

ÔN TẬP LẬP TRÌNH TRÍ TUỆ NHÂN TẠO

Câu 1 . So sánh List , Tuple , Set và Dictionary trong ngôn ngữ Python . Cho ví dụ từng loại ?

List

- List (hay danh sách) là một cấu trúc dữ liệu có thể thay đổi (mutable) và cho phép lưu trữ các phần tử có thứ tự.
- Các phần tử trong List được phân tách bằng dấu phẩy và được đặt trong dấu ngoặc vuông [].
- Có thể thay đổi giá trị của các phần tử trong List.

Vd : `ten = ['apple', 'banana', 'orange']`

Tuple :

- Tuple là một cấu trúc dữ liệu không thay đổi (immutable) và cho phép lưu trữ các phần tử có thứ tự.
- Các phần tử trong Tuple được phân tách bằng dấu phẩy và được đặt trong dấu ngoặc đơn () .
- Không thể thay đổi giá trị của các phần tử trong Tuple sau khi đã khai báo.

Vd : `point = (3, 5)`

Set :

- Set là một cấu trúc dữ liệu không có thứ tự và không chứa các phần tử trùng lặp.
- Các phần tử trong Set không được phân tách bằng dấu phẩy và được đặt trong dấu ngoặc nhọn {} .
- Set không cho phép truy cập các phần tử bằng chỉ số và không duy trì thứ tự.

Vd : `colors = {'red', 'green', 'blue'}`

Dictionary :

- Dictionary là một cấu trúc dữ liệu có thể thay đổi (mutable) và lưu trữ các cặp key-value.
- Mỗi key trong Dictionary phải là duy nhất và không thay đổi, trong khi value có thể thay đổi.
- Các cặp key-value được phân tách bằng dấu hai chấm và các cặp được đặt trong dấu ngoặc nhọn {}.

Vd : student = {'name': 'John', 'age': 20, 'grade': 'A'}

Câu 2. Viết chương trình nhập vào từ bàn phím một chuỗi S . Hãy tạo từ điển D trong đó key là các chữ cái xuất hiện trong S (chữ cái không được trùng nhau) , value tương ứng với số lần xuất hiện của các chữ cái trong S .

```
s = input("Nhập chuỗi S :")
dem = {}
for i in s :
    if i in dem :
        dem[i] +=1
    else:
        dem[i] = 1
print(dem)
```

Câu 3. Cho tập tin text.txt với nội dung : dsak1257\$%3*&4pk6819

Xây dựng hàm đọc nội dung tập tin text.txt .Thực hiện lọc và lấy số lẻ đưa vào danh sách (List) SoLe và các số chẵn đưa vào danh sách list SoChan

```
chan = []
le = []
try:
    with open("../data/test.txt", encoding="utf-8") as f:
        a = f.read()
        for i in a:
            if (i.isdigit()):
                i = int(i)
                if (i % 2 == 0):
                    chan.append(i)
            else:
```

```

        le.append(i)
except FileNotFoundError:
    print("Không tìm thấy file")

print(" số chẵn " , chan)
print("Số lẻ : " , le)

```

Câu 4. Viết tính tần suất xuất hiện của các từ được nhập vào từ input . In ra màn hình danh sách các từ sau khi đã được sắp xếp theo chữ cái .

```

nhap = input("Nhập chuỗi chữ : ")
def xuấtHien ( nhap ):
    dem = {}
    for i in nhap:
        if i in dem:
            dem[i] += 1
        else:
            dem[i] = 1
    SapXep = sorted(dem.keys())
    return dem , SapXep
dem , SapXep = xuấtHien(nhap)
print("Đã được sắp xếp",SapXep)

```

Câu 5. Viết chương trình cho phép nhập vào từ bàn phím họ và tên đầy đủ của sinh viên trong lớp , mỗi sinh viên được nhập trên một dòng .Việc nhập sẽ kết thúc khi người dùng gõ vào dòng trống .Sau đó , hãy in ra tất cả các họ và tên không trùng nhau của sinh viên trong lớp.

```

student = []
while True:
    fullname = input("Ho & ten :")
    if fullname == "":
        break
    else:
        student.append(fullname)

list_Student = set(student)      # ham set thuc hien loc cac ten trung nhau
for i in list_Student:
    print(i)

```

Câu 6. Viết chương trình nhập vào một mảng hai chiều các số thực A (m hàng , n cột) từ bàn phím .

a. Tìm giá trị lớn nhất và nhỏ nhất trên mỗi cột

- b. Tìm phần tử lớn nhất và phần tử nhỏ nhất của mảng A cùng các chỉ số hàng và cột của 2 phần tử này.
- c. Trong mảng A có bao nhiêu phần tử lớn nhất .

```
d. import numpy as np

# A ) Tìm giá trị lớn nhất và nhỏ nhất trên mỗi cột
def max_min_rows (matrix):
    max_cols = np.max(matrix, axis=0)
    min_cols = np.min(matrix, axis=0)
    return max_cols , min_cols

# B) Tìm phần tử lớn nhất và phần tử nhỏ nhất của mảng A cùng các
    chỉ số hàng và cột của 2 phần tử này.
def max_min_cols_chiSo (matrix):
    max_matrix = np.max(matrix)
    seach_matrix_max = np.where(matrix == max_matrix)
    seach_matrix_max = seach_matrix_max[0][0],
    seach_matrix_max[1][0]

    min_maxtrix = np.min(matrix)
    seach_matrix_min = np.where(matrix==min_maxtrix)
    seach_matrix_min = seach_matrix_min[0][0] ,
    seach_matrix_min[1][0]

    return max_matrix, seach_matrix_max , min_maxtrix ,
    seach_matrix_min
# C ) Trong mảng A có bao nhiêu phần tử lớn nhất .
def count_max (matrix):
    max_matrix = np.max(matrix)
    count_max_matrix = np.count_nonzero(matrix == max_matrix)
    return count_max_matrix

m = int(input("hang :"))
n = int(input("Cot : "))

matrix = np.zeros((n , m ))

for i in range(n):
    for j in range(m):
        matrix[i][j] = float(input("Phan tu trong mang "))

max_cols , min_cols = max_min_rows(matrix)
max_matrix , seach_max , min_matrix , seach_min =
max_min_cols_chiSo(matrix)
count_max = count_max(matrix)

print("Gia tri lon nhat cua cot",max_cols )
print("Gia tri lon nhat cua cot",min_cols)
print("-----")
print("Gia tri lon nhat cua matrix ",max_matrix)
print("Chi so cua phan tu lon nhat ",seach_max)
print("Gia tri nho nhat cua matrix ",min_matrix)
print("Chi so cua phan tu nho nhat ",seach_min)
print("-----")
print("So luong phan tu lon nhat ",count_max)
```

Câu 7. Viết chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào .Các tiêu chí kiểm tra mật khẩu bao gồm :

1. Ít nhất 1 chữ cái nằm trong [a-z]
2. Ít nhất 1 số nằm trong [0-9]
3. Ít nhất 1 kí tự nằm trong [A-Z]
4. Ít nhất 1 ký tự nằm trong [S#@]
5. Độ dài mật khẩu tối thiểu là 6
6. Độ dài tối đa là 12

Chương trình phải chấp nhận mật khẩu phân cách nhau bởi dấu , và kiểm tra xem chúng có đáp ứng những tiêu chí trên hay không . Mật khẩu hợp lệ sẽ được in , mỗi mật khẩu cách nhau bởi dấu phẩy

Vd : mật khẩu nhập vào chương trình là :

Abd1234@1,aF1#,2wE*,2We3345

Thì đầu ra là Abd1234@1

```
import numpy as np

def checkPassword (matkhau):

    # Ít nhất 1 chữ cái nằm trong [a-z]
    if not np.any([char.islower() for char in matkhau]):
        return False

    # 2. Ít nhất 1 số nằm trong [0-9]
    if not np.any([char.isdigit() for char in matkhau]):
        return False

    # 3. Ít nhất 1 kí tự nằm trong [A-Z]
    if not np.any([char.isupper() for char in matkhau]):
        return False

    # 4. Ít nhất 1 ký tự nằm trong [S#@]
    ky_Tu = ['$','@','#']
    if not np.any([char in ky_Tu for char in matkhau]):
        return False

    # 5. Độ dài mật khẩu tối thiểu là 6
    # 6. Độ dài tối đa là 12
    if len(matkhau) < 6 or len(matkhau)>12:
        return False
```

```

        return True

matKhai = input("Nhap mat khai : ").split(",")

TinhHopLe = []
for i in matKhai:
    i = i.strip()
    if checkPassword(i):
        TinhHopLe.append(i)
if len(TinhHopLe) == 0:
    print("Mat khai khong hop le ")
else:
    print("mat khai hop le /n".join(TinhHopLe))

```

Câu 8. Viết chương trình Python nhập một dãy số nguyên cách nhau bởi khoảng trắng , sau đó kiểm tra xem nó có đối xứng hay không ? Nếu dãy số nhập vào không đối xứng , kiểm tra xem có thể biến đổi nó để trở thành đối xứng hay không ? Nếu có khả năng đối xứng thì biến đổi nó trở thành đối xứng , ngược lại xuất ra màn hình KHÔNG CÓ KHẢ NĂNG ĐỐI XỨNG .

```

"""
Câu 8. Viết chương trình Python nhập một dãy số nguyên cách nhau bởi khoảng
trắng ,
sau đó kiểm tra xem nó có đối xứng hay không ?
Nếu dãy số nhập vào không đối xứng , kiểm tra xem có thể biến đổi nó để trở
thành đối xứng hay không ?
Nếu có khả năng đối xứng thì biến đổi nó trở thành đối xứng , ngược lại
xuất ra màn hình KHÔNG CÓ KHẢ NĂNG ĐỐI XỨNG .
"""

def check_DoiXungKhong (number):
    if number == number[::-1]:
        return True
    return False

def delete_Giong (numbuer):
    mang = []
    for i in number :
        mang.append(i)
    for i in numbuer:
        for j in range (mang.count(i)):
            mang.remove(i)

def co_The_Doi_Xung_Khong (number ):
    mang = []
    mang_kem = []
    dem = 0
    dem3 = 0
    for i in number :
        mang.append(i)
    mang = sorted(mang)
    mang_copy = mang
    if check_DoiXungKhong(number) == False:
        for i in mang :
            if mang_copy.count(i) == 1 :
                dem+=1

```

```

elif mang_copy.count(i) == 3:
    dem3+=1
if (dem > 1 or dem3 >= 2 )or (dem == 1 & dem3==1):
    return ("Khong the bien doi ")
else:
    for i in mang:
        if mang.count(i) % 2 == 0 :
            if i == mang[mang.index(i) + 1]:
                mang.remove(mang[mang.index(i) + 1])

        elif mang.count(i) % 3 == 0:

            if mang_copy.count(i)//3 ==0:
                mang_kem.append(i)
                del mang[mang.index(i) or mang.index(i) + 1]
                mang_kem.append(i)
    return (mang[::-1] + mang_kem + mang)

number = int(input("day so"))
number_str = str(number)
if check_DoiXungKhong(number_str) :
    print("Day so " ,number_str , " doi xung nhau" )
else:
    print("day so " , number_str," khong doi xung nhau")
    print("Bien doi ", co_The_Doi_Xung_Khong(number_str))

```

Cách 2 :

```

"""
Câu 8. Viết chương trình Python nhập một dãy số nguyên cách nhau bởi khoảng
trắng ,
sau đó kiểm tra xem nó có đối xứng hay không ?
Nếu dãy số nhập vào không đối xứng , kiểm tra xem có thể biến đổi nó để trở
thành đối xứng hay không ?
Nếu có khả năng đối xứng thì biến đổi nó trở thành đối xứng , ngược lại
xuất ra màn hình KHÔNG CÓ KHẢ NĂNG ĐỐI XỨNG .
"""
def check_DoiXungKhong (number):
    if number == number[::-1]:
        return True
    return False

def doi_Str_thanh_List (number ):
    mang = []
    for i in number :
        mang.append(i)
    mang.sort()
    return mang

def not_edit_Doi_Xung (number):
    dem1 = 0
    dem_Le = 0
    for i in number:
        if number.count(i) == 1:

            dem1+=1
    elif number.count(i) % 3 == 0:

```

```

        dem_Le+=1
    if dem1 >= 2 or dem_Le>3:
        return False
    return True
def edit_DoiXung (number):
    right = []
    left = []
    mid = []
    for i in number:
        if number.count(i) % 2 == 0 :
            if number.count(i) == 2 :
                if i not in right :
                    right.append(i)
                elif i not in left:
                    left.append(i)
            else:
                if right.count(i) < (number.count(i)//2) :
                    right.append(i)
                elif left.count(i) < (number.count(i)//2):
                    left.append(i)
        else:
            mid.append(i)
    return (left+mid+right[::-1])

number = int(input("day so"))
number_str = str(number)
mang = doi_Str_thanh_List(number_str)

if check_DoiXungKhong(number_str) :
    print("Day so " ,number_str , " doi xung nhau" )
else:
    print("day so " , number_str, " khong doi xung nhau")
    print(not_edit_Doi_Xung(mang))
    if not_edit_Doi_Xung(mang) == True :
        print("Bien doi " , edit_DoiXung(mang))
    else:
        print("Khong the bien doi")

```

Câu 9. Định nghĩa một hàm có thể tạo ra một dictionary chứa key là những số từ 1 đến 10 (bao gồm cả 1 và 10) và các value là các số nguyên tố khác nhau (nếu người dùng nhập vào một số không phải số nguyên tố hoặc số nguyên tố đã được nhập vào trước đó thì yêu cầu người dùng nhập lại đến khi nào hoàn tất việc nhập 10 value .In ra value lon nhat của dictionary .

```

def check (soNT):
    if soNT > 2 :

```



```

for i in (2 , soNT):
    if soNT % i != 0 :
        return soNT
    else:
        soNT = 1

d = dict()

for i in range(1,10):
    while True:
        value = int(input("Nhap value ( la so nguyen to ) : "))
        print(check(value))
        if (check(value) == value):
            d[i] = value
            break
        else:
            print("Khong phai so nguyen to ")
print("Max : " ,max(d.values()))

```

Câu 10. Hiện thực lớp đối tượng DIENTHOAI với các đặc điểm sau:

Thuộc tính :

HANGSX: chuỗi .

MODEL : chuỗi

GIATIEN : số nguyên

MANHINH (kích thước màn hình): số thực

Phương thức :

- Thêm thông tin điện thoại
 - Sửa thông tin điện thoại
 - Xóa thông tin điện thoại
 - Xuất thông tin điện thoại
- a) Hãy xây dựng các hàm cần thiết để cho phép nhận vào từ bàn phím các thông tin để thực hiện danh sách (List) các đối tượng DIENTHOAI
- b) Sắp xếp và xuất ra danh sách (List) các đối tượng DIENTHOAI theo thứ tự kích thước màn hình lớn đến nhỏ

c)

```
class DienThoai:

    def __init__(self ,hangsx, model, giatien, manhin):
        self.HANGSX = hangsx
        self.MODEL = model
        self.GIATIEN = giatien
        self.MANHINH = manhin

    def hien_Thi_Thong_Tin (self):
        print("Hang sản xuất : ",self.HANGSX)
        print("Model : " , self.MODEL)
        print("Gia tien : " , self.GIATIEN)
        print("Kích thước màn hình " , self.MANHINH)

danhSach_SP = []
def them_thong_tin ():
    hangsx = input("Hang SX : ")
    model = input("Model : ")
    giatien = int(input("Gia Tien : "))
    manhin = float(input("Màn hình :"))
    dien_Thoai = DienThoai(hangsx , model , giatien ,
manhin)
    danhSach_SP.append(dien_Thoai)
def sua_Thong_Tin ():
    model = input("Sua model ")
    for i in danhSach_SP:
        if i.MODEL == model:
            hangsx = input("Sua hang sản xuất : ")
            giatien = input("Sua gia tien : ")
            manhin = input("Sua thông số Màn Hình : ")

            i.HANGSX = hangsx
            i.GIATIEN = giatien
            i.MANHINH = manhin
            print("Thông tin điện sản phẩm đã được cập nhật
")
        return
    print("Không tìm thấy địa chỉ model")

# Xóa thông tin điện thoại
def delete_SP ():
    model = input("Thông số model : ")
    for i in danhSach_SP:
        if i.MODEL == model:
            danhSach_SP.remove(i)
            print("Sản phẩm đã được xóa !!! ")
            return
    print("Không tìm thấy model ???")

# Sắp xếp thông tin điện thoại
def sort_SP ():
    sort_SanPham = sorted(danhSach_SP,key=lambda i :
i.MANHINH , reverse=True)
    for i in sort_SanPham:
        i.hien_Thi_Thong_Tin()
while True:
    print("----- MENU -----")
    print("1. Thêm điện thoại")
    print("2. Sửa thông tin điện thoại")
    print("3. Xóa điện thoại")
```

```

print("4. Xuất danh sách điện thoại")
print("5. Sắp xếp và xuất danh sách theo kích thước màn hình")
print("0. Thoát chương trình")

choice = input("Nhập lựa chọn của bạn: ")

if choice == "1":
    them_thong_tin()
elif choice == "2":
    sua_Thong_Tin()
elif choice == "3":
    delete_SP()
elif choice == "4":
    for dt in danhSach_SP:
        dt.hien_Thi_Thong_Tin()
elif choice == "5":
    sort_SP()
elif choice == "0":
    break

```

Câu 11. Hiện thức lớp đối tượng SINHVIEN với các thuộc tính và phương thức sau :

Thuộc tính :

MSSV : chuỗi

HOTEN :chuỗi

DIEMKT1 : số thực

DIEMKT2 : số thực

DIEMTB : (DIEMKT1+DIEMKT2)/2

Phương thức :

- Thêm thông tin sinh viên
- Sửa thông tin sinh viên
- Xuất thông tin sinh viên

a) Hãy xây dựng các hàm cần thiết để cho phép nhập vào từ bàn phím các thông tin để thực danh sách (List) các đối tượng SINHVIEN

b) Sắp xếp và xuất ra danh sách (List) các đối tượng SINHVIEN theo thứ tự DIEMTB từ cao đến thấp

```
class SINHVIEN:
    def __init__(self , MSSV , HOTEN , DIEMKT1 , DIEMKT2 ):
        self.MSSV =MSSV
        self.HOTEN = HOTEN
        self.DIEMKT1 = DIEMKT1
        self.DIEMKT2 = DIEMKT2
        self.DIEMTB = (DIEMKT1+DIEMKT2)/2

    def nhap_Thong_Tin_SV (self):
        self.MSSV = input("Ma so sinh vien : ")
        self.HOTEN = input("Ten sinh vien : ")
        test = True
        while test == True:
            self.DIEMKT1 = float(input("Diem kiem tra 1 : "))
            self.DIEMKT2 = float(input("Diem kiem tra 2 : "))
            if (0<= self.DIEMKT1 <= 10) & ( 0 <= self.DIEMKT2 <= 10):
                break
            else:
                print("diem kiem tra >=0 va <= 10")
                test = True
        self.DIEMTB = (self.DIEMKT1 + self.DIEMKT2) / 2

    def sua_Thong_Tin_SV (self):
        self.HOTEN = input("Nhap ho ten : ")
        test = True
        while test == True:
            self.DIEMKT1 = float(input("Diem kiem tra 1 : "))
            self.DIEMKT2 = float(input("Diem kiem tra 2 : "))
            if (0 <= self.DIEMKT1 <= 10) & (0 <= self.DIEMKT2 <= 10):
                break
            else:
                print("diem kiem tra >=0 va <= 10")
                test = True
        self.DIEMTB = (self.DIEMKT1 + self.DIEMKT2) / 2

    def hien_Thi_Thong_Tin (self):
        print("Ma so sinh vien : " , self.MSSV)
        print("Ho & ten : " , self.HOTEN)
        print("Diem kiem tra 1 : " , self.DIEMKT1)
        print("Diem kiem tra 2 : " , self.DIEMKT2)
        print("Diem Trung binh : " , ((self.DIEMKT1 + self.DIEMKT2) / 2))
        print(" ")
danh_sach_sinh_vien = []

def them_Thong_tin_vao_danh_sach ():
    number_sv = int(input("So luong sinh vien "))
    for i in range (number_sv):
        print(f"Nhap thong tin sinh vien {i + 1} : ")
        sinh_Vien = SINHVIEN("", "", 0 , 0 )
        sinh_Vien.nhap_Thong_Tin_SV()
        danh_sach_sinh_vien.append(sinh_Vien)
    return danh_sach_sinh_vien

def sua_Thong_Tin_SV ():
    mssv = input("Nhap ma sinh vien : ")
```

```

for i in danh_sach_sinh_vien :
    if i.MSSV == mssv:
        i.sua_Thong_Tin_SV ()
        print("Da cap nhat thong tin ")
        return
    print("Khong tim thay Ma so sinh vien nay !!!")
def delete_SV ():
    mssv = input("Nhap ma sinh vien ")
    for i in danh_sach_sinh_vien:
        try:
            if i.mssv == mssv:
                danh_sach_sinh_vien.remove(mssv)
                print("Thong tin sinh vien ", mssv, " da duoc xoa ")
                return
        except ValueError:
            print("Khong tin thay ma sinh vien !!! ")

def sap_Xep ():
    danh_sach = sorted(danh_sach_sinh_vien , key=lambda sv : sv.DIENTB ,
reverse=True )
    for i in danh_sach :
        i.hien_Thi_Thong_Tin ()
while True:
    print("-----MENU-----")
    print("1 : Them thong tin sv")
    print("2 : Sua thong tin sv")
    print("3 : Xoa thong tin sv")
    print("4 : Hien thi thong tin sv")
    print("5 : Sap xep diem sv ")
    print("0 : Thoat chuong trinh ")

    chon = int(input("Nhap so chon : "))

    if chon ==1 :
        them_Thong_tin_vao_danh_sach()
    elif chon == 2 :
        sua_Thong_Tin_SV()
    elif chon == 3:
        delete_SV()
    elif chon == 4 :
        for i in danh_sach_sinh_vien:
            i.hien_Thi_Thong_Tin ()
    elif chon == 5 :
        sap_Xep()
    elif chon ==0 :
        break

```

Câu 12: Thực hiện lớp đối tượng QUEUE và các thuộc tính và phương thức sau:

Thuộc tính :

Element : danh sách (List)

Front : số nguyên

Phương thức :

- Kiểm tra hàng đợi rỗng
- Trả về phần tử ở hàng đợi
- Thêm một phần tử vào hàng đợi
- Xóa phần tử ở hàng đợi
- Xuất hàng đợi ra màn hình

- a) Hãy xây dựng các hàm cần thiết bắt buộc người nhập vào một số nguyên tố (nếu giá trị nhập vào không phải số nguyên tố thì cần thiết yêu cầu phải nhập vào) . Thêm các số nguyên tố được nhập vào hàng đợi Q là đối tượng kế thừa từ lớp QUEUE . Xuất ra màn hình hàng đợi Q .
- b) Tuần tự thực hiện các thao tác sao khi hàng đợi Q rỗng : Tìm và xuất ra màn hình phần tử đầu danh sách Q : Xóa phần tử đầu danh sách Q và xuất ra màn hình hàng đợi Q.

```
class Queue :
    def __init__(self):
        self.Element = []
        self.Front = None

    # kiểm tra hàng đợi có rỗng hay không
    def is_empty (self):
        return len(self.Element) == 0

    # phần tử đầu trong hàng đợi
    def get_front(self):
        return self.Front

    # thêm một phần tử vào cuối hàng đợi
    def enqueue (self , item):
        self.Element.append(item)
        if self.Front is None :
            self.Front = item

    # loại bỏ phần tử đầu tiên khỏi hàng đợi
    def dequeue (self):
        if not self.is_empty():
            self.Element.pop(0)
            # pop(0) là hàm bỏ phần tử đầu
            # trong hàng đợi
            if not self.is_empty():
                self.Front = self.Element[0]
            else:
                self.Front = None
        else:
```

```

        print("Hang doi rong")
# hiển thị các phần tử trong hàng đợi
def dispalay (seft):
    if not seft.is_empty():
        print("hang doi")
        for i in seft.Element:
            print(i)
    else :
        print("hang doi rong ")
def is_so_Nguyen_To ( n ) :
    if n < 2:
        return False
    for i in range(2, int(n**0.5 )+1):
        if n % i == 0:
            return False
    return True

Q = Queue()

while True :
    num = int(input("Nhap so nguyen to "))
    try :
        if is_so_Nguyen_To(num):
            Q.enqueue (num)
        else:
            print("Vui long nhap lai so nguyen to ")
    except ValueError :
        print("Vui long nhap lai so nguyen to")

    chon = input("Tiep tục hay không ? (y , n )")
    if chon.lower() == "n":
        # ham lower dung de chuyen doi ki tu
        viet hoa thanh viet thuong
        break

Q.dispalay()

while not Q.is_empty():
    print("Phan tu dau trong danh sach " , Q.get_front())
    Q.dequeue()
    Q.dispalay()

```

Câu 13. Viết chương trình python theo OOP.

- Class InHoaDon (in ra hóa đơn) bán hàng Rau Quả trong đó các thuộc tính .
- Ngay , LoaiRauQua , SoLuong , DonGia , ThanhTien.

```

from datetime import datetime

class InHoaDon:
    def __init__(self , ngay , LoaiRauQua , SoLuong , DonGia ):
        self.ngay = ngay
        self.LoaiRauQua = LoaiRauQua
        self.Soluong = SoLuong
        self.DonGia = DonGia

```

```

        self.ThanhTien = self.SoLuong * self.DonGia
    def hien_Thi (seft):
        print("----- Hoa Don -----")
        print("Ngày mua hàng : " , seft.ngay)
        print("Loại rau quả : " , seft.LoaiRauQua)
        print("Số lượng : " , seft.SoLuong)
        print("Đơn Giá : " , seft.DonGia)

ngay = str(datetime.now())
loaiRauQua = input("Loại rau quả : ")
while True:
    soLuong = int(input("Số lượng : "))
    donGia = int(input("Đơn giá : "))
    if soLuong < 0 or donGia < 0:
        print("Vui lòng nhập loại")
    else:
        break
hoaDon = InHoaDon(ngay , loaiRauQua , soLuong , donGia)
hoaDon.hien_Thi()

```

Câu 14. Cho tập tin thống kê môn học với tên k2021.csv nội dung như sau :

- Viết câu lệnh đọc file k2021.csv >
- In 5 dòng đầu tiên và 5 dòng cuối cùng của dữ liệu ra màn hình?
- Thống kê xem lớp có bao nhiêu bạn trượt môn (điểm dưới 4 hoặc không có điểm)
- Thống kê xem lớp có bao nhiêu bạn điểm loại giỏi (điểm từ 8 trở lên)

```

import pandas as pd

a = pd.read_csv("./data/k2020.csv" , index_col=0)

#B

print(a.head(5))
print(a.tail(5))

# C

print(f"Số lượng học sinh đạt loại giỏi là : " , len(a[a.Diem >=8]) , " sinh viên ")
print(f"Số lượng sinh viên trước môn là : " , len(a[(a.Diem < 4 )|(a.Diem.isnull())]) , " sinh viên ")

```


Câu 15. Dữ liệu : winequality-red.csv có các số đo và chất lượng của rượu vang đỏ , các thuộc tính .

a. Viết chương trình Python :

In ra xem bao nhiêu dòng và bao nhiêu cột trong file.

b. Vẽ biểu đồ minh họa Dataset với thuộc tính alcohol và điểm của quality.

c. Sử dụng hồi quy tuyến tính để xây dựng quan giữa thuộc tính alcohol và quality:

- In ra độ lệch chuẩn (căn bậc 2 phương sai)

Hồi số hồi quy

Sai số

Dự báo về chất lượng rượu khi cho nồng độ alcohol thay đổi
(Nhập)

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model , metrics
data = pd.read_csv("./data/winequality-red.csv" , index_col=0)

print("Cot trong data winequality-red.csv : ",len(data.columns))
print(data.columns)
alcohol = data['alcohol']
quality = data['quality']
plt.plot(alcohol , quality , "go")
plt.xlabel('Alcohol')
plt.ylabel('Quality')
plt.show()

x = alcohol.values.reshape(-1,1)
y = quality.values.reshape(-1 , 1)

model = linear_model.LinearRegression()
model.fit(x,y)

mse = metrics.mean_squared_error(model.predict(x) , y)

print("Tong binh phuong sai so tren tap mau " , mse)
print("he so hoi quy : " , model.coef_)
print("Sai so " , model.intercept_)

while True :
    z = float(input("nhap nồng độ alcohol (nhập 0 để dừng )"))
    if z == 0:
```

```
        break
    print("Nong do ruou " , z , " do , du bao chat luong " ,
model.predict([[z]]))
```