|  |  |  |
| --- | --- | --- |
| Задачки с Codewars: | Моё решение: | Топовое решение: |
| Round up to the next multiple of 5  Input may be any positive or negative  integer(including 0).  You can assume that all inputs are  valid  integers. | def round\_to\_next5(n):      while n%5 != 0:          n += 1      return n | def round\_to\_next5(n):      return n + (5 - n) % 5 |
| Write a function getDrinkByProfession/get\_drink\_by\_profession() that receives as input parametera string, and produces outputs according to the following table:  Input    Output  "Jabroni" -> "Patron Tequila"  "School Counselor" -> "Anything with Alcohol"  "Programmer" -> "Hipster Craft Beer"  "Bike Gang Member" -> "Moonshine"  "Politician" -> "Your tax dollars"  "Rapper" -> "Cristal"  \*anything else\* -> "Beer"  Note: anything else is the default case: if the input to the function is not any of the values  in the table, then the return value should  be "Beer."  Make sure you cover the cases where certain  words do not show up with correct capitalization  For example, getDrinkByProfession("pOLitiCIaN") should still return "Your tax dollars". | def get\_drink\_by\_profession(param):      p = param.lower()      p2 = p.title()      if p2 == "Jabroni":          return ("Patron Tequila")      elif p2 == "School Counselor":          return ("Anything with Alcohol")      elif p2 == "Programmer":          return ("Hipster Craft Beer")      elif p2 == "Bike Gang Member":          return ("Moonshine")      elif p2 == "Politician":          return ("Your tax dollars")      elif p2 == "Rapper":          return ("Cristal")      else:          return ("Beer") | d = {      "jabroni": "Patron Tequila",      "school counselor": "Anything with Alcohol",      "programmer": "Hipster Craft Beer",      "bike gang member": "Moonshine",      "politician": "Your tax dollars",      "rapper": "Cristal"  }  def get\_drink\_by\_profession(s):      return d.get(s.lower(), "Beer") |
| Return the number (count) of vowels in the given string.  We will consider a, e, i, o, u as vowels for this Kata (but not y).  The input string will only consist of lower case letters and/or spaces. | def get\_count(input\_str):      num\_vowels = 0      for i in input\_str:          if i == "a" or i == "e" or\              i == "i" or i == "o" or\              i == "u":                  num\_vowels += 1        return num\_vowels | def getCount(inputStr):      return sum(1 for let in inputStr if let in  "aeiouAEIOU") |
| Description:  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. | def digital\_root(n):      while n > 9:          n = n%10 + digital\_root(n//10)      return n | def digital\_root(n):      return n if n < 10 else digital\_root  (sum(map(int,str(n)))) |
|  |  |  |
|  |  |  |
|  |  |  |