**1911170- Nguyễn Hữu Thành Nam**

**CTK43**

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

[**https://www.w3resource.com/python-exercises/python-basic-exercises.php**](https://www.w3resource.com/python-exercises/python-basic-exercises.php)

**1.**

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!")

**2.**

import sys

print("Phiên bản python")

print (sys.version)

print("Thông tin phiên bản.")

print (sys.version\_info)

**3.**

import datetime

now = datetime.datetime.now()

print ("ngày và giờ : ")

print (now.strftime("%Y-%m-%d %H:%M:%S"))

**4.**

from math import pi

r = float(input ("Nhập bán kính : "))

print ("Diện tích có bán kính là : " + str(pi \* r\*\*2))

**5.**

firtName = input("Nhập họ : ")

lastName = input("Nhập tên : ")

print ("Xin chào  " + firtName + " " + lastName)

**6.**

values = input("Nhập số cách nhau bằng dấu phẩy : ")

list = values.split(",")

tuple = tuple(list)

print('List : ',list)

print('Tuple : ',tuple)

**7.**

filename = input("Nhập tên file: ")

f\_extns = filename.split(".")

print ("Đuôi file là : " + repr(f\_extns[-1]))

**8.**

color\_list = ["Red","Green","White" ,"Black"]

print( "%s %s"%(color\_list[0],color\_list[-1]))

**9.**

dealine = (11,12,2022)

print( "bắt đầu từ : %i / %i / %i"%dealine)

**10.**

a = int(input("nhập 1 số : "))

n1 = int( "%s" % a )

n2 = int( "%s%s" % (a,a) )

n3 = int( "%s%s%s" % (a,a,a) )

print ("Giai thừa là " , n1+n2+n3)

**11.**

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

**12.**

import calendar

y = int(input("Nhập năm : "))

m = int(input("Nhập tháng : "))

print(calendar.month(y, m))

**13.**

print("""

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

This

is a  ....... multi-line

heredoc string --------> example

""")

**14.**

from datetime import date

firtdate = date(2022, 6, 1)

lastdate = date(2022, 8, 15)

deadline = lastdate - firtdate

print(deadline.days)

**15.**

pi = 3.14

r= 6.0

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

print('khối lượng quả cầu: ',V)

**16.**

def match(n):

    if n <= 17:

        return 17 - n

    else:

        return (n - 17) \* 2

print(match(25))

print(match(6))

**17.**

def sad(n):

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

print(sad(1000))

print(sad(900))

print(sad(800))

print(sad(2200))

**18.**

def Tinh(x, y, z):

     sum = x + y + z

     if x == y == z:

      sum = sum \* 3

     return sum

print(Tinh(5, 6, 3))

print(Tinh(6, 6, 6))

**19.**

def new\_string(str):

  if len(str) >= 2 and str[:2] == "Is":

    return str

  return "Is" + str

print(new\_string("Array"))

print(new\_string("IsEmpty"))

**20.**

def larger\_string(str, n):

   result = ""

   for i in range(n):

      result = result + str

   return result

print(larger\_string('abc', 2))

print(larger\_string('.py', 3))

**21.**

num = int(input("Nhập một số: "))

if(num % 2 == 0):

    print('số chẵn')

else :

    print('số lẻ')

**22.**

def find4(nums):

  count = 0

  for num in nums:

    if num == 4:

      count = count + 1

  return count

print(find4([1, 4, 6, 7, 4, 5, 4, 7, 8, 6]))

print(find4([1, 4, 6, 4, 7, 4, 5, 4, 4, 4]))

**23.**

import string

def stringx(str, n):

  length = 2

  if length > len(str):

    length = len(str)

  substr = str[:length]

  result = ""

  for i in range(n):

    result = result + substr

  return result

print(stringx('nam', 2))

print(stringx('p', 6));

**24.**

from ctypes.wintypes import WORD

def check(char):

    work = 'aeiou'

    return char in work

print(check('y'))

print(check('e'))

**25.**

from gettext import find

def finddata(group\_data, n):

   for value in group\_data:

       if n == value:

           return True

   return False

print(finddata([1, 5, 8, 3], 3))

print(finddata([5, 8, 3], -1))

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

**1. Tính:**

**a) (a + b),**

**b) a/b,**

**c) a^b**

a = int(input('Nhập số a :'))

b = int(input('Nhập số b : '))

print(a + b)

print(a/b\*1.0)

print(a\*\*b)

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

chieudai = int(input('Nhập chiều dài :'))

chieurong = int(input('Nhập chiều rộng :'))

dientich = chieudai \* chieurong

print('diện tích hcn : ', dientich)

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

n = int(input('Nhập n: '))

for num in range(2,n+1):

    for i in range(2,num):

        if (num%i==0):

            break

    else:

        print(num)

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

import math

def check\_perfect\_square (m):

    n = int(math.sqrt(m))

    return n \* n == m

def check\_fibo(m):

    check\_perfect\_square (5 \* m \* m + 4)

    check\_perfect\_square (5 \* m \* m - 4)

    if ((n) == True):

        print (n, "là số Fibonacci")

    else:

        print (n , "không phải là số Fibnacci")

n = int(input("Vui lòng nhập giá trị nguyên để kiểm tra số Fibonacci:"))

check\_fibo(n)

**5. Tìm số Fibonacci thứ n (dùng đệ quy và không đệ quy) | 6. Tính tổng n số Fibonacci đầu tiên (dùng đệ quy và không đệ quy)**

from importlib.abc import Finder

import math

# de quy

def Fibonacci(n):

    if n < 0:

        print('số nhập không đúng')

    elif n == 0:

        return 0

    elif n == 1 or n == 2:

        return 1

    else:

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

n = int(input('Nhập n:'))

print('Số Fibonacci thứ n là: ', Fibonacci(n))

def sumOfnFibonacci(m):

    sum  = 0

    for i in range(m):

        sum += Fibonacci(i)

    return sum

m = int(input('Nhập m số Fibonacci:'))

print('Tổng n số fibonacci đầu tiên là: ', sumOfnFibonacci(m))

# khong dung de quy

def Fibonacci\_1(n):

    a = 0

    b = 1

    if n < 0:

        print('Đầu vào không đúng')

    elif n == 0:

        return 0

    elif n == 1:

        return b

    else:

        for i in range(1, n):

            c = a + b

            a = b

            b = c

        return b

print('Số Fibonacci thứ n la: ', Fibonacci\_1(n))

def sumOfnFibonacci\_1(n):

    a = 0

    b = 1

    if n < 0:

        print('Incorrect Input')

    elif n == 0 or n == 1:

        return n

    else:

        s = 0

        while b <= n:

            s += b

            a, b = b, a + b

        return s

p = int(input('Nhập m số Fibonacci:'))

print('Tổng n số Fibonacci đầu tiên là ', sumOfnFibonacci\_1(p))

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

from cmath import sqrt

def canBacHai(n):

    sum = 0

    for i in range(n):

        sum += sqrt(i)

    return sum

n = int(input('Nhập n số nguyên đầu tiên: '))

print('Tổng căn bậc 2 của N số nguyên dương dầu tiên là: ', canBacHai(n))

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

import math

def giaiPTbac2(a,b,c):

    if(a == 0):

        if(b ==0):

            print('Phương trình vô nghiệm');

        else:

            print("Phương trình có một nghiệm: x = ", + (-c / b));

        return;

    delta = b \* b - 4 \* a \* c;

    if (delta > 0):

        x1 = (float)((-b + math.sqrt(delta)) / (2 \* a));

        x2 = (float)((-b - math.sqrt(delta)) / (2 \* a));

        print ("Phương trình có 2 nghiệm là: x1 = ", x1, " và x2 = ", x2);

    elif (delta == 0):

        x1 = (-b / (2 \* a));

        print("Phương trình có nghiệm kép: x1 = x2 = ", x1);

    else:

        print("Phương trình vô nghiệm!");

a = float(input("Nhập hệ số bậc 2, a = "));

b = float(input("Nhập hệ số bậc 1, b = "));

c = float(input("Nhập hằng số tự do, c = "));

giaiPTbac2(a, b, c)

**9. Tính n!**

a = int(input('Nhập 1 số n : '))

def GiaiThua(a):

    if a ==0 :

        return 1

    return a \* GiaiThua(a -1)

print(GiaiThua(a))

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

n = int(input('Nhập số hàng :'))

for hang in range(1, n+1):

    for cot in range(1, hang+1):

        print('\*' , end=' ')

    print()

**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.**

x = int(input('Nhập số giây : '))

gio = x // 3600

x = x % 3600

phut = x // 60

x = x % 60

print('kết quả : ', gio,'giờ : ',phut,'phút : ',x , 'giây')

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

**a) Xuất 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

firts\_list = [1,3,5,6,9,8,7,45,10,23,4,9,9,5,2,45,1]

list\_prime = []

list\_fibonacci = []

list\_odd = []

def isOdd(x):

    return (x & 1)

def notDivisibleBy4():

    new\_list = list(filter(lambda x: x % 5 != 0 and isOdd(x), firts\_list))

    print('Số lẻ không chia hết cho 5 là : ', new\_list)

def check\_perfect\_square(x):

    n = int(math.sqrt(x))

    return n \* n == x

def check\_fibonacci(a):

    return check\_perfect\_square(5 \* a \* a - 4) or check\_perfect\_square(5 \* a \* a + 4)

def printFibonacci():

    print('Số Fibonacci có trong mảng la:', end=' ')

    for i in firts\_list:

        if check\_fibonacci(i) == True:

            list\_fibonacci.append(i)

    print(list\_fibonacci , end=' ')

def check\_prime\_number(n):

    flag = 1

    if n < 2:

        flag = 0

        return flag

    for i in range(2, n):

        if n % i == 0:

            flag = 0

            break

    return flag

def check\_prime\_in\_list():

    print('\nSố nguyên tố có trong mảng:', end=' ')

    for i in firts\_list:

        if check\_prime\_number(i) == 1:

            list\_prime.append(i)

    print(list\_prime)

def find\_prime\_max():

    print('Số nguyên tố lớn nhất trong mảng:', max(list\_prime))

def find\_fib\_min():

    print('Số Fibonacci nhỏ nhất là: ', min(list\_fibonacci))

def average\_odd():

    sum = 0

    average = 0

    for i in firts\_list:

        if isOdd(i) == 1:

            list\_odd.append(i)

    for i in list\_odd:

        sum += i

    average = sum / len(list\_odd)

    return average

def oddNumberNotDivisibleBy3():

    result = 1

    odd\_Number\_Not\_DivisibleBy3 = list(filter(lambda x: x % 3 != 0, list\_odd))

    for i in odd\_Number\_Not\_DivisibleBy3:

        result \*= i

    return result

def ChangeLocation(number1, number2):

    firts\_list[number1], firts\_list[number2] = firts\_list[number2], firts\_list[number1]

    return firts\_list

def max2Number():

    current\_list\_sort = sorted(firts\_list)

    print('Số lớn thứ 2 trong danh sách là : ', current\_list\_sort[-2])

def sumLastNumber():

    sum = 0

    for i in firts\_list:

        sum = sum + i % 10

        i = int(i / 10)

    return sum

def countAppearsOfNumbers():

    n = int(input('Nhập sốc cần đếm số lần xuất hiện '))

    print('Số lần xuất hiện số {0} là: {1} lần'.format(n, firts\_list.count(n) ))

def printNumberAppears():

    n = int(input('Nhập số lần xuất hiện của một số:'))

    for i in firts\_list:

        if n == firts\_list.count(i):

            print('Số xuất hiện số {0} lần là: {1}'.format(n, i))

            break

def countMaxAppearsOfNumbers():

    counter = 0

    num = firts\_list[0]

    for i in firts\_list:

        curr\_frequency = firts\_list.count(i)

        if curr\_frequency > counter:

            counter = curr\_frequency

            num = i

    return num

notDivisibleBy4()

printFibonacci()

check\_prime\_in\_list()

find\_prime\_max()

find\_fib\_min()

print('Trung bình các số lẻ :', average\_odd(), end=' ')

print('\nCác phần tử không chia hết cho 3: ', oddNumberNotDivisibleBy3())

number1 = int(input('Nhập số hạng đầu: '))

number2 = int(input('nhập số hạng thứ 2: '))

print('Danh sach ban dau', firts\_list)

print('Danh sach sau khi swap', ChangeLocation(number1 - 1, number2 - 1))

firts\_list.reverse()

print('Danh sách sau khi đảo ngược là :', firts\_list)

print('Số lớn thứ nhì trong danh sách',max2Number())

print('Tổng chữ số còn lại ', sumLastNumber())

print('số lần xuất hiện của một số trong danh sách',countAppearsOfNumbers())

print('Số xuất hiện n lần trong danh sách',printNumberAppears())

print('Số xuất hiện nhiều nhất là:',countMaxAppearsOfNumbers())