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 dụng : 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("Khong tin thay file")  
  
print(" số chẳn " , chan)  
print("Số lẽ : " , 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("Nhâp chuỗi chữ : ")  
def xuatHien ( 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 = xuatHien(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 dung gõ vào dòng trống .Sau đó , hãy in ra tất cả các họ và tên không trùng nhau của sinh viện 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 nhạp vào một mảng hai chiều các số thực A ( m hàng , n cột ) từ bàn phím .

1. Tìm giá trị lớn nhất và nhỏ nhất trên mỗi cột
2. 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.
3. Trong mảng A có bao nhiêu phần tử lớn nhất .
4. 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 tròng [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 da là 12

Chương trình phải chấp nhận mật khẩu phân cách nhâu bởi dấu , và kiểm tra xe chúng có đap ứng những tiêu trí 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 tròng [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 da là 12  
 if len(matkhau) < 6 or len(matkhau)>12:  
 return False  
  
 return True  
  
matKhau = input("Nhap mat khau : ").split(",")  
  
TinhHopLe = []  
for i in matKhau:  
 i = i.strip()  
 if checkPassword(i):  
 TinhHopLe.append(i)  
if len(TinhHopLe) == 0:  
 print("Mat khau khong hop le ")  
else:  
 print("mat khau 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 .

def check\_DoiXungKhong (number):  
 if number == number[::-1]:  
 return True  
 return False  
  
def bienDoi (number):  
 if check\_DoiXungKhong(number) == False:  
 list\_number = number+number[::-1]  
 return list\_number  
  
number = int(input("day so"))  
number\_str = str(number)  
number\_list = list(number\_str)  
if check\_DoiXungKhong(number\_list) :  
 print("Day so " ,number\_str ," doi xung nhau" )  
else:  
 print("day so " , number\_str," khong doi xung nhau")  
 print("Bien doi day so tren thanh doi xung : " , (bienDoi(number\_list)))

*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

1. 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
2. Sắp xếp và xuất ra danh sách (List) các đối tượng DIENTHOAI theo thứ tự kích thước màn hình lớn đến nhỏ
3. class DienThoai:  
     
    def \_\_init\_\_(self ,hangsx, model, giatien, manhinh):  
    self.HANGSX = hangsx  
    self.MODEL = model  
    self.GIATIEN = giatien  
    self.MANHINH = manhinh  
     
    def hien\_Thi\_Thong\_Tin (self):  
    print("Hang san xuat : ",self.HANGSX)  
    print("Model : " , self.MODEL)  
    print("Gia tien : " , self.GIATIEN)  
    print("Kich thuoc man hinh " , self.MANHINH)  
     
   danhSach\_SP = []  
   def them\_thong\_tin ():  
    hangsx = input("Hang SX : ")  
    model = input("Model : ")  
    giatien = int(input("Gia Tien : "))  
    manhinh = float(input("Man hinh :"))  
    dien\_Thoai = DienThoai(hangsx , model , giatien , manhinh)  
    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 san xuat : ")  
    giatien = input("Sua gia tien : ")  
    manhinh = input("Sua thong so Man Hinh : ")  
     
    i.HANGSX = hangsx  
    i.GIATIEN = giatien  
    i.MANHINH = manhinh  
    print("Thong tin dien san pham da duoc cap nhat ")  
    return  
    print("Khong tin thay dia chi model")  
     
   # Xóa thông tin điện thoại  
   def delete\_SP ():  
    model = input("Thong so model : ")  
    for i in danhSach\_SP:  
    if i.MODEL == model:  
    danhSach\_SP.remove(i)  
    print("San pham da duoc xoa !!! ")  
    return  
    print("Khong tim thay model ???")  
     
   # Sap Xep 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 thức

DIEMTB : (DIEMKT1+DIEMKT2)/2

Phương thức :

* Thêm thông tin sinh viên
* Sửa thông tin sinh viên
* Xuất thông tin sinh viên

1. 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
2. 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 (seft):  
 seft.MSSV = input("Ma so sinh vien : ")  
 seft.HOTEN = input("Ten sinh vien : ")  
 test = True  
 while test == True:  
 seft.DIEMKT1 = float(input("Diem kiem tra 1 : "))  
 seft.DIEMKT2 = float(input("Diem kiem tra 2 : "))  
 if (0<= seft.DIEMKT1 <= 10) & ( 0 <= seft.DIEMKT2 <= 10):  
 break  
 else:  
 print("diem kiem tra >=0 va <= 10")  
 test = True  
 seft.DIEMTB = (seft.DIEMKT1 + seft.DIEMKT2) / 2  
  
 def sua\_Thong\_Tin\_SV (seft):  
 seft.HOTEN = input("Nhap ho ten : ")  
 test = True  
 while test == True:  
 seft.DIEMKT1 = float(input("Diem kiem tra 1 : "))  
 seft.DIEMKT2 = float(input("Diem kiem tra 2 : "))  
 if (0 <= seft.DIEMKT1 <= 10) & (0 <= seft.DIEMKT2 <= 10):  
 break  
 else:  
 print("diem kiem tra >=0 va <= 10")  
 test = True  
 seft.DIEMTB = (seft.DIEMKT1 + seft.DIEMKT2) / 2  
  
  
 def hien\_Thi\_Thong\_Tin (seft):  
 print("Ma so sinh vien : " , seft.MSSV)  
 print("Ho & ten : " , seft.HOTEN)  
 print("Diem kiem tra 1 : " , seft.DIEMKT1)  
 print("Diem kiem tra 2 : " , seft.DIEMKT2)  
 print("Diem Trung binh : " , ((seft.DIEMKT1 + seft.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.DIEMTB , 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 rổng
* 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

1. 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 .
2. 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  
  
 # kiem tra hang doi co rong hay khong  
 def is\_empty (sefl):  
 return len(sefl.Element) == 0  
  
 # phan tu dau trong hang doi  
 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 (seft ):  
 if not seft.is\_empty():  
 seft.Element.pop(0) # pop(0) la ham bo phan tu dau trong hang doi  
 if not seft.is\_empty():  
 seft.Front =seft.Element[0]  
 else:  
 seft.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 tuc hay khong ? (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 Qủa 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("Ngay mua hang : " , seft.ngay)  
 print("Loai rau qua : " , seft.LoaiRauQua)  
 print("So luong : " , seft.SoLuong)  
 print("Don Gia : " , seft.DonGia)  
  
ngay = str(datetime.now())  
loaiRauQua = input("Loai rau qua : ")  
while True:  
 soLuong = int(input("So luong : "))  
 donGia = int(input("Don gia : "))  
 if soLuong < 0 or donGia < 0:  
 print("Vui long nhap loai")  
 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 sao :

1. Viết câu lệnh đọc file k2021.csv >
2. In 5 dòng đầu tiên và 5 dòng cuối cùng của dữ liệu ra màn hình?
3. Thống kê xem lớp có bao nhiêu bạn trượt môn (điểm dưới 4 hoặc không có điểm )
4. 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"So luong hoc sinh dat loai gioi la : " ,len(a[a.Diem >=8]) , " sinh vien ")  
print(f"So luong sinh vien truoc mon la : " , len(a[(a.Diem < 4 )|( a.Diem.isnull())]) , " sinh vien ")

Câu 15. Dữ liệu : winequality-red.csv có các số đo và chắc lượng của rựu vang đó , các thuộc tính .

1. Viết chương trình Python :   
   In ra xem bao nhiêu dòng và bao nhiêu cột trong file.
2. Vẽ biểu đồ mình họa Dataset với thuộc tính alcohol và điểm của quality.
3. Sử dụng hồi quy tuyến tính để xây dựng quan giữa thuộc tính alcohol và quality:  
    - In ra độ lệch chuẩn ( căn bặc 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 nong do alcohol (nhap 0 de dung )"))  
 if z == 0:  
 break  
 print("Nong do ruou " , z , " do , du bao chat luong " , model.predict([[z]]))