1911136 – Nguyễn Việt Duy Danh – CTK43 – Lab1

Github: <https://github.com/gominn1604/LT_Python/tree/main/Lab/Lab_1>

Bài 1: Hoàn thành 25/150 bài tập cơ bản ở trang:

<https://www.w3resource.com/python-exercises/python-basic-exercises.php>

Ex1:

print("Twinkle, twinkle, little star, \n\tHow I wonder what you are! \n\t\tUp above the world so high, \n\t\tLike a diamond in the sky. \nTwinkle, twinkle, little star, \n\tHow I wonder what you are!")

Ex2:

import platform

print(platform.python\_version())

Ex3:

import datetime

print(datetime.datetime.now())

Ex4:

from cmath import pi

radius = float(input("Input the radius of a circle: "))

area = pi \* radius \* radius

print(f"Circle shape area is: {area}")

Ex5:

fname = input("Input your first name: ")

lname = input("Input your last name: ")

print(f"{lname} {fname}")

Ex6:

values = input("Input some comma seprated numbers: ")

list = values.split(",")

tuple = tuple(list)

print("List: ", list)

print("Tuple: ", tuple)

Ex7:

from fileinput import filename

filename = input("Input a filename: ")

extOfFile = filename.split('.')

print("The extension of the file is: " + repr(extOfFile[-1]))

Ex8:

color\_list = ["Black", "Green", "White", "Blue", "Pink"]

print(f"{color\_list[0]} {color\_list[-1]}")

Ex9:

st\_date = (21, 12, 2022)

print("The examination will start from: %i/%i/%i"%st\_date)

Ex10:

n = int(input("Input n: "))

nn = int("%s%s" %(n,n))

nnn = int("%s%s%s" %(n,n,n))

print(n+nn+nnn)

Ex11:

print(abs.\_\_doc\_\_)

Ex12:

import calendar

y = int(input("Year: "))

m = int(input("Month: "))

print(calendar.month(y, m))

Ex13:

print("""

a string that you "don't" have to escape

This

is a ....... multi-line

heredoc string -------->""")

Ex14:

from datetime import date

lDate = date(2020, 5, 20)

fDate = date(2019, 4, 16)

result = lDate - fDate

print(result)

Ex15:

from cmath import pi

r = 6.0

V = 4.0/3.0 \* pi \* r\*\*3

print(f"The volume is: {V}")

Ex16:

n = int(input("Input n: "))

def diff(n):

    if n < 17:

        return 17-n

    else:

        return abs(17-n)\*2

print(diff(n))

Ex17:

def near\_thousand(n):

      return ((abs(1000 - n) <= 100) or (abs(2000 - n) <= 100))

print(near\_thousand(1100))

Ex18:

def sum(x,y,z):

    sum = x + y + z

    if x == y == z:

        sum = sum \* 3

    return sum

print(sum(3,3,3))

Ex19:

def newString(string):

    return string if len(string) >= 2 and string[:2] == "Is" else "Is" + string

print(newString("danh"))

Ex20:

def getString(str, n):

   result = ""

   for i in range(n):

      result = result + str

   return result

print(getString("danh",4))

Ex21:

n = int(input("Enter a number: "))

print ("Odd number") if n % 2 != 0 else print ("Even number")

Ex22:

def count(list):

    count = 0

    for num in list:

        if num == 4:

            count +=1

    return count

print(count([1,4,2,4,2,1,4]))

Ex23:

def subStringCopy(str, n):

    flen = 2

    result = ""

    if flen > len(str):

        flen = len(str)

    for i in range(n):

        result = result + str[:flen]

    return result

print(subStringCopy("danh",4))

Ex24:

def isVowel(char):

    allVowel = 'ueoai'

    return char in allVowel

print(isVowel('c'))

Ex25:

def isGrMem(grData, n):

    for value in grData:

        if n == value:

            return True

    return False

print(isGrMem([1,2,3,4], 4))

Bài 2: Viết hàm thực hiện các chức năng sau:

1. Tính:

a) (a + b),

b) a/b,

c) a\*\*b.

a = int(input("Enter a number a: "))

b = int(input("Enter a number b: "))

print(f"a + b = {a+b}")

print(f"a / b = {a/b}")

print(f"a \*\* b = {a\*\*b}")

2. Tính diện tích hình chữ nhật khi biết bán kính

from cmath import pi

r = float(input("Enter a radius: "))

s = float(pi \* r\*\*2)

print("S: ", s)

3. Xuất tất cả các số nguyên tố trong 1 khoảng cho trước

import math

fNumber = int(input("Enter a first number: "))

lNumber = int(input("Enter a last number: "))

def isPrimeNum(lNumber):

    if(lNumber<2):

        return False

    squareRoot = int(math.sqrt(lNumber))

    for value in range(2, squareRoot+1):

        if(lNumber % value == 0):

            return False

    return True

sb=""

for i in range (fNumber, lNumber+1):

    if (isPrimeNum(i)):

        sb = sb + str(i) + " "

    i = i + 1

print(sb)

4. Kiểm tra 1 số nguyên n có phải là số Fibonacci hay không

import math

n = int(input("Enter a number: "))

def checkPerfectSquare(m):

    n = int (math.sqrt(m))

    return n \* n == m

def checkFibo(m):

    return checkPerfectSquare(5 \* m \* m + 4) or checkPerfectSquare(5 \* m \* m - 4)

if (checkFibo(n) == True):

    print (n, "is the Fibonacci number")

else:

    print (n , "is not the Fibonacci number")

5. Tìm số Fibonacci thứ n (dùng đệ quy và không đệ quy)

# Khong dung de quy

def fibonacciWithoutRecursive(n):

    a = 0

    b = 1

    c = 1

    if (n < 0):

        return -1

    elif (n == 0 or n == 1):

        return n

    else:

        for i in range(2, n):

            a = b

            b = c

            c = a + b

        return c

# Dung de quy

def fibonacciWithRecursive(n):

    if (n < 0):

        return -1

    elif (n == 0 or n == 1):

        return n

    else:

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

n = int(input("Enter a number = "))

print(f"The {n}th Fibonacci number is: {str(fibonacciWithoutRecursive(n))}")

print(f"The {n}th Fibonacci number is: {str(fibonacciWithRecursive(n))}")

6. Tính tổng n số Fibonacci đầu tiên (dùng đệ quy và không đệ quy)

n = int(input("Enter a number: "))

# Khong dung de quy

def fibonacciWithoutRecursive(n):

    a = 0

    b = 1

    c = 1

    if (n < 0):

        return -1

    elif (n == 0 or n == 1):

        return n

    else:

        for i in range(2, n):

            a = b

            b = c

            c = a + b

        return c

listFibonacciWithoutRecursive = []

for i in range(0, n+1):

    listFibonacciWithoutRecursive.append(fibonacciWithoutRecursive(i))

# Dung de quy

def fibonacciWithRecursive(n):

    if (n < 0):

        return -1

    elif (n == 0 or n == 1):

        return n

    else:

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

listFibonacciWithRecursive = []

for i in range(0, n+1):

    listFibonacciWithRecursive.append(fibonacciWithRecursive(i))

print(f"Sum of the fibonacci sequence is: {sum(listFibonacciWithoutRecursive)}")

print(f"Sum of the fibonacci sequence is: {sum(listFibonacciWithRecursive)}")

7. Tính tổng căn bậc 2 của n số nguyên đầu tiên

import math

def sumNFirstInteger(n):

    sum = 0

    for i in range(0, n+1):

        sum += i

    return sum

n = int(input("Enter a number = "))

result = math.sqrt(sumNFirstInteger(n))

print(f"Sum of square roots of first {n} integers is: {result}")

8. Giải phương trình bậc 2: ax2 + bx + c=0

from math import sqrt

def ptBac2(a, b, c):

    x1 = 0

    x2 = 0

    delta = b\*\*2 - 4\*a\*c

    if delta < 0:

        print("Phương trình vô nghiệm")

    elif delta == 0:

        print(f"Phương trình có nghiệm kép x1=x2={-(b/(2\*a))}")

    else:

        x1 = (-(b) + sqrt(delta))/(2\*a)

        x2 = (-(b) - sqrt(delta))/(2\*a)

        print(f"Phương trình có 2 nghiệm phân biệt x1 = {x1}, x2 = {x2}")

a = float(input("Nhập hệ số a: "))

b = float(input("Nhập hệ số b: "))

while True:

    if a == 0 and b == 0:

        print("Một trong hai hệ số a, b phải khác 0")

        a = float(input("Nhập lại hệ số a: "))

        b = float(input("Nhập lại hệ số b: "))

    else:

        break

c = float(input("Nhập hệ số c: "))

ptBac2(a, b, c)

9. Tính n!

n = int(input("Enter a number: "))

def factorial(n):

    if n == 0:

        return 1

    return n \* factorial(n-1)

print(f"The factorial of {n} is: {factorial(n)}")

10.In \* dạng tam giác dưới như hình bên, đầu vào là số hàng(cột)

row = int(input("Enter the row number: "))

def printTriangle(row):

    listStar = []

    for i in range (1, row + 1):

        listStar.append("\*\t" \* i)

    print("\n".join(listStar))

printTriangle(row)

11. Đổi giờ - phút – giây: thời gian đầu vào là giây được đổi thành giờ, phút, giây.

Xuất kết quả ra màn hình dưới dạng: giờ:phút:giây. Ví dụ: soGiay = 3770 thì xuất

ra màn hình 1:2:50.

nOfSeconds = int(input("Enter a number of seconds: "))

def convertSec(s):

    sec = 0

    min = 0

    hour = 0

    if (s < 60):

        sec = s

        print(f"{hour}:{min}:{sec}")

    elif (s < 3600 and s >= 60):

        min = int((s-s % 60)/60)

        sec = int(s % 60)

        print(f"{hour}:{min}:{sec}")

    else:

        hour = int((s-s % 3600)/3600)

        min = int(((s % 3600) - (sec % 3600) % 60)/60)

        sec = int(s-min\*60-hour\*3600)

        print(f"{hour}:{min}:{sec}")

convertSec(nOfSeconds)

12.Cho một mảng số nguyên: (nên viết 2-3 cách)

a) Xuât tất cả các số lẻ không chia hết cho 5

b) Xuất tất cả các số Fibonacci

c) Tìm số nguyên tố lớn nhất

d) Tìm số Fibonacci bé nhất

e) Tính trung bình các số lẻ

f) Tính tích các phần tử là số lẻ không chia hết cho 3 trong mảng

g) Đổi chỗ 2 phần tử của danh sách, đầu vào là 2 vị trí cần đổi chỗ

h) Đảo ngược trật tự các phần tử của danh sách

i) Xuất tất cả các số lớn thứ nhì của danh sách

j) Tính tổng các chữ số của tất cả các số trong danh sách

k) Đếm số lần xuất hiện của một số trong danh sách

l) Xuất các số xuất hiện n lần trong danh sách

m) Xuất các số xuất hiện nhiều lần nhất trong danh sách

import math

import msvcrt

from os import system

import os

list = [1, 2, 3, 4, 5, 8, 9, 12, 15, 18, 21, 25, 30, 35, 37, 41, 21]

def CheckPerfectSquare(m):

    n = int(math.sqrt(m))

    return n \* n == m

def CheckFibo(m):

    return CheckPerfectSquare(5 \* m \* m + 4) or CheckPerfectSquare(5 \* m \* m - 4)

def isOdd(i):

    return True if i % 2 != 0 else False

def SoLeKhongChiaHetCho5(list):

    sb = ""

    for i in list:

        if i % 5 != 0 and i % 2 != 0:

            sb = sb + str(i) + " "

    print(sb)

def XuatTatCaSoFibonacci(list):

    sb = ""

    for i in list:

        if CheckFibo(i) == True:

            sb = sb + str(i) + " "

    print(sb)

def isPrimeNum(lNumber):

    if(lNumber<2):

        return False

    squareRoot = int(math.sqrt(lNumber))

    for value in range(2, squareRoot+1):

        if(lNumber % value == 0):

            return False

    return True

def TimSoNguyenToLonNhat(list):

    result = []

    for i in list:

        if isPrimeNum(i):

            result.append(i)

    print(max(result))

def TimSoFibonacciBeNhat(list):

    result = []

    for i in list:

        if CheckFibo(i) == True:

            result.append(i)

    print(min(result))

def TinhTrungBinhCacSoLe(list):

    result = []

    sum = 0

    for i in list:

        if isOdd(i):

            result.append(i)

    print(f"Danh sách các số lẻ: {result}")

    for i in result:

        sum+=i

    print(sum/len(result))

def TichSoLeKhongChiaHetCho3(list):

    result = []

    mul = 1

    for i in list:

        if isOdd(i) and i %3 != 0:

            result.append(i)

    print(f"Danh sách các số lẻ không chia hết cho 3: {result}")

    for i in result:

        mul\*=i

    print(mul)

def Swap(list, x, y):

    list[x], list[y] = list[y], list[x]

def Rev(list):

    print("Mảng sau khi đổi: ", list[::-1])

def TimSoLonThuNhi(list):

    maxst = max(list[0], list[1])

    maxnd = max(list[0], list[1])

    for i in range(2, len(list)):

        if list[i] > maxst:

            maxnd = maxst

            maxst = list[i]

        elif (list[i]>maxnd) and (maxst > list[i]):

            maxnd=list[i]

    print(f"Số lớn thứ nhì trong mảng là: {maxnd}")

def TongCacChuSo(n):

    total = 0

    while (n>0):

        total = total + n%10

        n = int(n/10)

    return total

def TongCacSo(list):

    sum = 0

    for i in list:

        sum += TongCacChuSo(i)

    print(f"Tổng các chữ số của tất cả các số trong danh sách là: {sum}")

def DemSoLanXuatHien(list, n):

    if(n in list):

        print(f"Số lần xuất hiện của {n} trong mảng là: ",list.count(n))

    else:

        print('Không tồn tại phần tử trong mảng!')

def XuatSoXuatHienNLan(list, n):

    result = []

    for i in list:

        if list.count(i) == n and i not in result:

            result.append(i)

    print(f"Số xuất hiện {n} lần trong mảng là: {result}")

def XuatHienNhieuNhat(list):

    b = []

    c = []

    for i in range(len(list)-1):

        b.append(list.count(list[i]))

    for i in range(len(b)-1):

        if b[i] == max(b):

            c.append(list[i])

    print(f"Số xuất hiện nhiều lần nhất trong danh sách là: ",c[0])

while True:

    system('CLS')

    print('Chọn chức năng muốn thực hiện: ')

    print('1: Xuất tất cả các số lẻ không chia hết cho 5')

    print('2: Xuất tất cả các số Fibonacci')

    print('3: Tìm số nguyên tố lớn nhất')

    print('4: Tìm số Fibonacci bé nhất')

    print('5: Tính trung bình các số lẻ')

    print('6: Tính tích các phần tử là số lẻ không chia hết cho 3 trong mảng')

    print('7: Đổi chỗ 2 phần tử của danh sách, đầu vào là 2 vị trí cần đổi chỗ')

    print('8: Đảo ngược trật tự các phần tử của danh sách')

    print('9: Xuất tất cả các số lớn thứ nhì của danh sách')

    print('10: Tính tổng các chữ số của tất cả các số trong danh sách')

    print('11: Đếm số lần xuất hiện của một số trong danh sách')

    print('12: Xuất các số xuất hiện n lần trong danh sách')

    print('13: Xuất các số xuất hiện nhiều lần nhất trong danh sách')

    print('0: Thoát chương trình')

    try:

        action = int(input("Nhập vào số của chức năng muốn chọn: "))

        if action == 0:

            break

        elif type(action) != int:

            print('Xin mời nhập lại')

            action = int(input())

            print('Chọn chức năng muốn thực hiện: ')

            print('1: Xuất tất cả các số lẻ không chia hết cho 5')

            print('2: Xuất tất cả các số Fibonacci')

            print('3: Tìm số nguyên tố lớn nhất')

            print('4: Tìm số Fibonacci bé nhất')

            print('5: Tính trung bình các số lẻ')

            print('6: Tính tích các phần tử là số lẻ không chia hết cho 3 trong mảng')

            print('7: Đổi chỗ 2 phần tử của danh sách, đầu vào là 2 vị trí cần đổi chỗ')

            print('8: Đảo ngược trật tự các phần tử của danh sách')

            print('9: Xuất tất cả các số lớn thứ nhì của danh sách')

            print('10: Tính tổng các chữ số của tất cả các số trong danh sách')

            print('11: Đếm số lần xuất hiện của một số trong danh sách')

            print('12: Xuất các số xuất hiện n lần trong danh sách')

            print('13: Xuất các số xuất hiện nhiều lần nhất trong danh sách')

            print('0: Thoát chương trình')

    except:

        print('Xin mời nhập lại')

        action = 0

    match action:

        case 1:

            system('CLS')

            print('===========================')

            print("Xuất tất cả các số lẻ không chia hết cho 5")

            print('===========================')

            SoLeKhongChiaHetCho5(list)

            print('===========================')

            msvcrt.getch()

        case 2:

            system('CLS')

            print('===========================')

            print("Xuất tất cả các số Fibonacci")

            print('===========================')

            XuatTatCaSoFibonacci(list)

            print('===========================')

            msvcrt.getch()

        case 3:

            system('CLS')

            print('===========================')

            print("Tìm số nguyên tố lớn nhất")

            print('===========================')

            TimSoNguyenToLonNhat(list)

            print('===========================')

            msvcrt.getch()

        case 4:

            system('CLS')

            print('===========================')

            print("Tìm số Fibonacci bé nhất")

            print('===========================')

            TimSoFibonacciBeNhat(list)

            print('===========================')

            msvcrt.getch()

        case 5:

            system('CLS')

            print('===========================')

            print("Tính trung bình các số lẻ")

            print('===========================')

            TinhTrungBinhCacSoLe(list)

            print('===========================')

            msvcrt.getch()

        case 6:

            system('CLS')

            print('===========================')

            print("Tính tích các phần tử là số lẻ không chia hết cho 3 trong mảng")

            print('===========================')

            TichSoLeKhongChiaHetCho3(list)

            print('===========================')

            msvcrt.getch()

        case 7:

            system('CLS')

            print('===========================')

            print("Đổi chỗ 2 phần tử của danh sách, đầu vào là 2 vị trí cần đổi chỗ")

            print('===========================')

            n1 = int(input("Nhập vào vị trí đầu tiên: "))

            n2 = int(input("Nhập vào vị trí cuối cùng: "))

            print('Mảng trước khi đổi: ', list)

            Swap(list, n1, n2)

            print('Mảng sau khi đổi: ', list)

            print('===========================')

            msvcrt.getch()

        case 8:

            system('CLS')

            print('===========================')

            print("Đảo ngược trật tự các phần tử của danh sách")

            print('===========================')

            print('Mảng trước khi đổi: ', list)

            Rev(list)

            print('===========================')

            msvcrt.getch()

        case 9:

            system('CLS')

            print('===========================')

            print("Xuất tất cả các số lớn thứ nhì của danh sách")

            print('===========================')

            print('Mảng hiện hành: ', list)

            TimSoLonThuNhi(list)

            print('===========================')

            msvcrt.getch()

        case 10:

            system('CLS')

            print('===========================')

            print("Tính tổng các chữ số của tất cả các số trong danh sách")

            print('===========================')

            print('Mảng hiện hành: ', list)

            TongCacSo(list)

            print('===========================')

            msvcrt.getch()

        case 11:

            system('CLS')

            print('===========================')

            print("Đếm số lần xuất hiện của một số trong danh sách")

            print('===========================')

            n= int(input("Nhập vào số n: "))

            print('Mảng hiện hành: ', list)

            DemSoLanXuatHien(list, n)

            print('===========================')

            msvcrt.getch()

        case 12:

            system('CLS')

            print('===========================')

            print("Xuất các số xuất hiện n lần trong danh sách")

            print('===========================')

            n = int(input("Nhập vào số n: "))

            print('Mảng hiện hành: ', list)

            XuatSoXuatHienNLan(list, n)

            print('===========================')

            msvcrt.getch()

        case 13:

            system('CLS')

            print('===========================')

            print("Xuất các số xuất hiện nhiều lần nhất trong danh sách")

            print('===========================')

            print('Mảng hiện hành: ', list)

            XuatHienNhieuNhat(list)

            print('===========================')

            msvcrt.getch()