



الجامعة الافتراضية السورية
SYRIAN VIRTUAL UNIVERSITY

PBR

S24

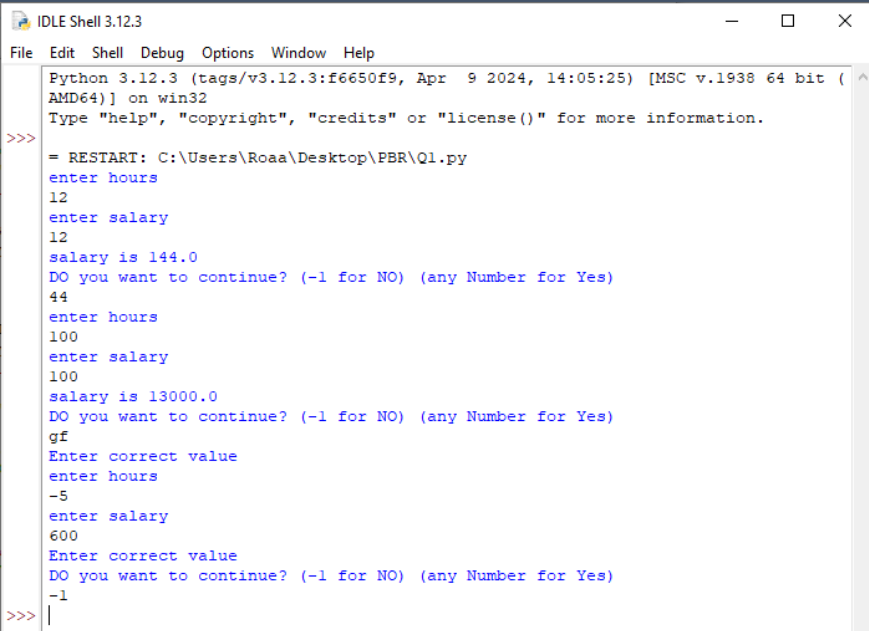
د. وسيم يوسف

إعداد الطلاب:

الاسم	الصف
Rouaa_298365	C1
Ranim_326483	C1

المسألة الأولى:

(1) تحديد الدخل والخرج والمعالجة:

الدخل	يقوم المستخدم بإدخال hours_salary - hours - continue_input
المعالجة	<p>ما دام (الشرط دائماً محققاً) كرر: بداية</p> <p>اقرأ hours , hours_salary إذا كان (hours > 40 and hours > 0) نفذ بداية</p> <p>Salary ← hours * hours_salary اكتب "salary is:", salary نهاية وإلا</p> <p>إذا كان (hours > 40) نفذ بداية</p> <p>extra_hours ← hours - 40 salary ← ((extra_hours * (hours_salary * 1.5)) + ((hours - extra_hours) * hours_salary)) اكتب "salary is:", salary نهاية وإلا</p> <p>إذا كان (hours < 0) نفذ بداية</p> <p>اكتب "enter correct value" نهاية</p> <p>اقرأ continue_input إذا كان (continue_input == -1) نفذ بداية</p> <p>إنهاء الحلقة نهاية نهاية</p>
الخرج	 <pre> IDLE Shell 3.12.3 File Edit Shell Debug Options Window Help Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information. >>> = RESTART: C:\Users\Roaa\Desktop\FBR\Q1.py enter hours 12 enter salary 12 salary is 144.0 DO you want to continue? (-1 for NO) (any Number for Yes) 44 enter hours 100 enter salary 100 salary is 13000.0 DO you want to continue? (-1 for NO) (any Number for Yes) gf Enter correct value enter hours -5 enter salary 600 Enter correct value DO you want to continue? (-1 for NO) (any Number for Yes) -1 >>> </pre>

(2) الكود البرمجي:

```
while True:
    ##### يقوم المستخدم بإدخال ساعات العمل وقيمة ساعات العمل #####
    try:
        hours = float (input('enter hours\n'))
        hours_salary = float (input ('enter salary\n'))
        ##### حساب الراتب إذا كانت ساعات العمل اقل من 40 ولكن أكبر من 0 #####
        if hours <=40 and hours>0:
            salary = hours *hours_salary
            print ('salary is', salary)
            ### حساب الراتب إذا كانت ساعات العمل أكبر من 40 ###
        elif hours >40 :
            extra_hours =hours -40
            salary =((extra_hours * (hours_salary* 1.5))+((hours -
            extra_hours)*hours_salary))
            print ('salary is', salary)
            ### إذا ادخل المستخدم قيمة أقل من 0 في ساعات العمل ###
        elif hours <0 :
            print('Enter correct value')

    ### إذا أراد المستخدم انهاء البرنامج يدخل -1 أو يدخل إيا رقم للإدخال مرة أخرى ###
    continue_input= int (input('DO you want to continue? (-1 for NO) (any
    Number for Yes)\n'))
    if(continue_input== -1):
        break
    except:
        ## إذا ادخل المستخدم قيمة خاطئة في ساعات العمل أو قيمة ساعات العمل او عند ##
        ##### سؤال إنهاء البرنامج #####
        print('Enter correct value')
```

```
while True:
    ##### يقوم المستخدم بإدخال ساعات العمل وقيمة ساعات العمل #####
    try:
        hours = float (input('enter hours\n'))
        hours_salary = float (input ('enter salary\n'))
        ##### حساب الراتب إذا كانت ساعات العمل اقل من 40 ولكن أكبر من 0 #####
        if hours <=40 and hours>0:
            salary = hours *hours_salary
            print ('salary is', salary)
            ### حساب الراتب إذا كانت ساعات العمل أكبر من 40 ###
        elif hours >40 :
            extra_hours =hours -40
            salary =((extra_hours * (hours_salary* 1.5))+((hours -extra_hours)*hours_salary))
            print ('salary is', salary)
            ### إذا ادخل المستخدم قيمة أقل من 0 في ساعات العمل ###
        elif hours <0 :
            print('Enter correct value')

    ### إذا أراد المستخدم انهاء البرنامج يدخل -1 أو يدخل إيا رقم للإدخال مرة أخرى ###
    continue_input= int (input('DO you want to continue? (-1 for NO) (any Number for Yes)\n'))
    if(continue_input== -1):
        break
    except:
        ##### إذا ادخل المستخدم قيمة خاطئة في ساعات العمل أو قيمة ساعات العمل او عند سؤال إنهاء البرنامج #####
        print('Enter correct value')
```

المسألة الثانية:

(1) تحديد الدخل والخرج والمعالجة:

الدخل	يقوم المستخدم بإدخال name و number
المعالجة	<p>Max1 ← None , Max2 ← None , Max3 ← None</p> <p>max_name1← None , max_name2← None , max_name3← None</p> <p>i ← 1</p> <p>ما دام (i ≤ 10) كرر:</p> <p>بداية</p> <p>اقرأ name , number</p> <p>إذا كان (max1 is None OR number > max1) نفذ</p> <p>بداية</p> <p>max3 ← max2</p> <p>max2 ← max1</p> <p>max1 ← number</p> <p>max_name3 ← max_name2</p> <p>max_name2 ← max_name1</p> <p>max_name1 ← name</p> <p>نهاية</p> <p>وإلا</p> <p>إذا كان (max2 is None OR number > max2) نفذ</p> <p>بداية</p> <p>max3 ← max2</p> <p>max2 ← number</p> <p>max_name3 ← max_name2</p> <p>max_name2 ← name</p> <p>نهاية</p> <p>وإلا</p> <p>إذا كان (max3 is None OR number > max3) نفذ</p> <p>بداية</p> <p>max3 ← number</p> <p>max_name3 ← name</p> <p>نهاية</p> <p>i ← i + 1</p> <p>نهاية</p> <p>اكتب</p>
	<p>"first 3 student :", "name :"+max_name1, ", max1,"name</p> <p>:"+max_name2, ", max2,"name :"+max_name3, ",max3</p>

```
IDLE Shell 3.14.0a1
File Edit Shell Debug Options Window Help
Python 3.14.0a1 (tags/v3.14.0a1:8cdaca8, Oct 15 2024, 20:08:21) [MSC v.1941 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ASUS\Desktop\FBR\Q2.py =====
enter 10 student to give you the first 3
enter number
70.5
enter name
yahia
enter number
87.8
enter name
roaa
enter number
75.3
enter name
osama
enter number
77.2
enter name
doaa
enter number
70.3
enter name
nour
enter number
85.4
enter name
ranim
enter number
69.7
enter name
lama
enter number
60.2
enter name
mazen
enter number
80.3
enter name
enas
enter number
74
enter name
ahmad
first 3 student : name :roaa 87.8 name :ranim 85.4 name :enas 80.3
>>>
```

الخرج

(2) الكود البرمجي:

المتغيرات

```
max1 =None
max2 = None
max3 = None
max_name1 =None
max_name2 = None
max_name3 = None
i=1
print('enter 10 student to give you the first 3 ')
while (i<=10):
    try:
        ##### إدخال معدل الطالب واسمه #####
        number = float(input('enter number\n'))
        name = input('enter name\n')
        ##### إيجاد الطالب الأول #####
        if max1 is None or number > max1:
            max3 = max2
            max2 =max1
            max1 = number
            max_name3 = max_name2
            max_name2 =max_name1
            max_name1 = name
            ##### إيجاد الطالب الثاني #####
        elif max2 is None or number > max2:
            max3 = max2
            max2 = number
            max_name3 =max_name2
            max_name2 = name
            ##### إيجاد الطالب الثالث #####
        elif max3 is None or number > max3:
            max3 = number
            max_name3 = name
        i=i+1
```

```

except :
    # إذا ادخل المستخدم قيمة خاطئة ##
    print ('enter correct value')
    ##### طباعة الطلاب الأوائل #####
print('first 3 student :','name :'+max_name1,',', max1,'name :'+max_name2,',',
max2,'name :'+max_name3,',',max3)

```

Q2.py - C:\Users\Roaa\Desktop\PBR2\Q2.py (3.12.3)

File Edit Format Run Options Window Help

```

##### المتغيرات #####
max1 = None
max2 = None
max3 = None
max_name1 = None
max_name2 = None
max_name3 = None
i=1
print('enter 10 student to give you the first 3 ')
while (i<=10):
    try:
        ##### إدخال معدل الطالب واسمه #####
        number = float(input('enter number\n'))
        name = input('enter name\n')
        ##### إيجاد الطالب الأول #####
        if max1 is None or number > max1:
            max3 = max2
            max2 = max1
            max1 = number
            max_name3 = max_name2
            max_name2 = max_name1
            max_name1 = name
            ##### إيجاد الطالب الثاني #####
        elif max2 is None or number > max2:
            max3 = max2
            max2 = number
            max_name3 = max_name2
            max_name2 = name
            ##### إيجاد الطالب الثالث #####
        elif max3 is None or number > max3:
            max3 = number
            max_name3 = name
        i=i+1
    except :
        # إذا ادخل المستخدم قيمة خاطئة ##
        print ('enter correct value')
##### طباعة الطلاب الأوائل #####
print('first 3 student :','name :'+max_name1,',', max1,'name :'+max_name2,',', max2,'name :'+max_name3,',',max3)

```

المسألة الثالثة:

(1) تحديد الدخل والخرج والمعالجة:

الدخل	يقوم المستخدم بإدخال continue_input - result_number
المعالجة	i←0
	L←0
	continue_input← "y"
	ما دام (continue_input=="y") كرر:
	بداية
	result=number1*number2
	اكتب "result =",number1,"*", number2
	اقرأ result_number
	إذا كان (result_number==result) نفذ
	بداية
	اكتب "true"
	i←i+1
	نهاية
	وإلا نفذ
	بداية
	اكتب "false \nthe answer is:",result
	L←L+1
	نهاية
	اقرأ continue_input
	نهاية
	إذا كان (continue_input=="n") نفذ
	بداية
	اكتب i, "the true answer is:"
	اكتب L, "the false answer is:"
	نهاية
	وإلا نفذ
	بداية
	اكتب "Did you mean 'n' "
	اكتب i, "the true answer is:"
	اكتب L, "the false answer is:"
	نهاية

```

IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Roaa\Desktop\FBR\Q3.py
result= 3 * 5
enter the result
15
true
Do you want to continue? y/n
y
result= 10 * 2
enter the result
20
true
Do you want to continue? y/n
y
result= 1 * 2
enter the result
23
false
the answer is: 2
Do you want to continue? y/n
n
the true answer is: 2
the false answer is: 1
>>>

```

الخروج

(2) الكود البرمجي:

```

import random
i=0
L=0
continue_input= 'y'
while (continue_input.lower()=='y'):
    ## الحصول على ارقام عشوائية ##
    number1=random.randint(1,10)
    number2=random.randint(1,10)
    result=number1*number2
    print ('result=',number1,'*', number2)
    try:
        result_number = float(input('enter the result\n'))
        ## إذا ادخل المستخدم قيمة الناتج صحيحة ##
        if (result_number ==result ):
            print ('true')
            i=i+1
        else:
            ## إذا ادخل المستخدم الناتج بقيمة خاطئة ##
            print ('false \nthe answer is:',result)
            L=L+1
    except :
        print('Enter correct value')
        ## لإكمال البرنامج أو انهاءه ##
        continue_input=input('Do you want to continue? y/n\n')
        ## لإنهاء البرنامج إذا ادخل المستخدم المحرف N ##
if (continue_input.lower()=='n'):
    print('the true answer is:',i)
    print('the false answer is:',L)
else:
    ## Y/N إذا لم يدخل المستخدم احد المحرفين ##
    print("Did you mean 'n'")
    print('the true answer is:',i)
    print('the false answer is:',L)

```



```

import random
i=0
L=0
continue_input= 'y'
while (continue_input.lower()=='y'):
    ## الحصول على ارقام عشوائية ##
    number1=random.randint(1,10)
    number2=random.randint(1,10)
    result=number1*number2
    print ('result=',number1,'*', number2)
    try:
        result_number = float(input('enter the result\n'))
        ## إذا ادخل المستخدم قيمة الناتج صحيحة ##
        if (result_number ==result ):
            print ('true')
            i=i+1
        else:
            ## إذا ادخل المستخدم الناتج بقيمة خاطئة ##
            print ('false \nthe answer is:',result)
            L=L+1
    except :
        print('Enter correct value')
        ## أو انتهائه لإكمال البرنامج ##
        continue_input=input('Do you want to continue? y/n\n')
        ## لإنهاء البرنامج N إذا ادخل المستخدم المحرف ##
if (continue_input.lower()=='n'):
    print('the true answer is:',i)
    print('the false answer is:',L)
else:
    ## Y/N إذا لم يدخل المستخدم احد المحرفين ##
    print("Did you mean 'n'")
    print('the true answer is:',i)
    print('the false answer is:',L)

```

المسألة الرابعة:

(1) تحديد الدخل والخرج والمعالجة:

الدخل	يقوم المستخدم بإدخال name - phone_number - local_sms - continue_input - international_call - local_call - international_sms
المعالجة	<p>❖ إدخال البيانات:</p> <p>customers ← {}</p> <p>مادام (الشرط دائماً محققاً) كرر:</p> <p>بداية</p> <p>أقرأ</p> <p>name,Phone_number,local_sms_count,international_sms_count,local_call_minuts ,international_call_minuts</p> <p>Customers[Phone_number]← { 'name': name, 'local_sms_count': local_sms_count, 'international_sms_count': international_sms_count, 'local_call_minuts': local_call_minuts, 'international_call_minuts': international_call_minuts, }</p> <p>إقرأ continue_input</p> <p>إذا كان (continue_input!='y') نفذ</p> <p>بداية</p> <p>إنهاء الحلقة</p> <p>نهاية</p>

```

أكتب customers
نهاية
❖ حساب الفاتورة:
local_sms ← customer['local_sms_count']
local_sms_cost ← 0
إذا كان (local_sms ≤ 5) نفذ
    بداية
    local_sms_cost ← local_sms * 50
    نهاية
وإلا نفذ
    بداية
    local_sms_cost ← (5 * 50) + ((local_sms - 5) * 35)
    نهاية
international_sms ← customer['international_sms_count']
international_sms_cost ← international_sms * 100
local_minutes ← customer['local_call_minutes']
local_call_cost ← 0
إذا كان (local_minutes ≤ 5) نفذ
    بداية
    local_call_cost ← local_minutes * 30
    نهاية
وإلا نفذ
    إذا كان (local_minutes ≤ 10) نفذ
        بداية
        local_call_cost ← 5 * 30
        نهاية
    وإلا نفذ
        بداية
        local_call_cost ← (5 * 30) + ((local_minutes - 10) * 25)
        نهاية
international_minutes ← customer['international_call_minutes']
international_call_cost ← international_minutes * 200
total_bill ← local_sms_cost + international_sms_cost + local_call_cost
+ international_call_cost
total_bill اكتب
❖ حفظ الفاتورة:
مادام phone_number, customer في customers كرر
    بداية
    bill ← total_bill
    customer['bill'] ← bill
    نهاية
    اكتب customers

```

❖ الفاتورة لرقم هاتف يدخله المستخدم:
إذا كان (customer != None) نفذ
بداية

اكتب phone_number , bill
نهاية

❖ رقم هاتف لأكبر قيمة فاتورة:
max_customer ← ''
max_bill_value ← 0
مادام phone_number, customer في customers كرر
بداية

إذا كان (bill > max_bill_value) نفذ
بداية

max_bill_value ← bill
max_customer ← phone_number
نهاية

❖ اسم العميل لأكبر قيمة فاتورة:
max_bill_value ← 0
مادام phone_number, customer في customers كرر
بداية

إذا كان (bill > max_bill_value) نفذ
بداية

max_bill_value ← bill
max_customer ← customer['name']
❖ ترتيب العملاء تنازلياً حسب قيمة فاتورة:
sort_bills ← {}
lst ← list()
name_lst ← list()
مادام phone_number, customer في customers كرر
بداية

bills ← customer('bill')
name_customer ← customer('name')
sort_bills[phone_number] ← (bills, name_customer)
نهاية

مادام key, val في sort_bills كرر
بداية

lst ← val
نهاية

رتب lst
اكتب name

❖ أسماء أول 3 عملاء لهم الفاتورة الأعلى مرتبين تصاعدياً حسب الفاتورة

```
sort_bills←{}
```

```
lst←list()
```

```
name_lst←list()
```

مادام phone_number, customer في customers كرر
بداية

```
bills ← customer('bill')
```

```
name_customer←customer('name')
```

```
sort_bills[phone_number] ←( bills ,name_customer)
```

نهاية

مادام val, key في sort_bills كرر
بداية

```
lst← val
```

نهاية

رتب lst

اكتب name[0],name [1],name[2]

❖ حفظ معلومات العميل في ملف:

افتح ملف File=

مادام phone_number, customer في customers كرر
بداية

```
line← phone_number + customer
```

نهاية

```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Enter customer name: enas
Enter phone number: 096355
Enter number of local SMS: 5
Enter number of international SMS: 3
Enter number of local call minutes: 6
Enter number of international call minutes: 9
Do you want to add another customer? (y/n): y
Enter customer name: lama
Enter phone number: 097365
Enter number of local SMS: 5
Enter number of international SMS: 6
Enter number of local call minutes: 8
Enter number of international call minutes: 9
Do you want to add another customer? (y/n): y
Enter customer name: doaa
Enter phone number: 034685
Enter number of local SMS: 6
Enter number of international SMS: 22
Enter number of local call minutes:
Enter correct value
Enter customer name: 8
Enter phone number: 5
Enter number of local SMS: 6
Enter number of international SMS: 6
Enter number of local call minutes: 6
Enter number of international call minutes: 6
Do you want to add another customer? (y/n): n
enter phone number to give you the bill
093554666
('093554666', 18400)
phone number has the max bills
093554666
customer name has the max bills
roaa
list has customers name from max to min
['roaa', 'lama', 'enas', 'ranim', '8']
list has 3 customers name from min to max
['enas', 'lama', 'roaa']
save info :done
>>>
```

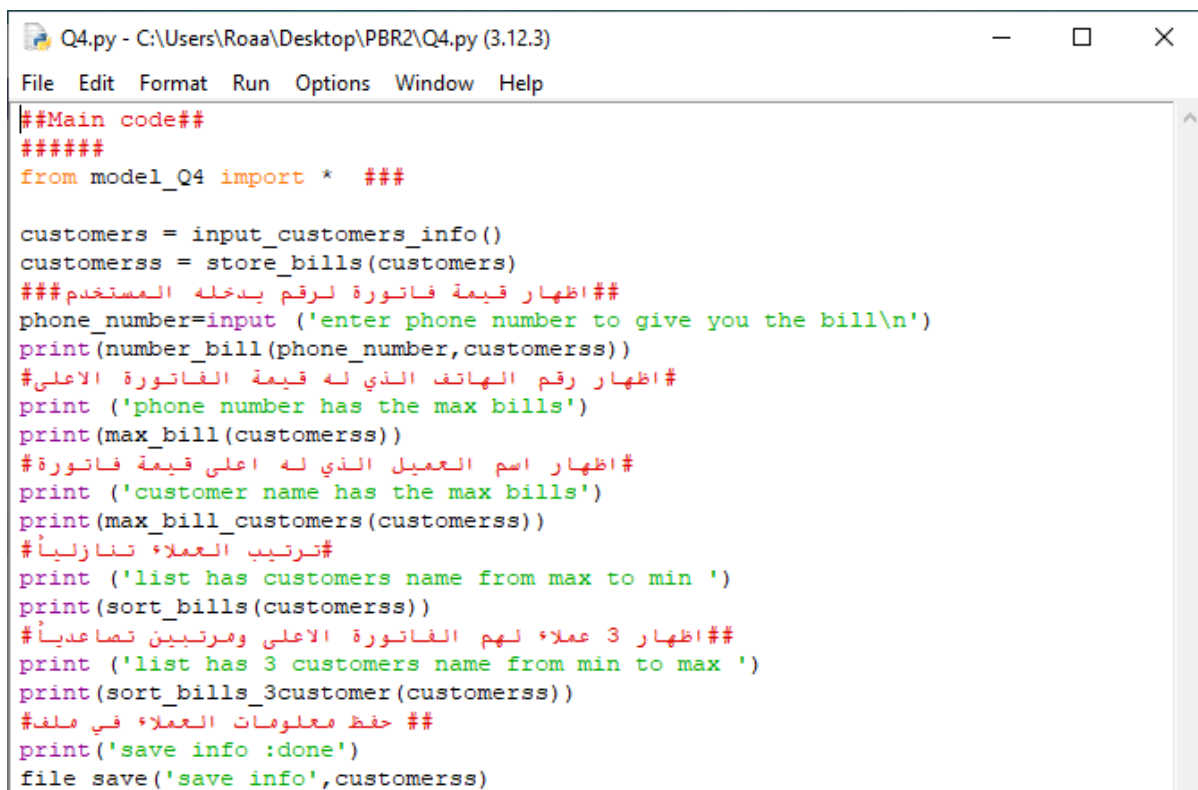
الخرج

(2) الكود البرمجي:

• Main code

```
##Main code##
#####
from model_Q4 import *

customers = input_customers_info()
customerss = store_bills(customers)
#####
# اظهر قيمة فاتورة لرقم يدخله المستخدم
phone_number=input ('enter phone number to give you the bill\n')
print(number_bill(phone_number,customerss))
# اظهر رقم الهاتف الذي له قيمة الفاتورة الاعلى
print ('phone number has the max bills')
print(max_bill(customerss))
# اظهر اسم العميل الذي له اعلى قيمة فاتورة
print ('customer name has the max bills')
print(max_bill_customers(customerss))
# ترتيب العملاء تنازلياً
print ('list has customers name from max to min ')
print(sort_bills(customerss))
# اظهر 3 عملاء لهم الفاتورة الاعلى ومرتبين تصاعدياً
print ('list has 3 customers name from min to max ')
print(sort_bills_3customer(customerss))
# حفظ معلومات العملاء في ملف
print('save info :done')
file_save('save_info',customerss)
```



The screenshot shows a window titled "Q4.py - C:\Users\Roaa\Desktop\PBR2\Q4.py (3.12.3)". The window contains the same Python code as the previous block, with syntax highlighting. The code is as follows:

```
##Main code##
#####
from model_Q4 import *

customers = input_customers_info()
customerss = store_bills(customers)
#####
# اظهر قيمة فاتورة لرقم يدخله المستخدم
phone_number=input ('enter phone number to give you the bill\n')
print(number_bill(phone_number,customerss))
# اظهر رقم الهاتف الذي له قيمة الفاتورة الاعلى
print ('phone number has the max bills')
print(max_bill(customerss))
# اظهر اسم العميل الذي له اعلى قيمة فاتورة
print ('customer name has the max bills')
print(max_bill_customers(customerss))
# ترتيب العملاء تنازلياً
print ('list has customers name from max to min ')
print(sort_bills(customerss))
# اظهر 3 عملاء لهم الفاتورة الاعلى ومرتبين تصاعدياً
print ('list has 3 customers name from min to max ')
print(sort_bills_3customer(customerss))
# حفظ معلومات العملاء في ملف
print('save info :done')
file_save('save_info',customerss)
```

```

##### العميل تابع إدخال البيانات #####
def input_customers_info():
    customers = {}
    while True:
        try:
            name = input('Enter customer name: ')
            phone_number = input('Enter phone number: ')
            local_sms_count = int(input('Enter number of local SMS: '))
            international_sms_count = int(input('Enter number of international SMS:
'))
            local_call_minutes = int(input('Enter number of local call minutes: '))
            international_call_minutes = int(input('Enter number of international
call minutes: '))

            customers[phone_number] = {
                'name': name,
                'local_sms_count': local_sms_count,
                'international_sms_count': international_sms_count,
                'local_call_minutes': local_call_minutes,
                'international_call_minutes': international_call_minutes,
            }

            continue_input = input('Do you want to add another customer? (y/n): ')
            if continue_input.lower() != 'y':
                break
        except:
            print('Enter correct value')

    return customers

#####تابع حساب الفاتورة#####
def calculate_bill(customer):

    local_sms = customer['local_sms_count']
    local_sms_cost = 0
    if local_sms <= 5:
        local_sms_cost = local_sms * 50
    else:
        local_sms_cost = (5 * 50) + ((local_sms - 5) * 35)

    international_sms = customer['international_sms_count']
    international_sms_cost = international_sms * 100

    local_minutes = customer['local_call_minutes']
    local_call_cost = 0
    if local_minutes <= 5:
        local_call_cost = local_minutes * 30
    elif local_minutes <= 10:
        local_call_cost = 5 * 30
    else:
        local_call_cost = (5 * 30) + ((local_minutes - 10) * 25)

    international_minutes = customer['international_call_minutes']
    international_call_cost = international_minutes * 200

    total_bill = local_sms_cost + international_sms_cost + local_call_cost +
international_call_cost
    return total_bill

```

```

#####تابع لحفظ الفاتورة في قاموس معلومات العميل#####
def store_bills(customers):
    for phone_number, customer in customers.items():
        bill = calculate_bill(customer)
        customer['bill'] = bill
    return customers

#####تابع لإظهار فاتورة رقم يدخله المستخدم إذا وجد#####
def number_bill(phone, customers):
    customer = customers.get(phone, None)
    if customer != None:
        bills = customer.get('bill')
        return phone, bills

#####تابع يظهر رقم العميل الذي له أكبر فاتورة#####
def max_bill(customers):
    max_customer = ''
    max_bill_value = 0
    for phone_number, customer in customers.items():
        bill = customer.get('bill', 0)
        if bill > max_bill_value:
            max_bill_value = bill
            max_customer = phone_number

    return max_customer

#####تابع يظهر اسم العميل الذي له أكبر فاتورة#####
def max_bill_customers(customers):
    # max_customer = ''
    max_bill_value = 0
    for phone_number, customer in customers.items():
        bill = customer.get('bill', 0)
        if bill > max_bill_value:
            max_bill_value = bill
            max_customer = customer['name']

    return max_customer

#####ترتيب العملاء حسب قيمة الفاتورة تنازلياً#####
def sort_bills(customers):
    sort_bills = {}
    lst=list()
    name_lst=list()
    # customer_name=''
    for phone_number, customer in customers.items():
        bills = customer.get('bill')
        name_customer=customer.get('name')
        sort_bills[phone_number] = ( bills ,name_customer)

    for key,val in sort_bills.items():
        lst.append(val)

    lst.sort(reverse=True)
    for bill,name in lst:
        name_lst.append(name)

    return name_lst

#####ترتيب العملاء تصاعدياً حسب قيمة الفاتورة#####
#####إظهار أول 3 عملاء لهم الفاتورة الأعلى#####
def sort_bills_3customer(customers):
    sort_bills = {}
    lst=list()
    name_lst=list()
    ## customer_name=''
    for phone_number, customer in customers.items():
        bills = customer.get('bill')
        name_customer=customer.get('name')

```

```

        sort_bills[phone_number] =( bills ,name_customer)

for key,val in sort_bills.items():
    lst.append(val)

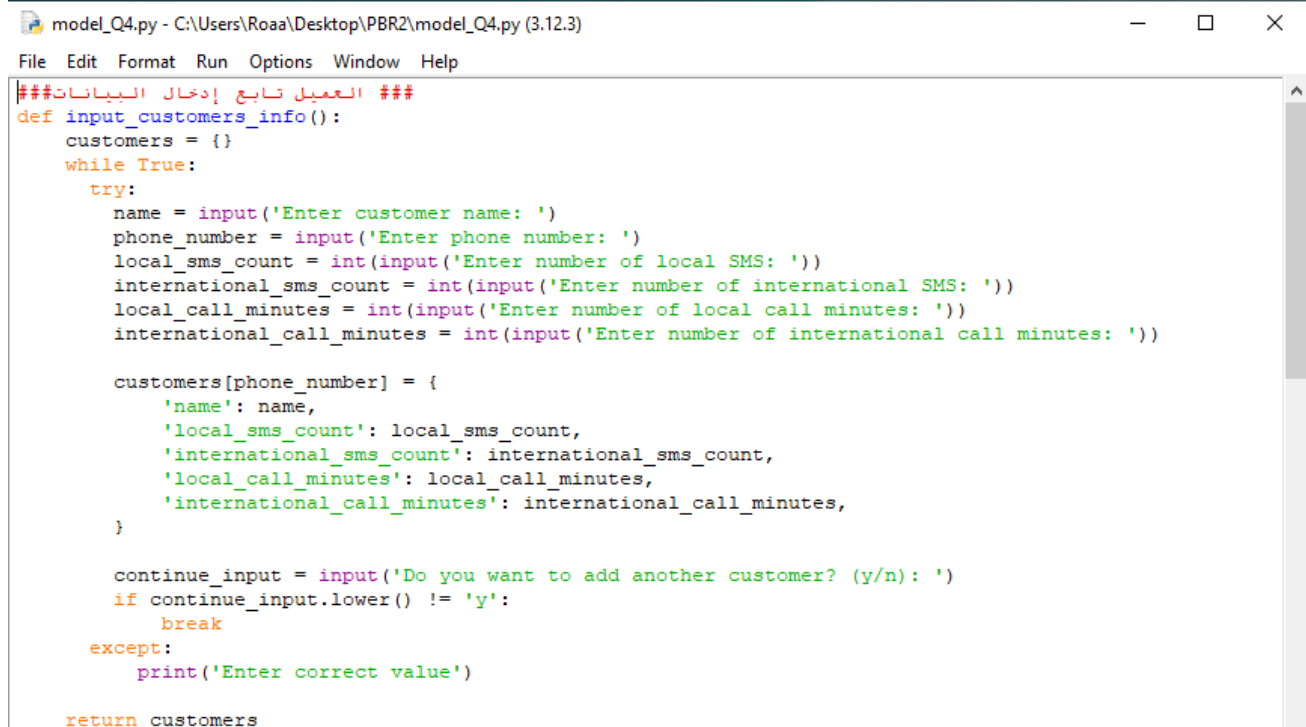
lst.sort()
for bill,name in lst:
    name_lst.append(name)
return name_lst[len(name_lst)-3:]

##### حفظ معلومات العميل في ملف #####
def file_save(filename,customers) :

    try:
        file = open(filename, 'a')
    except:
        print('File cannot be opened:', filename)

    for phone_number, customer in customers.items():
        line='phone number: '+phone_number+' info: '+str(customer)+'\n'
        file.write(line)
    file.close()

```



The screenshot shows a Python IDE window titled "model_Q4.py - C:\Users\Roaa\Desktop\PBR2\model_Q4.py (3.12.3)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```

##### العميل تابع إدخال البيانات #####
def input_customers_info():
    customers = {}
    while True:
        try:
            name = input('Enter customer name: ')
            phone_number = input('Enter phone number: ')
            local_sms_count = int(input('Enter number of local SMS: '))
            international_sms_count = int(input('Enter number of international SMS: '))
            local_call_minutes = int(input('Enter number of local call minutes: '))
            international_call_minutes = int(input('Enter number of international call minutes: '))

            customers[phone_number] = {
                'name': name,
                'local_sms_count': local_sms_count,
                'international_sms_count': international_sms_count,
                'local_call_minutes': local_call_minutes,
                'international_call_minutes': international_call_minutes,
            }

            continue_input = input('Do you want to add another customer? (y/n): ')
            if continue_input.lower() != 'y':
                break
        except:
            print('Enter correct value')

    return customers

```


model_Q4.py - C:\Users\Roaa\Desktop\PBR2\model_Q4.py (3.12.3)

File Edit Format Run Options Window Help

###تابع حساب الفاتورة##

def calculate_bill(customer):

local_sms = customer['local_sms_count']

local_sms_cost = 0

if local_sms <= 5:

local_sms_cost = local_sms * 50

else:

local_sms_cost = (5 * 50) + ((local_sms - 5) * 35)

international_sms = customer['international_sms_count']

international_sms_cost = international_sms * 100

local_minutes = customer['local_call_minutes']

local_call_cost = 0

if local_minutes <= 5:

local_call_cost = local_minutes * 30

elif local_minutes <= 10:

local_call_cost = 5 * 30

else:

local_call_cost = (5 * 30) + ((local_minutes - 10) * 25)

international_minutes = customer['international_call_minutes']

international_call_cost = international_minutes * 200

total_bill = local_sms_cost + international_sms_cost + local_call_cost + international_call_cost

return total_bill

###تابع لحظ الفاتورة في قاموس معلومات العميل###

def store_bills(customers):

for phone_number, customer in customers.items():

bill = calculate_bill(customer)

customer['bill'] = bill

return customers

model_Q4.py - C:\Users\Roaa\Desktop\PBR2\model_Q4.py (3.12.3)

File Edit Format Run Options Window Help

##تابع لإظهار فاتورة رقم يدخله المستخدم إذا وجد##

def number_bill(phone, customers):

customer = customers.get(phone, None)

if customer != None:

bills = customer.get('bill')

return phone, bills

###تابع يظهر رقم العميل الذي له أكبر فاتورة###

def max_bill(customers):

max_customer = ''

max_bill_value = 0

for phone_number, customer in customers.items():

bill = customer.get('bill', 0)

if bill > max_bill_value:

max_bill_value = bill

max_customer = phone_number

return max_customer

###تابع يظهر اسم العميل الذي له أكبر فاتورة###

def max_bill_customers(customers):

max_customer = ''

max_bill_value = 0

for phone_number, customer in customers.items():

bill = customer.get('bill', 0)

if bill > max_bill_value:

max_bill_value = bill

max_customer = customer['name']

return max_customer

```

#####ترتيب العملاء حسب قيمة الفاتورة تنازلياً#####
def sort_bills(customers):
    sort_bills = {}
    lst=list()
    name_lst=list()
    # customer_name=''
    for phone_number, customer in customers.items():
        bills = customer.get('bill')
        name_customer=customer.get('name')
        sort_bills[phone_number] =( bills ,name_customer)

    for key,val in sort_bills.items():
        lst.append(val)

    lst.sort(reverse=True)
    for bill,name in lst:
        name_lst.append(name)

    return name_lst
##ترتيب العملاء تصاعدياً حسب قيمة الفاتورة##
##إظهار أول 3 عملاء لهم الفاتورة الأعلى##
def sort_bills_3customer(customers):
    sort_bills = {}
    lst=list()
    name_lst=list()
    ## customer_name=''
    for phone_number, customer in customers.items():
        bills = customer.get('bill')
        name_customer=customer.get('name')
        sort_bills[phone_number] =( bills ,name_customer)

    for key,val in sort_bills.items():
        lst.append(val)

    lst.sort()
    for bill,name in lst:
        name_lst.append(name)
    return name_lst[len(name_lst)-3:]

```

```

|
| #####حفظ معلومات العميل في ملف#####
def file_save(filename,customers) :
|
|     try:
|         file = open(filename, 'a')
|     except:
|         print('File cannot be opened:', filename)
|
|     for phone_number, customer in customers.items():
|         line='phone number: '+phone_number+' info: '+str(customer)+'\n'
|         file.write(line)
|     file.close()

```