**ASSIGNMENT\_6 [ADITYA ANAND\_21108048]**

# -\*- coding: utf-8 -\*-

"""

Created on Mon Jul 4 09:10:07 2022

@author: ADITYA ANAND

"""

#1

def perfect\_number(n):

sum = 0

for x in range(1, n):

if n % x == 0:

sum += x

return sum == n

print(perfect\_number(6))

#2

def isPalindrome(string):

left\_pos = 0

right\_pos = len(string) - 1

while right\_pos >= left\_pos:

if not string[left\_pos] == string[right\_pos]:

return False

left\_pos += 1

right\_pos -= 1

return True

print(isPalindrome('aza'))

#3

def pascal\_triangle(n):

trow = [1]

y = [0]

for x in range(max(n,0)):

print(trow)

trow=[l+r for l,r in zip(trow+y, y+trow)]

return n>=1

pascal\_triangle(6)

#4

import string, sys

def ispangram(str1, alphabet=string.ascii\_lowercase):

alphaset = set(alphabet)

return alphaset <= set(str1.lower())

print ( ispangram('The quick brown fox jumps over the lazy dog'))

#5

items=[n for n in input().split('-')]

items.sort()

print('-'.join(items))

#6

def student\_data(student\_id, \*\*kwargs):

print(f'\nStudent ID: {student\_id}')

if 'student\_name' in kwargs:

print(f"Student Name: $ {kwargs['student\_name']}")

if 'student\_name' and 'student\_class' in kwargs:

print(f"\nStudent Name: $ {kwargs['student\_name']}")

print(f"Student Class: $ {kwargs['student\_class']}")

student\_data(student\_id='SV12', student\_name='Jean Garner')

student\_data(student\_id='SV12', student\_name='Jean Garner', student\_class ='V')

#7

class Student:

pass

class Marks:

pass

student1 = Student()

marks1 = Marks()

print(isinstance(student1, Student))

print(isinstance(marks1, Student))

print(isinstance(marks1, Marks))

print(isinstance(student1, Marks))

print("\nCheck whether the said classes are subclasses of the built-in object class or not.")

print(issubclass(Student, object))

print(issubclass(Marks, object))

#8

class py\_solution:

def threeSum(self, nums):

nums, result, i = sorted(nums), [], 0

while i < len(nums) - 2:

j, k = i + 1, len(nums) - 1

while j < k:

if nums[i] + nums[j] + nums[k] < 0:

j += 1

elif nums[i] + nums[j] + nums[k] > 0:

k -= 1

else:

result.append([nums[i], nums[j], nums[k]])

j, k = j + 1, k - 1

while j < k and nums[j] == nums[j - 1]:

j += 1

while j < k and nums[k] == nums[k + 1]:

k -= 1

i += 1

while i < len(nums) - 2 and nums[i] == nums[i - 1]:

i += 1

return result

print(py\_solution().threeSum([-25, -10, -7, -3, 2, 4, 8, 10])

#9

class py\_solution:

def is\_valid\_parenthese(self, str1):

stack, pchar = [], {"(": ")", "{": "}", "[": "]"}

for parenthese in str1:

if parenthese in pchar:

stack.append(parenthese)

elif len(stack) == 0 or pchar[stack.pop()] != parenthese:

return False

return len(stack) == 0

print(py\_solution().is\_valid\_parenthese("(){}[]"))

print(py\_solution().is\_valid\_parenthese("()[{)}"))

print(py\_solution().is\_valid\_parenthese("()"))