Họ và tên: NGUYỄN ĐÌNH TIẾN

MSSV: 20168533

Lớp: VUWIT15

Project 1

Program Language: Python

Tuần: 6

Project Euler

Problem 21:

1.Đề bài:

Let d(*n*) be defined as the sum of proper divisors of *n* (numbers less than *n* which divide evenly into *n*).  
If d(*a*) = *b* and d(*b*) = *a*, where *a* ≠ *b*, then *a* and *b* are an amicable pair and each of *a* and *b* are called amicable numbers.

For example, the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110; therefore d(220) = 284. The proper divisors of 284 are 1, 2, 4, 71 and 142; so d(284) = 220.

Evaluate the sum of all the amicable numbers under 10000.

2.Thuật toán:

- Duyệt các số từ 1 đến 10000, thêm các ước của số đang xét vào 1 list và tính tổng các số trong list

- Xét các ước của giá trị tổng đó, giống như trên, nếu tổng các ước đó lại bằng số đang xét bên trên thì ta được 1 cặp (2 số) thỏa mãn

3.Đáp án: 31626

4.Mã nguồn:

# Tổng tất cả các số Amicable dưới 10000

list\_amicable\_number = []

for n in range(1,10000):

if n not in list\_amicable\_number:

list\_divisor\_1 = []

list\_divisor\_2 = []

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

if n % i == 0 and i not in list\_divisor\_1:

list\_divisor\_1.append(int(i))

if n/i not in list\_divisor\_1 and i != 1:

list\_divisor\_1.append(int(n/i))

pair = sum(list\_divisor\_1)

if n != pair:

for j in range(1, pair):

if pair % j == 0 and j not in list\_divisor\_2:

list\_divisor\_2.append(int(j))

if n/i not in list\_divisor\_2 and j != 1:

list\_divisor\_2.append(int(pair/j))

if sum(list\_divisor\_2) == n:

list\_amicable\_number.append(n)

list\_amicable\_number.append(pair)

print(sum(list\_amicable\_number))

Problem 22:

1.Đề bài:

Using [names.txt](https://projecteuler.net/project/resources/p022_names.txt) (right click and 'Save Link/Target As...'), a 46K text file containing over five-thousand first names, begin by sorting it into alphabetical order. Then working out the alphabetical value for each name, multiply this value by its alphabetical position in the list to obtain a name score.

For example, when the list is sorted into alphabetical order, COLIN, which is worth 3 + 15 + 12 + 9 + 14 = 53, is the 938th name in the list. So, COLIN would obtain a score of 938 × 53 = 49714.

What is the total of all the name scores in the file?

Link file names.txt: https://projecteuler.net/project/resources/p022\_names.txt

2. Thuật toán:

- Đọc file và đưa toàn bộ tên vào list

- Xét từng tên trong list, đưa các kí tự của tên vào 1 list và tính giá trị value của kí tự đó dựa vào dictionary alphabet, sau đó update tên đó trong list ban đầu bằng giá trị, sau khi xét toàn bộ tên thì t được list mới đã trở thành list các giá trị của từng tên cũ

- Xét list ban đầu từng phần tử và nhân nó với index + 1, ta được giá trị sum cần tìm

3.Đáp án: 871198282

4.Mã nguồn:

# Tính tổng Name Score của danh sách 5000 tên

file\_names = open("problem022\_names.txt","r")

total\_names = sorted(file\_names.read().replace('"','').split(','))

alphabet = {

"a": 1,

"b": 2,

"c": 3,

"d": 4,

"e": 5,

"f": 6,

"g": 7,

"h": 8,

"i": 9,

"j": 10,

"k": 11,

"l": 12,

"m": 13,

"n": 14,

"o": 15,

"p": 16,

"q": 17,

"r": 18,

"s": 19,

"t": 20,

"u": 21,

"v": 22,

"w": 23,

"x": 24,

"y": 25,

"z": 26

}

for index, item in enumerate(total\_names):

value = 0

for index1, item1 in enumerate(list(item)):

value += alphabet[item1.lower()]

total\_names[index] = value

sum = 0

for index, item in enumerate(total\_names):

sum += item \* (index+1)

print(sum)

Problem 25:

1.Đề bài:

The Fibonacci sequence is defined by the recurrence relation:

F*n* = F*n*−1 + F*n*−2, where F1 = 1 and F2 = 1.

Hence the first 12 terms will be:

F1 = 1  
F2 = 1  
F3 = 2  
F4 = 3  
F5 = 5  
F6 = 8  
F7 = 13  
F8 = 21  
F9 = 34  
F10 = 55  
F11 = 89  
F12 = 144

The 12th term, F12, is the first term to contain three digits.

What is the index of the first term in the Fibonacci sequence to contain 1000 digits?

2.Thuật toán:

- Thêm các số fibonacci vào dãy

- Đến khi số có giá trị vượt quá 10 mũ 999 thì dừng lại

- Số phần tử của dãy chính là số thứ tự của số fibonacci đầu tiên cần tìm

3.Đáp án: 4782

4.Mã nguồn:

fibonacci = [1,1]

loop = True

while loop:

number = fibonacci[-1] + fibonacci[-2]

fibonacci.append(number)

if number >= 10 \*\* 999:

loop = False

print(len(fibonacci))