#Shihab Muhtasim

#IID:21301610

# TASK 1(a) Function to find factorial of given number

def factorial(n):

if n == 0:

return 1

return n \* factorial(n-1)

print(factorial(3))

# TASK 1 (b) Fibonacci series

def fibonacci(n):

if n<=1:

return n

return fibonacci(n-1)+fibonacci(n-2)

print(fibonacci(8))

#TASK 1(c) Print all the elements of a given array recursively.

def PrintArray(arr, idx=0):

if idx<len(arr):

print(arr[idx],end=" ")

idx+=1

PrintArray(arr,idx)

arr=[10,20,30,40]

PrintArray(arr)

#TASK 1(d) base to the n power

def powerN(base, n):

if n==0:

return 1

else:

return base\* powerN(base, n-1)

print(powerN(3, 2))

#TASK 2(a) decimal to binary

def binary(deci):

if deci==0:

return

else:

binary(deci//2)

print(deci%2, end="")

binary(115)

#TASK 2(b)

class Node:

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

self.data=data

self.next=n

class Singly:

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

self.head=Node(arr[0],None)

temp=self.head

for i in range (1, len(arr)):

n=Node(arr[i],None)

temp.next=n

temp=n

arr=[10,20,30,40]

s=Singly(arr)

hold= s.head

def add(head):

temp=head

if temp==None:

return 0

else:

return temp.data+ add(temp.next)

print(add(hold))

#TASK 2(c)

class Node:

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

self.data=data

self.next=n

class Singly:

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

self.head=Node(arr[0],None)

temp=self.head

for i in range (1, len(arr)):

n=Node(arr[i],None)

temp.next=n

temp=n

def printli(self):

temp=self.head

while temp!=None:

print(temp.data)

temp=temp.next

arr=[10,20,30,40]

s=Singly(arr)

hold= s.head

def backwards(head):

if head==None:

return

else:

backwards(head.next)

print(head.data,end=" ")

backwards(hold)

#TASK 3 HOUSE OF CARDS

def house(height):

if height==0:

print("No House")

elif height ==1:

return 8

else:

return 5+ house(height-1)

print(house(4))

#TASK 4(A)

def uptoN(n):

if n==0:

return

else:

uptoN(n-1)

print(n,end="")

def pattern(val):

if val==0:

return

else:

pattern(val-1)

uptoN(val)

print()

pattern(5)

#TASK 4(B)

def space(n):

if n==0:

return

else:

space(n-1)

print(" ",end="")

def uptoN(n):

if n==0:

return

else:

uptoN(n-1)

print(n,end="")

def pattern(val,n=0):

if val==0:

return

else:

space(val-1)

uptoN(n+1)

print()

pattern(val-1,n+1)

pattern(5)

#TASK 5

class FinalQ:

def print(self,array,idx):

if(idx<len(array)):

profit = self.calcProfit(array[idx])

print(f"{idx+1}. Investment: {array[idx]} Profit: {float(profit)}")

self.print(array,idx+1)

else:

return 0

def calcProfit(self,investment):

if investment<=25000:

return 0

elif investment<=100000:

if investment%1000!=0:

return 4.5+calcProfit(investment-100)

else:

return 45+calcProfit(investment-1000)

elif investment>100000:

return 80+calcProfit(investment-1000)

#Tester

array=[25000,100000,250000,350000]

f = FinalQ()

f.print(array,0)