

- [1.Count letters in string](#)
- [2.Find all pairs](#)
- [3.Return substring instance count](#)
- [4.Alphabet symmetry](#)
- [5.Longest vowel chain](#)
- [6.Non-even substrings](#)
- [7.Substring fun](#)
- [8.Repeated Substring](#)
- [9.Typoglycemia Generator](#)
- [10.Most frequently used words in a text\]](#)
- [11.Extract the domain name from a URL](#)
- [12. Detect Pangram](#)
- [13. Kebabize](#)
- [14. Dubstep](#)
- [15. Take a Ten Minute Walk](#)
- [16. Stop gninnipS My sdroW!](#)
- [17. Write Number in Expanded Form](#)
- [18. Persistent Bugger.](#)
- [19. Title Case](#)
- [20. Count and Group Character Occurrences in a String](#)
- [21. Find the Mine!](#)
- [22. Scramblies](#)
- [23. Longest alphabetical substring](#)
- [24. The Hashtag Generator](#)
- [25. Pete, the baker](#)
- [26. Mean Square Error](#)
- [27. Exponent method](#)
- [28. Where my anagrams at?](#)
- [29. Split Strings](#)
- [30. Anagram difference](#)
- [31. Anagram Detection](#)
- [32. Highest Scoring Word](#)

[33. Replace With Alphabet Position](#)

[34. Sherlock on pockets](#)

[35. Mexican Wave](#)

[36. Delete a Digit](#)

[37. Multiples of 3 or 5](#)

[38. String transformer](#)

[39. Largest product in a series](#)

[40. Duplicate Encoder](#)

[41. Backspaces in string](#)

[42. Sort Arrays \(Ignoring Case\)](#)

[43. Transform To Prime](#)

[44. Counting Duplicates](#)

[45. Alphabetized](#)

[46. Sum of Digits / Digital Root](#)

[47. Array.diff](#)

[48. Where is my parent!?\(cry\)](#)

[49. Playing with digits](#)

[50. Equal Sides Of An Array](#)

[51. Reverse or rotate?](#)

[51. Decipher this!](#)

[52. Bouncing Balls](#)

[53. WeIrD StRiNg CaSe](#)

[54. Are they the "same"?](#)

[55. Grouping and Counting](#)

[56. Find the Nexus of the Codewars Universe](#)

[57. Count letters in string](#)

[58. Triple trouble](#)

[59. Which are in?](#)

[60. Format a string of names like 'Bart, Lisa & Maggie'.](#)

[61. Find the missing letter](#)

[62. Who likes it?](#)

[63. Find The Parity Outlier](#)

[64. Is Integer Array?](#)

[65. Reversing and Combining Text](#)

[66. Integer reduction](#)

## 1.Count letters in string

(<https://www.codewars.com/kata/5808ff71c7cfa1c6aa00006d/train/ruby>)

6 kyu

=begin

Count letters in string

In this kata, you've to count lowercase letters in a given string and return the letter count in a hash with 'letter' as key and count as 'value'. The key must be 'symbol' instead of string in Ruby and 'char' instead of string in Crystal.

Example:

```
letterCount('arithmetics') #=> {:a=>1, :c=>1, :e=>1, :h=>1, :i=>2, :m=>1, :r=>1, :s=>1, :t=>2}
```

=end

```
p letter_count('codewars') == {:a=>1, :c=>1, :d=>1, :e=>1, :o=>1, :r=>1, :s=>1, :w=>1})
```

```
p letter_count('activity') == {:a=>1, :c=>1, :i=>2, :t=>2, :v=>1, :y=>1}
```

```
p letter_count('arithmetics') == {:a=>1, :c=>1, :e=>1, :h=>1, :i=>2, :m=>1, :r=>1, :s=>1, :t=>2})
```

## 2.Find all pairs

(<https://www.codewars.com/kata/5c55ad8c9d76d41a62b4ede3/train/ruby>)

7 kyu

=begin

Find all pairs

You are given array of integers, your task will be to count all pairs in that array and return their count.

Notes:

Array can be empty or contain only one value; in this case return 0

If there are more pairs of a certain number, count each pair only once. E.g.: for [0, 0, 0, 0] the return value is 2 (= 2 pairs of 0s)

Random tests: maximum array length is 1000, range of values in array is between 0 and 1000

Examples

[1, 2, 5, 6, 5, 2] --> 2

...because there are 2 pairs: 2 and 5

[1, 2, 2, 20, 6, 20, 2, 6, 2] --> 4

...because there are 4 pairs: 2, 20, 6 and 2 (again)

=end

```
p pairs([1, 2, 5, 6, 5, 2]) == 2
p pairs([1, 2, 2, 20, 6, 20, 2, 6, 2]) == 4
p pairs([0, 0, 0, 0, 0, 0, 0]) == 3
p pairs([1000, 1000]) == 1
p pairs([]) == 0
p pairs([54]) == 0
```

=begin

### 3.Return substring instance count

(<https://www.codewars.com/kata/5168b125faced29f66000005/train/ruby>)

7 kyu

=begin

Return substring instance count

Complete the solution so that it returns the number of times the search\_text is found within the full\_text.

Usage example:

solution('aa\_bb\_cc\_dd\_bb\_e', 'bb') # should return 2 since bb shows up twice

solution('aaabbbcccc', 'bbb') # should return 1

=end

```
p solution('abcdeb','b') == 2
```

```
p solution('abcdeb', 'a') == 1
```

```
p solution('abbc', 'bb') == 1
```

### 4.Alphabet symmetry

(<https://www.codewars.com/kata/59d9ff9f7905dfeed50000b0>)

7 kyu

=begin

Alphabet symmetry

Consider the word "abode". We can see that the letter a is in position 1 and b is in position 2. In the alphabet, a and b are also in positions 1 and 2. Notice also that d and e in abode occupy the positions they would occupy in the alphabet, which are positions 4 and 5.

Given an array of words, return an array of the number of letters that occupy their positions in the alphabet for each word. For example,

```
solve(["abode","ABc","xyzD"]) = [4, 3, 1]
```

See test cases for more examples.

Input will consist of alphabet characters, both uppercase and lowercase. No spaces.

Good luck!

If you like this Kata, please try:

Last digit symmetry

Alternate capitalization

```
=end
p solve(["abode","ABc","xyzD"]) == [4,3,1]
p solve(["abide","ABc","xyz"]) == [4,3,0]
p solve(["IAMDEFANDJKL","thedefgh","xyzDEFghijabc"]) == [6,5,7]
p solve(["encode","abc","xyzD","ABmD"]) == [1, 3, 1, 3]
```

## 5.Longest vowel chain

(<https://www.codewars.com/kata/59c5f4e9d751df43cf000035/train/ruby>)

7 kyu

=begin

Longest vowel chain

The vowel substrings in the word codewarriors are o,e,a,io. The longest of these has a length of 2. Given a lowercase string that has alphabetic characters only and no spaces, return the length of the longest vowel substring. Vowels are any of aeiou.

```
=end
p solve("codewarriors") == 2
p solve("suoidea") == 3
p solve("iuuvgheaae") == 4
p solve("ultrarevolutionariees") == 3
p solve("strengthlessneses") == 1
p solve("cuboideonavicuare") == 2
p solve("chrononhotonthuooaos") == 5
p solve("iiihoovaeaaaooougjyaw") == 8
```

## 6.Non-even substrings

(<https://www.codewars.com/kata/59da47fa27ee00a8b90000b4/train/ruby>)

6 kyu

=begin

Given a string of integers, return the number of odd-numbered substrings that can be formed.

For example, in the case of "1341", they are 1, 1, 3, 13, 41, 341, 1341, a total of 7 numbers.

solve("1341") = 7. See test cases for more examples.

=end

```
p solve("1341") == 7
p solve("1357") == 10
p solve("13471") == 12
p solve("134721") == 13
p solve("1347231") == 20
p solve("13472315") == 28
```

## 7.Substring fun

(<https://www.codewars.com/kata/565b112d09c1adfd500019c/train/ruby>)

7 kyu

=begin

Complete the function that takes an array of words.

You must concatenate the nth letter from each word to construct a new word which should be returned as a string, where n is the position of the word in the list.

For example:

["yoda", "best", "has"] --> "yes"

^     ^     ^

n=0   n=1   n=2

Note: Test cases contain valid input only - i.e. a string array or an empty array; and each word will have enough letters.  
=end

p nth\_char(['yoda', 'best', 'has']) == 'yes'

p nth\_char([]) == ''

p nth\_char(['X-ray']) == 'X'

p nth\_char(['No', 'No']) == 'No'

p nth\_char(['Chad', 'Morocco', 'India', 'Algeria', 'Botswana', 'Bahamas', 'Ecuador', 'Micronesia']) == 'Codewars'

## 8.Repeated Substring

(<https://www.codewars.com/kata/5491689aff74b9b292000334/train/ruby>)

6 kyu

=begin

For a given nonempty string s find a minimum substring t and the maximum number k, such that the entire string s is equal to t repeated k times. The input string consists of lowercase latin letters. Your function should return a tuple (in Python) (t, k) or an array (in Ruby and JavaScript) [t, k]

Example #1:

for string

s = "ababab"

the answer is

["ab", 3]

Example #2:

for string

s = "abcde"

the answer is

because for this string "abcde" the minimum substring t, such that s is t repeated k times, is itself.

=end

```
p f("ababab") == ["ab", 3]
p f("abcde") == ["abcde", 1]
```

## 9. Typoglycemia Generator

(<https://www.codewars.com/kata/55953e906851cf2441000032/train/ruby>)

5 kyu

=begin

Background

There is a message that is circulating via public media that claims a reader can easily read a message where the inner letters of each words is scrambled, as long as the first and last letters remain the same and the word contains all the letters.

Another example shows that it is quite difficult to read the text where all the letters are reversed rather than scrambled.

In this kata we will make a generator that generates text in a similar pattern, but instead of scrambled or reversed, ours will be sorted alphabetically

Requirement

return a string where:

- 1) the first and last characters remain in original place for each word
- 2) characters between the first and last characters must be sorted alphabetically
- 3) punctuation should remain at the same place as it started, for example: shan't -> sahn't

Assumptions

- 1) words are seperated by single spaces
- 2) only spaces separate words, special characters do not, for example: tik-tak -> tai-ktk
- 3) special characters do not take the position of the non special characters, for example: -dcba -> -dbca
- 4) for this kata puctuation is limited to 4 characters: hyphen(-), apostrophe('), comma(,) and period(.)
- 5) ignore capitalisation

for reference: <http://en.wikipedia.org/wiki/Typoglycemia>

=end

```
p scramble_words('professionals') == 'paefilnoorsss'
p scramble_words('i') == 'i'
p scramble_words('') == ''
p scramble_words('me') == 'me'
p scramble_words('you') == 'you'
p scramble_words('card-carrying') == 'caac-dinrryg'
p scramble_words("shan't") == "sahn't"
p scramble_words('-dcba') == '-dbca'
p scramble_words('dcba.') == 'dbca.'
```

p scramble\_words("you've gotta dance like there's nobody watching, love like you'll never be hurt, sing like there's nobody listening, and live like it's heaven on earth.") == "you've gotta dance like there's nobody watching, love like you'll never be hurt, sing like there's nobody listening, and live like it's heaven on earth."

## 10. Most frequently used words in a text

(<https://www.codewars.com/kata/51e056fe544cf36c410000fb/train/ruby>)

4 kyu

=begin

Write a function that, given a string of text (possibly with punctuation and line-breaks), returns an array of the top-3 most occurring words, in descending order of the number of occurrences.

Assumptions:

A word is a string of letters (A to Z) optionally containing one or more apostrophes (') in ASCII. (No need to handle fancy punctuation.)

Matches should be case-insensitive, and the words in the result should be lowercased.

Ties may be broken arbitrarily.

If a text contains fewer than three unique words, then either the top-2 or top-1 words should be returned, or an empty array if a text contains no words.

Examples:

```
top_3_words("In a village of La Mancha, the name of which I have no desire to call to
mind, there lived not long since one of those gentlemen that keep a lance
in the lance-rack, an old buckler, a lean hack, and a greyhound for
coursing. An olla of rather more beef than mutton, a salad on most
nights, scraps on Saturdays, lentils on Fridays, and a pigeon or so extra
on Sundays, made away with three-quarters of his income.")
# => ["a", "of", "on"]
```

```
top_3_words("e e e e DDD ddd DdD: ddd ddd aa aA Aa, bb cc cC e e e")
# => ["e", "ddd", "aa"]
```

```
top_3_words(" //wont won't won't")
# => ["won't", "wont"]
```

Bonus points (not really, but just for fun):

Avoid creating an array whose memory footprint is roughly as big as the input text.

Avoid sorting the entire array of unique words.

=end

```
p top_3_words("a a a b c c d d d d e e e e") == ["e", "d", "a"]
p top_3_words("e e e e DDD ddd DdD: ddd ddd aa aA Aa, bb cc cC e e e") == ["e", "ddd", "aa"]
p top_3_words(" //wont won't won't ") == ["won't", "wont"]
p top_3_words(" , e .. ") == ["e"]
p top_3_words(" ... ") == []
p top_3_words(" ' ") == []
p top_3_words(" "" ") == []
p top_3_words("""In a village of La Mancha, the name of which I have no desire to call to
mind, there lived not long since one of those gentlemen that keep a lance
in the lance-rack, an old buckler, a lean hack, and a greyhound for
coursing. An olla of rather more beef than mutton, a salad on most
nights, scraps on Saturdays, lentils on Fridays, and a pigeon or so extra
```



on Sundays, made away with three-quarters of his income."") == ["a", "of", "on"]

## 11. Extract the domain name from a URL

(<https://www.codewars.com/kata/514a024011ea4fb54200004b/train/ruby>)

5 kyu

=begin

Write a function that when given a URL as a string, parses out just the domain name and returns it as a string. For example:

```
domain_name("http://github.com/carbonfive/raygun") == "github"
domain_name("http://www.zombie-bites.com") == "zombie-bites"
domain_name("https://www.cnet.com") == "cnet"
=end
```

```
p domain_name("http://google.com") == "google"
p domain_name("http://google.co.jp") == "google"
p domain_name("www.xakep.ru") == "xakep"
p domain_name("https://youtube.com") == "youtube"
```

## 12. Detect Pangram

(<https://www.codewars.com/kata/545cedaa9943f7fe7b000048/train/ruby>)

6 kyu

A pangram is a sentence that contains every single letter of the alphabet at least once. For example, the sentence "The quick brown fox jumps over the lazy dog" is a pangram, because it uses the letters A-Z at least once (case is irrelevant).

Given a string, detect whether or not it is a pangram. Return True if it is, False if not. Ignore numbers and punctuation.

=end

```
p panagram?("The quick brown fox jumps over the lazy dog.") == true
p panagram?("This is not a pangram.") == false
```

## 13. Kebabize

(<https://www.codewars.com/kata/57f8ff867a28db569e000c4a/train/ruby>)

6 kyu

Modify the kebabize function so that it converts a camel case string into a kebab case.

```
kebabize('camelsHaveThreeHumps') // camels-have-three-humps
kebabize('camelsHave3Humps') // camels-have-humps
```

Notes:

the returned string should only contain lowercase letters

=end

```
p kebabize('myCamelCasedString') == 'my-camel-cased-string'
p kebabize('myCamelHas3Humps') == 'my-camel-has-humps'
```

## 14. Dubstep

(<https://www.codewars.com/kata/551dc350bf4e526099000ae5/train/ruby>)

6 kyu

Polycarpus works as a DJ in the best Berland nightclub, and he often uses dubstep music in his performance. Recently, he has decided to take a couple of old songs and make dubstep remixes from them.

Let's assume that a song consists of some number of words (that don't contain WUB). To make the dubstep remix of this song, Polycarpus inserts a certain number of words "WUB" before the first word of the song (the number may be zero), after the last word (the number may be zero), and between words (at least one between any pair of neighbouring words), and then the boy glues together all the words, including "WUB", in one string and plays the song at the club.

For example, a song with words "I AM X" can transform into a dubstep remix as "WUBWUBIWUBAMWUBWUBX" and cannot transform into "WUBWUBIAMWUBX".

Recently, Jonny has heard Polycarpus's new dubstep track, but since he isn't into modern music, he decided to find out what was the initial song that Polycarpus remixed. Help Jonny restore the original song.

Input

The input consists of a single non-empty string, consisting only of uppercase English letters, the string's length doesn't exceed 200 characters

Output

Return the words of the initial song that Polycarpus used to make a dubstep remix. Separate the words with a space.

Examples

```
song_decoder("WUBWEWUBAREWUBWUBTHEWUBCHAMPIONSWUBMYWUBFRIENDWUB") => WE ARE THE CHAMPIONS MY FRIEND
```

```
=end
```

```
p song_decoder("AWUBBWUBC") == "A B C"
```

```
p song_decoder("AWUBWUBWUBBWUBWUBWUBC") == "A B C"
```

```
p song_decoder("WUBAWUBBWUBCWUB") == "A B C"
```

## 15. Take a Ten Minute Walk

(<https://www.codewars.com/kata/54da539698b8a2ad76000228/train/ruby>)

6 kyu

You live in the city of Cartesia where all roads are laid out in a perfect grid. You arrived ten minutes too early to an appointment, so you decided to take the opportunity to go for a short walk. The city provides its citizens with a Walk Generating App on their phones -- everytime you press the button it sends you an array of one-letter strings representing directions to walk (eg. ['n', 's', 'w', 'e']). You always walk only a single block in a direction and you know it takes you one minute to traverse one city block, so create a function that will return true if the walk the app gives you will take you exactly ten minutes (you don't want to be early or late!) and will, of course, return you to your starting point. Return false otherwise.

Note: you will always receive a valid array containing a random assortment of direction letters ('n', 's', 'e', or 'w' only). It will never give you an empty array (that's not a walk, that's standing still!).

```
=end
```

```
p is_valid_walk(['n','s','n','s','n','s','n','s','n','s']) == true
```

```
p is_valid_walk(['w','e','w','e','w','e','w','e','w','e']) == false
p is_valid_walk(['w']) == false
p is_valid_walk(['n','n','n','s','n','s','n','s','n','s']) == false
```

=begin

## 16. Stop gninnipS My sdroW!

(<https://www.codewars.com/kata/5264d2b162488dc400000001>)

6 kyu

Write a function that takes in a string of one or more words, and returns the same string, but with all five or more letter words reversed (Just like the name of this Kata). Strings passed in will consist of only letters and spaces. Spaces will be included only when more than one word is present.

Examples: spinWords( "Hey fellow warriors" ) => returns "Hey wollef sroirraw"

=end

```
p spinWords("Hey fellow warriors") == "Hey wollef sroirraw"
p spinWords("This is a test") == "This is a test"
p spinWords("This is another test") == "This is rehtona test"
P spinWords('test') == 'test'
```

## 17. Write Number in Expanded Form

(<https://www.codewars.com/kata/5842df8ccbd22792a4000245>)

6 kyu

Write Number in Expanded Form

You will be given a number and you will need to return it as a string in Expanded Form. For example:

```
expanded_form(12); # Should return '10 + 2'
expanded_form(42); # Should return '40 + 2'
expanded_form(70304); # Should return '70000 + 300 + 4'
NOTE: All numbers will be whole numbers greater than 0.
```

If you liked this kata, check out part 2!!

=end

```
p expanded_form(12) == '10 + 2'
p expanded_form(42) == '40 + 2'
p expanded_form(70304) == '70000 + 300 + 4'
```

## 18. Persistent Bugger.

(<https://www.codewars.com/kata/55bf01e5a717a0d57e0000ec/train/ruby>)

6 kyu

Write a function, persistence, that takes in a positive parameter num and returns its multiplicative persistence, which is the number of times you must multiply the digits in num until you reach a single digit.

For example:

persistence(39) # returns 3, because  $3*9=27$ ,  $2*7=14$ ,  $1*4=4$   
# and 4 has only one digit

persistence(999) # returns 4, because  $9*9*9=729$ ,  $7*2*9=126$ ,  
#  $1*2*6=12$ , and finally  $1*2=2$

persistence(4) # returns 0, because 4 is already a one-digit number  
=end

```
p persistence(39) == 3
p persistence(4) == 0
p persistence(25) == 2
p persistence(999) == 4
```

## 19. Title Case

(<https://www.codewars.com/kata/5202ef17a402dd033c000009>)

6 kyu

A string is considered to be in title case if each word in the string is either (a) capitalised (that is, only the first letter of the word is in upper case) or (b) considered to be an exception and put entirely into lower case unless it is the first word, which is always capitalised.

Write a function that will convert a string into title case, given an optional list of exceptions (minor words). The list of minor words will be given as a string with each word separated by a space. Your function should ignore the case of the minor words string -- it should behave in the same way even if the case of the minor word string is changed.

```
=end
p title_case('a clash of KINGS', 'a an the of') == 'A Clash of Kings'
p title_case('THE WIND IN THE WILLOWS', 'The In') == 'The Wind in the Willows'
p title_case('the quick brown fox') == 'The Quick Brown Fox'
```

## 20. Count and Group Character Occurrences in a String

(<https://www.codewars.com/kata/543e8390386034b63b001f31>)

6 kyu

Write a method that takes a string as an argument and groups the number of times each character appears in the string as a hash sorted by the highest number of occurrences.

The characters should be sorted alphabetically e.g:

```
get_char_count("cba") => {1=>["a", "b", "c"]}
You should ignore spaces, special characters and count uppercase letters as lowercase ones.
=end
p get_char_count("Mississippi") == {4=>["i", "s"], 2=>["p"], 1=>["m"]}
p get_char_count("Hello. Hello? HELLO!!") == {6=>["l"], 3=>["e", "h", "o"]}
p get_char_count("aaa...bb...c!") == {3=>["a"], 2=>["b"], 1=>["c"]}
p get_char_count("aaabbbccc") == {3=>["a", "b", "c"]}
p get_char_count("abc123") == {1=>["1", "2", "3", "a", "b", "c"]}
```

## 21. Find the Mine!

(<https://www.codewars.com/kata/528d9adf0e03778b9e00067e/train/ruby>)

6 kyu

You've just discovered a square (NxN) field and you notice a warning sign. The sign states that there's a single bomb in the 2D grid-like field in front of you.

Write a function `mineLocation`/`MineLocation` that accepts a 2D array, and returns the location of the mine. The mine is represented as the integer 1 in the 2D array. Areas in the 2D array that are not the mine will be represented as 0s.

The location returned should be an array (Tuple<int, int> in C#) where the first element is the row index, and the second element is the column index of the bomb location (both should be 0 based). All 2D arrays passed into your function will be square (NxN), and there will only be one mine in the array.

=end

```
p mineLocation( [ [1, 0, 0], [0, 0, 0], [0, 0, 0] ] ) == [0, 0]
```

```
p mineLocation( [ [0, 0, 0], [0, 1, 0], [0, 0, 0] ] ) == [1, 1]
```

```
p mineLocation( [ [0, 0, 0], [0, 0, 0], [0, 1, 0] ] ) == [2, 1]
```

```
p mineLocation([ [1, 0], [0, 0] ]) == [0, 0]
```

```
p mineLocation([ [1, 0, 0], [0, 0, 0], [0, 0, 0] ]) == [0, 0]
```

```
p mineLocation([ [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 1, 0], [0, 0, 0, 0] ]) == [2, 2]
```

## 22. Scramblies

(<https://www.codewars.com/kata/55c04b4cc56a697bb0000048/train/ruby>)

5 kyu

Complete the function `scramble(str1, str2)` that returns true if a portion of `str1` characters can be rearranged to match `str2`, otherwise returns false.

Notes:

Only lower case letters will be used (a-z). No punctuation or digits will be included.

Performance needs to be considered

Input strings `s1` and `s2` are null terminated.

=end

```
p scramble('rkqodlw', 'world') == true
```

```
p scramble('cedewaraaossoqqyt', 'codewars') == true
```

```
p scramble('katas', 'steak') == false
```

```
p scramble('rkqodlw','world') == true
```

```
p scramble('cedewaraaossoqqyt','codewars') == true
```

```
p scramble('katas','steak') == false
```

```
p scramble('scriptjava','javascript') == true
```

```
p scramble('scriptingjava','javascript') == true
```

## 23. Longest alphabetical substring

(<https://www.codewars.com/kata/5a7f58c00025e917f30000f1>)

6 kyu

Find the longest substring in alphabetical order.

Example: the longest alphabetical substring in "asdfaaaabbbbcttavvffffdf" is "aaaabbbbctt".

There are tests with strings up to 10 000 characters long so your code will need to be efficient.

The input will only consist of lowercase characters and will be at least one letter long.

If there are multiple solutions, return the one that appears first.

```
=end
```

```
p longest('asd') == 'as'
```

```
p longest('nab') == 'ab'
```

```
p longest('abcdeapbcdef') == 'abcde'
```

```
p longest('asdfaaaabbbbcttavvffffdf') == 'aaaabbbbctt'
```

```
p longest('asdfbyfgiklag') == 'fgikl'
```

```
p longest('z') == 'z'
```

```
p longest('zyba') == 'z'
```

## 24. The Hashtag Generator

(<https://www.codewars.com/kata/52449b062fb80683ec000024>)

5 kyu

The marketing team is spending way too much time typing in hashtags.

Let's help them with our own Hashtag Generator!

Here's the deal:

It must start with a hashtag (#).

All words must have their first letter capitalized.

If the final result is longer than 140 chars it must return false.

If the input or the result is an empty string it must return false.

## Examples

" Hello there thanks for trying my Kata" => "#HelloThereThanksForTryingMyKata"

```
" Hello World "      => "#HelloWorld"
```

```
'''          => false
```

```
=end
```

```
p generateHashtag("") == false
```

```
p generateHashtag(" " * 200) == false
```

```
p generateHashtag("Do We have A Hashtag") == "#DoWeHaveAHashtag"
```

```
p generateHashtag("Codewars") == "#Codewars"
```

```
p generateHashtag("Codewars Is Nice") == "#CodewarsIsNice"
```

```
p generateHashtag("Codewars is nice") == "#CodewarsIsNice"
```

```
p generateHashtag("code" + " " * 140 + "wars") == "#CodeWars"
```

p

[illegible]

```
p generateHashtag("a" * 139) == "#A" + "a" * 138
```

```
p generateHashtag("a" * 140) == false
```

## 25. Pete, the baker

(<https://www.codewars.com/kata/525c65e51bf619685c000059/train/ruby>)

5 kyu

Pete likes to bake some cakes. He has some recipes and ingredients. Unfortunately he is not good in maths. Can you help him to find out, how many cakes he could bake considering his recipes?

Write a function `cakes()`, which takes the recipe (object) and the available ingredients (also an object) and returns the maximum number of cakes Pete can bake (integer). For simplicity there are no units for the amounts (e.g. 1 lb of flour or 200 g of sugar are simply 1 or 200). Ingredients that are not present in the objects, can be considered as 0.

Examples:

```
// must return 2
```

```
cakes({flour: 500, sugar: 200, eggs: 1}, {flour: 1200, sugar: 1200, eggs: 5, milk: 200});
```

```
// must return 0
```

```
cakes({apples: 3, flour: 300, sugar: 150, milk: 100, oil: 100}, {sugar: 500, flour: 2000, milk: 2000});
```

```
=end
```

```
p cakes({"flour"=>500, "sugar"=>200, "eggs"=>1}, {"flour"=>1200, "sugar"=>1200, "eggs"=>5, "milk"=>200}) == 2
```

```
p cakes({"cream"=>200, "flour"=>300, "sugar"=>150, "milk"=>100, "oil"=>100}, {"sugar"=>1700, "flour"=>20000, "milk"=>20000, "oil"=>30000, "cream"=>5000}) == 11
```

```
p cakes({"apples"=>3, "flour"=>300, "sugar"=>150, "milk"=>100, "oil"=>100}, {"sugar"=>500, "flour"=>2000, "milk"=>2000}) == 0
```

```
p cakes({"apples"=>3, "flour"=>300, "sugar"=>150, "milk"=>100, "oil"=>100}, {"sugar"=>500, "flour"=>2000, "milk"=>2000, "apples"=>15, "oil"=>20}) == 0
```

```
p cakes({"eggs"=>4, "flour"=>400}, {}) == 0
```

```
p cakes({"cream"=>1, "flour"=>3, "sugar"=>1, "milk"=>1, "oil"=>1, "eggs"=>1}, {"sugar"=>1, "eggs"=>1, "flour"=>3, "cream"=>1, "oil"=>1, "milk"=>1}) == 1
```

## 26. Mean Square Error

(<https://www.codewars.com/kata/51edd51599a189fe7f000015/train/ruby>)

Complete the function that

accepts two integer arrays of equal length

compares the value each member in one array to the corresponding member in the other

squares the absolute value difference between those two values

and returns the average of those squared absolute value difference between each member pair.

Examples

```
[1, 2, 3], [4, 5, 6] --> 9 because (9 + 9 + 9) / 3
```

```
[10, 20, 10, 2], [10, 25, 5, -2] --> 16.5 because (0 + 25 + 25 + 16) / 4
```

```
[-1, 0], [0, -1] --> 1 because (1 + 1) / 2
```

```
=end
```

```
p solution([1, 2, 3], [4, 5, 6]) == 9
```

```
p solution([10, 20, 10, 2], [10, 25, 5, -2]) == 16.5
```

```
p solution([-1, 0], [0, -1]) == 1
```

## 27. Exponent method

(<https://www.codewars.com/kata/5251f63bdc71af49250002d8>)

5 kyu

Create a method called "power" that takes two integers and returns the value of the first argument raised to the power of the second. Return nil if the second argument is negative.

Note: The \*\* operator has been disabled.

Examples:

```
p power(2, 3) == 8
p power(10, 0) == 1
p power(-5, 3) == -125
p power(-4, 2) == 16
p power(10, 0) == 1
p power(2, 3) == 8
p power(3, 2) == 9
p power(-5, 3) == -125
p power(-4, 2) == 16
p power(8, -2) == nil
```

## 28. Where my anagrams at?

(<https://www.codewars.com/kata/523a86aa4230ebb5420001e1>)

5 kyu

What is an anagram? Well, two words are anagrams of each other if they both contain the same letters. For example:

```
'abba' & 'baab' == true
```

```
'abba' & 'bbaa' == true
```

```
'abba' & 'abbba' == false
```

```
'abba' & 'abca' == false
```

Write a function that will find all the anagrams of a word from a list. You will be given two inputs a word and an array with words. You should return an array of all the anagrams or an empty array if there are none. For example:

```
=end
p anagrams('abba', ['aabb', 'abcd', 'bbaa', 'dada']) == ['aabb', 'bbaa']
p anagrams('racer', ['crazer', 'carer', 'racar', 'caers', 'racer']) == ['carer', 'racer']
p anagrams('laser', ['lazing', 'lazy', 'lacer']) == []
```

## 29. Split Strings

(<https://www.codewars.com/kata/515de9ae9dcfc28eb6000001>)

6 kyu

Complete the solution so that it splits the string into pairs of two characters. If the string contains an odd number of characters then it should replace the missing second character of the final pair with an underscore ('\_').

```
=end
```



```
p solution('abc') == ['ab', 'c_']
p solution('abcdef') == ['ab', 'cd', 'ef']
p solution("abcdef") == ["ab", "cd", "ef"]
p solution("abcdefg") == ["ab", "cd", "ef", "g_"]
p solution("") == []
```

## 30. Anagram difference

(<https://www.codewars.com/kata/5b1b27c8f60e99a467000041>)

6 kyu

Given two words, how many letters do you have to remove from them to make them anagrams?

Example

First word : c o d e w a r s (4 letters removed)

Second word : h a c k e r r a n k (6 letters removed)

Result : 10

Hints

A word is an anagram of another word if they have the same letters (usually in a different order).

Do not worry about case. All inputs will be lowercase.

=end

```
p anagram_difference("", "") == 0
p anagram_difference('a', "") == 1
p anagram_difference("", 'a') == 1
p anagram_difference('ab', 'a') == 1
p anagram_difference('ab', 'ba') == 0
p anagram_difference('ab', 'cd') == 4
p anagram_difference('aab', 'a') == 2
p anagram_difference('a', 'aab') == 2
p anagram_difference('codewars', 'hackerrank') == 10
```

## 31. Anagram Detection

(<https://www.codewars.com/kata/529eef7a9194e0cbc1000255>)

7 kyu

An anagram is the result of rearranging the letters of a word to produce a new word (see wikipedia).

Note: anagrams are case insensitive

Complete the function to return true if the two arguments given are anagrams of each other; return false otherwise.

Examples

"foefet" is an anagram of "toffee"

"Buckethead" is an anagram of "DeathCubeK"

=end

```
p is_anagram('Creative', 'Reactive') == true
p is_anagram("foefet", "toffee") == true
p is_anagram("Buckethead", "DeathCubeK") == true
p is_anagram("Twoo", "WooT") == true
p is_anagram("dumble", "bumble") == false
```

```
p is_anagram("ound", "round") == false
p is_anagram("apple", "pale") == false
```

## 32. Highest Scoring Word

(<https://www.codewars.com/kata/57eb8fcdf670e99d9b000272>)

6 kyu

Given a string of words, you need to find the highest scoring word.

Each letter of a word scores points according to its position in the alphabet: a = 1, b = 2, c = 3 etc.

You need to return the highest scoring word as a string.

If two words score the same, return the word that appears earliest in the original string.

All letters will be lowercase and all inputs will be valid.

=end

```
p high('man i need a taxi up to ubud') == 'taxi'
```

```
p high('what time are we climbing up the volcano') == 'volcano'
```

```
p high('take me to semynak') == 'semynak'
```

```
p high('aaa b') == 'aaa'
```

## 33. Replace With Alphabet Position

(<https://www.codewars.com/kata/546f922b54af40e1e90001da>)

6 kyu

In this kata you are required to, given a string, replace every letter with its position in the alphabet.

If anything in the text isn't a letter, ignore it and don't return it.

"a" = 1, "b" = 2, etc.

Example

```
alphabet_position("The sunset sets at twelve o' clock.")
```

Should return "20 8 5 19 21 14 19 5 20 19 5 20 19 1 20 20 23 5 12 22 5 15 3 12 15 3 11" (as a string)

=end

```
p alphabet_position("The sunset sets at twelve o' clock.") == "20 8 5 19 21 14 19 5 20 19 5 20 19 1 20 20 23 5 12 22 5 15 3 12 15 3 11"
```

```
p alphabet_position("-.-'") == ""
```

## 34. Sherlock on pockets

(<https://www.codewars.com/kata/53bb1201392478fefc000746>)

6 kyu

Sherlock has to find suspects on his latest case. He will use your method, dear Watson. Suspect in this case is a person which has something not allowed in his/her pockets.

Allowed items are defined by array of numbers.

Pockets contents are defined by map entries where key is a person and value is one or few things represented by an array of numbers (can be nil or empty array if empty), example:

```
pockets = {  
  bob: [1],  
  tom: [2, 5],  
  jane: [7]  
}
```

Write a method which helps Sherlock to find suspects. If no suspect is found or there are no pockets (pockets == nil), the method should return nil.

```
p find_suspects(pockets, [1, 2]) == [:tom, :jane]  
p find_suspects(pockets, [1, 7, 5, 2]) == nil  
p find_suspects(pockets, []) == [:bob, :tom, :jane]  
p find_suspects(pockets, [7]) == [:bob, :tom]  
=end
```

## 35. Mexican Wave

(<https://www.codewars.com/kata/58f5c63f1e26ecda7e000029/train/ruby>)

6 kyu

In this simple Kata your task is to create a function that turns a string into a Mexican Wave. You will be passed a string and you must return that string in an array where an uppercase letter is a person standing up.

Rules

1. The input string will always be lower case but maybe empty.
2. If the character in the string is whitespace then pass over it as if it was an empty seat.

=end

```
p wave("hello") == ["Hello", "hEllo", "heLlo", "helLo", "hellO"]  
p wave("codewars") == ["Codewars", "cOdewars", "coDewars", "codEwars", "codeWars", "codewArs", "codewaRs",  
  "codewarS"]  
p wave("") == []  
p wave("two words") == ["Two words", "tWo words", "twO words", "two Words", "two wOrds", "two woRds", "two worDs",  
  "two wordS"]  
p wave(" gap ") == [" Gap ", " gAp ", " gaP "]
```

## 36. Delete a Digit

(<https://www.codewars.com/kata/5894318275f2c75695000146/train/ruby>)

6 kyu

Task

Given an integer n, find the maximal number you can obtain by deleting exactly one digit of the given number.

Example

For n = 152, the output should be 52;

For n = 1001, the output should be 101.

Input/Output

[input] integer n

Constraints:  $10 \leq n \leq 1000000$ .

[output] an integer

=end

p delete\_digit(152) == 52

p delete\_digit(1001) == 101

p delete\_digit(10) == 1

## 37. Multiples of 3 or 5

(<https://www.codewars.com/kata/514b92a657cdc65150000006/train/ruby>)

6 kyu

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Finish the solution so that it returns the sum of all the multiples of 3 or 5 below the number passed in.

Note: If the number is a multiple of both 3 and 5, only count it once.

=end

p solution(10) == 23

p solution(20) == 78

p solution(200) == 9168

## 38. String transformer

(<https://www.codewars.com/kata/5878520d52628a092f0002d0/train/ruby>)

6 kyu

Given a string, return a new string that has transformed based on the input:

Change case of every character, ie. lower case to upper case, upper case to lower case.

Reverse the order of words from the input.

Note: You will have to handle multiple spaces, and leading/trailing spaces.

For example:

"Example Input" ==> "iNPUT eXAMPLE"

You may assume the input only contain English alphabet and spaces.

=end

## 39. Largest product in a series

(<https://www.codewars.com/kata/529872bdd0f550a06b00026e/train/ruby>)

6 kyu

Complete the greatestProduct method so that it'll find the greatest product of five consecutive digits in the given string of digits.

For example:

```

greatestProduct("123834539327238239583") // should return 3240
The input string always has more than five digits.
=end
p greatest_product("123834539327238239583") == 3240
p greatest_product("395831238345393272382") == 3240
p greatest_product("9249473782824422222111111532909999") == 5292
p greatest_product("9249473782824422222111111532909999") == 5292
p greatest_product("02494037820244202221011110532909999") == 0

```

## 40. Duplicate Encoder

(<https://www.codewars.com/kata/54b42f9314d9229fd6000d9c/train/ruby>)

6 kyu

The goal of this exercise is to convert a string to a new string where each character in the new string is "(" if that character appears only once in the original string, or ")" if that character appears more than once in the original string. Ignore capitalization when determining if a character is a duplicate.

Examples

```

"din"    => "((("
"recede" => "()()()"
"Success" => ")()()()"
"(( @"   => "))(("

```

```

=end
p duplicate_encode("din") == "((("
p duplicate_encode("recede") == "()()()"
p duplicate_encode("Success") == ")()()()"
p duplicate_encode("(( @") == "))(("

```

## 41. Backspaces in string

(<https://www.codewars.com/kata/5727bb0fe81185ae62000ae3/train/ruby>)

6 kyu

Assume "#" is like a backspace in string. This means that string "a#bc#d" actually is "bd"

Your task is to process a string with "#" symbols.

Examples

```

"abc#d##c"    ==> "ac"
"abc##d#####" ==> ""
"#####"      ==> ""
""             ==> ""

```

```

=end
p clean_string('abc#d##c') == "ac"
p clean_string('abc#####d##c#') == ""

```

```

=begin

```

## 42. Sort Arrays (Ignoring Case)

(<https://www.codewars.com/kata/51f41fe7e8f176e70d0002b9/train/ruby>)

6 kyu

Sort the given strings in alphabetical order, case insensitive. For example:

```
["Hello", "there", "I'm", "fine"] --> ["fine", "Hello", "I'm", "there"]
["C", "d", "a", "B"]           --> ["a", "B", "C", "d"]
=end
p sortme(["Hello", "there", "I'm", "fine"]) == ["fine", "Hello", "I'm", "there"]
p sortme(["C", "d", "a", "Ba", "be"]) == ["a", "Ba", "be", "C", "d"]
p sortme(["CodeWars"]) == ["CodeWars"]
```

## 43. Transform To Prime

(<https://www.codewars.com/kata/5a946d9fba1bb5135100007c/train/ruby>)

6 kyu

Task :

Given a List [] of n integers , find the minimum number to be inserted in a list, so that the sum of all elements of the list should equal the closest prime number .

Notes

List size is at least 2 .

List's numbers will only have positives (n > 0) .

Repetition of numbers in the list could occur .

The newer list's sum should equal the closest prime number .

Input >> Output Examples

1- minimumNumber ({3,1,2}) ==> return (1)

Explanation:

Since , the sum of the list's elements equal to (6) , the minimum number to be inserted to transform the sum to prime number is (1) , which will make \*the sum of the List\*\* equal the closest prime number (7)\* .

2- minimumNumber ({2,12,8,4,6}) ==> return (5)

Explanation:

Since , the sum of the list's elements equal to (32) , the minimum number to be inserted to transform the sum to prime number is (5) , which will make \*the sum of the List\*\* equal the closest prime number (37)\* .

3- minimumNumber ({50,39,49,6,17,28}) ==> return (2)

Explanation:

Since , the sum of the list's elements equal to (189) , the minimum number to be inserted to transform the sum to prime number is (2) , which will make \*the sum of the List\*\* equal the closest prime number (191)\* .

=end

p minimum\_number([3,1,2]) == 1

p minimum\_number([5,2]) == 0

p minimum\_number([1,1,1]) == 0

p minimum\_number([2,12,8,4,6]) == 5

p minimum\_number([50,39,49,6,17,28]) == 2

## 44. Counting Duplicates

(<https://www.codewars.com/kata/54bf1c2cd5b56cc47f0007a1/train/ruby>)

6 kyu

Count the number of Duplicates

Write a function that will return the count of distinct case-insensitive alphabetic characters and numeric digits that occur more than once in the input string. The input string can be assumed to contain only alphabets (both uppercase and lowercase) and numeric digits.

Example

"abcde" -> 0 # no characters repeats more than once

"aabbcd" -> 2 # 'a' and 'b'

"aabBcd" -> 2 # 'a' occurs twice and 'b' twice ('b' and 'B')

"indivisiBility" -> 1 # 'i' occurs six times

"IndivisiBilities" -> 2 # 'i' occurs seven times and 's' occurs twice

"aA11" -> 2 # 'a' and '1'

"ABBA" -> 2 # 'A' and 'B' each occur twice

=end

p duplicate\_count("") == 0

p duplicate\_count("abcde") == 0

p duplicate\_count("abcdeaa") == 1

p duplicate\_count("abcdeaB") == 2

p duplicate\_count("IndivisiBilities") == 2

## 45. Alphabetized

(<https://www.codewars.com/kata/5970df092ef474680a0000c9/train/ruby>)

6 kyu

The alphabetized kata

Re-order the characters of a string, so that they are concatenated into a new string in

"case-insensitively-alphabetical-order-of-appearance" order. Whitespace and punctuation shall simply be removed!

The input is restricted to contain no numerals and only words containing the english alphabet letters.

Example:

alphabetized("The Holy Bible") # "BbeehHilloTy"

=end

p alphabetized("") == ""

p alphabetized(" ") == ""

p alphabetized(" a") == "a"

p alphabetized("a ") == "a"

p alphabetized(" a ") == "a"

p alphabetized("A b B a") == "AabB"

p alphabetized(" a b c d e f g h i j k l m n o p q r s t u v w x y z A B C D E F G H I J K L M N O P Q R S T U V W X Y Z")  
== "aAbBcCdDeEfGhHiljJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ"

## 46. Sum of Digits / Digital Root

(<https://www.codewars.com/kata/541c8630095125aba6000c00/train/ruby>)

6 kyu

In this kata, you must create a digital root function.

A digital root is the recursive sum of all the digits in a number. Given n, take the sum of the digits of n. If that value has more than one digit, continue reducing in this way until a single-digit number is produced. This is only applicable to the natural numbers.

Here's how it works:

```
digital_root(16)
```

```
=> 1 + 6
```

```
=> 7
```

```
digital_root(942)
```

```
=> 9 + 4 + 2
```

```
=> 15 ...
```

```
=> 1 + 5
```

```
=> 6
```

```
digital_root(132189)
```

```
=> 1 + 3 + 2 + 1 + 8 + 9
```

```
=> 24 ...
```

```
=> 2 + 4
```

```
=> 6
```

```
digital_root(493193)
```

```
=> 4 + 9 + 3 + 1 + 9 + 3
```

```
=> 29 ...
```

```
=> 2 + 9
```

```
=> 11 ...
```

```
=> 1 + 1
```

```
=> 2
```

```
=end
```

```
Test.assert_equals( digital_root(16), 7 )
```

```
Test.assert_equals( digital_root(456), 6 )
```

## 47. Array.diff

(<https://www.codewars.com/kata/523f5d21c841566fde000009/train/ruby>)

6 kyu

Your goal in this kata is to implement a difference function, which subtracts one list from another and returns the result.

It should remove all values from list a, which are present in list b.

```
array_diff([1,2],[1]) == [2]
```

If a value is present in b, all of its occurrences must be removed from the other:



```

array_diff([1,2],[1]) == [2]
=end
p array_diff([1,2], [1]) == [2]
p array_diff([1,2,2], [1]) == [2,2]
p array_diff([1,2,2], [2]) == [1]
p array_diff([1,2,2], []) == [1,2,2]
p array_diff([], [1,2]) == []

```

## 48. Where is my parent!?(cry)

(<https://www.codewars.com/kata/58539230879867a8cd00011c/train/ruby> )

Where's my parent?

6 kyu

Mothers arranged a dance party for the children in school. At that party, there are only mothers and their children. All are having great fun on the dance floor when suddenly all the lights went out. It's a dark night and no one can see each other. But you were flying nearby and you can see in the dark and have ability to teleport people anywhere you want.

Legend:

-Uppercase letters stands for mothers, lowercase stand for their children, i.e. "A" mother's children are "aaaa".

-Function input: String contains only letters, uppercase letters are unique.

Task:

Place all people in alphabetical order where Mothers are followed by their children, i.e. "aAbaBb" => "AaaBbb".

=end

```

p find_children("abBA") == "AaBb"
p find_children("AaaaaZazzz") == "AaaaaaZzzz"
p find_children("CbcBcbaA") == "AaBbbCcc"
p find_children("xXfuUuuF") == "FfUuuuXx"
p find_children("") == ""

```

## 49. Playing with digits

(<https://www.codewars.com/kata/5552101f47fc5178b1000050/train/ruby>)

6 kyu

Some numbers have funny properties. For example:

$$89 \rightarrow 8^1 + 9^2 = 89 * 1$$

$$695 \rightarrow 6^2 + 9^3 + 5^4 = 1390 = 695 * 2$$

$$46288 \rightarrow 4^3 + 6^4 + 2^5 + 8^6 + 8^7 = 2360688 = 46288 * 51$$

Given a positive integer n written as abcd... (a, b, c, d... being digits) and a positive integer p

we want to find a positive integer k, if it exists, such as the sum of the digits of n taken to the successive powers of p is equal to k \* n.

In other words:

$$\text{Is there an integer k such as : } (a^p + b^{(p+1)} + c^{(p+2)} + d^{(p+3)} + \dots) = n * k$$

If it is the case we will return k, if not return -1.

Note: n and p will always be given as strictly positive integers.

`dig_pow(89, 1)` should return 1 since  $8^1 + 9^2 = 89 = 89 * 1$

`dig_pow(92, 1)` should return -1 since there is no k such as  $9^1 + 2^2$  equals  $92 * k$

`dig_pow(695, 2)` should return 2 since  $6^2 + 9^3 + 5^4 = 1390 = 695 * 2$

`dig_pow(46288, 3)` should return 51 since  $4^3 + 6^4 + 2^5 + 8^6 + 8^7 = 2360688 = 46288 * 51$

`=end`

`p dig_pow(89, 1) == 1`

`p dig_pow(92, 1) == -1`

`p dig_pow(46288, 3) == 51`

`p dig_pow(695, 2) == 2`

## 50. Equal Sides Of An Array

(<https://www.codewars.com/kata/5679aa472b8f57fb8c000047/train/ruby>)

Equal sides of an array

6 kyu

You are going to be given an array of integers. Your job is to take that array and find an index N where the sum of the integers to the left of N is equal to the sum of the integers to the right of N. If there is no index that would make this happen, return -1.

For example:

Let's say you are given the array `{1,2,3,4,3,2,1}`: Your function will return the index 3, because at the 3rd position of the array, the sum of left side of the index (`{1,2,3}`) and the sum of the right side of the index (`{3,2,1}`) both equal 6.

Let's look at another one.

You are given the array `{1,100,50,-51,1,1}`: Your function will return the index 1, because at the 1st position of the array, the sum of left side of the index (`{1}`) and the sum of the right side of the index (`{50,-51,1,1}`) both equal 1.

Last one:

You are given the array `{20,10,-80,10,10,15,35}`

At index 0 the left side is `{}`

The right side is `{10,-80,10,10,15,35}`

They both are equal to 0 when added. (Empty arrays are equal to 0 in this problem)

Index 0 is the place where the left side and right side are equal.

Note: Please remember that in most programming/scripting languages the index of an array starts at 0.

Input:

An integer array of length  $0 < \text{arr} < 1000$ . The numbers in the array can be any integer positive or negative.

Output:

The lowest index N where the side to the left of N is equal to the side to the right of N. If you do not find an index that fits these rules, then you will return -1.

Note:

If you are given an array with multiple answers, return the lowest correct index.

```

=end
p find_even_index([1,2,3,4,3,2,1]) == 3
p find_even_index([1,100,50,-51,1,1]) == 1
p find_even_index([1,2,3,4,5,6]) == -1
p find_even_index([20,10,30,10,10,15,35]) == 3
p find_even_index([20,10,-80,10,10,15,35]) == 0
p find_even_index([10,-80,10,10,15,35,20]) == 6
p find_even_index(Array(1..100)) == -1
p find_even_index([0,0,0,0,0]) == 0
p find_even_index([-1,-2,-3,-4,-3,-2,-1]) == 3
p find_even_index(Array(-100..-1)) == -1

```

## 51. Reverse or rotate?

(<https://www.codewars.com/kata/56b5afb4ed1f6d5fb0000991>)

6 kyu

Reverse or rotate?

The input is a string str of digits. Cut the string into chunks (a chunk here is a substring of the initial string) of size sz (ignore the last chunk if its size is less than sz).

If a chunk represents an integer such as the sum of the cubes of its digits is divisible by 2, reverse that chunk; otherwise rotate it to the left by one position. Put together these modified chunks and return the result as a string.

If

sz is <= 0 or if str is empty return ""

sz is greater (>) than the length of str it is impossible to take a chunk of size sz hence return "".

Examples:

```

revrot("123456987654", 6) --> "234561876549"
revrot("123456987653", 6) --> "234561356789"
revrot("66443875", 4) --> "44668753"
revrot("66443875", 8) --> "64438756"
revrot("664438769", 8) --> "67834466"
revrot("123456779", 8) --> "23456771"
revrot("", 8) --> ""
revrot("123456779", 0) --> ""
revrot("563000655734469485", 4) --> "0365065073456944"
=end
p revrot("1234", 0) == ""
p revrot("", 0) == ""
p revrot("1234", 5) == ""
p revrot("733049910872815764", 5) == "330479108928157"
p revrot("123456987654", 6) == "234561876549"
p revrot("123456987653", 6) == "234561356789"
p revrot("66443875", 4) == "44668753"
p revrot("66443875", 8) == "64438756"
p revrot("664438769", 8) == "67834466"
p revrot("123456779", 8) == "23456771"
p revrot("", 8) == ""
p revrot("123456779", 0) == ""

```

```
p revrot("563000655734469485", 4) == "0365065073456944"
```

## 51. Decipher this!

<https://www.codewars.com/kata/581e014b55f2c52bb00000f8>

Decipher this!

6 kyu

You are given a secret message you need to decipher. Here are the things you need to know to decipher it:

For each word:

the second and the last letter is switched (e.g. Hello becomes Holle)

the first letter is replaced by its character code (e.g. H becomes 72)

Note: there are no special characters used, only letters and spaces

Examples

```
decipherThis('72olle 103doo 100ya'); // 'Hello good day'
```

```
decipherThis('82yade 115te 103o'); // 'Ready set go'
```

```
=end
```

```
p decipher_this("65 119esi 111dl 111lw 108dvei 105n 97n 111ka") == "A wise old owl lived in an oak"
```

```
p decipher_this("84eh 109ero 104e 115wa 116eh 108sse 104e 115eokp") == "The more he saw the less he spoke"
```

```
p decipher_this("84eh 108sse 104e 115eokp 116eh 109ero 104e 104dare") == "The less he spoke the more he heard"
```

```
p decipher_this("87yh 99na 119e 110to 97ll 98e 108eki 116tah 119esi 111dl 98dri") == "Why can we not all be like that  
wise old bird"
```

```
p decipher_this("84kanh 121uo 80roti 102ro 97ll 121ruo 104ple") == "Thank you Piotr for all your help"
```

## 52. Bouncing Balls

<https://www.codewars.com/kata/5544c7a5cb454edb3c000047>

6 kyu

A child is playing with a ball on the  $n$ th floor of a tall building. The height of this floor,  $h$ , is known.

He drops the ball out of the window. The ball bounces (for example), to two-thirds of its height (a bounce of 0.66).

His mother looks out of a window 1.5 meters from the ground.

How many times will the mother see the ball pass in front of her window (including when it's falling and bouncing?)

Three conditions must be met for a valid experiment:

Float parameter " $h$ " in meters must be greater than 0

Float parameter "bounce" must be greater than 0 and less than 1

Float parameter "window" must be less than  $h$ .

If all three conditions above are fulfilled, return a positive integer, otherwise return -1.

Note:

The ball can only be seen if the height of the rebounding ball is strictly greater than the window parameter.

Example:

-  $h = 3$ , bounce = 0.66, window = 1.5, result is 3

- h = 3, bounce = 1, window = 1.5, result is -1

(Condition 2) not fulfilled).

=end

p bouncingBall(3, 0.66, 1.5) == 3

p bouncingBall(30, 0.66, 1.5) == 15

p bouncingBall(30, 0.75, 1.5) == 21

p bouncingBall(30, 0.4, 10) == 3

p bouncingBall(40, 1, 10) == -1

p bouncingBall(-5, 0.66, 1.5) == -1

p bouncingBall(1, 0.66, 0.66) == 1

p bouncingBall(1, 0.66, 1) == -1

## 53. WeIrD StRiNg CaSe

(<https://www.codewars.com/kata/52b757663a95b11b3d00062d>)

6 kyu

Write a function toWeirdCase (weirdcase in Ruby) that accepts a string, and returns the same string with all even indexed characters in each word upper cased, and all odd indexed characters in each word lower cased. The indexing just explained is zero based, so the zero-ith index is even, therefore that character should be upper cased.

The passed in string will only consist of alphabetical characters and spaces(' '). Spaces will only be present if there are multiple words. Words will be separated by a single space(' ').

=end

p weirdcase( "String" ) == "StRiNg"

p weirdcase( "Weird string case" ) == "WeIrD StRiNg CaSe"

## 54. Are they the "same"?

(<https://www.codewars.com/kata/550498447451fbbd7600041c>)

6 kyu

Given two arrays a and b write a function comp(a, b) that checks whether the two arrays have the "same" elements, with the same multiplicities. "Same" means, here, that the elements in b are the elements in a squared, regardless of the order.

Examples

Valid arrays

a = [121, 144, 19, 161, 19, 144, 19, 11]

b = [121, 14641, 20736, 361, 25921, 361, 20736, 361]

comp(a, b) returns true because in b 121 is the square of 11, 14641 is the square of 121, 20736 the square of 144, 361 the square of 19, 25921 the square of 161, and so on. It gets obvious if we write b's elements in terms of squares:

a = [121, 144, 19, 161, 19, 144, 19, 11]

b = [11\*11, 121\*121, 144\*144, 19\*19, 161\*161, 19\*19, 144\*144, 19\*19]

Invalid arrays

If we change the first number to something else, comp may not return true anymore:

a = [121, 144, 19, 161, 19, 144, 19, 11]

b = [132, 14641, 20736, 361, 25921, 361, 20736, 361]

comp(a,b) returns false because in b 132 is not the square of any number of a.

```
a = [121, 144, 19, 161, 19, 144, 19, 11]
b = [121, 14641, 20736, 36100, 25921, 361, 20736, 361]
comp(a,b) returns false because in b 36100 is not the square of any number of a.
```

#### Remarks

a or b might be [] (all languages except R, Shell).

a or b might be nil or null or None or nothing (except in Haskell, Elixir, C++, Rust, R, Shell, PureScript).

If a or b are nil (or null or None), the problem doesn't make sense so return false.

#### Note for C

The two arrays have the same size (> 0) given as parameter in function comp.

=end

```
p comp([121, 144, 19, 161, 19, 144, 19, 11], [121, 14641, 20736, 361, 25921, 361, 20736, 361]) == true
```

```
p comp([121, 144, 19, 161, 19, 144, 19, 11], [132, 14641, 20736, 361, 25921, 361, 20736, 361]) == false
```

```
p comp(nil, [1, 2, 3]) == false
```

```
p comp([1, 2], []) == false
```

```
p comp([1, 2], [1, 4, 4]) == false
```

## 55. Grouping and Counting

(<https://www.codewars.com/kata/53a452dd0064085711001205>)

6 kyu

Your goal is to write the group\_and\_count method, which should receive an array as a unique parameter and return a hash. Empty or nil input must return nil instead of a hash. This hash returned must contain as keys the unique values of the array, and as values the counting of each value.

Example usage:

```
input = [1,1,2,2,2,3]
```

```
group_and_count(input)# == {1=>2, 2=>3, 3=>1}
```

The following methods were disabled:

```
Array#count
```

```
Array#length
```

=end

```
p group_and_count([1,1,2,2,2,3]) == {1=>2, 2=>3, 3=>1}
```

```
p group_and_count([]) == nil
```

```
p group_and_count(nil) == nil
```

```
p group_and_count([1, 7, 5, -1]) == {1=>1, 7=>1, 5=>1, -1=>1}
```

## 56. Find the Nexus of the Codewars Universe

(<https://www.codewars.com/kata/5453dce502949307cf000bff>)

6 kyu

Not to brag, but I recently became the nexus of the Codewars universe! My honor and my rank were the same number. I cried a little.

Complete the method that takes a hash/object/directory/association list of users, and find the nexus: the user whose rank is the closest is equal to his honor. Return the rank of this user. For each user, the key is the rank and the value is the honor.

If nobody has an exact rank/honor match, return the rank of the user who comes closest. If there are several users who come closest, return the one with the lowest rank (numeric value). The hash will not necessarily contain consecutive rank numbers; return the best match from the ranks provided.

Example

```
rank  honor
users = { 1 => 93,
          10 => 55,
          15 => 30,
          20 => 19, <--- nexus
          23 => 11,
          30 => 2 }
=end
```

```
p nexus({1 => 3, 3 => 3, 5 => 1}) == 3
p nexus({1 => 10, 2 => 6, 3 => 4, 5 => 1}) == 3
p nexus({1 => 10, 2 => 3, 3 => 4, 5 => 1}) == 2
```

## 57. Count letters in string

(<https://www.codewars.com/kata/5808ff71c7cfa1c6aa00006d>)

6 kyu

In this kata, you've to count lowercase letters in a given string and return the letter count in a hash with 'letter' as key and count as 'value'. The key must be 'symbol' instead of string in Ruby and 'char' instead of string in Crystal.

```
=end
p letter_count('arithmetics') == { :a=>1, :c=>1, :e=>1, :h=>1, :i=>2, :m=>1, :r=>1, :s=>1, :t=>2 }
```

## 58. Triple trouble

(<https://www.codewars.com/kata/55d5434f269c0c3f1b000058>)

6 kyu

Write a function

```
triple_double(num1, num2)
```

which takes numbers num1 and num2 and returns 1 if there is a straight triple of a number at any place in num1 and also a straight double of the same number in num2.

If this isn't the case, return 0

Examples

```
triple_double(451999277, 41177722899) == 1
# num1 has straight triple 999s and num2 has straight double 99s
```

```
triple_double(1222345, 12345) == 0
# num1 has straight triple 2s but num2 has only a single 2
=end
```

```
p triple_double(12345, 12345) == 0
p triple_double(666789, 12345667) == 1
```

## 59. Which are in?

(<https://www.codewars.com/kata/550554fd08b86f84fe000a58>)

6 kyu

Given two arrays of strings a1 and a2 return a sorted array r in lexicographical order of the strings of a1 which are substrings of strings of a2.

```
#Example 1: a1 = ["arp", "live", "strong"]
```

```
a2 = ["lively", "alive", "harp", "sharp", "armstrong"]
```

```
returns ["arp", "live", "strong"]
```

```
#Example 2: a1 = ["tarp", "mice", "bull"]
```

```
a2 = ["lively", "alive", "harp", "sharp", "armstrong"]
```

```
returns []
```

Notes:

Arrays are written in "general" notation. See "Your Test Cases" for examples in your language.

In Shell bash a1 and a2 are strings. The return is a string where words are separated by commas.

Beware: r must be without duplicates.

Don't mutate the inputs.

=end

## 60. Format a string of names like 'Bart, Lisa & Maggie'.

(<https://www.codewars.com/kata/53368a47e38700bd8300030d2>)

6 kyu

Given: an array containing hashes of names

Return: a string formatted as a list of names separated by commas except for the last two names, which should be separated by an ampersand.

Example:

```
list([ {name: 'Bart'}, {name: 'Lisa'}, {name: 'Maggie'} ])
```

```
# returns 'Bart, Lisa & Maggie'
```

```
list([ {name: 'Bart'}, {name: 'Lisa'} ])
```

```
# returns 'Bart & Lisa'
```

```
list([ {name: 'Bart'} ])
```

```
# returns 'Bart'
```



```
list([])
# returns "
Note: all the hashes are pre-validated and will only contain A-Z, a-z, '-' and '.'.
=end
```

## 61. Find the missing letter

(<https://www.codewars.com/kata/5839edaa6754d6fec10000a2>)

6 kyu  
#Find the missing letter

Write a method that takes an array of consecutive (increasing) letters as input and that returns the missing letter in the array.

You will always get an valid array. And it will be always exactly one letter be missing. The length of the array will always be at least 2.

The array will always contain letters in only one case.

Example:

['a','b','c','d','f'] -> 'e' ['O','Q','R','S'] -> 'P'

["a","b","c","d","f"] -> "e"

["O","Q","R","S"] -> "P"

(Use the English alphabet with 26 letters!)

Have fun coding it and please don't forget to vote and rank this kata! :-)

I have also created other katas. Take a look if you enjoyed this kata!

=end

## 62. Who likes it?

(<https://www.codewars.com/kata/5266876b8f4bf2da9b000362>)

6 kyu  
You probably know the "like" system from Facebook and other pages. People can "like" blog posts, pictures or other items. We want to create the text that should be displayed next to such an item.

Implement a function likes :: [String] -> String, which must take in input array, containing the names of people who like an item. It must return the display text as shown in the examples:

likes [] // must be "no one likes this"

likes ["Peter"] // must be "Peter likes this"

likes ["Jacob", "Alex"] // must be "Jacob and Alex like this"

likes ["Max", "John", "Mark"] // must be "Max, John and Mark like this"

likes ["Alex", "Jacob", "Mark", "Max"] // must be "Alex, Jacob and 2 others like this"

=end

## 63. Find The Parity Outlier

(<https://www.codewars.com/kata/5526fc09a1bbd946250002dc>)

6 kyu

You are given an array (which will have a length of at least 3, but could be very large) containing integers. The array is either entirely comprised of odd integers or entirely comprised of even integers except for a single integer N. Write a method that takes the array as an argument and returns this "outlier" N.

Examples

[2, 4, 0, 100, 4, 11, 2602, 36]

Should return: 11 (the only odd number)

[160, 3, 1719, 19, 11, 13, -21]

Should return: 160 (the only even number)

=end

## 64. Is Integer Array?

(<https://www.codewars.com/kata/52a112d9488f506ae7000b95>)

6 kyu

Write a function with the signature shown below:

```
def is_int_array(arr)
```

```
  true
```

```
end
```

returns true / True if every element in an array is an integer or a float with no decimals.

returns true / True if array is empty.

returns false / False for every other input.

```
=end
```

## 65. Reversing and Combining Text

(<https://www.codewars.com/kata/56b861671d36bb0aa8000819/train/ruby>)

Your task is to Reverse and Combine Words.

Input: String containing different "words" separated by spaces

1. More than one word? Reverse each word and combine first with second, third with fourth and so on...

(odd number of words => last one stays alone, but has to be reversed too)

2. Start it again until there's only one word without spaces

3. Return your result...

```
=end
```

```
p reverse_and_combine_text("abc def") == "cbafed"
```

```
p reverse_and_combine_text("abc def ghi jkl") == "defabcjklghi"
```

```
p reverse_and_combine_text("dfghrtcbafed") == "dfghrtcbafed"
```

```
p reverse_and_combine_text("234hh54 53455 sdfqwzrt rteetrt hjhjh llll12 44") ==
```

```
"trzwqfdstrteettr45hh4325543544hjhjh21llll"
```

p reverse\_and\_combine\_text("sdfsdf wee sdffg 342234 ftt") == "gffds432243fdsfdseewttf"

## 66. Integer reduction

(<https://www.codewars.com/kata/59fd6d2332b8b9955200005f/ruby>)