Họ và tên: NGUYỄN ĐÌNH TIẾN

MSSV: 20168533

Lớp: VUWIT15

Project 1

Program Language: Python

Tuần: 13

Project Euler

Problem 67:

1.Đề bài:

By starting at the top of the triangle below and moving to adjacent numbers on the row below, the maximum total from top to bottom is 23.

**3**  
**7** 4  
2 **4** 6  
8 5 **9** 3

That is, 3 + 7 + 4 + 9 = 23.

Find the maximum total from top to bottom in [triangle.txt](https://projecteuler.net/project/resources/p067_triangle.txt) (right click and 'Save Link/Target As...'), a 15K text file containing a triangle with one-hundred rows.

**NOTE:** This is a much more difficult version of [Problem 18](https://projecteuler.net/problem=18). It is not possible to try every route to solve this problem, as there are 299 altogether! If you could check one trillion (1012) routes every second it would take over twenty billion years to check them all. There is an efficient algorithm to solve it. ;o)

2.Thuật toán:

- Chia mỗi dòng thành list riêng và tách từng số trong dòng thành số riêng

- Duyệt mảng từ dưới đi lên, thay thế dần mỗi số bằng tổng của nó với số bên dưới lớn hơn

- Khi duyệt xong thì mảng chỉ còn 1 số là đường đi lớn nhất cần tìm

3.Đáp án: 7273

4.Mã nguồn:

# Maximum path sum II

file = open("problem067.txt","r")

#Chia mỗi dòng thành 1 chuỗi nhỏ

number = file.read().strip().split('\n')

#Tiếp tục tách mỗi số trong chuỗi nhỏ ra và ép kiểu thành int, chuyển các chuỗi nhỏ thành list

for i in range(0,len(number)):

    number[i] = number[i].split(' ')

    number[i] = [int(x) for x in number[i]]

#Tính các đường đi từ chân tháp

for i in range(len(number)-2,-1,-1):

    for j in range(len(number[i])):

        number[i][j] = number[i][j] + max(number[i+1][j], number[i+1][j+1])

print(number[0][0])

Problem 18:

1.Đề bài:

By starting at the top of the triangle below and moving to adjacent numbers on the row below, the maximum total from top to bottom is 23.

**3**  
**7** 4  
2 **4** 6  
8 5 **9** 3

That is, 3 + 7 + 4 + 9 = 23.

Find the maximum total from top to bottom of the triangle below:

75  
95 64  
17 47 82  
18 35 87 10  
20 04 82 47 65  
19 01 23 75 03 34  
88 02 77 73 07 63 67  
99 65 04 28 06 16 70 92  
41 41 26 56 83 40 80 70 33  
41 48 72 33 47 32 37 16 94 29  
53 71 44 65 25 43 91 52 97 51 14  
70 11 33 28 77 73 17 78 39 68 17 57  
91 71 52 38 17 14 91 43 58 50 27 29 48  
63 66 04 68 89 53 67 30 73 16 69 87 40 31  
04 62 98 27 23 09 70 98 73 93 38 53 60 04 23

**NOTE:** As there are only 16384 routes, it is possible to solve this problem by trying every route. However, [Problem 67](https://projecteuler.net/problem=67), is the same challenge with a triangle containing one-hundred rows; it cannot be solved by brute force, and requires a clever method! ;o)

2. Thuật toán:

- Chia mỗi dòng thành list riêng và tách từng số trong dòng thành số riêng

- Duyệt mảng từ dưới đi lên, thay thế dần mỗi số bằng tổng của nó với số bên dưới lớn hơn

- Khi duyệt xong thì mảng chỉ còn 1 số là đường đi lớn nhất cần tìm

3.Đáp án: 1074

4.Mã nguồn:

# Maximum path sum I

number = '''75

95 64

17 47 82

18 35 87 10

20 04 82 47 65

19 01 23 75 03 34

88 02 77 73 07 63 67

99 65 04 28 06 16 70 92

41 41 26 56 83 40 80 70 33

41 48 72 33 47 32 37 16 94 29

53 71 44 65 25 43 91 52 97 51 14

70 11 33 28 77 73 17 78 39 68 17 57

91 71 52 38 17 14 91 43 58 50 27 29 48

63 66 04 68 89 53 67 30 73 16 69 87 40 31

04 62 98 27 23 09 70 98 73 93 38 53 60 04 23'''

#Chia mỗi dòng thành 1 chuỗi nhỏ

number = number.split('\n')

#Tiếp tục tách mỗi số trong chuỗi nhỏ ra và ép kiểu thành int, chuyển các chuỗi nhỏ thành list

for i in range(0,len(number)):

    number[i] = number[i].split(' ')

    number[i] = [int(x) for x in number[i]]

#Tính các đường đi từ chân tháp

for i in range(len(number)-2,-1,-1):

    for j in range(len(number[i])):

        number[i][j] = number[i][j] + max(number[i+1][j], number[i+1][j+1])

print(number[0][0])

Problem 97:

1.Đề bài:

The first known prime found to exceed one million digits was discovered in 1999, and is a Mersenne prime of the form 26972593−1; it contains exactly 2,098,960 digits. Subsequently other Mersenne primes, of the form 2*p*−1, have been found which contain more digits.

However, in 2004 there was found a massive non-Mersenne prime which contains 2,357,207 digits: 28433×27830457+1.

Find the last ten digits of this prime number.

2.Thuật toán:

- Lần lượt tính các phép lũy thừa, nhân và cộng nhưng sau mỗi phép tính đều lấy phần dư khi chia cho 10 tỉ để chỉ lấy ra 10 số cuối cùng

3.Đáp án: 8739992577

4.Mã nguồn:

# Large non-Mersenne prime

n = 2

for i in range(7830456):

n = (2 \* n) % 10000000000

n \*= 28433

n += 1

n = n % 10000000000

print(n)