Họ và tên: NGUYỄN ĐÌNH TIẾN

MSSV: 20168533

Lớp: VUWIT15

Project 1

Program Language: Python

Tuần: 10

Project Euler

Problem 48:

1.Đề bài:

The series, 11 + 22 + 33 + ... + 1010 = 10405071317.

Find the last ten digits of the series, 11 + 22 + 33 + ... + 10001000.

2.Thuật toán:

- Tính giá trị của dãy số

- Hiển thị ra 10 kí tự cuối của giá trị đó, ta được kết quả cần tìm

3.Đáp án: 9110846700

4.Mã nguồn:

# Self powers

sum = 0

for i in range(1,1001):

sum += i\*\*i

print(''.join(list(str(sum))[-10:-1])+list(str(sum))[-1])

Problem 50:

1.Đề bài:

The prime 41, can be written as the sum of six consecutive primes:

41 = 2 + 3 + 5 + 7 + 11 + 13

This is the longest sum of consecutive primes that adds to a prime below one-hundred.

The longest sum of consecutive primes below one-thousand that adds to a prime, contains 21 terms, and is equal to 953.

Which prime, below one-million, can be written as the sum of the most consecutive primes?

2. Thuật toán:

- Tìm tất cả số nguyên tố dưới 1 triệu

- Lần lượt phân tích từng số ra các số nguyên tố nhỏ hơn bằng cách trừ dần đi các số nguyên tố từ nhỏ đến lớn

- Lấy ra số nguyên tố có thể phân tích ra tổng nhiều số nguyên tố nhất và hiển thị

3.Đáp án: 997651

4.Mã nguồn:

# Circular primes

from math import ceil

list\_primes = []

# Tìm tất cả số nguyên tố dưới 1 triệu

n = 1000000

primes = [True] \* n

primes[0] = False

primes[1] = False

roundUp = lambda n, prime: int(ceil(n / prime))

for currentPrime in range(2, n):

if not primes[currentPrime]:

continue

list\_primes.append(currentPrime)

for i in range(2, roundUp(n, currentPrime)):

primes[i \* currentPrime] = False

# Ta được list\_primes là list các số nguyên tố dưới 1 triệu

# Tìm số nguyên tố bằng tổng nhiều số nguyên tố nhất

# list\_length: List số các số nguyên tố mà 1 số nguyên tố có thể phân tích ra

# list\_dictionary: List các bộ giá trị, key là số nguyên tố có thể phân tích được, value là số các số nguyên tố mà key đó có thể phân tích ra

list\_length = []

list\_dictionary = []

for number in list\_primes:

# list\_check: List các số nguyên tố mà 1 số có thể phân tích ra

list\_check = []

number\_real = number

for i in list\_primes:

if i <= number:

number -= i

list\_check.append(i)

elif i > number:

for j in list\_primes:

if j < i - number:

number += j

list\_check.remove(j)

elif j == i - number:

number = 0

list\_check.append(i)

list\_check.remove(j)

break

elif j > i - number:

break

break

if number == 0:

break

if number == 0:

list\_length.append(len(list\_check))

list\_dictionary.append({number\_real:len(list\_check)})

# Duyệt từng bộ giá trị trong list\_dictionary, đến key có value bằng với giá trị lớn nhất trong list\_length thì dừng lại và key là số nguyên tố cần tìm

loop = True

for index, item in enumerate(list\_dictionary):

if loop == True:

for key, value in item.items():

if value == max(list\_length):

print(key)

loop = False

break

elif loop == False:

break

Problem 52:

1.Đề bài:

It can be seen that the number, 125874, and its double, 251748, contain exactly the same digits, but in a different order.

Find the smallest positive integer, *x*, such that 2*x*, 3*x*, 4*x*, 5*x*, and 6*x*, contain the same digits.

2.Thuật toán:

- Duyệt từng số từ 1

- Kiểm tra lần lượt các điều kiện: số kí tự của số đó khi nhân 2, 3, 4, 5, 6 có giữ nguyên không

- Xóa từng chữ số có trong số ban đầu với ra khỏi số sau khi nhân nếu có trong đó, dùng biến check để kiểm tra, mỗi lần xóa được hết kí tự thì cộng thêm 1, nếu check bằng 5 (5 lần kiểm tra 2, 3, 4, 5, 6) thì dừng lại và hiển thị ra giá trị cần tìm

3.Đáp án: 142857

4.Mã nguồn:

# Permuted multiples

i = 1

loop = True

while loop:

check = 0

for j in range(2,7):

if len(str(i)) == len(str(j\*i)):

list1 = list(str(i))

list2 = list(str(j\*i))

for index, item in enumerate(list1):

if item in list2:

list2.remove(item)

if len(list2) == 0:

check += 1

else:

i += 1

if check == 5:

print(i)

loop = False

else:

i += 1