Họ và tên: NGUYỄN ĐÌNH TIẾN

MSSV: 20168533

Lớp: VUWIT15

Project 1

Program Language: Python

Tuần: 8

Project Euler

Problem 36:

1.Đề bài:

The decimal number, 585 = 10010010012 (binary), is palindromic in both bases.

Find the sum of all numbers, less than one million, which are palindromic in base 10 and base 2.

(Please note that the palindromic number, in either base, may not include leading zeros.)

2.Thuật toán:

Kiểm tra một số có là số đối xứng hay không, nếu có thì tiếp tục kiểm tra xem số đó ở dạng nhị phân có đối xứng hay không, nếu thỏa mãn thì đếm.

3.Đáp án: 872187

4.Mã nguồn:

# Double-base palindromes

n = 0

for i in range(1,1000000):

    if str(i) == str(i)[::-1]:

        if bin(i)[2:] == bin(i)[2:][::-1]:

            n += i

print(n)

Problem 40:

1.Đề bài:

An irrational decimal fraction is created by concatenating the positive integers:

0.123456789101112131415161718192021...

It can be seen that the 12th digit of the fractional part is 1.

If *dn* represents the *n*th digit of the fractional part, find the value of the following expression.

*d*1 × *d*10 × *d*100 × *d*1000 × *d*10000 × *d*100000 × *d*1000000

2. Thuật toán:

- Dùng vòng while để lặp thêm dần vào chuỗi các số lần lượt từ 1 tăng dần, kiểm tra đến khi nào số kí tự của chuỗi vượt quá 1000001 thì dừng lại

- Đưa chuỗi vào list và lấy ra các kí tự ở vị trí 1, 10, 100, ... theo yêu cầu và nhân lại, ta được kết quả

3.Đáp án: 210

4.Mã nguồn:

# Champernowne's constant

sequence = " "

loop = True

i = 1

while loop:

sequence += str(i)

i += 1

list\_check = list(sequence)

if len(list\_check) >= 1000001:

break

value = int(list\_check[1]) \* int(list\_check[100]) \* int(list\_check[1000]) \* int(list\_check[10000]) \* int(list\_check[100000]) \* int(list\_check[1000000])

print(value)

Problem 35:

1.Đề bài:

The *n*th term of the sequence of triangle numbers is given by, *tn* = ½*n*(*n*+1); so the first ten triangle numbers are:

1, 3, 6, 10, 15, 21, 28, 36, 45, 55, ...

By converting each letter in a word to a number corresponding to its alphabetical position and adding these values we form a word value. For example, the word value for SKY is 19 + 11 + 25 = 55 = *t*10. If the word value is a triangle number then we shall call the word a triangle word.

Using [words.txt](https://projecteuler.net/project/resources/p042_words.txt) (right click and 'Save Link/Target As...'), a 16K text file containing nearly two-thousand common English words, how many are triangle words?

Link download file [words.txt](https://projecteuler.net/project/resources/p042_words.txt): https://projecteuler.net/project/resources/p042\_words.txt

2.Thuật toán:

Xét từng tên trong file, tính giá trị tương ứng của tên đó dựa trên alphabets, kiểm tra nếu giá trị đó là triangle numbers thì đếm

3.Đáp án: 162

4.Mã nguồn:

# How many are triangle words?

import math

file\_names = open("problem042\_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

}

count = 0

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

value = 0

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

value += alphabet[item1.lower()]

if value \* 2 == int(math.sqrt(value \* 2)) \* (int(math.sqrt(value \* 2)) + 1):

count += 1

print(count)