

**INDIVIDUAL ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**CT010-3-1-FSD**

**FUNDAMENTALS OF SOFTWARE DEVELOPMENT**

**APU/UC1F2002-CGD/IT/SE/CS(DA)/IT(IOT)/CS**

**APU1F2002-MMT/MMT(VRAR)/CE/CS(DF)/CS(CYB)/CS(IS)**

**HAND OUT DATE: 15TH MAY 2020**

**HAND IN DATE: 8 TH JUNE 2020**

**WEIGHTAGE: 100%**

**INSTRUCTIONS TO CANDIDATES:**

1. Submit your assignment online in MS Teams unless advised otherwise
2. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld
3. Cases of plagiarism will be penalized
4. You must obtain at least 50% in each component to pass this module



**Individual Assignment**

Module code : CT010-3-1-FSD

Student name : Dow Yi Cheng

TP number : TP059614

Intake Code : APU1F2002CS(DA)

Lecturer Name : Tanveer Khaleel Shaikh

Hand out Date           : 15th MAY 2020

Hand in Date : 8th JUNE 2020

Table of Contents

[Introduction 3](#_Toc63548397)

[Assumption 4](#_Toc63548398)

[Pseudocode 4](#_Toc63548399)

[Flow Chart 56](#_Toc63548400)

[Variable 57](#_Toc63548401)

[Control structure 57](#_Toc63548402)

[Looping structure 58](#_Toc63548403)

[.txt file 59](#_Toc63548404)

[Additional features 60](#_Toc63548405)

[Screenshots of I/O & Explanation 62](#_Toc63548406)

[Conclusion 69](#_Toc63548407)

[References 69](#_Toc63548408)

# Introduction

Nowadays, the epidemic of COVID-19 is getting worse. Thus, the purpose of this assignment is developing a program to helping hospital staff to recording patient data easily. With this program doctor and nurse can not getting waste any precious time to record, manage and find data of patient.

# Assumption

This program can record patient data while doing register, such as patient name, contact, specific zone and patient id but I found that patient id is not asking user to type in. It is needed to give a patient id automatically to the particular patient. That means I needed to read the file first and read the patient id from the .txt file.

Also, this program also able to record the result of patient. Example like, doctor or nurse just need enter patient id ,then this program can allow user to type in the result of the patient. When I am doing this part, I just realize a big problem which is overlapping.

When I doing test part, I thought I needed to record those data in 6 .txt file, but I just realize I just need recording into 4 txt file enough, because the when patient tested positive, it is not allow to doing any test anymore.

Besides that, this program will also give a case id while patient is infected COVID-19. That means, I not only want to give a case id to the particular patient, also needed display it in the test result so the user able to know the case id of the patient.

After that, when I was doing the status part , I just realize I need to change the status “ACTIVE” into “RECOVERED” or “DECEASED” , so I decided go to w3school to learn about the solution of the problem.

In the part of edit patient status, I need to read the case id of the patient and link it to the patient id for finding data in the patient data file. Which means I need to record the patient id in the case id file .

I am also keeping facing the problem of string and integer. I just realize I can’t save integer data in the .txt file. Then, when I read the data from .txt file. it will auto recognise these data into string format.

And also, when I am reading data from .txt file, I can’t use strip() when the data is in list format. That means I need to specific the data location or digit that I want to strip.

# Pseudocode

PROGRAM register

BEGIN

declare cnt,ptlist,mylist,list1,sp,ptnumber,ptname,ptcontact,ptgroup,r,f,list1

cnt1 = 1

ptlist = []

mylist = []

print ("\nEnter patient name:")

read ptname

#read patiend id from .txt file

r=open("patient.txt","r")

LOOP list1 FROM r:

sp = list1.split("><")

ptlist.append(str(sp[0]))

END LOOP

r.close()

ptnumber = int(ptlist[-1])

ptnumber = ptnumber + 1

print ("Enter patient contact number or email address :")

read ptcontact

print("Select one patient group(ATO/ACC/AEO/SID/AHS) in below:\nA.ATO\nB.ACC\nC.AEO\nD.SID\nE.AHS")

print ("Enter patient group:")

read ptgroup

while (ptgroup!= "ATO" and ptgroup!="ACC" and ptgroup!="AEO" and ptgroup!="SID" and ptgroup!="AHS"):

print ("Invalid group name")

print ("Please type again:")

read ptgroup

END while

print("\nPatient name is :",ptname,

"\nPatient id numebr is :",ptnumber,

"\nPatient personal contact number or email address is:",ptcontact,

"\nPatient group is :",ptgroup,"\n")

f=open("patient.txt","a")

f.write(str(ptnumber)+"><"+str(ptname)+"><"+str(ptcontact)+"><"+str(ptgroup)+"\n")

f.close ()

zone ()

END

PROGRAM zone

BEGIN

declare mylist,r,list1,table1,table2,table3,table4,cnt2

mylist=[]

r=open("patient.txt","r")

LOOP list1 FROM r:

sp = list1.split("><")

mylist.append(int(sp[0]))

table1 = []

table2 = []

table3 = []

table4 = []

LOOP cnt2 FROM mylist STEP 1:

if (cnt2%4==1):

table1.append(cnt2)

elif (cnt2%4==2):

table2.append(cnt2)

elif (cnt2%4==3):

table3.append(cnt2)

else:

table4.append(cnt2)

END if

END LOOP

print ("Patient ID in Zone A are: ",table1,

"\nPatient ID in Zone B are: ",table2,

"\nPatient ID in Zone C are: ",table3,

"\nPatient ID in Zone D are: ",table4 )

print ("Total patient ID list that have in hospital : ",mylist)

r.close()

END

PROGRAM test1

BEGIN

declare flag,ptlist,flag,r1,r2,r3,r4,r5,ptnumber,list1,list2,list3,list4,list5,a

flag = 1

ptlist =[]

while (flag == 1 or flag == 2 or flag == 3 or flag==4 or flag ==5):

print ("\nEnter patient ID:")

read ptnumber

flag = 0

r1 = open ("patient.txt","r") # checking range of patient

LOOP list1 FROM r1:

a = list1.split("><")

ptlist.append (a[0])

END LOOP

LOOP a FROM ptlist STEP 1:

if (ptnumber != int(a)):

flag = 1

elif (ptnumber == int(a)):

flag = 0

break

END if

END LOOP

r1.close()

r3 = open ("negativecase1.txt","r") # checking overlapping in negative

LOOP list3 FROM r3 :

a = list3.strip()

if (ptnumber == int(a)):

flag = 3

break

END if

END LOOP

r3.close()

r2 = open ("positivecase.txt","r") # checking overlapping in positive

LOOP list2 FROM r2 :

list2 = list2.split("\t")

a = list2[0].strip()

if (ptnumber == int(a)):

flag = 2

break

END if

END LOOP

r2.close()

r4 = open ("recover.txt","r")# checking overlapping in recover

LOOP list4 FROM r4 :

a = list4.strip()

if (ptnumber ==int(a)):

flag = 4

break

END if

END LOOP

r4.close()

r5 = open ("dead.txt","r") # checking overlapping in dead

LOOP list5 FROM r5 :

a = list5.strip()

if (ptnumber ==int(a)):

flag = 5

break

END if

END LOOP

r5.close()

if flag == 1:

print ("This patient id haven't register\nPlease enter again")

testinput()

elif flag == 2:

print ("This patient id already has been tested positive\nPlease enter again")

testinput()

elif flag == 3:

print ("This patient id already has been tested negative,preparing for test 2\nPlease enter again")

testinput()

elif flag == 4:

print ("This patient id already recovered\nPlease enter again")

testinput()

elif flag == 5:

print ("This patient id already dead\nPlease enter again")

testinput()

else :

r=open("patient.txt","r") # read patient data

declare sp , zone , r ,ptgroup , ptname ,ptcontact

LOOP list1 FROM r :

sp = list1.split("><")

if (ptnumber == int(sp[0])):

ptgroup = str(sp[3].strip("\n"))

ptname = str(sp[1])

ptcontact = str(sp[2])

if (ptnumber%4==1):

zone = str("zone A")

elif (ptnumber%4==2):

zone = str("zone B")

elif (ptnumber%4==3):

zone = str("zone C")

else:

zone = str("zone D")

END if

print ("Patient name is:",ptname,"\nPatient id is:",ptnumber,"\nPatient contact is:",ptcontact,"\nPatient group is:",ptgroup,"\nPatient zone is:",zone)

declare positivelist,totallist

positivelist=[]

totallist=[]

r2 = open ("positivecase.txt","r") #adding number from positive list

Declare a, b

LOOP cnt FROM r2 :

a = cnt.split("\t")

b = a[0].strip()

totallist.append (b.strip())

r2.close()

END if

END LOOP

declare test1,w

if(ptgroup == "ATO" or ptgroup=="ACC" or ptgroup=="AEO"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient QHNF")

print ("Stay at ICU/NW :")

read place

while place != "ICU" and place != "NW":

print ("Invalid input")

print ("Stay at ICU/NW :")

read place

END while

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient QDFR and preparing for test 2")

w = open ("negativecase1.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

print ("Please type again :")

read test1

testinput()

END if

elif(ptgroup =="SID"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient QHNF")

place = str(input("Stay at ICU/NW :"))

if place != "ICU" and place != "NW":

print ("Invalid input")

function()

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient HQFR and preparing for test 2")

w = open ("negativecase1.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

print ("Please type again :")

read test1

testinput()

END if

elif(ptgroup =="AHS"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient HQNF")

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient CWFR and preparing for test 2")

negativelist.append(ptnumber)

w = open ("negativecase1.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

test1= int(input("Please type again :"))

testinput()

END if

END if

declare caseidlist , adc , a , b , p1 , idc

caseidlist = []

if test1== 1:

r = open ("caseactive.txt","r")

LOOP cnt FROM r:

a=cnt.split("\t")

b=a[0].split()

caseidlist.append (b)

p1.close ()

idc = len(caseidlist)

p1 = open ("caseactive.txt","a")

p1.write(str(idc)+"\t"+str(ptnumber)+"\t"+"PENDING"+"\n")

print ("Case id for patient id",ptnumber,"is:",idc)

r.close()

END loop

END if

print("\n")

print("ˇ" \* 70)

print ("Patient ID in active case is :",totallist,"\nTotal number of active case is :",len(totallist))

print ("Patient information has been updated suscessful")

function()

END

PROGRAM test2

BEGIN

declare flag,ptlist,flag,r1,r2,r3,r4,r5,ptnumber,list1,list2,list3,list4,list5,a

flag = 1

ptlist =[]

while (flag == 1 or flag == 2 or flag == 3 or flag==4 or flag ==5):

print ("\nEnter patient ID:")

read ptnumber

flag = 0

r1 = open ("patient.txt","r") # checking range of patient

LOOP list1 FROM r1:

a = list1.split("><")

ptlist.append (a[0])

END LOOP

LOOP a FROM ptlist STEP 1:

if (ptnumber != int(a)):

flag = 1

elif (ptnumber == int(a)):

flag = 0

break

END if

END LOOP

r1.close()

r3 = open ("negativecase2.txt","r") # checking overlapping in negative

LOOP a FROM r3 :

a = list3.strip()

if (ptnumber == int(a)):

flag = 3

break

END id

END LOOP

r3.close()

r4 = open ("negativecase1.txt","r")

LOOP list4 FROM r4 :

a = list4.strip()

if (ptnumber == int(a)):

flag = 4

break

r2 = open ("positivecase.txt","r") # checking overlapping in positive

LOOP list2 FROM r2 :

list2 = list2.split("\t")

a = list2[0].strip()

if (ptnumber == int(a)):

flag = 2

break

END if

END LOOP

r2.close()

r5 = open ("recover.txt","r")# checking overlapping in recover

LOOP list5 FROM r5 :

a = list5.strip()

if (ptnumber ==int(a)):

flag = 5

break

END if

END LOOP

r6 = open ("dead.txt","r") # checking overlapping in dead

LOOP list6 FROM r6 :

a = list6.strip()

if (ptnumber ==int(a)):

flag = 6

break

END if

END LOOP

r6.close()

if flag == 1:

print ("This patient id haven't register\nPlease enter again")

testinput()

elif flag == 2:

print ("This patient id already has been tested positive\nPlease enter again")

testinput()

elif flag == 3:

print ("This patient id already has been tested negative,preparing for test 2\nPlease enter again")

testinput()

elif flag == 5:

print ("This patient id already recovered\nPlease enter again")

testinput()

elif flag == 6:

print ("This patient id already dead\nPlease enter again")

testinput()

elif flag == 4:

r=open("patient.txt","r") # read patient data

declare sp , zone , r ,ptgroup , ptname ,ptcontact

LOOP list1 FROM r :

sp = list1.split("><")

if (ptnumber == int(sp[0])):

ptgroup = str(sp[3].strip("\n"))

ptname = str(sp[1])

ptcontact = str(sp[2])

if (ptnumber%4==1):

zone = str("zone A")

elif (ptnumber%4==2):

zone = str("zone B")

elif (ptnumber%4==3):

zone = str("zone C")

else:

zone = str("zone D")

END if

print ("Patient name is:",ptname,"\nPatient id is:",ptnumber,"\nPatient contact is:",ptcontact,"\nPatient group is:",ptgroup,"\nPatient zone is:",zone)

declare positivelist,totallist

positivelist=[]

totallist=[]

r2 = open ("positivecase.txt","r") #adding number from positive list

Declare a, b

LOOP cnt FROM r2 :

a = cnt.split("\t")

b = a[0].strip()

totallist.append (b.strip())

r2.close()

END if

END LOOP

declare test1,w

if(ptgroup == "ATO" or ptgroup=="ACC" or ptgroup=="AEO"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient QHNF")

print ("Stay at ICU/NW :")

read place

while place != "ICU" and place != "NW":

print ("Invalid input")

print ("Stay at ICU/NW :")

read place

END while

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient QDFR and preparing for test 3")

w = open ("negativecase2.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

print ("Please type again :")

read test1

testinput()

END if

elif(ptgroup =="SID"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient HQNF")

place = str(input("Stay at ICU/NW :"))

if place != "ICU" and place != "NW":

print ("Invalid input")

function()

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient HQFR and preparing for test 3")

negativelist.append(ptnumber)

w = open ("negativecase2.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

print ("Please type again :")

read test1

testinput()

END if

elif(ptgroup =="AHS"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient HQNF")

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient CWFR and preparing for test 3")

negativelist.append(ptnumber)

w = open ("negativecase2.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

test1= int(input("Please type again :"))

testinput()

END if

END if

r.close()

declare caseidlist , adc , a , b , p1 , idc

caseidlist = []

if test1== 1:

r = open ("caseactive.txt","r")

LOOP cnt FROM r:

a=cnt.split("\t")

b=a[0].split()

caseidlist.append (b)

idc = len(caseidlist)

p1 = open ("caseactive.txt","a")

p1.write(str(idc)+"\t"+str(ptnumber)+"\t"+"PENDING"+"\n")

print ("Case id for patient id",ptnumber,"is:",idc)

p1.close ()

r.close()

END LOOP

END if

print("\n")

print("ˇ" \* 70)

print ("Patient ID in active case is :",totallist,"\nTotal number of active case is :",len(totallist))

print ("Patient information has been updated suscessful")

function()

else:

print ("This patient haven't enter test 1 or test 2 result ")

testinput()

END if

END

#TEST 3

PROGRAM test3():

BEGIN

declare flag,ptlist,flag,r1,r2,r3,r4,r5,ptnumber,list1,list2,list3,list4,list5,a

flag = 1

ptlist =[]

while (flag == 1 or flag == 2 or flag == 3 or flag==4 or flag ==5):

print ("\nEnter patient ID:")

read ptnumber

flag = 0

r1 = open ("patient.txt","r") # checking range of patient

LOOP list1 FROM r1:

a = list1.split("><")

ptlist.append (a[0])

END LOOP

LOOP a FROM ptlist STEP 1:

if (ptnumber != int(a)):

flag = 1

elif (ptnumber == int(a)):

flag = 0

break

END if

END LOOP

r1.close()

r3 = open ("negativecase3.txt","r") # checking overlapping in negative

LOOP a FROM r3 :

a = list3.strip()

if (ptnumber == int(a)):

flag = 3

break

END id

END LOOP

r3.close()

r4 = open ("negativecase2.txt","r")

LOOP list4 FROM rr :

a = list4.strip()

if (ptnumber == int(a)):

flag = 4

break

r4.close()

r2 = open ("positivecase.txt","r") # checking overlapping in positive

LOOP list2 FROM r2 :

list2 = list2.split("\t")

a = list2[0].strip()

if (ptnumber == int(a)):

flag = 2

break

END if

END LOOP

r2.close()

r5 = open ("recover.txt","r")# checking overlapping in recover

LOOP list5 FROM r5 :

a = list5.strip()

if (ptnumber ==int(a)):

flag = 5

break

END if

END LOOP

r5.close()

r6 = open ("dead.txt","r") # checking overlapping in dead

LOOP list6 FROM r6 :

a = list6.strip()

if (ptnumber ==int(a)):

flag = 6

break

END if

END LOOP

if flag == 1:

print ("This patient id haven't register\nPlease enter again")

testinput()

elif flag == 2:

print ("This patient id already has been tested positive\nPlease enter again")

testinput()

elif flag == 3:

print ("This patient id already has been tested negative,preparing for test 2\nPlease enter again")

testinput()

elif flag == 5:

print ("This patient id already recovered\nPlease enter again")

testinput()

elif flag == 6:

print ("This patient id already dead\nPlease enter again")

testinput()

elif flag == 4:

r=open("patient.txt","r") # read patient data

declare sp , zone , r ,ptgroup , ptname ,ptcontact

LOOP list1 FROM r :

sp = list1.split("><")

if (ptnumber == int(sp[0])):

ptgroup = str(sp[3].strip("\n"))

ptname = str(sp[1])

ptcontact = str(sp[2])

if (ptnumber%4==1):

zone = str("zone A")

elif (ptnumber%4==2):

zone = str("zone B")

elif (ptnumber%4==3):

zone = str("zone C")

else:

zone = str("zone D")

END if

print ("Patient name is:",ptname,"\nPatient id is:",ptnumber,"\nPatient contact is:",ptcontact,"\nPatient group is:",ptgroup,"\nPatient zone is:",zone)

declare positivelist,totallist

positivelist=[]

totallist=[]

r2 = open ("positivecase.txt","r") #adding number from positive list

Declare a, b

LOOP cnt FROM r2 :

a = cnt.split("\t")

b = a[0].strip()

totallist.append (b.strip())

r2.close()

END if

END LOOP

declare test1,w

if(ptgroup == "ATO" or ptgroup=="ACC" or ptgroup=="AEO"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient QHNF")

print ("Stay at ICU/NW :")

read place

while place != "ICU" and place != "NW":

print ("Invalid input")

print ("Stay at ICU/NW :")

read place

END while

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient RU")

w = open ("negativecase3.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

print ("Please type again :")

read test1

testinput()

END if

elif(ptgroup =="SID"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient QHNF")

place = str(input("Stay at ICU/NW :"))

while place != "ICU" and place != "NW":

print ("Invalid input")

function()

END while

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient RU")

negativelist.append(ptnumber)

w = open ("negativecase3.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

print ("Please type again :")

read test1

testinput()

END if

elif(ptgroup =="AHS"):

print ("\nEnter patient test 1 result :\n0.Return to option menu\n1.Positive \n2.Negative\nPlease select your option:")

read test1

if(test1 == 1):

print ("Advise patient HQNF")

positivelist.append(ptnumber)

LOOP positive FROM positivelist STEP 1:

totallist.append (str(ptnumber))

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

elif(test1 == 2):

print ("Advise patient CW")

negativelist.append(ptnumber)

w = open ("negativecase3.txt","a")

w.write (str(ptnumber)+"\n")

w.close()

elif(test1 == 0):

function()

else:

print ("Invalid input ")

test1= int(input("Please type again :"))

testinput()

END if

END if

r.close()

declare caseidlist , adc , a , b , p1 , idc

caseidlist = []

if test1== 1:

r = open ("caseactive.txt","r")

LOOP cnt FROM r:

a=cnt.split("\t")

b=a[0].split()

caseidlist.append (b)

idc = len(caseidlist)

p1 = open ("caseactive.txt","a")

p1.write(str(idc)+"\t"+str(ptnumber)+"\t"+"PENDING"+"\n")

print ("Case id for patient id",ptnumber,"is:",idc)

p1.close ()

r.close()

END LOOP

END if

print("\n")

print("ˇ" \* 70)

print ("Patient ID in active case is :",totallist,"\nTotal number of active case is :",len(totallist))

print ("Patient information has been updated suscessful")

function()

else:

print ("This patient haven't enter test 1 or test 2 result ")

testinput()

END if

END

PROGRAM testinput():

declare check,cnt

check = 0

cnt = 1

while (cnt == 1):

print("\nSelect test (0,1,2,3) result that you want upload\n0.Return to main option\n1.Enter test 1 result\n2.Enter test 2 result\n3.Enter test 3 result")

print("Please select your command :")

read check

while (check >3 or check<0):

print("\nInvalid value")

print("Please insert your value again:")

read check

END while

if (check >= 1 and check <=3):

if (check == 1):

test1()

elif (check == 2):

test2()

elif (check == 3):

test3()

elif (check == 0):

function()

else:

cnt=1

END if

else:

cnt=1

END if

END while

PROGRAM edit():

BEGIN

declare totallist,ptlist,k,u,caseid

totallist = []

ptlist = []

k = open ("positivecase.txt","r")

LOOP u FROM k:

u = u.split("\t")

totallist.append(int(u[0].strip()))

k.close()

END LOOP

print ("Enter patient case ID:")

read caseid

if (caseid>len(totallist)):

print ("Invalid case ID")

function()

END if

declare ptnumber

ptnumber = int(totallist[caseid-1])

declare r1,list1,a,flag

r1 = open ("patient.txt","r") # checking range of patient

LOOP list1 FROM r1:

a = list1.split("><")

ptlist.append (a[0])

LOOP a FROM ptlist(1st item) TO last item STEP 1:

if (ptnumber != int(a)):

flag = 1

elif (ptnumber == int(a)):

flag = 0

break

END if

END LOOP

r1.close()

if (flag == 1):

print ("This patient ID haven't register")

function ()

END if

declare f,line,a,flag

with open ("caseactive.txt","r")as f:

LOOP line FROM f:

a=line.split("\t")

if (str(caseid) == a[0]):

if(a[2].strip() != "PENDING"):

flag = 0

break

else:

flag = 1

END if

END if

END LOOP

if flag == 0:

print ("This ACTIVE case status already changed\nPlease enter again:")

function()

END if

declare sp,zone,ptgroup,ptname,ptcontact

r=open("patient.txt","r")

LOOP list1 FROM r:

sp = list1.split("><")

if (ptnumber == int(sp[0])):

ptgroup = str(sp[3].strip("\n"))

ptname = str(sp[1])

ptcontact = str(sp[2])

if (ptnumber%4==1):

zone = str("zone A")

elif (ptnumber%4==2):

zone = str("zone B")

elif (ptnumber%4==3):

zone = str("zone C")

else:

zone = str("zone D")

END if

END if

END LOOP

r.close()

declare positivelist,h,list1,cnt

positivelist = []

print ("Patient name is:",ptname,"\nPatient id is :",ptnumber,"\nPatient contact is :",ptcontact,"\nPatient group is :",ptgroup,"\nPatient zone is:",zone)

h = open ("positivecase.txt","r")

LOOP list1 FROM h:

list1 = list1.split("\t")

positivelist.append(list1[0].strip("\n")) #read value to positive list

END LOOP

h.close()

LOOP cnt FROM positivelist:#checking positivelist

cnt = cnt.split("\t")

cnt = cnt[0].strip()

if(ptnumber == int(cnt)):

print("\nChanging status to RECOVERED = 1\nChanging status to DECEASED = 2")

declare status,r,filedata,f,line,new,newdata,a

print ("Please enter your command")

read status

while (status < 1 or status > 2):

print ("Please enter your command")

read status

END while

if (status == 1 or status == 2):

if(status == 1):

r = open ("recover.txt","a")

r.write (str(ptnumber)+"\n")

r.close()

declare filedata

filedata = ""

with open ("caseactive.txt","r")as f:

LOOP line FROM f:

a=line.split("\t")

if (str(caseid) == a[0] ):

new = a[2].replace("PENDING","RECOVERED")

newdata = str(caseid)+"\t"+a[1]+"\t"+new

filedata = filedata + newdata

else:

filedata = filedata + line

END if

END LOOP

with open("caseactive.txt","w") as f:

f.write(filedata)

print ("Caseid :",caseid,"\tStatus : RECOVERED","\tPatient ID:",ptnumber)

print ("Cases of patient ID infected:",positivelist)

print ("Total case of infected status:",len(positivelist))

print ("Patient information has been updated suscessful")

function()

elif(status == 2):

r = open ("dead.txt","a")

r.write (str(ptnumber)+"\n")

r.close()

file\_data = ""

with open ("caseactive.txt","r")as f:

for line in f:

a=line.split("\t")

if (str(caseid) == a[0] ):

new = a[2].replace("PENDING","DECEASED")

newdata = str(caseid)+"\t"+a[1]+"\t"+new

file\_data = file\_data + newdata

else:

file\_data = file\_data + line

END if

END LOOP

with open("caseactive.txt","w") as f:

f.write(file\_data)

print ("Caseid :",caseid,"\tStatus : DECEASED","\tPatient ID:",ptnumber)

print ("Cases of patient ID infected:",positivelist)

print ("Total case of infected status:",len(positivelist))

print ("Patient information has been updated suscessful")

function()

END if

END if

END LOOP

print ("\nInvalid input\nThis patient ID havent insert to ACTICE case")

function()

END

#4.1

PROGRAM testoutput ():

BEGIN

declare ptlist,ptlist1,ptlist2,ptlist3,ptlist4

ptlist = []

ptlist1 = []

ptlist2 = []

ptlist3 = []

ptlist4 = []

print("ˇ" \* 70 )

declare r,cnt,a

print ("\n\n\t\t-----$$$ ALL POSITIVE TEST $$$------")

r = open ("positivecase.txt","r")

LOOP cnt FROM r:

a = cnt.split("\t")

a = a[0].strip("\n")

ptlist.append (a)

ptlist4.append (a)

END LOOP

r.close ()

print ("\n\tAll patient id in active case are ",ptlist)

print ("\n\tTotal number tested carry out in active case are ",len(ptlist))

print("\n\t\t------$$$ NEGATIVE TEST 1 $$$-------")

declare r,cnt,a

r = open ("negativecase1.txt","r")

LOOP cnt FROM r:

a = cnt.strip("\n")

ptlist1.append (a)

ptlist4.append (a)

END LOOP

print ("\n\tAll patient id tested negetive in test 1 ",ptlist1)

print ("\n\tNumber of tests carried out in NEGATIVE TEST 1 ",len(ptlist1))

r.close ()

print("\n\t\t------$$$ NEGATIVE TEST 2 $$$-------")

declare r,cnt,a

r = open ("negativecase2.txt","r")

LOOP cnt FROM r:

a = cnt.strip("\n")

ptlist2.append (a)

ptlist4.append (a)

END LOOP

print ("\n\tAll patient id tested negetive in test 2 ",ptlist2)

print ("\n\tNumber of tests carried out in NEGATIVE TEST 2 ",len(ptlist2))

r.close ()

print("\n\t\t------$$$ NEGATIVE TEST 3 $$$-------")

declare r,cnt,a

r = open ("negativecase3.txt","r")

LOOP cnt FROM r:

a = cnt.strip("\n")

ptlist3.append (a)

ptlist4.append (a)

END LOOP

print ("\n\tAll patient id tested negetive in test 3 ",ptlist3)

print ("\n\tNumber of tests carried out in NEGATIVE TEST 3 ",len(ptlist3))

r.close ()

print ("\n\t\tTotal number of tested carry out",len(ptlist4))

END

#4.2

PROGRAM patient():

BEGIN

declare ptlist,ptlist1

ptlist = []

ptlist1 = []

print("ˇ" \* 70 )

declare cnt,r,a

print ("\n\n\t\t-----$$$ TOTAL OF PATIENT TESTED $$$------")

r = open ("positivecase.txt","r")

LOOP cnt FROM r:

a = cnt.strip("\n")

a = a[0].strip("\n")

ptlist.append (a)

r.close ()

END LOOP

declare cnt,r,a

r = open ("negativecase1.txt","r")

LOOP cnt FROM r:

a = cnt.strip("\n")

ptlist.append (a)

END LOOP

r.close ()

LOOP cnt FROM ptlist STEP 1:

if cnt not in ptlist1:

ptlist1.append (cnt)

END if

END LOOP

print ("\n\t\tTotal number of patient tested is",len(ptlist1))

END

#4.3

PROGRAM recovered ():

BEGIN

declare ptlist,r,cnt,a

ptlist=[]

r = open ("recover.txt","r")

LOOP cnt FROM r:

a = cnt.strip("\n")

ptlist.append (a)

print ("\n\n\t\t-----$$$ RECOVERED CASE $$$------")

print ("\tAll patient id that had been recovered are",ptlist)

print ("\tTotal number case of patient that had been recovered are",len(ptlist))

r.close()

END

#4.4

PROGRAM positive():

BEGIN

declare caselist,ato,acc,aeo,sid,ahs

caselist = []

ato = []

acc = []

aeo = []

sid = []

ahs = []

declare cnt,a,r

r = open ("positivecase.txt","r")

LOOP cnt FROM r:

a = cnt.strip("\n")

a = a[0].strip("\n")

caselist.append(a)

r.close()

caselist.sort

declare cnt2,r1,cnt2,a,b

r1 = open ("patient.txt","r")

LOOP cnt2 FROM r1:

LOOP cnt FROM caselist STEP1:

a = cnt2.split("><")

b = a[3].strip()

if cnt == a[0]:

if b == "ATO":

a[0]=int(a[0])

ato.append (a[0])

elif b == "ACC":

a[0]=int(a[0])

acc.append (a[0])

elif b == "AEO":

a[0]=int(a[0])

aeo.append (a[0])

elif b == "SID":

a[0]=int(a[0])

sid.append (a[0])

elif b == "AHS":

a[0]=int(a[0])

ahs.append (a[0])

END if

END if

print("\n\n\t\t-----$$$ TOTAL POSITIVE CASE $$$------")

print("\tTotal cases if positive tested patient are",len(caselist))

print("\n\n\t\t-----$$$ TOTAL POSITIVE CASE IN ATO GROUP $$$------")

print("\n\tTotal cases in ATO group are",len(ato))

print("\n\n\t\t-----$$$ TOTAL POSITIVE CASE IN ATO GROUP $$$------")

print("\n\tTotal cases in ACC group are",len(acc))

print("\n\n\t\t-----$$$ TOTAL POSITIVE CASE IN ATO GROUP $$$------")

print("\n\tTotal cases in AEO group are",len(aeo))

print("\n\n\t\t-----$$$ TOTAL POSITIVE CASE IN ATO GROUP $$$------")

print("\n\tTotal cases in SID group are",len(sid))

print("\n\n\t\t-----$$$ TOTAL POSITIVE CASE IN ATO GROUP $$$------")

print("\n\tTotal cases in AHS group are",len(ahs))

r1.close()

END

#4.5

PROGRAM activezone():

BEGIN

declare ptlist,f,cnt,b

plist = []

f =open ("caseactive.txt","r")

LOOP cnt FROM f:

b = cnt.split("\t")

status = b[2].strip("\n")

if (status == "ACTIVE"):

plist.append (int(b))

END if

END LOOP

declare table1,table2,table3,table4

table1 = []

table2 = []

table3 = []

table4 = []

declare cnt2

LOOP cnt FROM ptlist STEP 1 :

if (cnt2%4==1):

table1.append(cnt2)

elif (cnt2%4==2):

table2.append(cnt2)

elif (cnt2%4==3):

table3.append(cnt2)

else:

table4.append(cnt2)

END if

END LOOP

print("\n\n -----$$$ ACTIVE CASE AVAILABLE IN SPECIFIC ZONE $$$------")

print ("\tInfected case in ZONE A",table1)

print ("\tInfected case in ZONE B",table2)

print ("\tInfected case in ZONE C",table3)

print ("\tInfected case in ZONE D",table4)

f.close()

END

#4

PROGRAM additional ():

BEGIN

declare cnt,check

cnt = 1

while (cnt == 1):

print ("\n")

print("ˇ" \* 70)

print("Enter (0/1/2/3/4) to select system function\n0.Return to main option\n1.Total of Test(1,2,3) carried out\n2.Total patients tested\n"

"3.Total recovered cases\n4.Total patients test positive for COVID-19 group wise\n5.Total active cases zone wise")

print("ˇ" \* 70)

print("Please select your command :")

read check

while (check >5 or check<0):

print("\nInvalid value")

print("Please insert your value again:")

read check

END while

if (check >= 0 and check <=5):

if (check == 1):

testoutput()

elif (check == 2):

patient()

elif (check == 3):

recovered()

elif (check == 4):

positive()

elif (check == 5):

activezone()

elif (check == 0):

function()

else:

cnt=1

END if

else:

cnt=1

END if

END while

END

#5.1

PROGRAM searchpt():

BEGIN

declare ptnumber,r,list1,sp,ptgroup,ptname,ptcontact,zone,flag

print("Enter patient ID that you want to search:")

read ptnumber

r=open("patient.txt","r")

LOOP list1 FROM r:

sp = list1.split("><")

if (ptnumber == int(sp[0])):

ptgroup = str(sp[3].strip("\n"))

ptname = str(sp[1])

ptcontact = str(sp[2])

if (ptnumber%4==1):

zone = str("zone A")

elif (ptnumber%4==2):

zone = str("zone B")

elif (ptnumber%4==3):

zone = str("zone C")

else:

zone = str("zone D")

print ("Patient name is:",ptname,"\nPatient id is:",ptnumber,"\nPatient contact is:",ptcontact,"\nPatient group is:",ptgroup,"\nPatient zone is:",zone)

flag = 0

break

END if

else:

flag = 1

END if

END LOOP

if (flag == 1):

print ("Invalid input")

END if

r.close()

if flag == 0:

declare r3,cnt,n1,r1,b,r2,c,cnt2,c

r3 = open ("negativecase1.txt","r")

LOOP cnt FROM r3:

n1 = cnt.strip()

if n1 == str(ptnumber):

flag = 1

break

END if

END LOOP

r3.close ()

if (flag == 0):

r1 = open ("positivecase.txt","r")

for cnt in r1:

b = cnt.split("\t")

if b[0] == str(ptnumber):

print ("Patient is tested positive")

print ("Patient is Quarantine in :",b[1].strip())

r2 = open ("caseactive.txt","r")

for cnt2 in r2:

c = cnt2.split ("\t")

if c[1] == str(ptnumber):

print ("Patient case id is :",c[0])

print ("Patient status is :",c[2].strip())

r2.close()

flag=2

break

END

END LOOP

END if

r2.close()

r1.close()

END LOOP

END if

if flag == 2:

add()

END if

if flag == 1:

print ("This patient is tested negative")

add()

END if

if flag == 0:

print ("This patient havent go through any test")

END if

END

#5.2

PROGRAM searchcase():

BEGIN

declare caseid,r,cnt,a,ptnumber,flag

print ("Enter Case ID:")

read caseid

r = open ("caseactive.txt","r")

LOOP cnt FROM r:

a = cnt.split ("\t")

ptnumber = a[1]

if str(caseid) == a[0].strip():

print ("Caseid :",caseid,"\nPatient ID:",a[1],"\nStatus : ",a[2].strip())

flag = 0

break

else:

flag = 1

END if

END LOOP

if flag == 1:

print ("Invalid input")

END if

r.close()

declare r2,cnt,b,ptgroup,ptname,ptcontact,zone

r2 = open ("patient.txt","r")

LOOP cnt FROM r2:

b = cnt.split ("><")

if ptnumber == b[0]:

ptnumber = int(ptnumber)

ptgroup = str(b[3].strip("\n"))

ptname = str(b[1])

ptcontact = str(b[2])

if (ptnumber%4==1):

zone = str("zone A")

elif (ptnumber%4==2):

zone = str("zone B")

elif (ptnumber%4==3):

zone = str("zone C")

else:

zone = str("zone D")

print ("Patient name is:",ptname,"\nPatient id is:",ptnumber,"\nPatient contact is:",ptcontact,"\nPatient group is:",ptgroup,"\nPatient zone is:",zone)

END if

END if

END LOOP

r2.close()

END

#5.3

PROGRAM searchggcase():

BEGIN

declare total,r,cnt

total = []

r = open ("dead.txt","r")

for cnt in r:

total.append (cnt.strip())

print ("\n\n\t\t-----$$$ TOTAL DECEASED CASE $$$------")

print ("\t\tTotal case of DECEASED case is",len(total))

r.close()

r = open ("patient.txt","r")

total.sort()

declare h,a,ptnumber,zone

LOOP cnt FROM r

a = cnt.split ("><")

LOOP h FROM total STEP 1:

if a[0]== h:

ptnumber = int(a[0])

if (ptnumber%4==1):

zone = str("zone A")

elif (ptnumber%4==2):

zone = str("zone B")

elif (ptnumber%4==3):

zone = str("zone C")

else:

zone = str("zone D")

END if

print ("\n\tPatient name is",a[1])

print ("\tPatient ID is",a[0])

print ("\tPatient email/contact is",a[2])

print ("\tPatient group is",a[3].strip(),"\n")

print ("\tPatient zone is",zone)

END if

END LOOP

END LOOP

r.close()

END

#5

PROGRAM add():

BEGIN

declare cnt,check

cnt = 1

while (cnt == 1):

print ("\n")

print("ˇ" \* 70)

print("Enter (0/1/2/3) to select system function\n0.Return to main option\n1.Searching patient detail by insert patient ID\n2.Searching patient detail by insert case ID\n"

"3.Total deceased cases")

print("ˇ" \* 70)

print("Please select your command :")

read check

while (check >3 or check<0):

print("\nInvalid value")

print("Please insert your value again:")

read check

END while

if (check >= 0 and check <=3):

if (check == 1):

searchpt()

elif (check == 2):

searchcase()

elif (check == 3):

searchggcase()

elif (check == 0):

function()

else:

cnt=1

END if

else:

cnt=1

END if

END while

END

PROGRAM function

BEGIN

declare cnt, check

cnt = 1

while( cnt = 1 )

print ("\n")

print ( "ˇ" \* 70)

print ("Enter (1/2/3/4/5) to select system function\n1.Register new list of patient\n2.Upload test (1,2,3) result\n3.Edit patient status""\n4.Additional function (1)\n5.Additional function (2)")

print ( "ˇ" \* 70)

print ("Please select your command :")

read check

while (check >5 or check<1):

print("\nInvalid value")

print("Please insert your value again:")

read check

END while

if (check >= 1 and check <=5):

if (check == 1):

register()

elif (check == 2):

testinput()

elif (check == 3):

edit()

elif (check == 4):

additional()

elif (check == 5):

add()

else:

cnt=1

END if

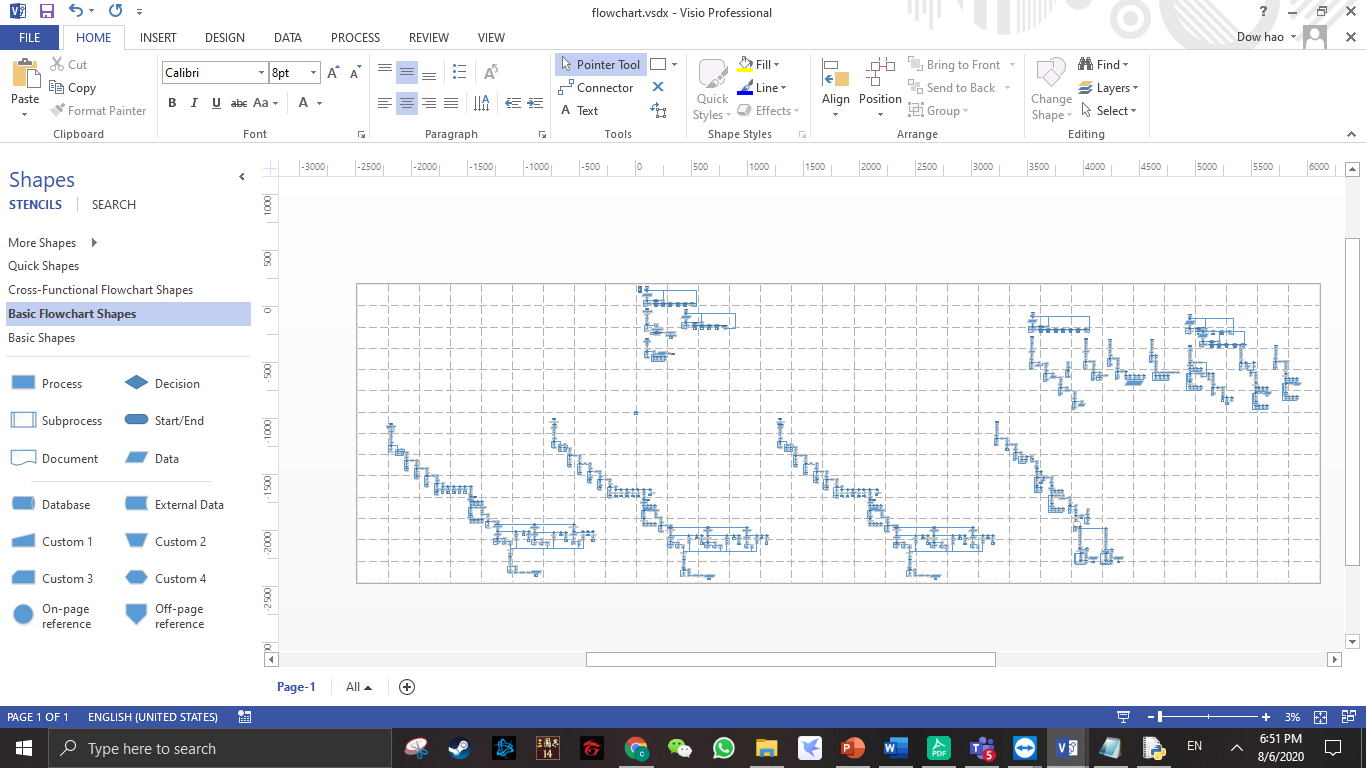
else:

cnt=1

END if

END

# Flow Chart



# Variable

Variable is a place to storing a value with a specific name of characters.

flag = 1

ptlist = []

ptname= str(input("\nEnter patient name:"))

r=open("patient.txt","r")

f=open("patient.txt","a")

Majority of this assignment, I used this variable to write my program. In the 1st example is I give a value for the specific variable to run of my program. In the 2nd example is determine the variable into list function. In the 3rd example is asking a value from the user to enter their value to run the program. In the next 2 example is open .txt file as read or append mode to ensure the python won’t getting confuse about that.

## Control structure

Control structure is letting user making decision to selecting their command or other function as they wanted.

* 1. If

if (flag == 1):

print ("This patient ID haven't register")

function ()

Control structure if is for using at while the condition is only 1, In this case of if condition, when flag equal to one, it is enable to run previous code and going to print ("This patient ID haven't register") and running the function () to return to the main menu letting user select their command again.

* 1. If – else

if(a[2].strip() != "ACTIVE"):

flag = 0

break

else:

flag = 1

In this case of if-else is when the status off the patient (a[2]) is not “ACTIVE”, the value of flag will become zero and stop it (break) , else flag value will become 1. Then the flag will be taken and continue run the program.

* 1. Nested if-else

for cnt2 in mylist:

if (cnt2%4==1):

table1.append(cnt2)

elif (cnt2%4==2):

table2.append(cnt2)

elif (cnt2%4==3):

table3.append(cnt2)

else:

table4.append(cnt2)

In this case of nested if-else is, all value of cnt2 is getting from mylist of 1st item till the last item of mylist. when cnt2 divided 4 and the remainder is 1, this patient id will append in the table1 list for separating their zone location. Same as other condition, when else if cnt2 divided 4 and the remainder is 2, this patient id will append in table 2 list. And also when the remainder is 3, it will append into table3 list. If not all the condition above it will append into table4 list. When the last item of item is read then which mean the whole process is ended.

## Looping structure

Looping structure is loop the specific code while archive the condition.

* 1. While loop

place = str(input("Stay at ICU/NW :"))

while place != "ICU" and place != "NW":

print ("Invalid input")

In this case of looping structure is asking user to enter ICU or NW. When the user type other word or invalid value, the program should able to asking the user retype the place again and print (invalid input) to ensure the program will record correctly .

* 1. For loop

r=open("patient.txt","r")

for list1 in r:

sp = list1.split("><")

ptlist.append(str(sp[0]))

r.close()

In this case, I opened the patient.txt and read the data from this file. Thus, the cade (for list1 in r) is mean read the data one line after another line. After that split them into a list and assign the value as sp. In the last I append it into ptlist and run the following code which is closing the .txt file.

1. Function without parameter

def recovered ():

ptlist=[]

r = open ("recover.txt","r")

for cnt in r:

a = cnt.strip("\n")

ptlist.append (a)

print ("\n\n\t\t-----$$$ RECOVERED CASE $$$------")

print ("\tAll patient id that had been recovered are",ptlist)

print ("\tTotal number case of patient that had been recovered are",len(ptlist))

r.close()

function means a specific code which only will execute while it is called another function or else. In this case, I declare the ptlist as a list method. Then I opened the recover.txt for reading the whole data which is store in this file. I read the file one line after the other then strip them into a list and in the last append them into ptlist. Then I print the patient id that store in ptlist and the length of ptlist to display the data to the user.

## .txt file

.txt file is a format of storing data in devices. The majority is separate by 3 modes which is “r”, “w” and “a”.

* 1. Read mode

r=open("patient.txt","r")

for list1 in r:

sp = list1.split("><")

ptlist.append(str(sp[0]))

r.close()

”r” is for telling python to read the particular the .txt file. Then open and read the data of txt.file. After that, python will assume the second code and reading the file data one line after another line. Then it will categorize them into list. And the last it will append the patient id(sp[0]) into ptlist for the following code and close the file.

* 1. Write mode

with open("caseactive.txt","w") as f:

f.write(filedata)

“w” is using for rewrite the whole .txt file and adding a new data into the file. In this case, I open the “caseactive.txt” file and assume a variable name as f. After that I replace the old file and adding new data (filedata) into it.

* 1. Append mode

w=open ("positivecase.txt","a")

w.write (str(ptnumber)+"\t"+place+"\n")

w.close()

“a” is for telling python to add new data in the text file and maintain the old data of the txt.flie. In this case , I record those ptnumner (patient id) and place (ICU/NW) in positivecase file.

## Additional features

* 1. Split

sp = list1.split("><")

The split method is split a string and categorize them in a list. In this following case, I split the data of named list1 and assume it a sp. Then I split them with using (“><”) to removing both the leading.

* 1. Strip

a = list2[0].strip()

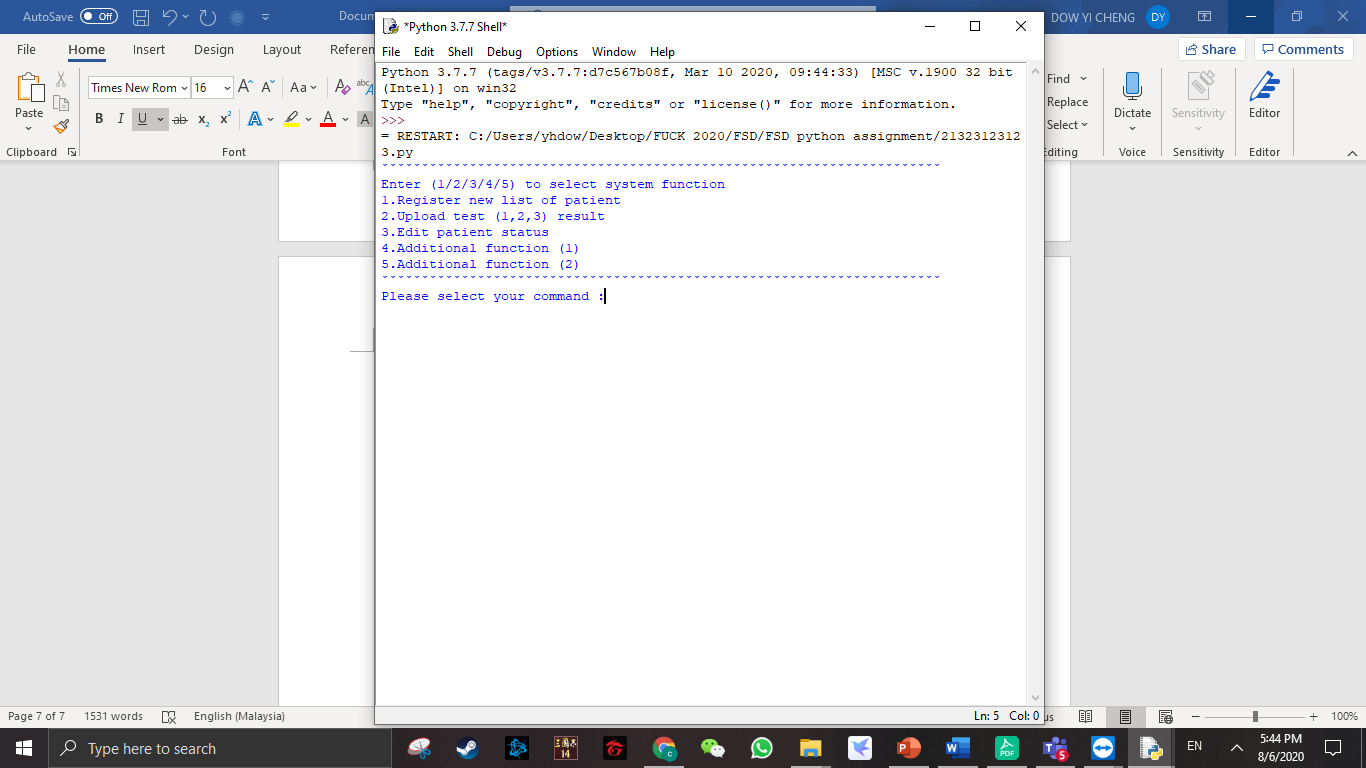
The strip method is returning a copy of the string by removing both the leading and the trailing characters (based on the string argument passed). In this case of strip, I strip the patient id (list2[0]) and assume it as “a”. Thus my “a” variable is the value of patient id (number) and without spacing and tabs(\n\t) .

* 1. Replace

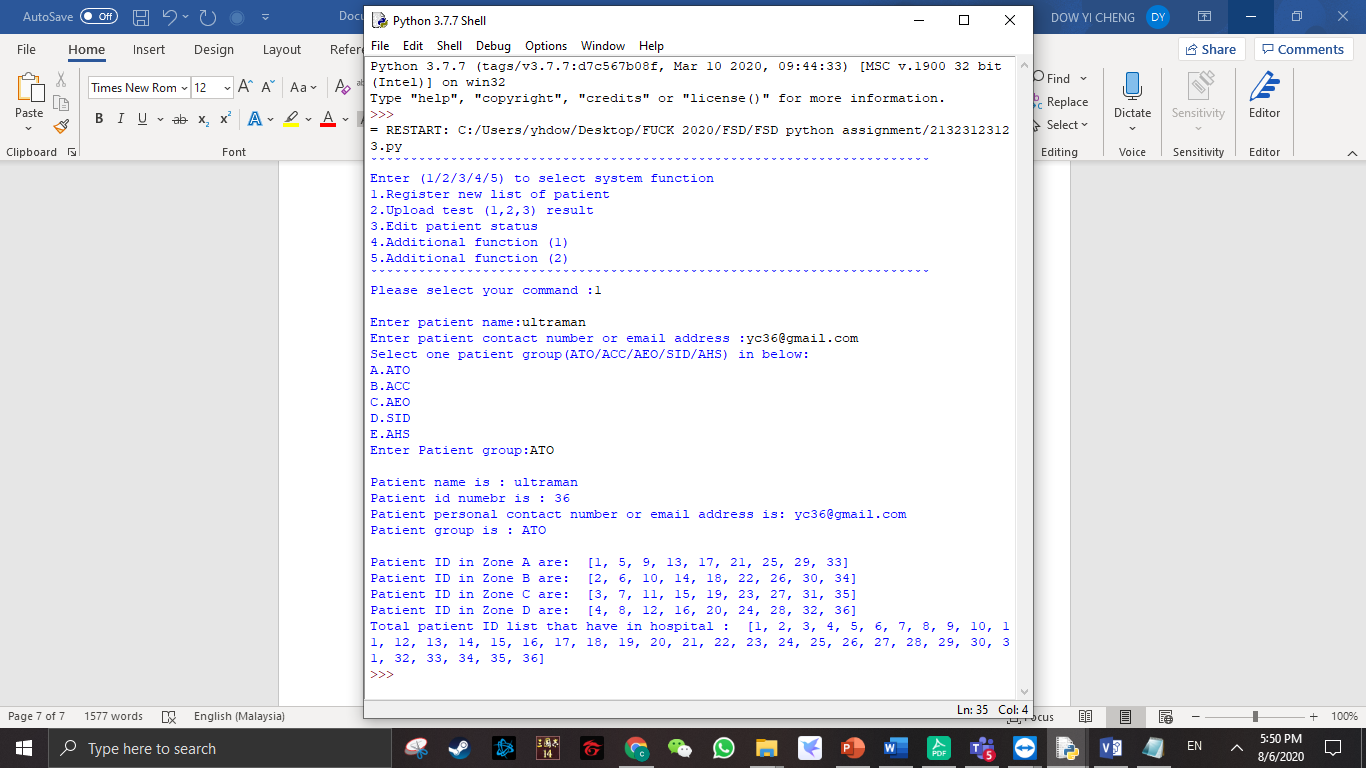
new = a[2].replace("ACTIVE","RECOVERED")

The replace method is for change an old word to another new word that you want to change. That means it is needed 2 argument to complete it. In this case of replace, I want to replace the status of patient (a[2]) and replace it from ACTIVE become RECOVERED. After that assume it as a new variable “new”.

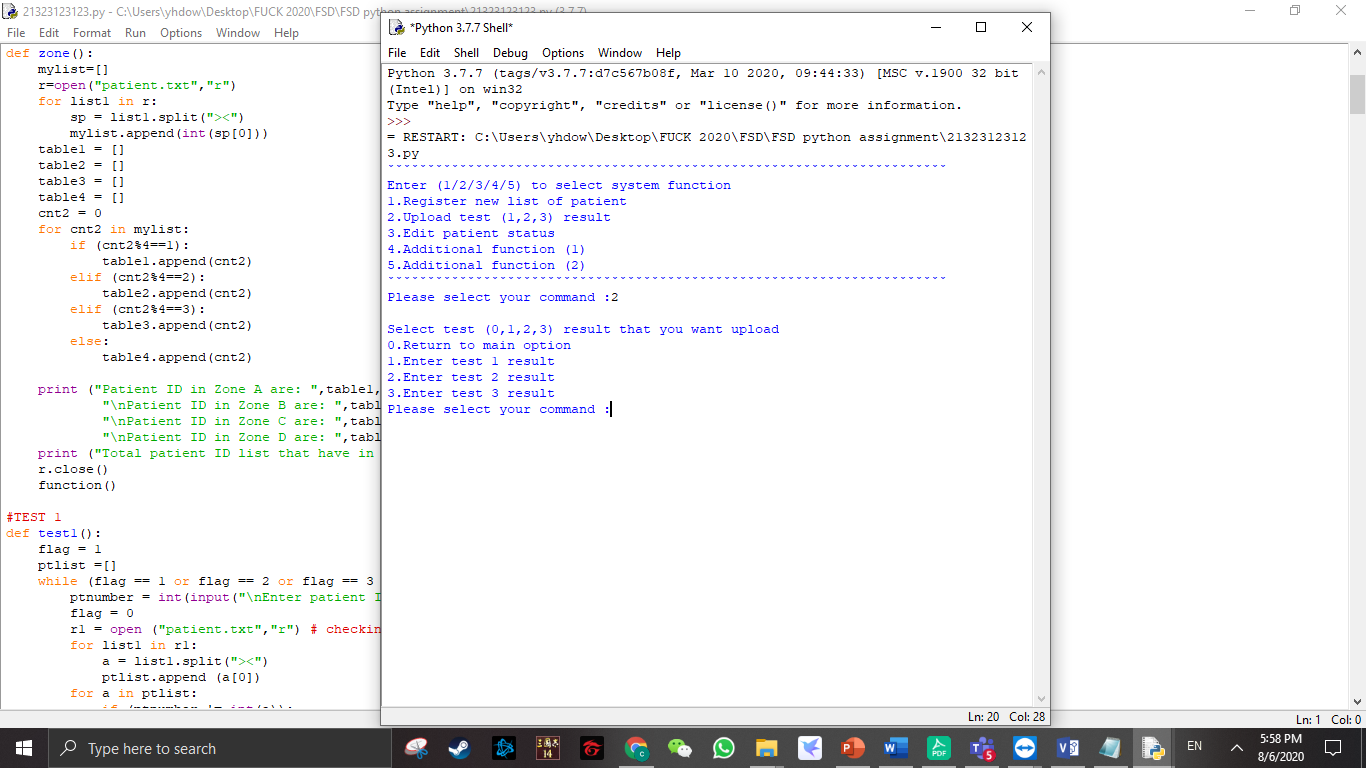
# Screenshots of I/O & Explanation



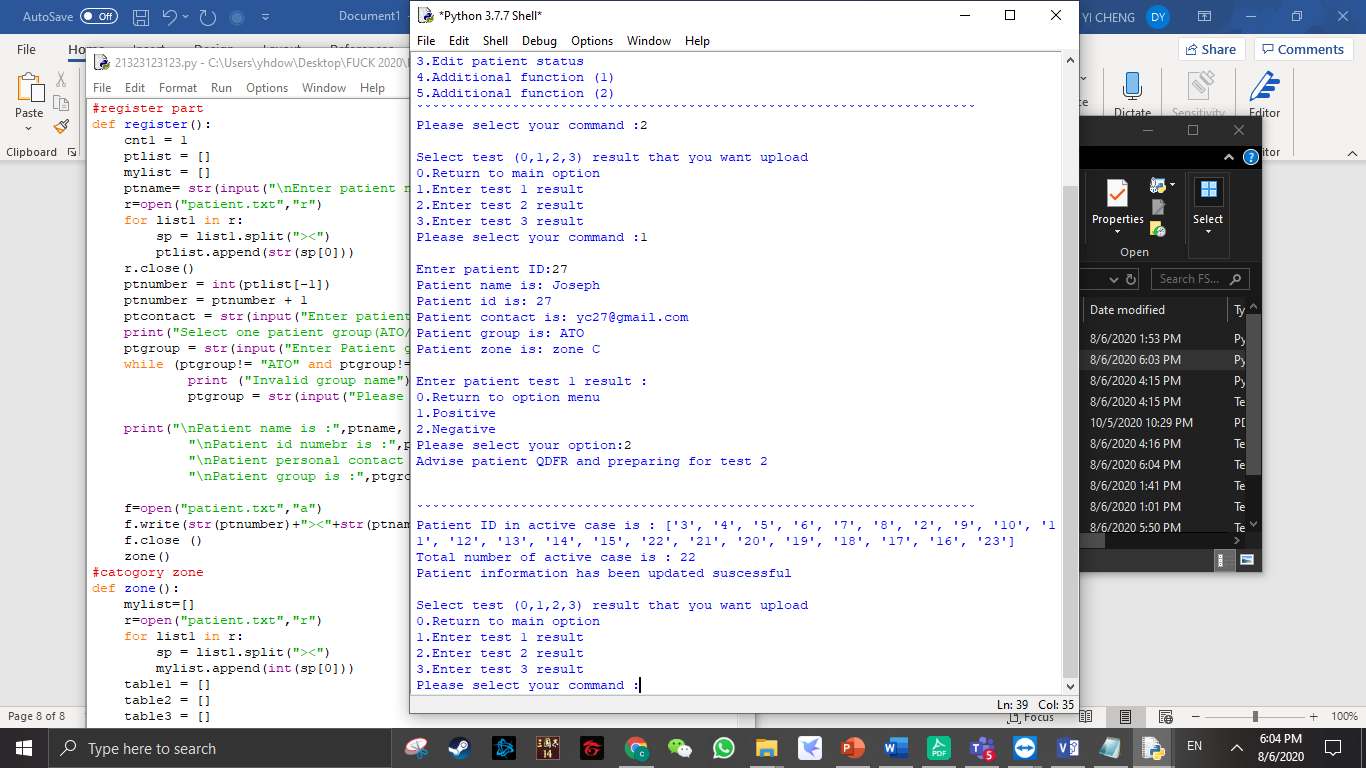
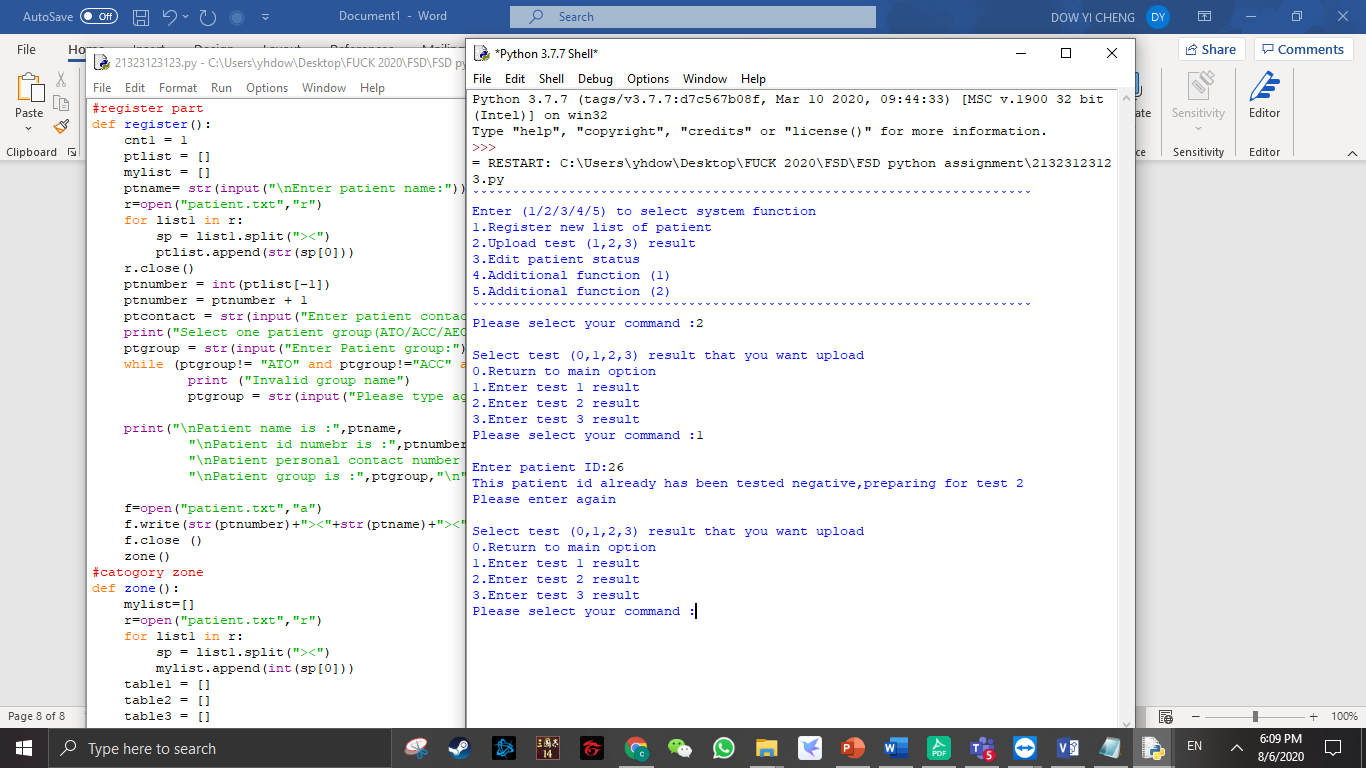
This is my main menu of my program. So, I will let the user to choosing their function as they wanted to do. They got 5 option to letting user to select by entering to particular number and it will display following code that user selected.



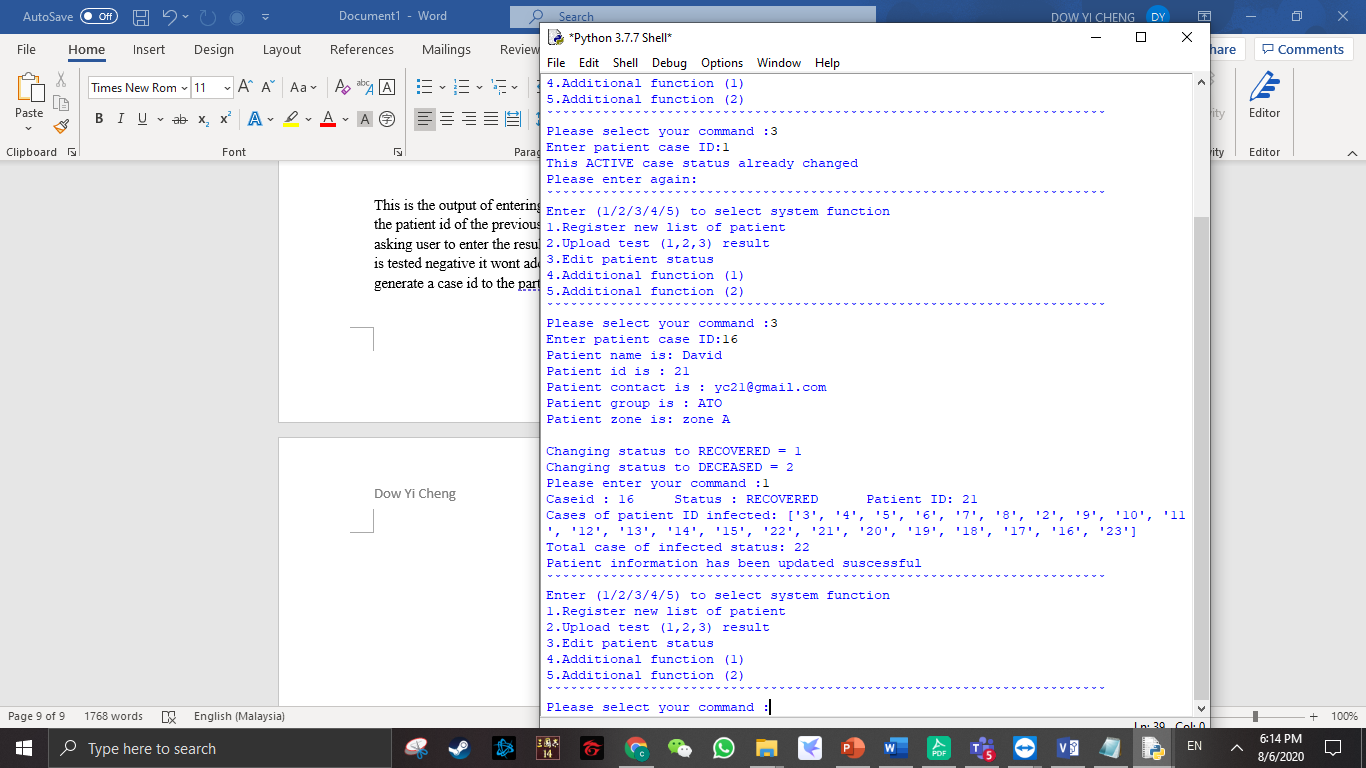
This is the generated output of register a patient (option1),user need to key in patient detail such as patient name , group ,contact. After that, program will give a patient id to the particular and display it again to the user to allow user to check information. The program also will auto generate a zone group to the patient and record in a txt.file.



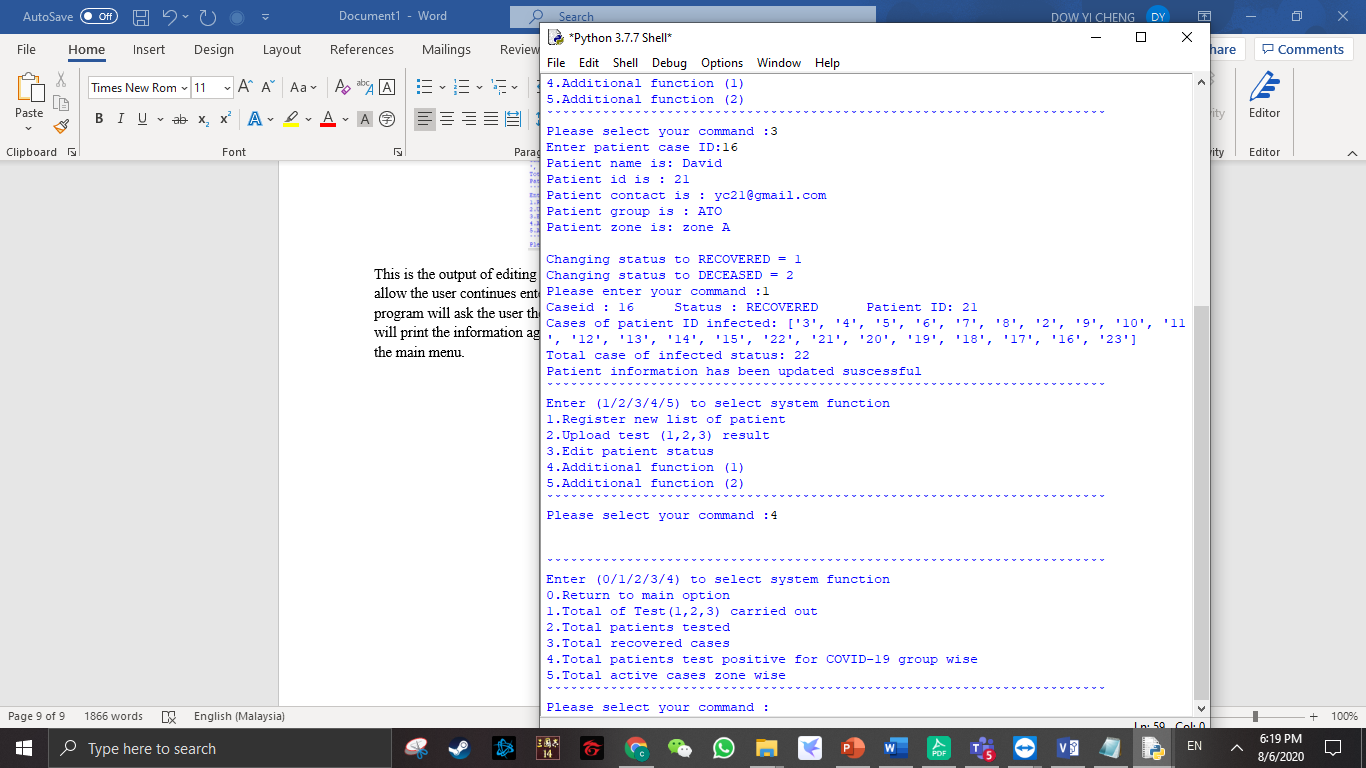
This is the output of uploading test result option (option2), After this program will asking user to enter a command to select specific test result (test1,2,3).

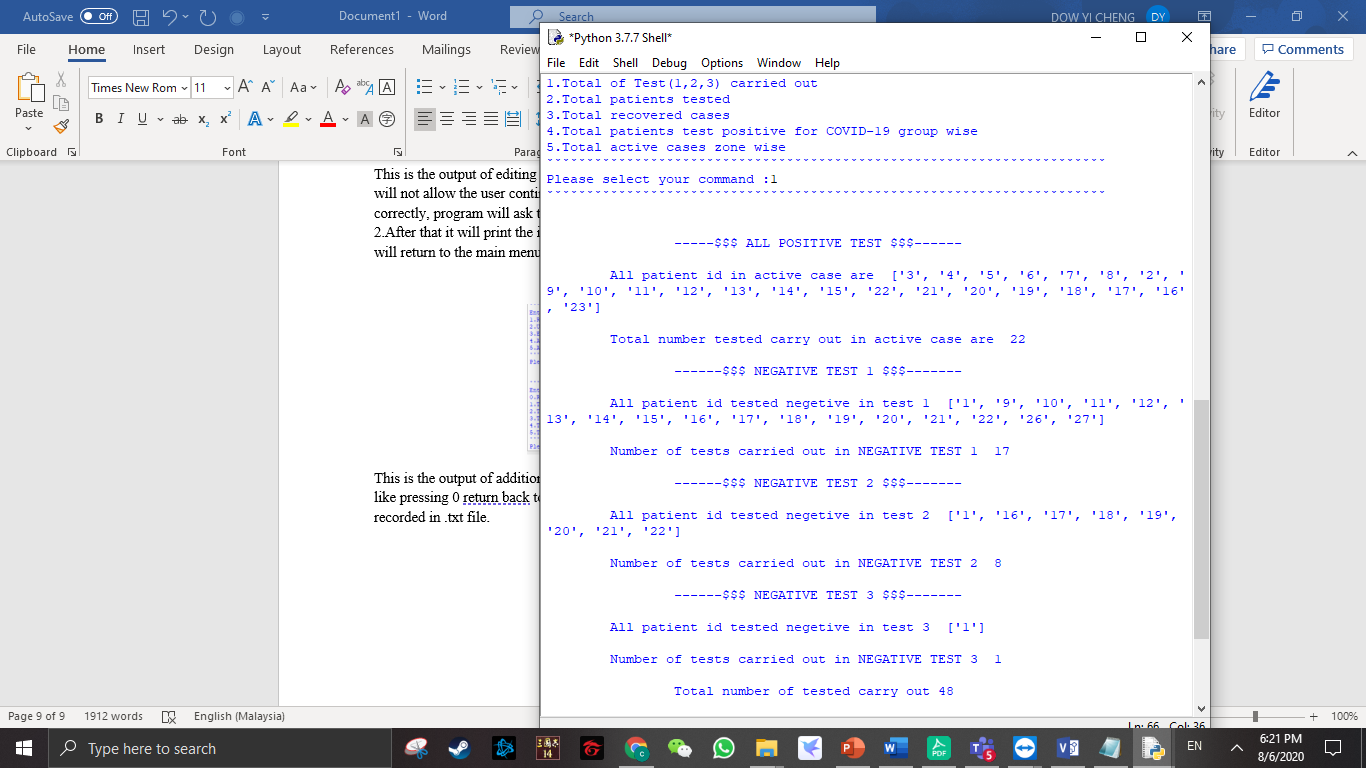
This is the output of entering result patient of test 1. So, if user type wrong command or overlapping the patient id of the previous recorded it will ask user to retype it again. If user enter correctly, its will asking user to enter the result of test1, and display it out to letting user checking. After that if patient is tested negative it wont added in case id list , if patient result is positive , the program will auto generate a case id to the particular user. After finish enter result, program will auto return to the test option to let user enter again. Test 2 and test 3 are similar to the test 1 above.



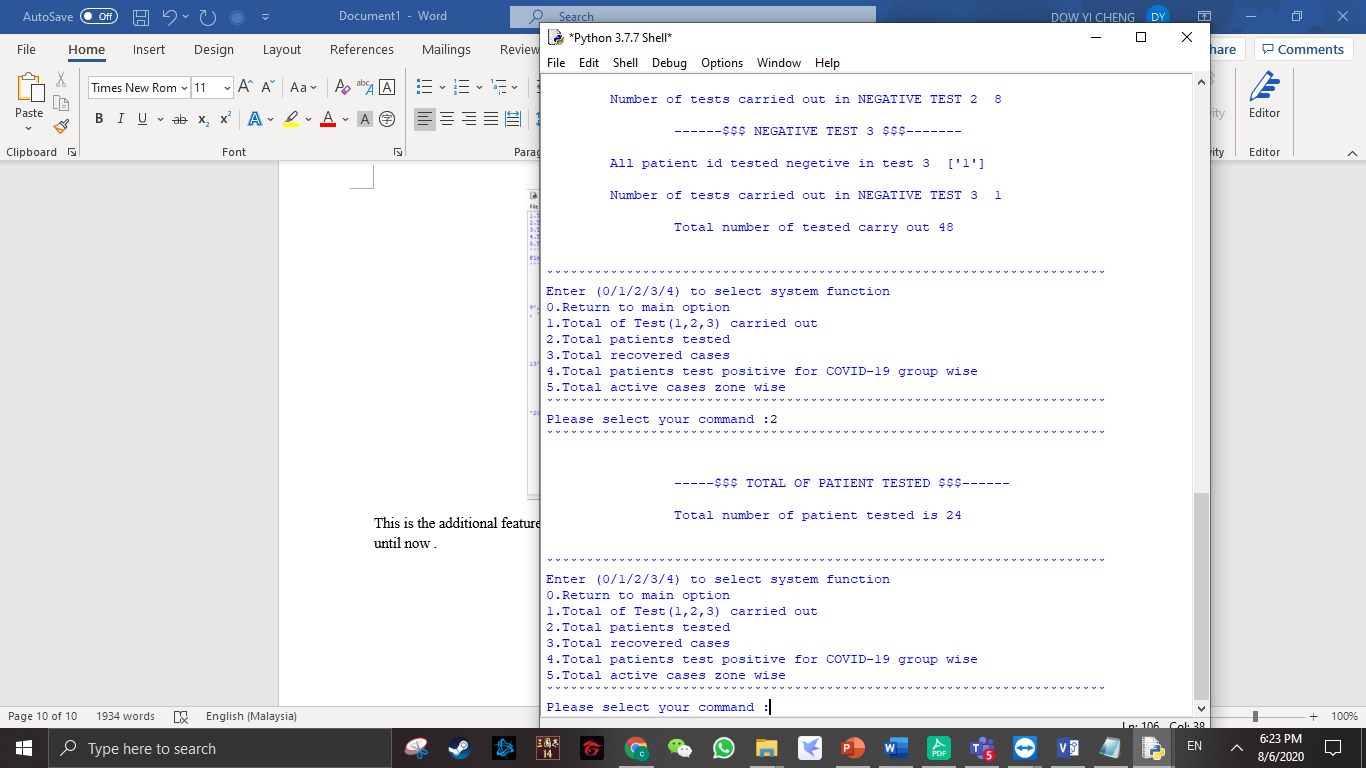
This is the output of editing patient result (option 3), if the case id is overlapping again , the program will not allow the user continues enter result and return back into the main option list. If user enter correctly, program will ask the user the status (recovered or deceased ) of the patient by select 1 or 2.After that it will print the information again to let user to checking those information. In the last it will return to the main menu.



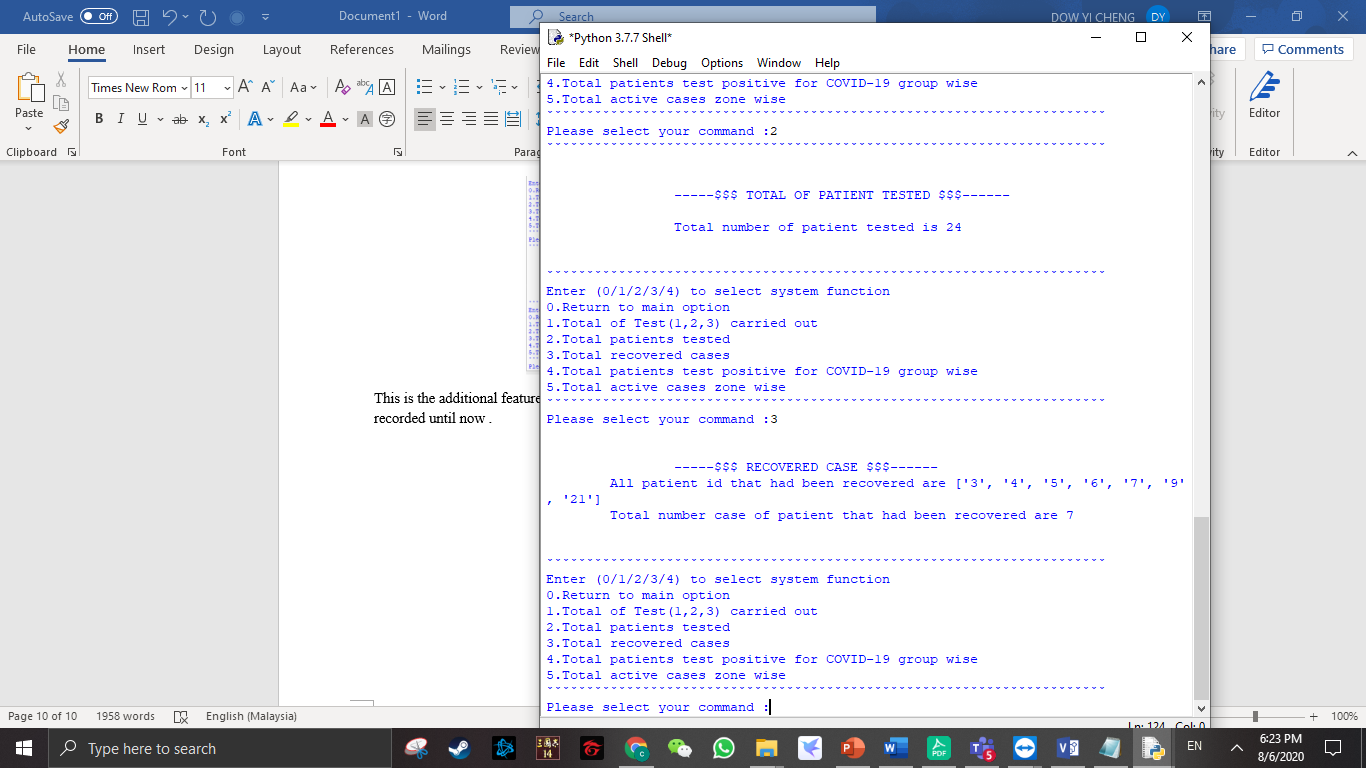
This is the output of additional features (option 4), there got 6 option to let user to select. Example like pressing 0 return back to the main menu. 1-5 option is print the total of the data that has been recorded in .txt file.



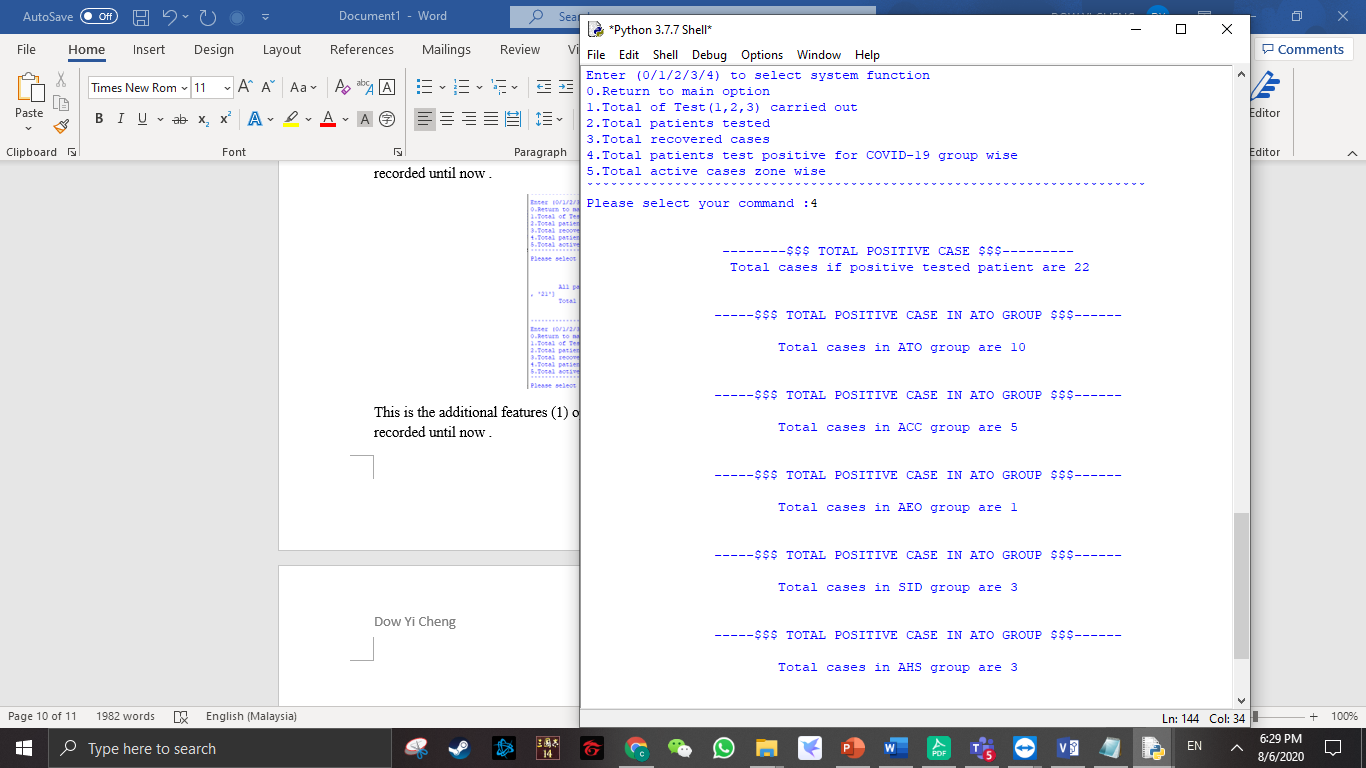
This is the additional features (1) option (1) output , It will display all total tested has been recorded until now .



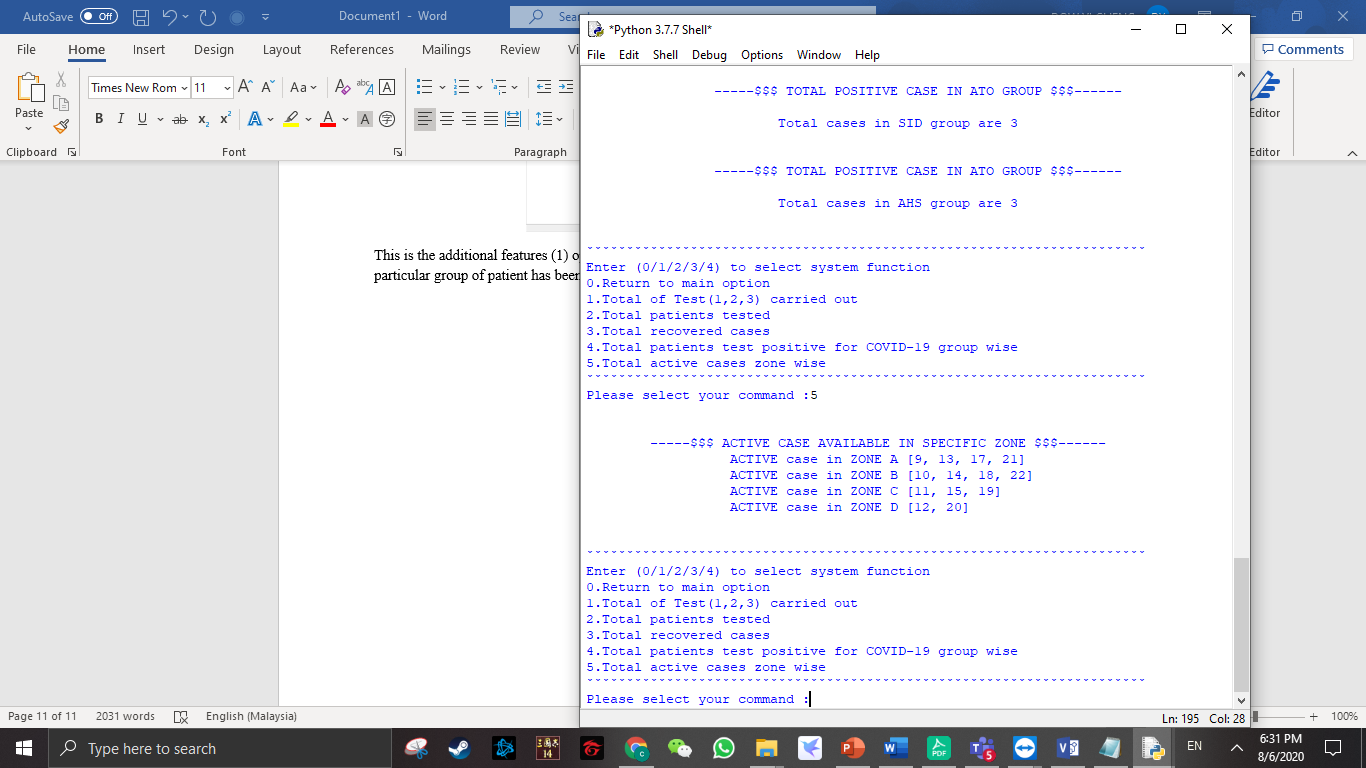
This is the additional features (1) option (2) output , It will display all total of patient tested has been recorded until now . After that return back into option list.



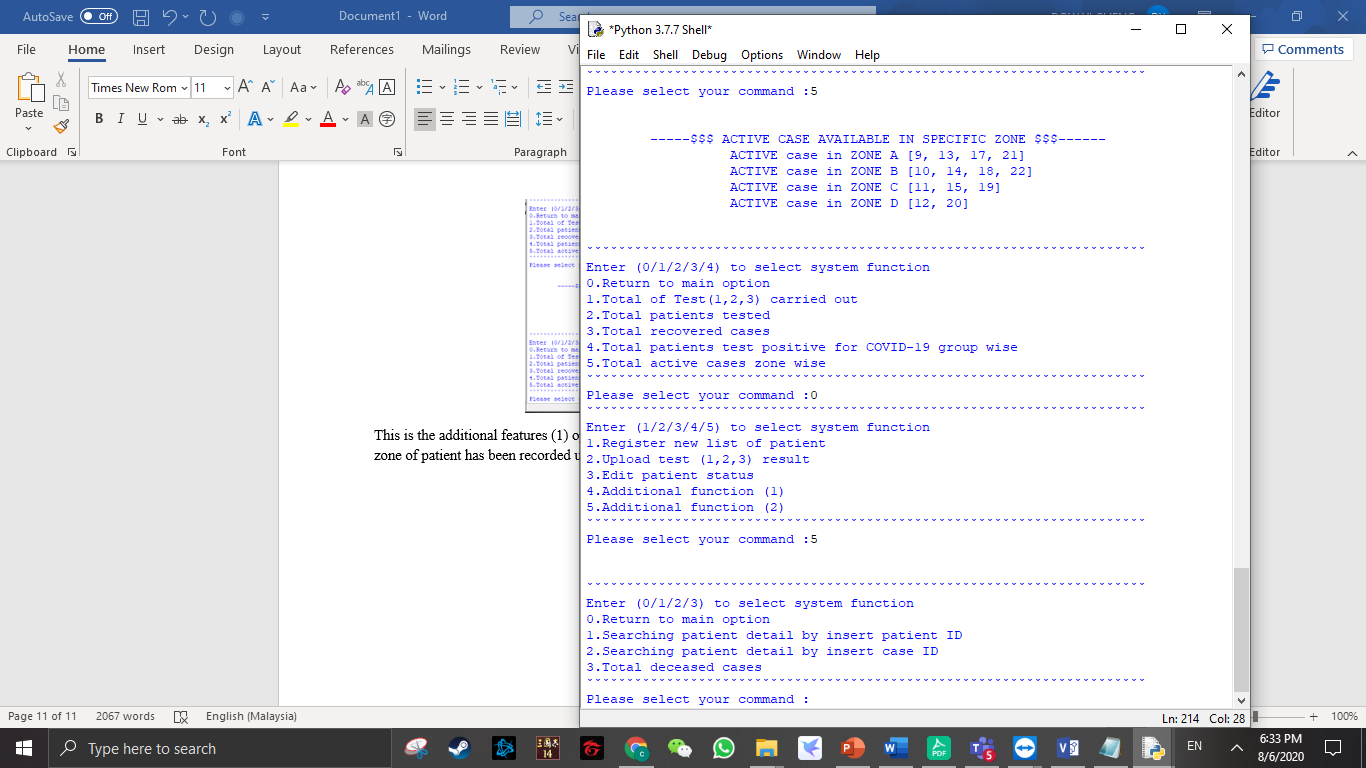
This is the additional features (1) option (3) output , It will display all total of patient tested has been recorded until now .After that return back into option list.



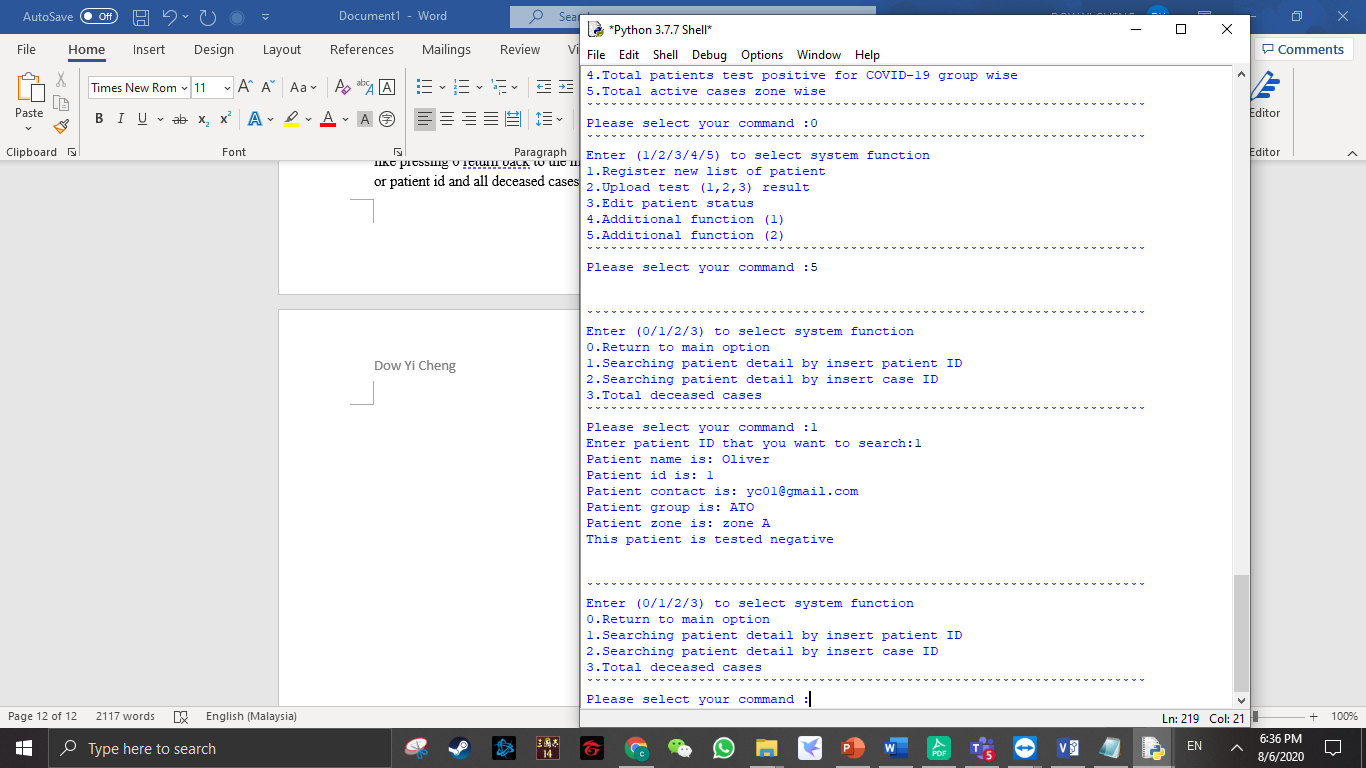
This is the additional features (1) option (4) output , It will display all total of positive case of particular group of patient has been recorded until now . After that return back into option list.



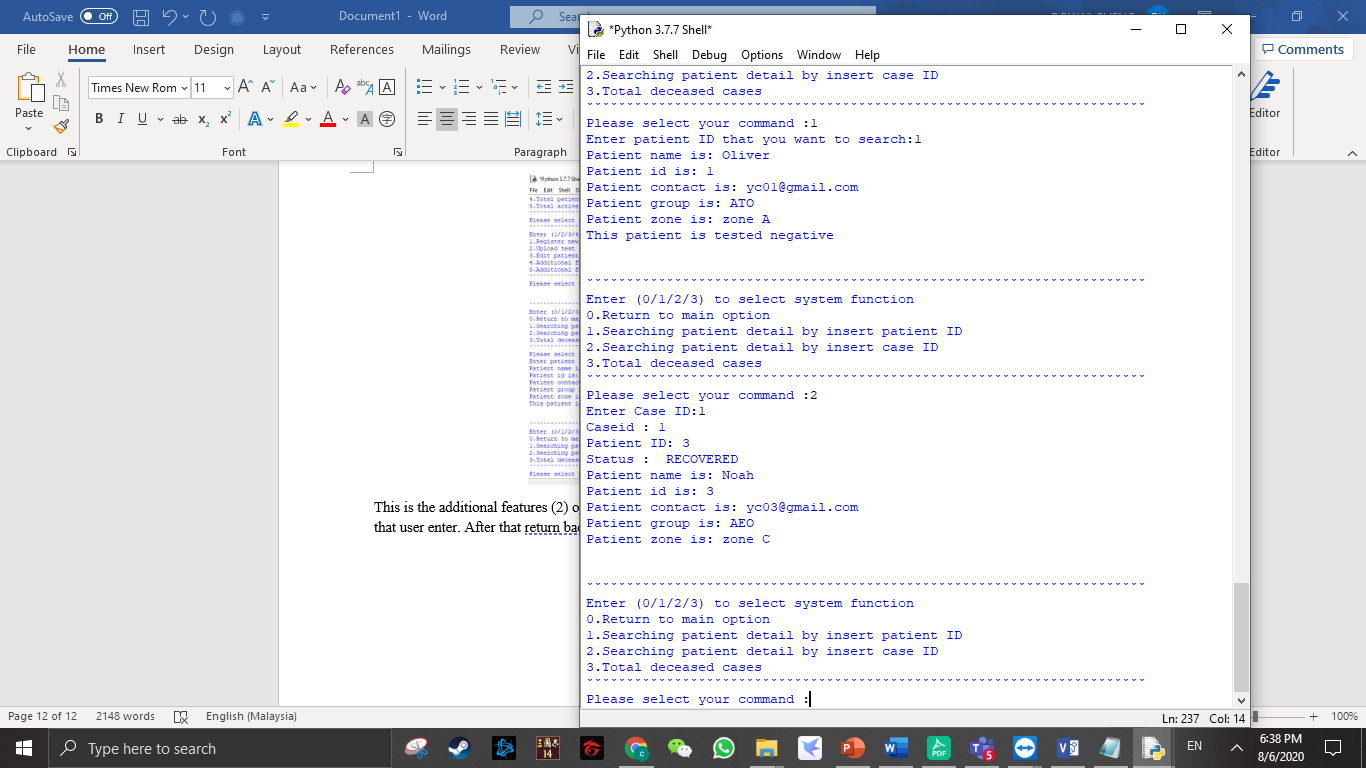
This is the additional features (1) option (5) output , It will display all total of active case of particular zone of patient has been recorded until now . After that return back into option list.



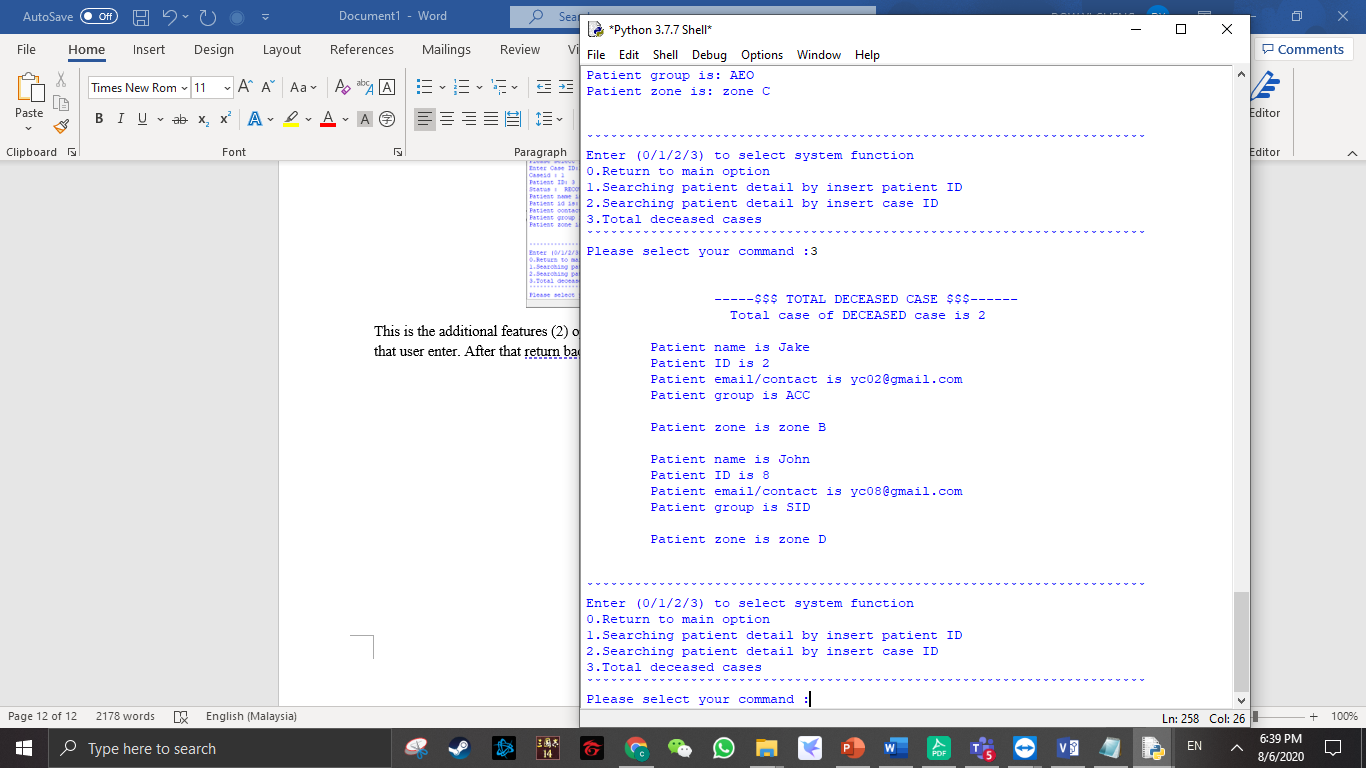
This is the output of additional (2) features (option 5), there got 4 option to let user to select. Example like pressing 0 return back to the main menu. 1-3 option is searching patient data by entering case id or patient id and all deceased cases has been recorded until now.



This is the additional features (2) option (1) output , It will display all detail of the particular patient id that user enter. After that return back into option list.



This is the additional features (2) option (2) output , It will display all detail of the particular case id that user enter. After that return back into option list.



This is the additional features (2) option (3) output , It will display all detail of the deceased case .After that return back into option list.

# Conclusion

As a conclusion, this program is develop for helping fighting against in frontline hospital staff for recording those patient information easier to save the time and helping those patient who infected COVID-19.

# References

1. W3 Schools.(Not Stated) *Python Tutorials* [Online]. Available at: <https://www.w3schools.com/python/ref_string_split.asp#:~:text=Definition%20and%20Usage,number%20of%20elements%20plus%20one.> [Accessed June 5th 2020]
2. Programiz. (Not Stated) *Python String Strip Method* [Online]. Available at: <https://www.programiz.com/python-programming/methods/string/strip#:~:text=The%20strip()%20method%20returns,of%20characters%20to%20be%20removed).>