**TUẦN 1 &2 : ÔN TẬP PYTHON**

**Bài tập thực hành tuần 1**

1. Viết các lệnh xuất các biến sau:

# Kiểu dữ liệu số

x = 10 #Python hiểu đây là số nguyên

y = 10.5 #Python hiểu đây là số thực

z = 5 #Python hiểu đây là số nguyên

t = x + z #Python hiểu t là số nguyên

t = x + y #Python hiểu t là số thực

# Kết luận

# số nguyên (+,-,\*,/) số nguyên = số nguyên

# số nguyên (+,-,\*,/) số thực = số thực

# số thực (+,-,\*,/) số thực = số thực

# Kiểu thực

a = 1.4

a = 31.4e-10

# Kiểu số phức: a +(-)ib

z = 2 + 7j

# Kiểu số dạng bát phân (8), thập lục phân (16) hoặc nhị phân (2)

x = 0b100100

y = 0xf400

# True và False

u = True # u -> 1

v = False # v -> 0

# Kiểu dữ liệu chuỗi

a = "Hello" # Python hiểu a là kiểu chuỗi

print(a)

b = 'World' # Python hiểu b là kiểu chuỗi

print(b)

# Sử dụng \n để xuống dòng

print("Python programming \n for Data Analysis")

# Sử dụng f để hiện thị giá trị

print(f'Giá trị biến a là: {a}')

print("Giá trị biến a là:",a)

1. Các kiểu dữ liệu List, Tuple

# Kiểu dữ liệu danh sách: LIST

# Đây kiểu dữ liệu có khả năng chứa nhiều phần tử khác nhau

# kể cả khác kiểu dữ liệu

listA = ['red',100,'blue',5.5,"yellow",2+4j]

# Có đánh chỉ chỉ số dương lẫn âm

# listA đánh chỉ số dương: 0,1,2,3,4,5

# listA đánh chỉ số âm: -6,-5,-4,-3,-2,-1

print(listA[2])

print(listA[-2])

# Kết quả trả về của việc lấy theo một dãy các

# phần tử theo chỉ sô

# trong danh sách cũng là danh sách

# --> Cái này cũng được gọi là kỹ thuật SLICE

subListA = listA[2:5]

print(subListA)

print(subListA[1])

print(listA[1:-2])

# Đếm số lượng phần tử của danh sách

soluongpt = len(listA)

print(soluongpt)

print(len(listA))

# Lấy phần tử cuối cùng trong list

print(listA[-1]) # Lấy theo chỉ số âm

print(listA[soluongpt-1]) # Lấy theo chỉ số dương

1. Các cấu trúc điều khiển
2. Viết chương trình giải phương trình bậc 1 ax + b = 0, trong đó, a, b là các số thưc được nhập từ bàn phím
3. Tính tổng s = 0 + 1 + ... + n, trong đó n là số tự nhiên nhập từ bàn phím. Yêu cầu viết bằng for…range, while.
4. Dùng for…in:
5. Liệt kê danh sách các ký tự có trong 1 từ, với từ được nhập vào từ bàn phím
6. Liệt kê danh sách các giá trị có trong 1 danh sách, với danh sách được nhập vào từ bàn phím.
7. In ra màn hình các số chẵn của 1 mảng được nhập vào từ bàn phím.
8. Nhập vào một chuỗi ký tự, nhập ký tự cần tìm. In ra ký tự đó xuất hiện bao nhiêu lần tại vị trí nào và cho biết chiều dài chuỗi ký tự nhập vào.
9. Viết chương trình tính tiền điện hàng tháng của một hộ gia đình. Người sử dụng sẽ nhập số kWh từ bàn phím (là một số nguyên). Tiền điện được tính như sau: Nếu số KhW nhỏ hơn bằng 100 thì đơn giá là 2000 VNĐ. Nếu số KWh vượt 100, thì đơn giá cho mỗi KWh sẽ được cộng dồn thêm 100 VNĐ

Ví dụ:

    Số Kwh = 90 thì tổng tiền = 180000 = 90 \* 2000

    Số KWh = 101 thì tổng tiền = 202100 = 100 \* 2000 + (2000 + 100)

    Số Kwh = 102 thì tổng tiền = 204300 = 100\*2000 + (2000+100) + (2000+100+100)

1. Viết hàm tính giá trị bình phương của một số Sau đó, gọi hàm.
2. Viết hàm tính các phép tính số học (+,-,\*,/) của hai số được nhập vào từ bàn phím.
3. Hãy nhập một số tự nhiên từ bàn phím sau đó, xuất màn hình thông báo số đó có phải là số nguyên tố hay không
4. Hãy nhập từ bàn phím số tự nhiên n và xuất ra màn hình:

    1. Các số nguyên tố nhỏ hơn n

2. Xuất ra tổng các số nguyên tố nhỏ hơn n

9) Hãy viết chương trình đọc file in.txt và hiển thị ra màn hình nội dung từng dòng trong file đó

10) Hãy viết chương trình xuất ra file out.txt các số chẵn nhỏ số n được nhập từ bàn phím. Biết rằng mỗi dòng thì lưu 1 số

11) Đọc file dữ liệu num\_input.csv Hãy lưu trữ các số trong file vào chương trình. Sau đó, xuất ra màn hình tổng của các số trong file.

12) Tạo ra một list mới mang tên myUpper chứa các chuỗi trong myCharacter nhưng được viết hoa

13) Hãy tạo danh sách myGreater5 là các số lớn hơn 5 được lấy từ myNumber

14) Hãy tạo danh sách mySquare là các số bình phương của các giá trị được lấy từ myNumber

15) Hãy tạo danh sách myEven là các số chẵn trong myNumber

16) Hãy tạo danh sách myEvenSquare là các bình phương các số chẵn trong myNumber

17) Hãy định nghĩa hàm tính lập phương của một số (Khi cần cài đặt nhanh chóng các hàm mà

không cần tốn thời gian định nghĩa hàm cụ thể ta có thể sử dụng dạng hàm nặc danh (không có tên và dùng ngay), được gọi là ngôn ngữ lambda)

ketqua = lambda x: x\*\*3

print(ketqua(10))

17) Hãy lấy ra danh sách các số là số chẵn

myEven = list(filter(lambda x: (x%2 == 0), myNumber))

print(myEven)

18) Hãy lấy ra danh sách myChanChuc các số là số chẵn và chia hết cho 5

myChanChuc = list(filter(lambda x: (x%2 == 0 and x%5 == 0), myNumber))

print(myChanChuc)

(Chúng ta dùng lambda trong filter khi điều kiện lọc là hàm phức tạp hoặc trên tập dự liệu có thể lặp (iterator) còn nếu hàm đơn giản và dùng trên list thì chúng ta có thể dùng list comprhension)

Tương tự chúng ta có thể sử dụng map như filter. Hãy tạo ra danh sách myLapPhuong lập phương các số trong myNumber

myLapPhuong = list(map(lambda x: x\*\*3, myNumber))

print(myLapPhuong)

19) Cho một danh sách bao gồm các chuỗi chứa các chữ số.

Ví dụ: A = [‘3’, ‘27’,’5’,’123’,’9’,’1’]. Viết hàm sắp xếp chuỗi tăng dần theo hai kiểu (string compare hoặc integer compare)

Ví dụ:

* string compare ( so sánh theo thứ tự từ điển)

Danh sách A sau khi được sắp xếp [‘1’, ‘123’,’27’,’3’,’5’,’9’]

* integer compare (so sánh theo số nguyên)

Danh sách A sau khi được sắp xếp [‘1’, ‘3’,’5’,’9’,’27’,’123’]

* Gợi ý dùng sorted() và key function

20) Viết chương trình in list sau khi đã xóa số tại vị trí thứ 1, thứ 2, thứ 3, thứ 6 trong [12,24,35,70,88,120,155].

\* Dùng list comprehension  và enumerate()

21) Xóa các phần tử trùng nhau trong list.

\* Yêu cầu không dùng hàm của kiểu dữ liệu dictionary và set

Ví dụ:

Input A = [1, 2, 3, 1, 2, 5, 6, 7, 8]

Output B = [1, 2, 3, 5, 6, 7, 8]

22) Đếm số lần xuất hiện của các phần tử trong list.

\* Yêu cầu không dùng hàm của kiểu dữ liệu dictionary và module collections

Ví dụ: A = [1,1,1,1,2,2,2,2,3,3,4,5,5]

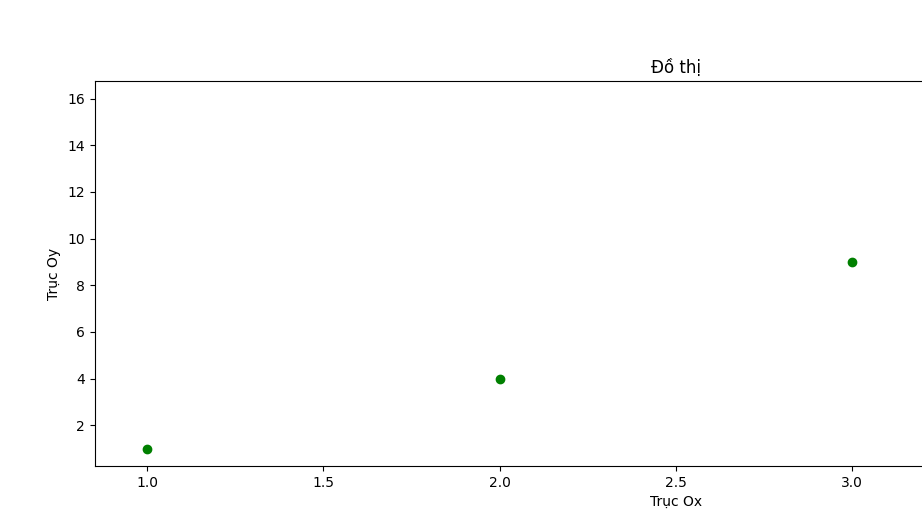
Output: 1: 4, 2: 4, 3:2, 4:1, 5:2

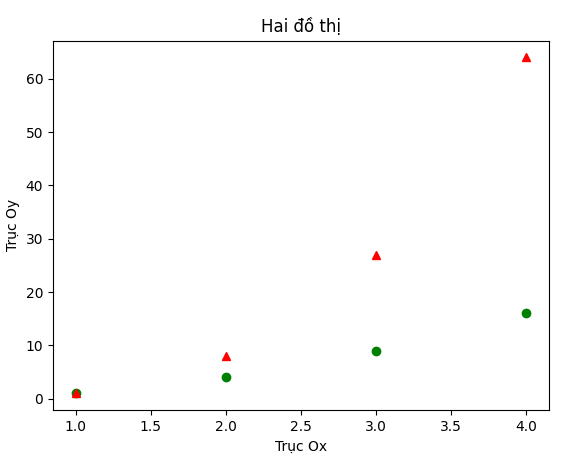
Bài thực hành tuần 1 – phần 2:

import matplotlib.pyplot as plt

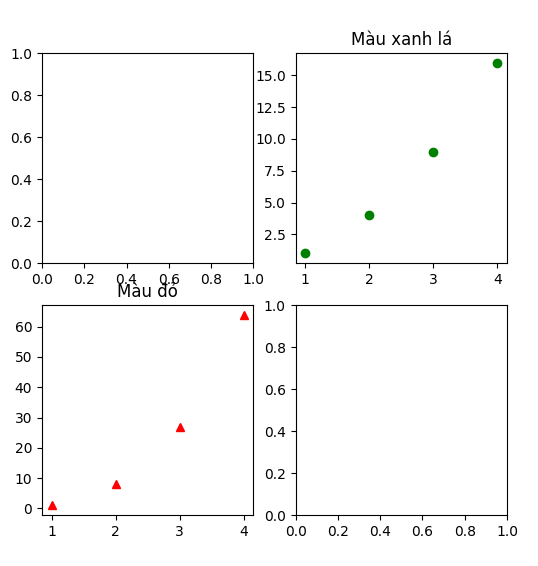
import numpy as np

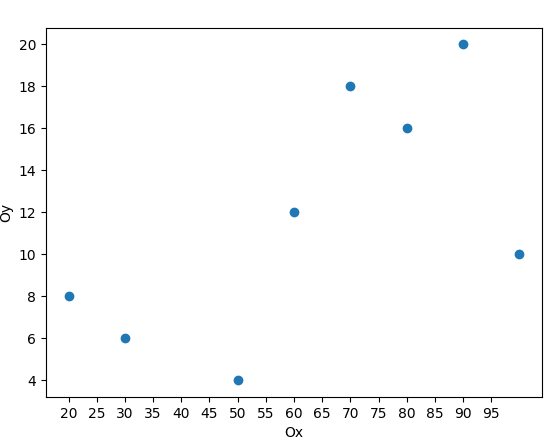
1. Vẽ biểu đồ theo các dạng sau:



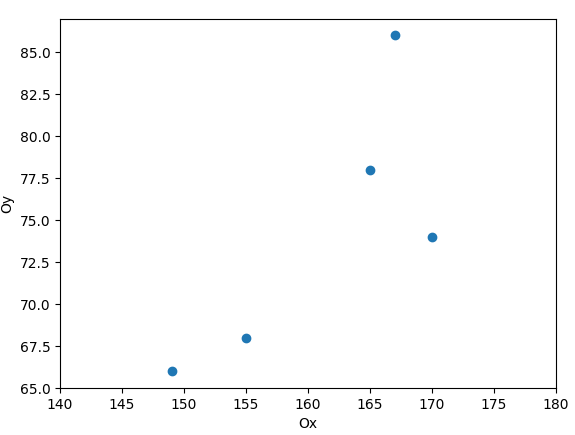


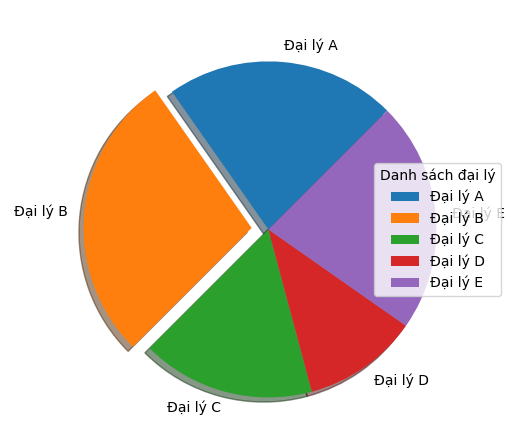
#Sinh viên tự viết 2 dòng 1 cột và hiện thị tương ứng

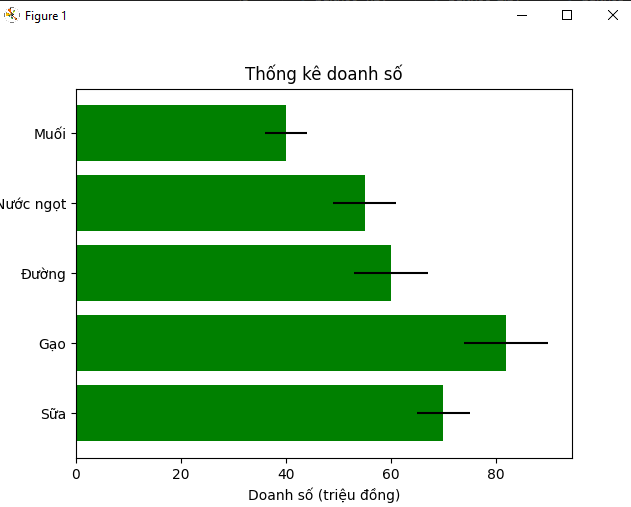


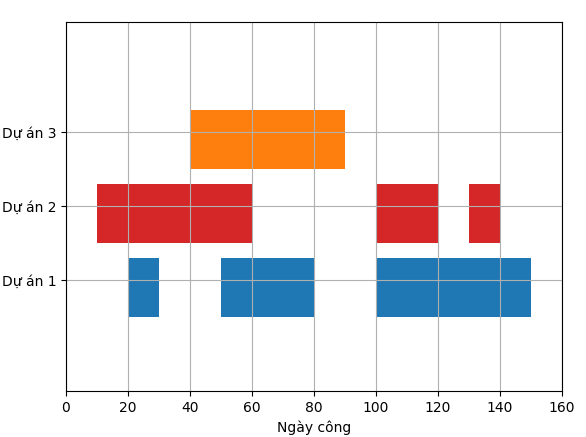


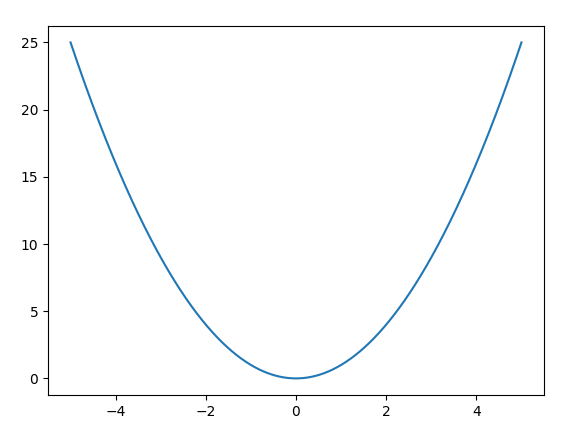
#Sinh viên tự viết yticks sao cho khoảng nhảy là 1

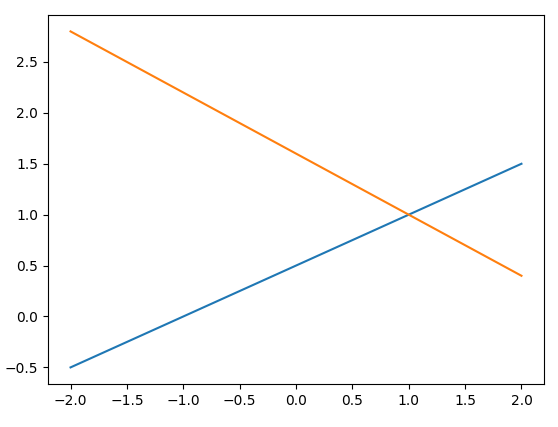


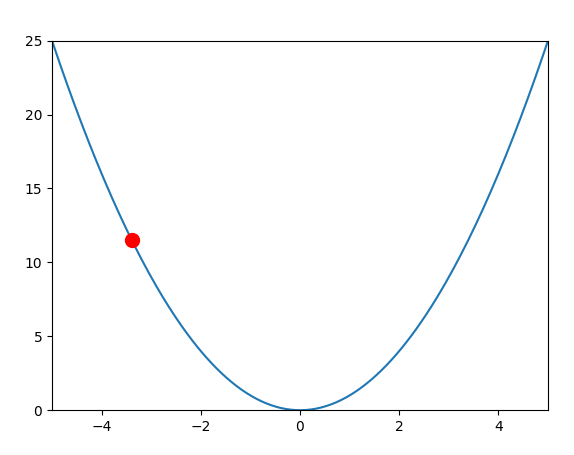


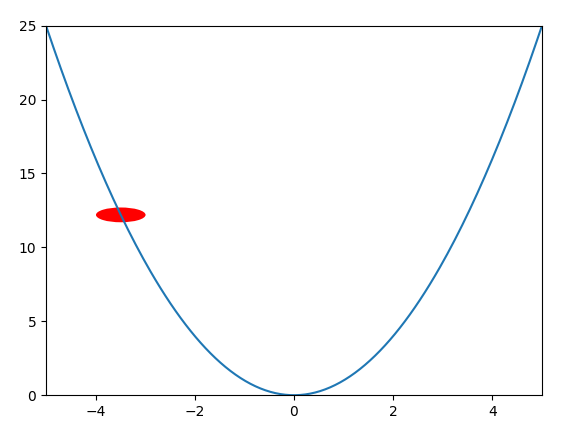


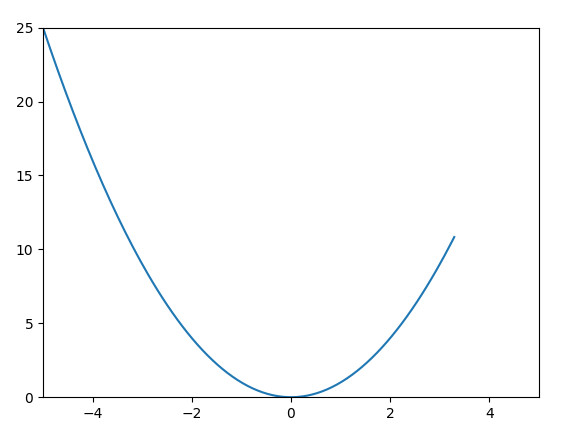












Bài 10\_1, Bài 12 không có đồ thị

**BÀI TẬP 3: CLASS**

Tạo file Rectangle.py

Cài đặt lớp hình chữ nhật theo thiết kế

Có 2 fields (thuộc tính): width, length

Có các phương thức:

- tính diện tích (area)

- tính chu vi (perimeter)

- hiện thị cơ bản (display)

Phạm vi khai báo class Rectangle được tính từ phím tab sau class Rectangle

'''

class Rectangle:

    '''

    Hàm (method) khởi tạo (constructor)

    Đây là method (phương thức) đặc biệt phải có khi khai báo class

    Mục đích: Để nạp những giá trị ban đầu cho các thể hiện (cụ thể)

    của đối tượng khi chạy chương trình

    '''

    def \_\_init\_\_(self,width,length):

        self.width = width

        self.length = length

    def area(self):

        result = self.width \* self.length

        return result

    def perimeter(self):

        result = 2\*(self.width + self.length)

        return result

    def display(self):

        print(f'rộng: {self.width}, dài: {self.length}, chu vi: {self.perimeter():.2f}, diện tích: {self.area():.2f}\n')

Tạo các file sau:

ChuNhat1.py

import Rectangle as hcn

# Hãy khai báo 2 hình chữ nhật cụ thể hay còn gọi là thể hiện (instances)

# Sau đó, in ra màn hình kích thước và diện tích , chu vi của mỗi cái

hcn1 = hcn.Rectangle(4,10)

hcn1.display()

hcn2 = hcn.Rectangle(7,25)

hcn2.display()

ChuNhat2.py

Chương trình menu

1- Thêm mới hình chữ nhật

2- Hiển thị danh sách hình chữ nhật

3- Tính tổng diện tích các hình chữ nhật

4- Hiển thị các hình chữ nhật có chu vi nhỏ nhất

Others- Thoát

'''

import Rectangle as rect

menu\_options = {

    1: 'Thêm mới hình chữ nhật',

    2: 'Hiển thị danh sách hình chữ nhật',

    3: 'Tính tổng diện tích các hình chữ nhật',

    4: 'Hiển thị các hình chữ nhật có chu vi nhỏ nhất',

    'Others': 'Thoát chương trình'

}

def print\_menu():

    for key in menu\_options.keys():

        print (key, '--', menu\_options[key] )

# Khai báo biến lưu trữ những hình chữ nhật

dsHCN = []

while(True):

        print\_menu()

        userChoice = ''

        try:

            userChoice = int(input('Nhập tùy chọn: '))

        except:

            print('Nhập sai định dạng, hãy nhập lại.....')

            continue

        #Check what choice was entered and act accordingly

        if userChoice == 1:

           cr = float(input("Nhập chiều rộng: "))

           cd = float(input("Nhập chiều dài: "))

           hcn = rect.Rectangle(cr,cd)

           dsHCN.append(hcn)

        elif userChoice == 2:

            for item in dsHCN:

                item.display()

        elif userChoice == 3:

            dientich = 0.0

            for item in dsHCN:

                dientich = dientich + item.area()

            print(f'Tổng diện tích là: {dientich}')

        elif userChoice == 4:

            if dsHCN.count == 0:

                print('Danh sách rỗng')

            else:

                chuvinn = dsHCN[0].perimeter()

                for item in dsHCN:

                    if chuvinn > item.perimeter():

                        chuvinn = item.perimeter()

                for item in dsHCN:

                    if item.perimeter() == chuvinn:

                        item.display()

        else:

            print('Thoát chương trình BYE BYE')

            break

ChuNhat3.py

Lưu trữ danh sách các hình chữ nhật từ file input.db

Lưu danh sách các hình chữ nhật xuống file output.db theo định dạng

    rong-dai-chuvi-dientich

Lưu ý: Trong các file mỗi hàng là thông tin một hình chữ nhật

'''

import Rectangle as rect

# Tải dữ liệu từ file vào listRectangle

fr = open('input.db',mode='r',encoding='utf-8')

listRectangle = []

for line in fr:

    stripLine = line.strip('\n')

    ds = stripLine.split(',')

    cr = float(ds[0])

    cd = float(ds[1])

    hcn = rect.Rectangle(cr,cd)

    listRectangle.append(hcn)

fr.close()

# Ghi dữ liệu từ listRectangle xuống file

fw = open('output.db',mode='w',encoding='utf-8')

for item in listRectangle:

    fw.write(f'{item.width}-{item.length}-{item.perimeter()}-{item.area()}\n')

fw.close()

'''

(\*) Sinh viên tự thực hành

Viết chương trình menu

1- Đọc dữ liệu từ file input.db

2- Thêm mới hình chữ nhật

3- Hiển thị danh sách hình chữ nhật

4- Lưu danh sách hình chữ nhật xuống file demo4output.db

Others- Thoát

'''

import Rectangle as rect

menu ={

    1:'1- Đọc dữ liệu từ file input.db'   ,

    2:'2- Thêm mới hình chữ nhật',

    3:'3- Hiển thị danh sách hình chữ nhật',

    4:'4- Lưu danh sách hình chữ nhật xuống file demo4output.db',

    'Others':'Thoát'

}

def print\_menu():

    for key in menu.keys():

        print (key,'--',menu[key])

#Khai báo biến lưu trữ hình chữ nhật

while(True):

    print\_menu()

    chon=''

    try:

        chon =int(input('Nhập tùy chọn:'))

    except:

        print('Nhập sai định dạng, hãy nhập lại:')

        continue

    #Kiểm tra các lựa chọn

    if chon ==1:

        #1- Đọc dữ liệu từ file input.db

        fr=open('input.db', mode='r',encoding ='utf-8')

        dsHCN =[]

        for line in fr:

            stripLine = line.strip('\n')

            ds =stripLine.split(',')

            cr =float(ds[0])

            cd =float(ds[1])

            hcn=rect.Rectangle(cr,cd)

            dsHCN.append(hcn)

        fr.close()

    elif chon ==2:

       # 2- Thêm mới hình chữ nhật

       cr =float(input("Nhập chiều rộng:"))

       cd =float(input("Nhập chiều dài:"))

       hcn =rect.Rectangle(cr,cd)

       dsHCN.append(hcn)

    elif chon ==3:

        #3- Hiển thị danh sách hình chữ nhật

        if dsHCN.count ==0:

            print('Danh sách rỗng')

        else:

            for item in dsHCN:

                item.display()

    elif chon ==4:

        #4- Lưu danh sách hình chữ nhật xuống file demo4output.db

        fw =open('outpudemo4.db',mode='w',encoding ='utf-8')

        for item in dsHCN:

            fw.write(f'{item.width}-{item.length}-{item.perimeter()}-{item.area()}\n)')

        fw.close()

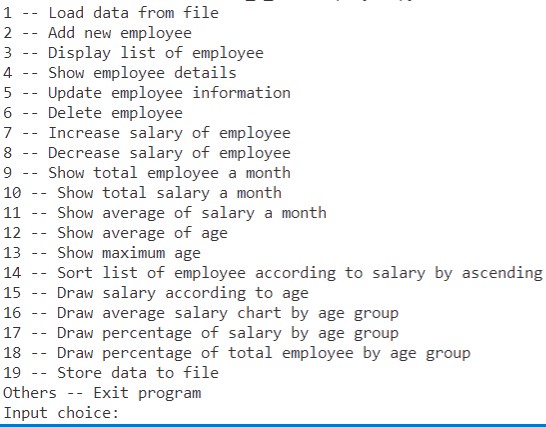
    else:

        print('Kết thúc chương trình')

        break

**Rule: Test cases is following by order, if failing any test case then the scoring process will be terminated, score is evaluated by percentage of number of test cases finishing.**

Warning: You must output the same test case to pass. DON’T ACCEPT HARD CODE /menu



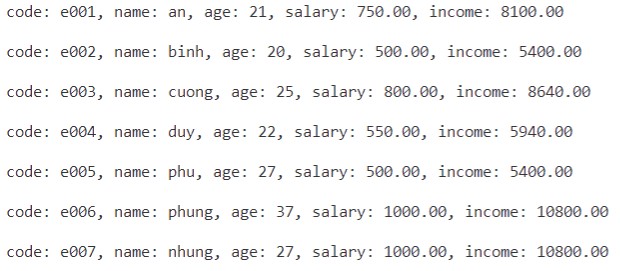
# Case-01

/menu

1

/menu

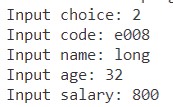
3



# Case-02

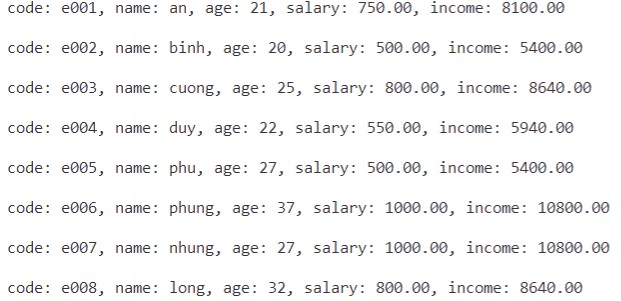
/menu

2



/menu

3



# Case-03

/menu

4

Input code: e006



/menu

4

Input code: e010

Nothing here

# Case-04

/menu

5

Input code: e020

Nothing here

/menu

5

Input code: e006

Input name: loan

Input age: 40

Input salary: 550

/menu

4

Input code: e006



/menu

5

Input code: e006

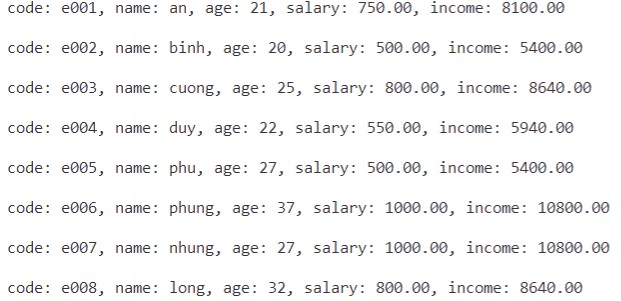
Input name: phung

Input age: 37

Input salary: 1000

/menu

3



# Case-05

/menu

6

Input code: e010

Nothing here

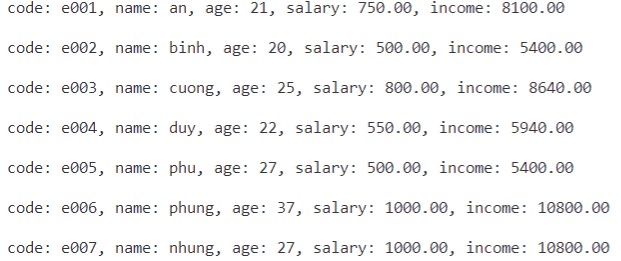
/menu

6

Input code: e008

/menu

3



# Case-06

/menu

7

Input code: e008

Nothing here

/menu

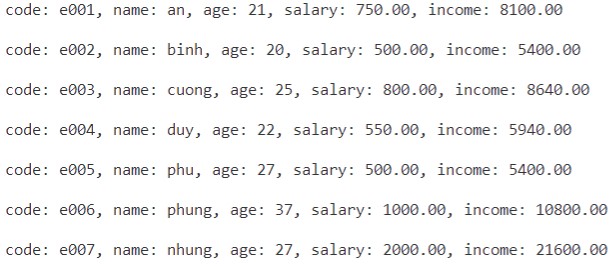
7

Input code: e007

Input amount: 1000

/menu

3



# Case-07

/menu

8

Input code: e008

Nothing here

/menu

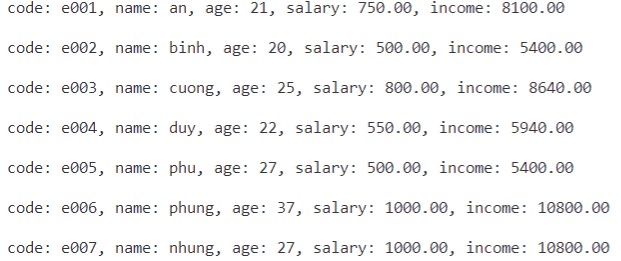
8

Input code: e007

Input amount: 1000

/menu

3



# Case-08

/menu

9

Number of employees: 7

/menu

10

Total salary a month: 5100.00

/menu

11

Average of salary: 728.57

/menu

12

Average of age: 25.57

# Case-09

/menu

13

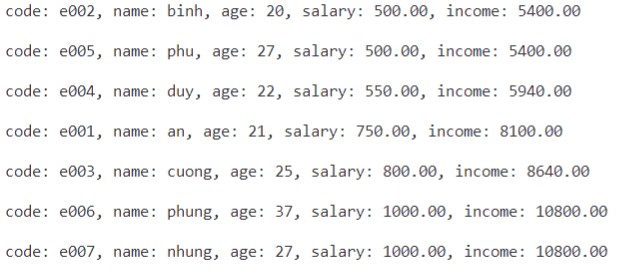
Maximum age: 20

/menu

14

/menu

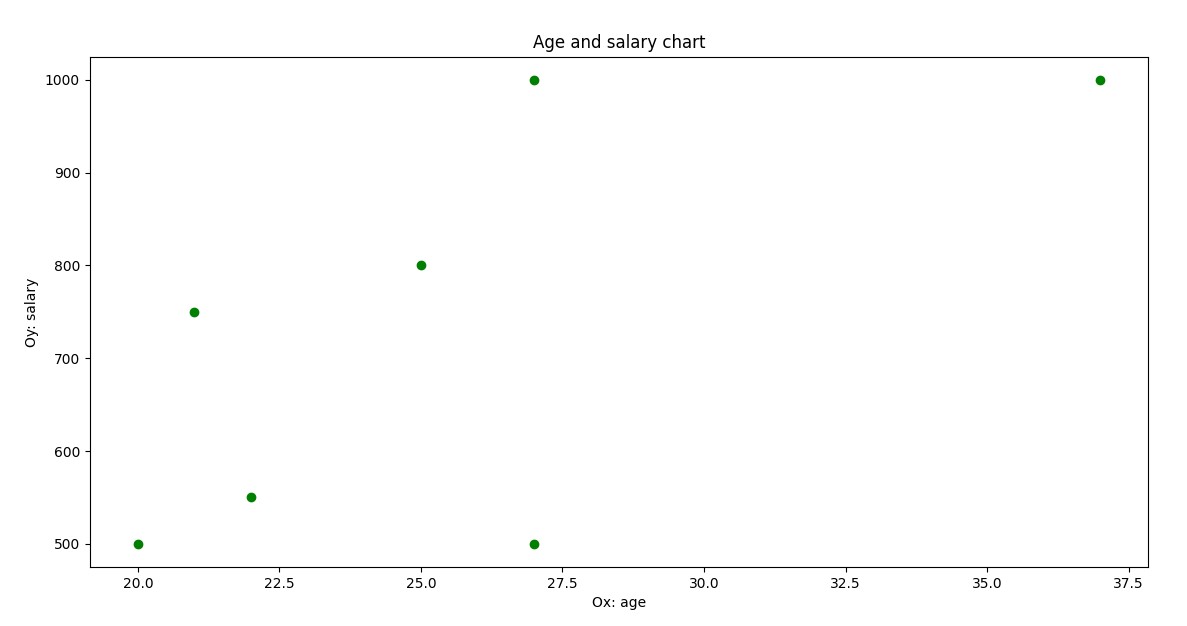
3



# Case-10

/menu

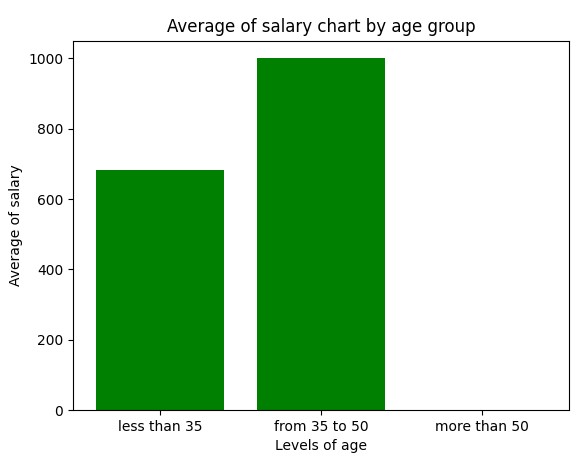
15



# Case-11

/menu

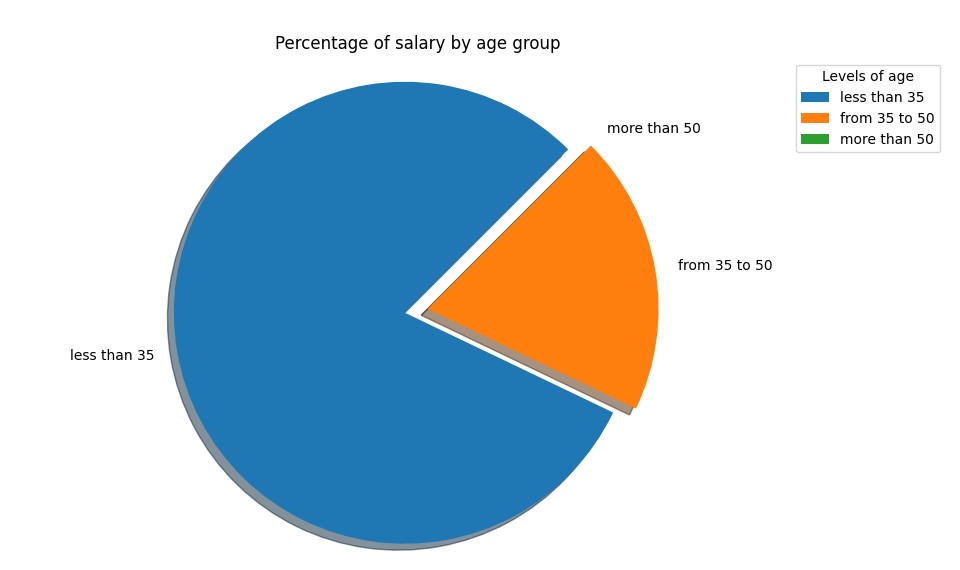
16



# Case-12

/menu

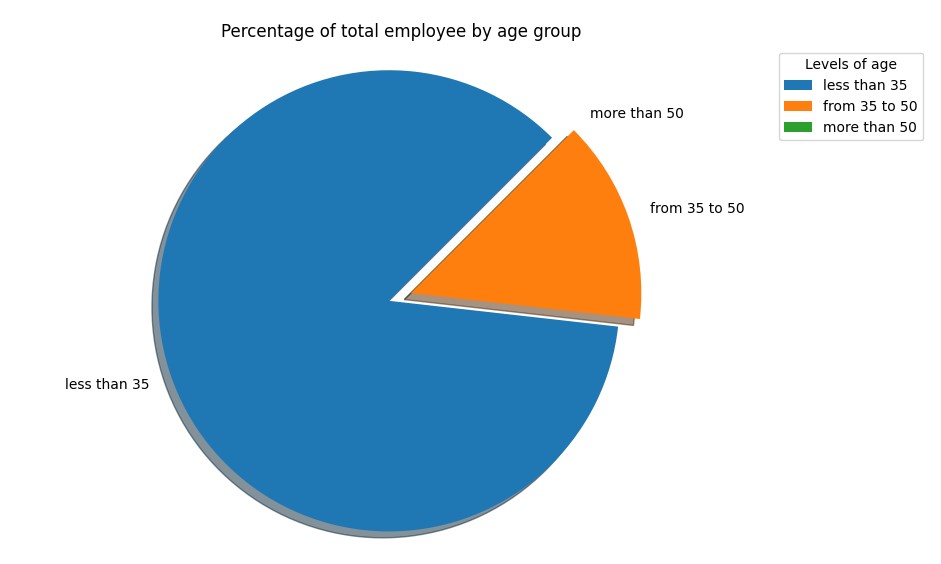
17



# Case-13

/menu

18



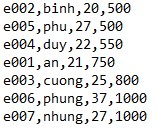
# Case-14

/menu

19

Store file successfully

// results



# Case-15

/menu

20

BYE BYE

// Exit program