CS-Midterm-Sylvia杜竞耘

#2 more problems solved (9 and 15), could get 5 bonus points right?

1.

def printAsterisks (n) : #defining the function

if isinstance (n, int) : #define if n is integer

if 0 < n <= 75 : #define if n is a positive integer no greater than 75

print ('\*' \* n + "I can't believe that this CS midterm is so easy especially for me!") #print a line of n asterisks(\*) and the sentence

elif n > 75 : #define if n is greater than 75

print ('\*' \* 75 + "I can't believe that this CS midterm is so easy especially for me!") #print 75 asterisks(\*) and the sentence

else :

print ('Invalid n') #if n is not positive or is not integer, print invalid n

2.

def modCount (n, m) : #define a function

if isinstance (n, int) and isinstance (m, int) : #define if n and m are integers

if m <= n : #define if m is not greater than n

f = int (n / m) #calculate the result

return f #return the result

else : #if m is greater than n, print invalid number

print ('Invalid number')

else : #if m or n is not integer, print invalid number

print ('Invalid number')

3.

def compareLists (a, b) : #define function

if isinstance (a, list) and isinstance (b, list) : #define if a and b are lists

la = len (a) #define length of a

lb = len (b) #define length of b

sa = sum (a) #define sum of a

sb = sum (b) #define sum of b

same = [] #define an empty list same

for val in a : #start a for loop

if val in b : #if val in a is also in b, add val to same

same.append (val)

if la == lb and sa == sb: #if a and b are same length and same sum

print ('same length, same sum')

elif la == lb and sa != sb : #if a and b are same length but not same sum

print ('same length, not same sum')

elif la != lb and sa == sb : #if a and b are not same length but same sum

print ('not same length, same sum')

else : #if a and b are not same length and not same sum

print ('not same length, same sum')

print (same) #print the same value in a and b

else : #if a or b is not a list, print invalid a or b

print ('Invalid a or b')

4.

def is\_anagram (a, b) : #define a function, a and b are strings

m = (list (a)).sort() #define m = sorted and listed a

n = (list (b)).sort() #define n = sorted and listed b

if m == n : #if m equals n, return True

return True

else : #if not, return False

return False

5.

def remove\_duplicates (a) : #define the function

b = [] #create an empty list

for val in a : #start a for loop

if val not in b : #add the values in a only once into b

b.append (val)

else :

pass

7.

def is\_power (a, b) : #define the function

while True : #start a while loop

if a % b == 0 : #define if a is divisible by b

m = a / b #check if m is power of b

a = m

return True #return the result

else : #if a is not divisible by b

return False #return the result

9.

def is\_sorted (a) : #define a function

index = 0 #define index as 1st item in a

back = len (a) – 1 #define back as last item in a

while index < back : #start a while loop

if a[index] <= a[index + 1] : #define if a is sorted

index += 1

else : #if not, return false

return False

return True #if sorted, return true

10.

def middle (a) : #define the function

b = len (a) – 1 #b represents the index of the last value in a

c = a[1 : b] #delete a[0] and the last value

return c #return result

11.

def chop (a) : #define the function

a.pop(0) #pop the 1st element

a.pop() #pop the last element

return None

13.

n = input ('enter a list of words splited by space\n') #ask input from user

m = n.split () #split input to form a list

a = input ('enter a new word\n') #ask new word from user

m[2] = a #change the 3rd word in the list

print (m) #print result

14.

def palindrome () : #define a function

a = input ('enter a word\n') #ask input word from user

index = 0 #define index

back = len(a) – 1 #define back

while index < back and back > 0: #start a while loop

m = a[index] #define m

n = a[back] #define n

if m == n :

index += 1

back -= 1

else :

return False

return True

15.

def has\_duplicates (a) : #define the function

b = [] #define an empty list

for val in a : #start a for loop

if val not in b : #if val in a not in b, add val to b

b.append (val)

else : #if val already in b, return True

return True

return False #if all val in a not appear more than once, return false