LAB 01

Bài 1:

1/ Write a Python program to display the current date and time.

import datetime

now = datetime.datetime.now()

print ("Ngay gio hien tai : ")

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

2/ Write a Python program that calculates the area of a circle based on the radius entered by the user

from math import pi

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

print ("The area of the circle with radius " + str(r) + " is: " + str(pi \* r\*\*2))

3/ Write a Python program that accepts the user's first and last name and prints them in reverse order with a space between them

Ho = str(input('Nhap ho cua ban: '))

Ten = str(input('Nhap ten cua ban: '))

print('Xin chao: '+ Ten +' '+Ho)

4/ Write a Python program to display the first and last colors from the following list.

color\_list = ['Xanh','Do','Tim','Vang']

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

5/ Write a Python program to display the examination schedule. (extract the date from exam\_st\_date)

thoigian = (9,8,2023)

print('Thoi gian thi bat dau tu ngay: %i/%i/%i'%thoigian)

6/ Write a Python program that accepts an integer (n) and computes the value of n+nn+nnn.

x = int(input('Nhap so nguyen bat ky: '))

x1 = int('%s'%x)

x2 = int('%s%s'%(x,x))

x3 = int('%s%s%s'%(x,x,x))

print(x1+x2+x3)

7/ Write a Python program that prints the calendar for a given month and year.

import calendar

y = int(input('Nhap thang: '))

m = int(input('Nhap nam: '))

print(calendar.month(m, y))

8/ Write a Python program to print all even numbers from a given list of numbers in the same order and stop printing any after 237 in the sequence.

numbers = [

    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,

    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,

    815, 67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445, 742, 717,

    958,743, 527

    ]

for x in numbers:

    if x == 237:

        print(x)

        break

    elif x % 2 == 0:

        print (x)

9/Write a Python program to calculate the number of days between two dates.

from datetime import date

f\_date  = date(2023, 6, 7)

l\_date  = date(2023, 6, 29)

delta = l\_date - f\_date

print(delta.days)

10/ Write a Python program to get the volume of a sphere with radius six.

pi = 3.14

r = 6.0

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

print('The tich cua hinh cau la: ', V)

11/ Write a Python program to solve (x + y) \* (x + y).

x = 6

y = 7

ketqua = x\*\*2 + 2\*x\*y + y\*2

print('({0} + {1}) ^ 2 = {2}'.format(x, y, ketqua))

12/ Write a Python program to calculate the difference between a given number and 17. If the number is greater than 17, return twice the absolute difference.

def So(n):

    if n <= 17:

        return 17 - n

    else:

        return (n-17)\*2

print(So(20))

print(So(10))

13/ Write a Python program to sum two given integers. However, if the sum is between 15 and 20 it will return 20.

def tong(x, y):

    tong = x + y

    if tong in range(15, 20):

        return 20

    else:

        return tong

print(tong(10, 12))

print(tong(6, 10))

print(tong(10, 8))

14/ Write a Python program to find out the number of CPUs used.

import multiprocessing

print(multiprocessing.cpu\_count())

15/ Write a Python program that returns a string that is n (non-negative integer) copies of a given string

def chuoilon(text, n):

    chuoi = ''

    for i in range(n):

        chuoi = chuoi + text

    return chuoi

print(chuoilon('abc', 5))

print(chuoilon('123', 4))

16/ Write a Python program to create a histogram from a given list of integers.

def bieudo(items):

    for n in items:

        output = ''

        times = n

        while(times > 0):

            output += '\*'

            times = times - 1

        print(output)

bieudo([1, 5, 6, 4])

17/ Write a Python program to print the following string in a specific format (see the output).

print(""" Twinkle, twinkle, little star,

    How I wonder what you are!

        Up above the world so high,

        Like a diamond in the sky.

Twinkle, twinkle, little star,

    How I wonder what you are """)

18/ Write a Python program that accepts a sequence of comma-separated numbers from the user and generates a list and a tuple of those numbers.

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

list = values.split(",")

tuple = tuple(list)

print('List : ',list)

print('Tuple : ',tuple)

19/ Write a Python program to display the first and last colors from the following list.

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

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

20/ Write a Python program to print the following 'here document'.

print("""

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

This

is a  ....... multi-line

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

""")

21/ Write a Python program to test whether a number is within 100 of 1000 or 2000

def near\_thousand(n):

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

print(near\_thousand(1000))

print(near\_thousand(900))

print(near\_thousand(800))

print(near\_thousand(2200))

22/ Write a Python program to count the number 4 in a given list.

nums = [4 , 1, 4, 5, 4, 1]

def list\_count\_4(nums):

  count = 0

  for num in nums:

    if num == 4:

      count = count + 1

  return count

print(list\_count\_4(nums))

23/Write a Python program that concatenates all elements in a list into a string and returns it.

def concatenate\_list\_data(list):

    result= ''

    for element in list:

        result += str(element)

    return result

print(concatenate\_list\_data(list))

24/ Write a Python program to sum three given integers. However, if two values are equal, the sum will be zero

def sum\_three(x, y, z):

    if x == y or y == z or x==z:

        sum = 0

    else:

        sum = x + y + z

    return sum

print(sum\_three(2, 1, 2))

print(sum\_three(3, 2, 2))

print(sum\_three(2, 2, 2))

print(sum\_three(1, 2, 3))

25/ Write a Python program to get OS name, platform and release information.

import platform

import os

print("Name of the operating system:",os.name)

print("\nName of the OS system:",platform.system())

print("\nVersion of the operating system:",platform.release())

Bài 2:

1. Tính:

a) (a + b)

a = int(input('Nhap so thu nhat: '))

b = int(input('Nhap so thu hai: ')))

c = a + b

print('Tong hai so la: ', c)

b) a/b

a = int(input('Nhap so tho thu nhat: '))

b = int(input('Nhap so thu hai: '))

c = float(a/b)

print('Thuong cua so la: ', c)

c) ab

a = int(input('Nhap so tho thu nhat: '))

b = int(input('Nhap so thu hai: '))

c = a\*\*b

print('Luy thua cua hai la: ', c)

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

a = int(input('Nhap chieu dai hinh chu nhat: '))

b = 20

c = a \* b

print('Dien tich hinh chu nhat khi co chieu rong la {0}: '.format(b), c)

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

lower = int(input("Nhap gia tri toi thieu: "))

upper = int(input("Nhap gia tri toi da: "))

for number in range(lower,upper+1):

    if number>1:

        for i in range(2,number):

            if (number%i)==0:

                break

        else:

            print(number)

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

n = int(input("Nhập vào một số: "))

phi = (1 + 5 \*\* 0.5) / 2

psi = (1 - 5 \*\* 0.5) / 2

F\_n = (phi \*\* n - psi \*\* n) / (5 \*\* 0.5)

if F\_n.is\_integer():

    print(n, "là số Fibonacci")

else:

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

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

Dùng đệ quy

def fibonacci(n):

    if n == 0 or n == 1:

        return 1

    else:

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

Không dùng đệ quy

n = int(input("Nhập vào một số nguyên dương: "))

f0 = 0

f1 = 1

fn = 1

if n < 0:

    print("Số nhập vào không hợp lệ")

elif n == 0 or n == 1:

    print("Số Fibonacci thứ", n, "là", n)

else:

    for i in range(2, n + 1):

        f0 = f1

        f1 = fn

        fn = f0 + f1

    print("Số Fibonacci thứ", n, "là", fn)

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

Dùng đệ quy

def fibonacci(n):

    if n == 0 or n == 1:

        return n

    else:

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

def sum\_fibonacci(n):

    total = 0

    for i in range(0, n + 1):

        total += fibonacci(i)

    return total

n = int(input("Nhập vào một số nguyên dương: "))

print("Tổng", n, "số Fibonacci đầu tiên là:", sum\_fibonacci(n))

Không dùng đệ quy

n = int(input("Nhập vào một số nguyên dương: "))

f0 = 0

f1 = 1

fn = 1

total = 0

if n < 0:

    print("Số nhập vào không hợp lệ")

elif n == 0 or n == 1:

    print("Tổng", n, "số Fibonacci đầu tiên là:", n)

else:

    for i in range(0, n + 1):

        total += fn

        f0 = f1

        f1 = fn

        fn = f0 + f1

    print("Tổng", n, "số Fibonacci đầu tiên là:", total)

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

n = int(input("Nhập vào một số nguyên dương: "))

import math

total = 0

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

    total += math.sqrt(i)

print("Tổng căn bậc 2 của", n, "số nguyên đầu tiên là:", total)

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

a = float(input("Nhập vào a: "))

b = float(input("Nhập vào b: "))

c = float(input("Nhập vào c: "))

import math

delta = b\*\*2 - 4\*a\*c

if delta < 0:

    print("Phương trình vô nghiệm")

elif delta == 0:

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

else:

    print("Phương trình có hai nghiệm phân biệt:")

    print("x1 =", (-b + math.sqrt(delta)) / (2\*a))

    print("x2 =", (-b - math.sqrt(delta)) / (2\*a))

9. Tính n!

n = int(input("Nhập vào một số nguyên không âm: "))

result = 1

if n < 0:

    print("Số nhập vào không hợp lệ")

else:

    for i in range(1, n + 1):

        result \*= i

    print(n, "! =", result)

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

n = int(input("Nhập vào số hàng (cột) của tam giác: "))

for i in range(0, n):

    for j in range(0, n - 1):

        print(" ", end="")

    for j in range(0, i+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.

seconds = int(input("Nhập số giây: "))

hours = seconds // 3600

minutes = (seconds % 3600) // 60

seconds = (seconds % 3600) % 60

print(f"{hours}:{minutes}:{seconds}")

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

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

array = [10, 15, 21, 25, 32, 37, 41, 45, 51, 55]

answer = []

for num in array:

    if num % 2 != 0 and num % 5 != 0:

        answer.append(num)

print("Các số lẻ không chia hết cho 5 trong mảng là: ", answer)

b) Xuất tất cả các số Fibonacci

import math

def is\_fibonacci(n):

    x = (5\*n\*n + 4)

    y = (5\*n\*n - 4)

    if math.sqrt(x) == int(math.sqrt(x)) or math.sqrt(y) == int(math.sqrt(y)):

        return True

    else:

        return False

array = [10, 15, 21, 25, 32, 37, 41, 45, 51, 55]

answer = []

for num in array:

    if is\_fibonacci(num):

        answer.append(num)

print("Các số Fibonacci trong mảng là: ", answer)

c) Tìm số nguyên tố lớn nhất

*# Khai báo một mảng số nguyên*

arr = [10, 15, 21, 25, 32, 37, 41, 45, 51, 55]

*# Tạo một biến để lưu trữ số nguyên tố lớn nhất*

max\_prime = -1

*# Viết một hàm để kiểm tra xem một số có phải là số nguyên tố hay không*

def is\_prime(n):

*# Nếu n < 2 thì không phải là số nguyên tố*

    if n < 2:

        return False

*# Nếu n == 2 thì là số nguyên tố*

    if n == 2:

        return True

*# Nếu n là số chẵn thì không phải là số nguyên tố*

    if n % 2 == 0:

        return False

*# Kiểm tra các số lẻ từ 3 đến căn bậc hai của n*

    for i in range(3, int(n\*\*0.5) + 1, 2):

*# Nếu n chia hết cho i thì không phải là số nguyên tố*

        if n % i == 0:

            return False

*# Nếu không có số nào chia hết cho n thì là số nguyên tố*

    return True

*# Duyệt qua các phần tử trong mảng*

for num in arr:

*# Gọi hàm kiểm tra trên mỗi phần tử*

    if is\_prime(num):

*# Nếu phần tử là số nguyên tố và lớn hơn max\_prime*

        if num > max\_prime:

*# Cập nhật max\_prime bằng phần tử đó*

            max\_prime = num

*# In kết quả ra màn hình*

print("Số nguyên tố lớn nhất trong mảng là: ", max\_prime)

d) Tìm số Fibonacci bé nhất

*# Hàm kiểm tra số Fibonacci*

def is\_fibonacci(n):

*# Sử dụng công thức dạng tường minh của dãy Fibonacci*

*# Nếu n là số chính phương của 5n^2 + 4 hoặc 5n^2 - 4, n là số Fibonacci*

    return (5 \* n\*\*2 + 4)\*\*0.5 % 1 == 0 or (5 \* n\*\*2 - 4)\*\*0.5 % 1 == 0

*# Hàm tìm số Fibonacci bé nhất trong mảng*

def min\_fibonacci(arr):

*# Khởi tạo biến lưu kết quả, ban đầu bằng None*

    result = None

*# Duyệt qua từng phần tử trong mảng*

    for num in arr:

*# Nếu phần tử là số Fibonacci và bé hơn kết quả hiện tại hoặc kết quả chưa có*

        if is\_fibonacci(num) and (result is None or num < result):

*# Cập nhật kết quả bằng phần tử đó*

            result = num

*# Trả về kết quả*

    return result

e) Tính trung bình các số lẻ

array = [11, 12, 40, 51, 81, 59, 53, 20, 5]

import statistics

oddnumber = list(filter(lambda x: x % 2 == 1, array))

oddavg = statistics.mean(oddnumber)

print('Trung bình của các số lẻ: ',oddavg)

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

def Tich\_so\_le(array):

    m = 1

    for x in array:

        if x % 2 != 0 and x % 3 != 0:

            m = m \* x

    return m

array = [2, 5, 7, 9, 11, 13, 15, 17]

print('Tich cac phan tu la so le khong chia het cho 3 trong mang la: ', Tich\_so\_le(array))

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

def Doi\_cho(array, slot1, slot2):

    slot= array[slot1]

    array[slot1] = array[slot2]

    array[slot2] = slot

    return array

array = [1,5,6,2,89,10,44]

slot1 = 1

slot2 = 6

print('Danh sach khi doi cho la: ',Doi\_cho(array, slot1, slot2))

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

array = [1, 6, 7, 8, 9, 10]

def Dao\_nguoc(array):

    if not array:

        return array

    ds\_phantucuoi = [array[-1]]

    tatca\_phantu = array[:-1]

    return ds\_phantucuoi + Dao\_nguoc(tatca\_phantu)

print('Dao nguoc phan tu trong mang la: ',Dao\_nguoc(array))

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

array = [1, 7, 8, 1, 6 , 5, 7]

def Tatcasolonthuhai(array):

    array.sort(reverse=True)

    return array[1]

print('Tat ca cac so lon thu hai la: ', Tatcasolonthuhai(array))

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

def Tong\_cac\_chu\_so(n):

  if n < 10:

    return n

  else:

    m = n % 10

    return m + Tong\_cac\_chu\_so(n // 10)

array = [1, 3, 2, 22, 11]

tong = 0

for num in array:

  tong += Tong\_cac\_chu\_so(num)

print('Tong cac chu so la: ',tong)

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

array = [1, 2, 3, 4, 5, 1, 2, 1]

def count\_number(lst, num):

  count = 0

  for x in lst:

    if x == num:

      count += 1

  return count

num = 1

tong = count\_number(array, num)

print("Số lần xuất hiện của {0} trong danh sách là {1}".format(num, tong))

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

def Xuat\_hien\_n\_lan(array, n):

  seen = set()

  for num in array:

    if num not in seen and array.count(num) == n:

      print(num)

      seen.add(num)

array = [1, 0, 3, 2, 5, 6, 2, 3, 1]

n = 2

Xuat\_hien\_n\_lan(array, n)

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

array = [int(input("Nhap mot so: ")) for i in range(10)]

from collections import Counter

c = Counter(array)

max\_count = max(c.values())

def Xuat\_cac\_so(array, max\_count):

    tong = []

    for num in array:

        if c[num] == max\_count:

            tong.append(num)

    return tong

print("Cac so xuat hien nhieu nhat trong danh sach:", Xuat\_cac\_so(array, max\_count))

Lab2

Bài 1:

from datetime import datetime

class SinhVien:

    truong = "Đại học Đà Lạt"

    def \_\_init\_\_(self, maSo: int, hoTen: str, ngaySinh: datetime):

        self.\_\_maSo = maSo

        self.\_\_hoTen = hoTen

        self.\_\_ngaySinh = ngaySinh

    @property

    def maSo(self):

        return self.\_\_maSo

    @maSo.setter

    def maSo(self, maso):

        if self.laMaSoHopLe(maso):

            self.\_\_maSo = maso

    @property

    def hoTen(self):

        return self.\_\_hoTen

    @hoTen.setter

    def hoTen(self, hoten):

        self.\_\_hoTen = hoten

    @property

    def ngaySinh(self):

        return self.\_\_ngaySinh

    @ngaySinh.setter

    def ngaySinh(self, ngaySinh):

        self.\_\_ngaySinh = ngaySinh

    @staticmethod

    def laMaSoHopLe(maso: int):

        return len(str(maso)) == 7

    @classmethod

    def doiTenTruong(self, tenmoi):

        self.truong = tenmoi

    def \_\_str\_\_(self):

        return f"{self.\_\_maSo}\t{self.\_\_hoTen}\t{self.\_\_ngaySinh}"

    def xuat(self):

        print(f"{self.\_\_maSo}\t{self.\_\_hoTen}\t{self.\_\_ngaySinh}")

class DanhSachSV:

    def \_\_init\_\_(self):

        self.dssv = []

    def themSinhVien(self, sv: SinhVien):

        self.dssv.append(sv)

    def xuat(self):

        for sv in self.dssv:

            print(sv)

    def timSvTheoMssv(self, mssv: int):

        return [sv for sv in self.dssv if sv.maSo == mssv]

    def timVTSvTheoMssv(self, mssv: int):

        for i in range(len(self.dssv)):

            if self.dssv[i].maSo == mssv:

                return i

        return -1

    def xoaSvTheoMssv(self, maSo: int) -> bool:

        vt = self.timVTSvTheoMssv(maSo)

        if vt != -1:

            del self.dssv[vt]

            return True

        else:

            return False

    def timSvTheoTen(self, ten: str):

        return [sv for sv in self.dssv if sv.hoTen.rsplit(' ', 1)[-1] == ten]

    def timSvSinhTruocNgay(self, ngay: datetime):

        return [sv for sv in self.dssv if sv.ngaySinh < ngay]

sv1 = SinhVien(1234511,'Nguyen Van A',datetime.strptime('12/04/2000','%d/%m/%Y'))

sv2 = SinhVien(1234512,'Nguyen Van B',datetime.strptime('13/04/2000','%d/%m/%Y'))

sv3 = SinhVien(1234513,'Nguyen Van C',datetime.strptime('14/04/2000','%d/%m/%Y'))

ds=DanhSachSV()

ds.themSinhVien(sv1)

ds.themSinhVien(sv2)

ds.themSinhVien(sv3)

ds.xuat()

msTim = int(input("Nhập vào mã số muốn tìm: "))

kq = ds.timSvTheoMssv(msTim)

print("Đã tìm thấy sinh viên có mã số: ", msTim)

for sv in kq:

    sv.xuat()

print(ds.timVTSvTheoMssv(msTim))

msXoa = int(input("Nhập vào mã số muốn xóa: "))

ds.xoaSvTheoMssv(msXoa)

ds.xuat()

ten = input("Nhập tên muốn tìm: ")

kq = ds.timSvTheoTen(ten)

print("Đã tìm thấy sinh viên có tên: ", ten)

for sv in kq:

    sv.xuat()

ngay = datetime.strptime("15/06/2000", "%d/%m/%Y")

kqNgay = ds.timSvSinhTruocNgay(ngay)

print("Đã tìm thấy ngày sinh trước 15/06/2000: ")

for sv in kqNgay:

    sv.xuat()

Bài 2: