S5700695 Riley Jordan

Programming Assignment

Task A:

Function	I/O	How it works	How did i test	Challenges
max_of_three()	Input: Num1, num2, num3 Output: Maximum (int,int,int)->(int)	If !(num2>max) && !(num3>max) == num1 = maximum	Biggest number (10,20,30)->30 (10,40,30)->40	Presumed userinput was needed
calculator()	Input: Num1, num2, operator Output: Results (int,int,str)->(float)	conditional assignments on parameters	(10,0,“/”)->”n/a”; (10,0,“*”)->0; #outputs of different operators	If num1 / 0
winning_number( )	Input: User_numbers, winning_numbers Output: Prize (list, list)->(str)	Match = len(set(list & set(list)	([3,5,10],[5,14,17])->3rd ([5,14,17],[5,14,17])->1s	No loops and using sets
sum_of_evens()	Input: Min_value, max_value Output: Total (int,int)->(int)	For i (min-1, max+1) % == 0 i+ total	(12,21)->60	Presumed needed ignore odds
is_prime()	Input: Num Output: Output (int)->(bool)	For i (2,int(num**0.5)+1	19->T 1->F	Originally if modulus == 0. Now check for factors

S5700695 Riley Jordan

are_anagrams()	Input: Str1, str2 Output: Output (str,str)->(Bool)	If (len!=len)=F If (sorted!=sorted)=F	(hi,ih)->T (we,eb)->F	checked if values was the same not sorted
calculate_average()	Input: Number Output: Average (list)->(float)	+list numbers / list length	[10,20,30]->20.0	If list values was 0 it didn't work
calculate_weekly_pay()	Input: Hours_worked Output: Total_pay (int)->(int)	Check if hours exceed standard for OT_P	(36)->(438)	hours -> (int)
km_to_miles()	Input: Kilometers Output: Miles (int)->(float)	km*0.62 = 1 mile	(120)->(74.4) (5)->(3.1)	Using round to 3 figs
celsius_to_fahrenheit()	Input: Celsius Output: Output (int)->(float)	celsius*9/5+32 = output	(20)->(68.0) (0)->(32.0)	N/A
net_annual_income()	Input: Gross_salary Output: Net_salary (int)->(float)	net _sarlary = grosalary * (conditional tax)	(30000)->(21000.0) (45000)->(22500.0)	G_Salary/tax was wrong
letter_occurrence()	Input: Input_string: Output: Count (string)->(int)	For loop letter (input_string) If “a” or “A” Count+1	(“Amazing”)->(2) (“AHH”)->(1)	Using and, not or

S5700695 Riley Jordan

fuel_cost()	Input: Distance_miles Output: Total_cost (int)->(float)	(d/mpg)*LPG*PPL	(50)->(6.705) (150)->(20.115)	Make d_miles positive
find_maximum_difference()	Input: A, b Output: Maximum (list,list)->(int)	For loop (a,b) then max(max, abs(i-j)	[1,5,600],[100,7,3,29,39#)->(597)	Didn't use min or max
is_golden_number()	Input: N Output: Boolean (int)->(bool)	(a*b%1000)&(a+b=n)==T	(70)->T (61)->F	Getting nested loops correct
decrypt_cypher_tex()	Input: Encrypted_text, key Output: Decyrpted_text (str,int)->(string)	For i (chr(ord(i)-key)%256)	(khoor)->(hello)	Using chr, ord

Task B:

FUNCTION	I/O	How it works	How I tested	Challenges
daily_sales()	Input Available_items, inventory records, Current_day Output Available_items (int,list,int)->(int)	(Current_day%7 !=0) (sold_unit=randint(200)-available_items==avavailable_items	unit_test()	Only update records with one function w/o merging
restock_inventory	Input Available_items, Inventory records Current day Output Available_items (int,list,int)->(int)	if(inventory_record) & current_day != %7==0 restocked=2000 - Inventory_record[-1][3]	unit_test()	Checking inventory_recordhad previous data

S5700695 Riley Jordan

Github Link

https://github.com/xmrenigmax/L4_Coursework_Programming

Github Repository