-of hard to get your head around (though thinking of them as pizza slices turned out to be very helpful for me), but even worse is that they're so hard to calculate with! Even adding them together is no picknick.

Take, for instance, the two fractions 1/6 and 3/10. If they had the same denominator, you could simply add the numerators together, but since they have different denominators, you can't do that. First, you have to make the denominators equal. The easiest way to do that is to use cross-multiplication to make both denominators 60 (i.e. the original denominators multiplied together, 6 * 10). Then the two fractions becomes 10/60 and 18/60, and you can then add those two together to get 28/60.

(if you were a bit more clever, you might have noticed that the lowest common denominator of those fractions is actually 30, not 60, but it doesn't really make much difference).

You might think you're done here, but you're not! 28/60 has not been reduced yet, those two numbers have factors in common! The greatest common divisor of both is 4, so we divide both numerator and denominator with 4 to get 7/15, which is the real answer.

For today's challenge, you will get a list of fractions which you will add together and produce the resulting fraction, reduced as far as possible.

NOTE: Many languages have libraries for rational arithmetic that would make this challenge really easy (for instance, Python's `fractions` module does exactly this). You are allowed to use these if you wish, but the spirit of this challenge is to try and implement the logic yourself. I highly encourage you to only use libraries like that if you can't figure out how to do it any other way.

#Formal inputs & outputs

##Inputs
The input will start with a single number N, specifying how many fractions there are to be added.

After that, there will follow N rows, each one containing a fraction that you are supposed to add into the sum. Each fraction comes in the form "X/Y", so like "1/6" or "3/10", for instance.

##Output
The output will be a single line, containing the resulting fraction reduced so that the numerator and denominator has no factors in common.

#Sample inputs & outputs

##Input 1

2
1/6
3/10

##Output 1

7/15

##Input 2

3
1/3
1/4
1/12

##Output 2

2/3

#Challenge inputs

##Input 1

5
2/9
4/35
7/34
1/2
16/33

##Input 2

10
1/7
35/192
61/124
90/31
5/168
31/51
69/179
32/5
15/188
10/17

Title: [2015-08-10] Challenge #227 [Easy] Square Spirals
Text: # [](#EasyIcon) __(Easy)__: Square Spirals

Take a square grid, and put a cross on the center point, like this:

+ + + + +

+ + + + +

+ + X + +

```
+ + + + +

+ + + + +
```

The grid is 5-by-5, and the cross indicates point 1. Let's call the top-left corner location (1, 1), so the center point is at location (3, 3). Now, place another cross to the right, and trace the path:

```
+ + + + +

+ + + + +

+ + X-X +

+ + + + +

+ + + + +
```

This second point (point 2) is now at location (4, 3). If you continually move around anticlockwise as much as you can from this point, you will form a square spiral, as this diagram shows the beginning of:

```
+ + + + +

+ X-X-X .
  |   | .
+ X X-X .
  |     |
+ X-X-X-X

+ + + + +
```

Your challenge today is to do two things: convert a point number to its location on the spiral, and vice versa.

Formal Inputs and Outputs

Input Specification

On the first line, you'll be given a number **S**. This is the size of the spiral. If **S** equals 5, then the grid is a 5-by-5 grid, as shown in the demonstration above. **S** will always be an odd number.

You will then be given one of two inputs on the next line:

* You'll be given a single number **N** - this is the point number of a point on the spiral.

* You'll be given two numbers **X** and **Y** (on the same line, separated by a space) - this is the location of a point on the spiral.

Output Description

If you're given the point number of a point, work out its location. If you're given a location, find out its point number.

Sample Inputs and Outputs

Example 1

(Where is 8 on this spiral?)

```
5-4-3
|   |
6 1-2
```

```
  |
  7-8-9
```

Input

```
3
8
```

Output

```
(2, 3)
```

Example 2

This corresponds to the top-left point (1, 1) in [this 7-by-7 grid](https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Ulam_spiral_howto_all_numbers.svg/811px-Ulam_spiral_howto_all_numbers.svg.png).

Input

```
7
1 1
```

Output

```
37
```

Example 3

Input

```
11
50
```

Output

```
(10, 9)
```

Example 4

Input

```
9
6 8
```

Output

```
47
```

If your solution can't solve the next two inputs before the heat death of the universe, don't worry.

Example 5

Let's test how fast your solution is!

Input

```
1024716039
```

557614022

Output

(512353188, 512346213)

Example 6

:D

Input

234653477
11777272 289722

Output

54790653381545607