**IMPLIMETING MULTIPLE INHERITENCE**

# the objective is to generate a febonocci series and to get nth term

# class 1 for generation of febonaccinumber

class febonaccinumber:

# a costructor

def \_\_init\_\_(self, x):

self.x = 0

self.a = 0

self.b = 1

self.c = 1

# defining method for febonacci series

def fib1(self):

if self.x == 1:

print(self.a)

else:

print(self.a)

print(self.b)

for i in range(2, self.x):

self.c = self.a + self.b

self.a = self.b

self.b = self.c

print(self.c)

# creating class for implimenting multiple inheritence

class extra:

pass

# creating class for getting Nth term of series

class nthterm(febonaccinumber, extra):

# defining constructors

def \_\_init\_\_(self, n):

self.n = n

self.arr = 0

@classmethod

def fibnth(cls):

arr = [0] \* (n + 1)

arr[1] = 1

for i in range(2, n + 1):

arr[i] = arr[i - 1] + arr[i - 2]

return arr[n]

# main function

# entering input and accesing class through object

x = int(input("enter number"))

object1 = febonaccinumber(x)

object3 = extra()

object1.fib1()

n = (int(input("Enter the term :")))

object2 = nthterm(n)

y = object2.fibnth()

print(y)

PEP8 SCREEN SHOT

