Company name

Description automatically generated

**INDIVIDUAL ASSIGNMENT**

# **032022-VAR**

# **Programming with Python**

**HAND OUT DATE: 8th April 2022 [Wk3]**

**HAND IN DATE: 27th June 2022 [Wk12]**

**Weightage: 50%**

**Online Submission Time before = 23: 59 PM Malaysia Time**

**Student Name:** WONG KANG SHIN

**Student ID:** TP068522

**INSTRUCTIONS TO CANDIDATES:**

1. Assignment is to be submitted through online submission (Moodle).
2. Students are advised to underpin their answers with the use of references ((cited using the APA name system of Referencing).
3. Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.
4. Cases of plagiarism will be penalized.
5. You must obtain 50% overall to pass this module.

Table of contents :

1.0 Introduction and assumptions

2.0 Design of the program

3.0 Pseudocode

4.0 Flowcharts

5.0 Program source code and explanation

6.0 Screenshots of sample input/output and explanation

7.0 Conclusion

**1.0 Introduction and assumptions**

OCEAN Sdn Bhd is a pharmacy where daily activities has been affected by Covid-19. Therefore, a program is needed to provide an efficient and smooth process of purchasing to the customers. The employees in the pharmacy should be able to refer to all information about customers and medicines using the program as well.

This report addresses the techniques and methods used to plan and construct the program

**2.0 Design of the program**

The entire program is split into functions. It starts with the main menu, branching into 3 types of users. Admin, new customers and registered customers. Then there are many sub functions under these 3 users. All navigation between branches are done using the “if and else method”. While loops are used when information given is incorrect.

The program uses files to store both temporary and permanent data. There are 2 files storing temporary information. The first one is the account file, the account file is only used in the Registered Customer mode. The file stores the name of the customer so that it can be used in all the subfunctions. The second temporary file is “Temporary Order file(torder.txt)” . This file is only used when a registered customer purchases an item. All items chosen by the customer will be listed in “torder.txt”. The file is then displayed to show all the items purchased. “torder.txt” is then copied into the long-term file for record.

The program has 3 files storing permanent information. The medicine, the information of registered customers and the history of all purchases. These files can be viewed and altered by the users according to the function they have chosen.

**3.0 Pseudocode**

**3.1 Menus and Logins :**

Start

menu():

Display Menu

Get mode

If mode == 1

admin\_menu()

Elif mode == 2

N.cust\_menu()

Elif mode == 3

R.cust\_menu()

Def admin\_login():

Display “Enter Username and password”

If user and password is correct :

admin\_menu()

Def admin\_menu():

Display admin Menu

Get option

If option == 1:

admin\_upload()

Elif option == 2:

admin\_view\_med()

Elif option == 3:

admin\_modify()

Elif option == 4:

admin\_delete()

Elif option == 5:

admin\_search()

Elif option == 6:

admin\_view\_order()

Elif option ==7:

admin\_view\_s\_order()

Else :

Display “Invalid input”

admin\_menu()

Else:

Display “Invalid Username or Password”

menu()

Def N.cust\_menu() :  
 Display menu

Get n.option

elIf option == 1:

N\_cust\_view()

elIf option == 2:

N\_cust\_register()

elIf option == 3:

menu()

Else :

Display “Invalid code”

menu()

Def R\_cust\_login():

Display login

Get r.username and password

If username & password matches :

R\_cust\_menu()

Else :

Display “Invalid User or Password”

Exit

Def R\_cust\_menu()

Get r.option

If option == 1:

r\_cust\_view()

elIf option == 2:

r\_cust\_order()

elIf option == 3:

r\_cust\_record()

elIf option == 4:

r\_cust\_info()

elIf option == 5:

menu()

Else :

Display “Invalid input”

menu()

**3.2 Admin functions:**

Def admin\_upload():

Get medicine name

Get medicine description

Get medicine price

write to ‘medicine.txt’

Print 'information uploaded’

admin\_menu()

Def admin\_view():

Open ‘medicine.txt’

Display ‘medicine.txt’

admin\_menu()

Def admin\_modify():

Get medicine name to configure

Open’medicine.txt’

Display ‘medicine.txt’

Get what category to configure

Identify which line to configure

Write that line with new information

Display configuration made

admin\_menu()

Def admin\_delete():

Open medicine file

Display medicine file

Get what medicine to delete

Identify line to remove

Add loop to rewrite file without next 4 lines

Display deleted

admin\_menu()

Def admin\_search():

Get what medicine to search

Define availability as no

Loop to search medicine:

If available :

Change availability to yes

Display lines

If availability remains no:

Display ‘no search found’

Else continue

admin\_menu()

Def admin\_view\_order():

Open order file

Display order file

admin\_menu()

Def admin\_s\_order():

Get which customer to view

Search for customer name’s line

For line in order list

Define done = 0

If done = 0:

If lines = ‘ customer ‘:

print(line)

Done = 1

If done = 1:

If line doesn’t start with ‘--’:

Print line

If line starts with ‘--’:

end

Else :

end

admin\_menu()

**3.3 New Customers Functions :**

Def N\_cust\_view():

Open medical file

Display medical file

N\_cust\_menu()

Def N\_cust\_register():

Get customer Name

Gender

Phone number

IC number

While true :

Get Password

Get confirm password

If password == confirm password :

Break

Else :

Display not same password

Continue

Display information recorded

Open member file

Write customer details

N\_cust\_menu()

**Registered Customer Functions:**

Def r\_cust\_view():

Open medicine file

Display medicine file

R\_cust\_menu()

Def r\_cust\_order():

Open medicine file

Open Temp Order file

Define price

Define available as 0

Open account file

Write name in account file into temp order file

Loop while true :

Display what medicine wanted

Get name of medicine wanted(<enter to break)

Get quantity

If name isn’t ‘enter’:

(get price)

Open medicine file

Search for medicine name

Available = 1

Get the price

Write user name, medicine name, quantity and price into temp order

Price = price + Calculated total price

Elif name is ‘enter’:

break

Display temp order

Display Price

Get payment

If price - payment >= 0:

Display ‘remaining needed

Elif price - payment <=0:

Write payment into order file

Display ‘ here’s ur change, payment - price’

R\_cust\_menu()

Def r\_cust\_record():

Open order file

Open Account file

Define customer name as name from account file

For line in file:

If line starts with customer name:

Print the entire section until ‘----’

R\_cust\_menu()

Def r.cust\_info():  
 Open member file

For line in file:

If line starts with member’s name:

Display member’s information

menu()

**4.0 Flowchart**

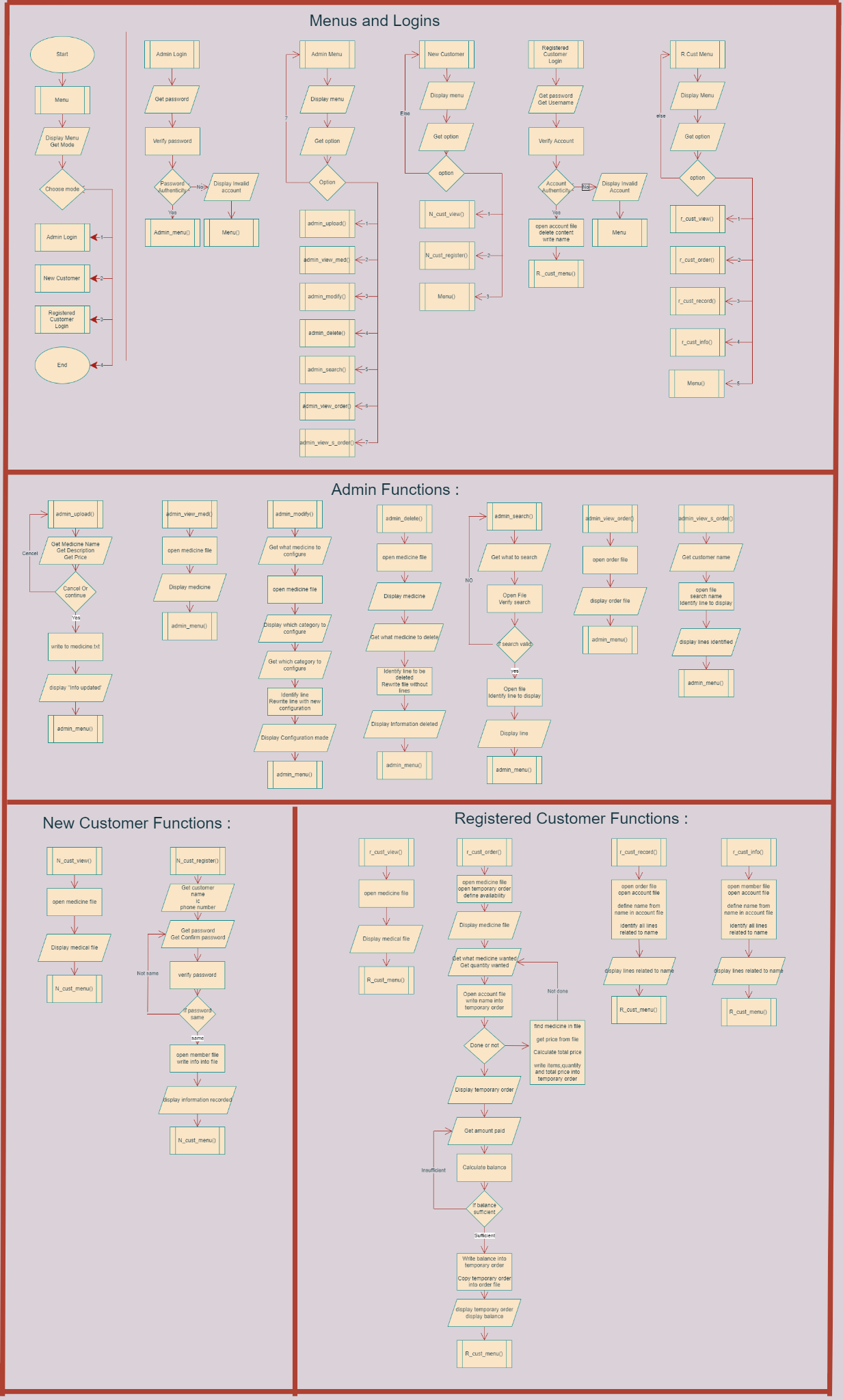
****

Figure 1.0 : Flowchart

**5.0 Program source code and explanation**

There are 3 main methods used to program the entire program. The basic if and else, and looping. A method I call “Tick-counter” is used to identify more than one line together in the file. The tick-counter is used to verify customer’s username and password, printing specific medicine information wanted, printing summary of specific customers and much more. The final method used is the method I call “RAM”. RAM is used to temporarily store and list information to be printed as a whole. Or to use the same temporary information (such as User’s name) in all multiple functions.

5.1 If and else + Loops (Menu)

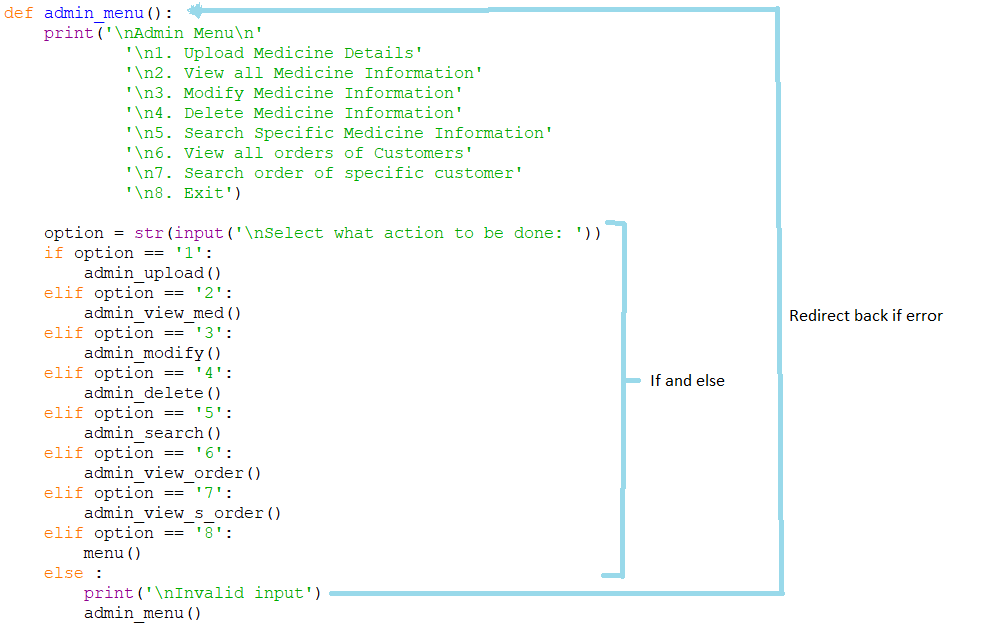
****

Figure 2.0 : Menu’s program

If and else and loops are used in all the menu functions. User is asked to give input of choice. If and else will redirect users to sub functions. While loop will be triggered if User chooses “exit” or types and invalid input

5.2 Tick Counter Method (identifying lines)

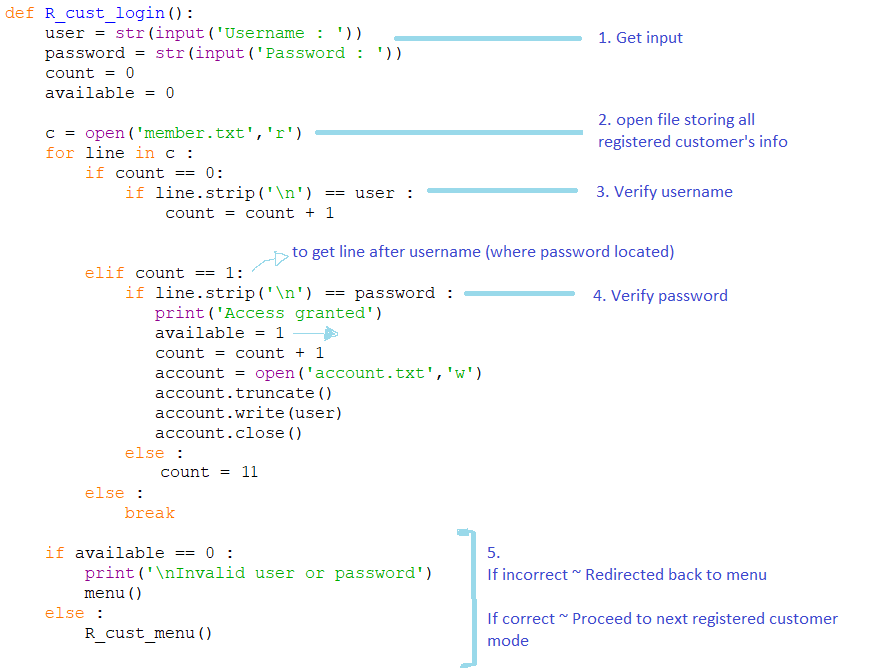


Figure 3.0 : Program used in login process

This diagram shows the process of verifying the customer’s username and password. The count is defined as an indicator of what information should be verified. Count = 0 is for searching Username. When the line in file matches the input username. It changes Count to 1. Which verifies the password.

Availability shows if the password matches. 1 = verified. 0 = password or username doesn’t match.

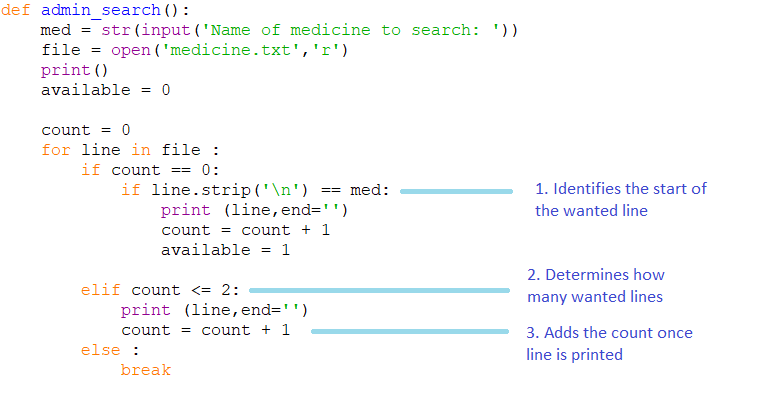


Figure 5.0 : Programming of Tick-Counter

This is another example of Tick-Counter. This is used as each medicine has 3 lines in a file. Therefore, when a user wants to display a specific medicine’s information. 3 lines needs to be printed. Tick-Counter is used to identify which line is wanted. Then print the next 2 lines. In this case 3 lines needs to be printed therefore count is set to <=2 which gives 3 ticks. 0,1,2. The counter is changed each time the wanted line is printed. Once all wanted lines are printed. The loop breaks

5.3 RAM (Using file to temporary store data)

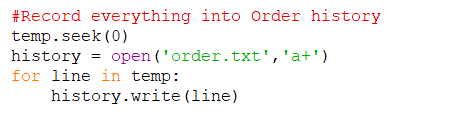


Figure 6.0 : Programming of copying data

This picture is extracted from the purchasing function. All purchasing items, quantity, paid amount and balance are listed in the *temp* file. At the end of the purchase. The *temp* file is printed to show the customer purchasing summary. The entire summary is then copied into the “order.txt” to be stored permanently

**6.0 Screenshots of sample input/output and explanation**

The following pictures below will show the output result from the coding above is run.

6.1 Menu

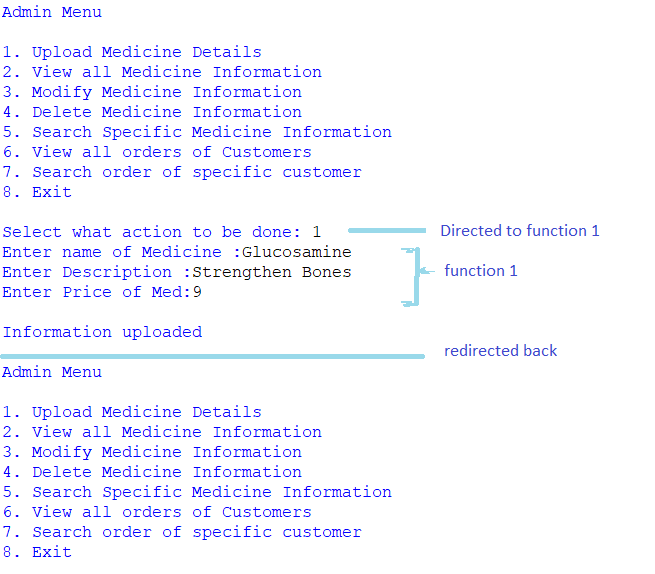
****

Figure 7.0 : Run perspective of Menu

6.2 Login

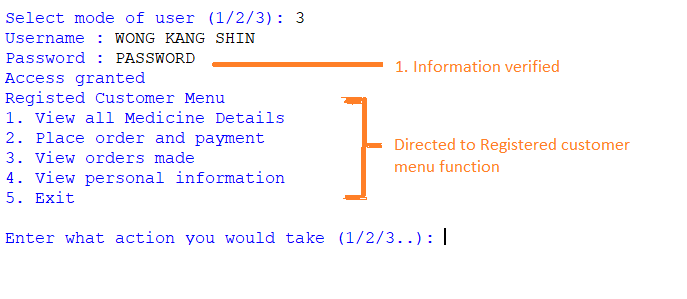
****

Figure 8.0 : Run perspective of Login

In step “1. Information verified”. The member’s file is opened. The username and password will be searched throughout the file and match with the input given. Access will be granted if it matches

6.3 Identifying lines (Medicine)

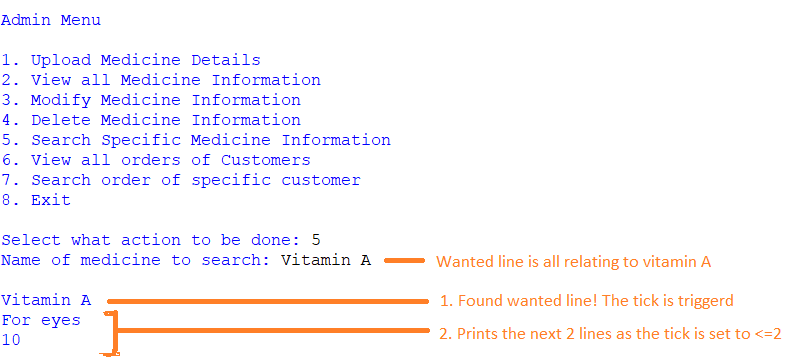
****

Figure 9.0 : Run perspective of Tick-Counter method

This is a scenario when an Admin User choose to find specific information in a file. Therefore, the tick-counter is used.

6.4 Transferring Data

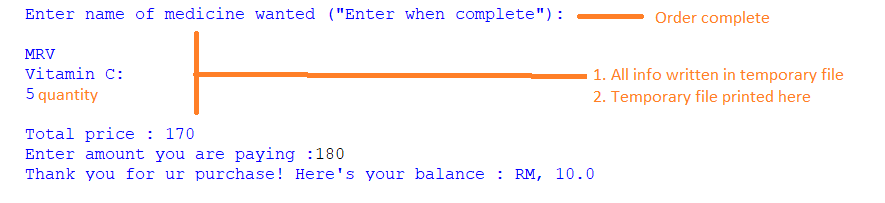
****

Figure 10.0 Run perspective of printed files

Once a payment is done. All information from temporary file is written in purchase history file. The figure below shows the purchase history when an admin requests it to be displayed

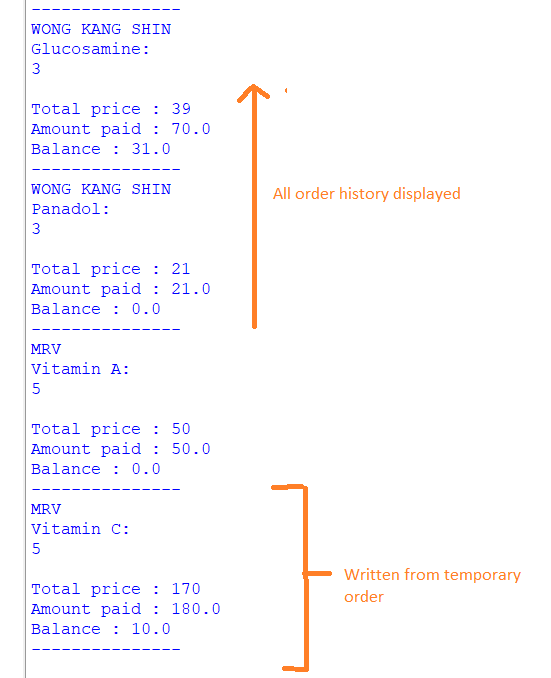
****

Figure 11.0 : Displaying rewritten data from Temporary file

**7.0 Conclusion**

In time of pandemic, many businesses are minimising human interaction with machines and software. OCEAN Sdn Bhd has decided to develop a software to assist their businesses as well. With a few techniques in Python programming such as if and else, while loops, tick counter and using files to store permanent and temporary information. A program can be made just for OCEAN Sdn Bhd for their daily activities.