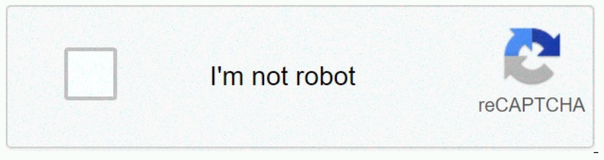

Continue

Codingbat python answers list 2

Item-2 opportunity memedium python problem list--1 loop of use a[0], a[1], ... to access the elements in the list len(a) is count_evens H length big_diff centered_average sum13 sum67 There is a Python HelpPython example, codepython stringspyon sample, Python list. If the BooleanCode Codingbat Python label, the question and answer section 2, this document is prepared and can only be used for educational purposes. All questions were taken from there are good questions about pythons and Javanese. Please go to the original website and answer the questions there. Codingbat also offers a reporting tool where educators can view students' grades. The answer was made by me in my spare time, Codingbat said, all the answers are true. I used some questions in the Internship application review. You can use the question usage details. You can use my answers or this document under the GNU General Public License Terms, as published by the Free Software Foundation, version 2 licenses, or (according to your options). Newer versions The document is made with Google Docs and the grammar highlighter, GeSHi, a great open source tool. If you have any questions, please email me through samet2@gmail.com enjoy python! This document contains 27 questions in these sections: Warm-up-2 Logic-2 String-2 List-2Warmup-2 1. string_timesGiven strings and non-string_times string_times string_times negative int. My solution: def string_times(str, n): Result =''i=0 while i4.string_splosionGiven a non-empty string, such as code, returns a string such as CCoCodCode string_splosion('code') 'CCoCodCode'string_splosion('abc') 'aababc'string_splosion('ab'aab'). My solution:def string_splosion(str): Result =" for i in range (len(str)+1): The result += str[i] returns the result 5. 1 (we do not count the end substring) Final2('hixxhi') 1last2('xaxxaxaxx') 1last2('axxxaaxx') 2 My solution: def last2(str): If lane (str) < 2: Return 0 Final2 = str[len(str)-2:] count = 0 for i in range (len(str)-2): sub = str[i:i+2] if sub ==last2:count = count + 1 return count 6. array_count9Given array of ints, returns a number of 9 in an array array_count9([1, 2, 9]) 1array_count9([1, 9,9]) 2array_count9([1), 3 My solution:def array_count9(nums): Result = 0 for i in nums: If i = 9: Result += 1, result 7 array_front9Given, array of ints returns True array_front9. 9, 3, 4]) Truearray_front9([1, 2, 3, 4, 9]) Falsearray_front9([1, 2, 3, 4, 5]) Result = false for i in range(4): If len(nums) > i: If nums[i] == 9: Result = actual return result 8 array123Given array of ints returns True if .1, Truearray123([1, 1, 2, 4, 1]) Falsearray123([1, 1, 2, 4, 1]) Falsearray123([1, 1, 1, 2, 2, 3]) string_matchGiven 2 strings a and b return the number of positions of the same length. 2 substrings, so xxcaazz and xxbaaz yield 3 because sub strings xx, aa and az appear in the same place in both strings string_match ('xxcaazz', 'xxaaz') 3string_match('abc', 'abc') 2string_match('abc', 'axc') 0 my solution: def string_match(a, r =a=a). If len(a) 11.lone_sumGiven 3 int values, b c, returns their total. However, if you do If one value is the same as the other value, lone_sum(1, 2, 3) 6lone_sum(3, 2, 3) 2lone_sum(3, 3, 3) 0, my resolution:def lone_sum(a, b, c): If == b ==c: returns 0 elif a ==b: return c elif a ==c: b elif b ==c: Returns another: returns + b + c 12.lucky_sumGiven 3 int, b c, returns their total. However, if you do If one of the values is 13, the value is not counted, and the right value is not counted. For example, if b is 13, then both b and c do not count lucky_sum(1, 2, 3) 6lucky_sum(1, 2, 13) 3lucky_sum(1, 13, 3) 1 My solution: def lucky_sum(a, b, c): If == 13: Returns 0 elif b == 13: returns elif c == 13: returns another + b: returns + b + c 13.no_teen_sumGiven 3 int, b c, returns their total. However, if any value is a teenager - in the range of 13..19 in total - then that value counts as 0 except 15 and 16, not count as no_teen_sum fix_teem a no_teen_sum teenager. 2, 3) 6no_teen_sum(2, 13, 1) 3no_teen_sum(2, 1, 14) 3 My Solution:def no_teen_sum(a, b, c): Returns fix_teem(a) + fix_teem(b) + fix_teem(c) def fix_teem(n): if n>=13 and nat the maximum 1), while the other value is far different from both other values by 2 or more. Note: ABS(num) calculates the absolute value of a number close_far(1, 2, 10) Trueclose_far(1), 2, 3) Falseclose_far(4, 1, 3) True My solution:def close_far(a, b, c): if abs(b-a) =2): Return True Elif (off ==c) and (abs(b-a) > = 2) and (abs(b-c)>2): Returns True else: returns false string-2 16.double_charGiven There are two characters double_char('The') 'TThhee'double_char count_hi 17.count_hiReturn 17.count_hiReturn('AAbb'), 'AAAAbb'double_char('Hi-There') 'HHii-TThheree'double_char. 1count_hi('abchi hi') 2count_hi('hihi') 2 My solution:def count_hi(str): a = str.split('hi') back lane(a) - 1 18.cat_dogReturn True If the string Cat_dog ('catdog') Truecat_dog ('catcat') Falsecat_dog ('1cat1cadodog'cat_dog). If len(a) == len(b): Returns True else: Returns False 19.count_codeReturn the number of times a code string appears anywhere in a given string, unless we accept any letter for 'd', so cope and cooe count count_code ('aaacodebbb') 1count_code ('codexxcode') 2count_code ('coxxxx.') my solution:def count_code(str): Result = 0 for i in range (len(str)-3): If str[i:i+2] == 'co' and str[i+3] == 'e': Results += 1 returns result20.end_otherGiven two strings Returns True if one string appears at the end of another string. Ignoring upper/lower case differences (in other words, calculations should not be case-sensitive). Note: s.lower() returns the lowercase version of the end_other string ('Hiabc', 'abc') Trueend_other ('AbC', 'HiaBc') Trueend_other ('abc', 'abXabc') True My solution:def end_other(a, b): a=a.lower() b=b.lower(), return (b.endswith(a) or a.endswith(b)) 21.xyz_thereReturn True. yz at xyz is not directly preceded by intervals (.b)) so xxyz counts, but x.xyz does not xyz_there ('abcxyz') Truexyz_there ('abc.xyz') Falsexyz_there ('xyz.abc') true my solution: def xyz_there(str): str = str.be('..xyz', '') If 'xyz' in str: True return else: Returns false-2 entries 22.count_evensReturn The number even ints in a given array. Note: The %mod operator calculates the rest, e.g. 5 % 2 is 1 count_evens([2, 1, 2, 3, 4]) 3count_evens([2, 2, 0]) 3count_evens([1, 3, 5]) My solution:def count_evens(nums): Result = 0 for i in nums: If i%2 = 0: Result += returns 1 23.big_diffGiven array length 1 or more, returns the largest difference between the value and the smallest value in the array. Note: The built-in min(v1, v2) and max(v1, v2) functions return smaller or larger values big_diff([10, 3, 5, 6]) 7big_diff([7, 2, 10, 9]) 8big_diff([2, 10, 7, 2]) 8 My solution: def big_diff(nums): maxx = nums[0] minn = nums[0] for i(nums):Lens:)): If nums[i] > maxx: maxx = nums[i] If nums[i] < minutes: minutes = nums[i] returns maxx - minn 24.centered_averageReturn average, centered of array of ints, which we say is the average of the excluded values, not the largest and smallest values in the array. Use the int section to create the centered_average final average. 2, 3, 4, 100]) 3centered_average([1, 1, 5, 5, 10, 8, 7]) 5centered_average([-10, -4, -2, -4, -2, 0]) -3 My solution: def centered_average(nums): Total = 0 = 2 Centers = nums centered.remove(max(nums)) Center.remove(min(nums)) for i in the center: Sum += i Sum of numbers in an array Returns 0 for an empty array, except for number 13, very unfortunate, so it does not count and the numbers that come immediately after 13 do not count, including 13([1, 2, 2, 1, 1]) 6 My solution: def sum13(nums): Sum = 0 for i in range(0, nums.count(13)): You're signed in with another tab or window. Reload to refresh your session You have signed out on another tab or window. Reload to refresh your session We use optional third-party analytics cookies to understand how GitHub .com so that we can create better products. We use optional third-party analytics cookies to understand how GitHub .com so that we can create better products. You can update your selection by clicking cookie settings at the bottom of the page. For more information, click the following Please refer to our privacy statement. We use cookies that are necessary to perform the necessary website functions, such as those used to log in to you. Always learn to use it. We use analytics cookies to understand how you use our website so that we can make it better, such as these cookies are used to collect information about the pages you visit and the number of clicks you need to get the job done. Learn more