

Syria Arab Rrpublic

Lattakia – Tishreen University

Department of Communication
and electrical engineering
Homework No 1



الجمهورية العربية السورية

اللاذقية – جامعة تشرين

كلية الهندسة الكهربائية و الميكانيكية

قسم هندسة الاتصالات و الالكترونيات

السنة الخامسة : وظيفة ١ برمجة

- مقرر برمجة و ادارة شبكات -

الدكتور مهند عيسى

إشراف :

صالح مروان شهيلة 2910

إعداد :

First Network Programming Homework

Question 1: Python Basics?

A- If you have two lists, L1=['HTTP','HTTPS','FTP','DNS'] L2=[80,443,21,53], convert it to generate this dictionary d={'HTTP':80,'HTTPS':443,'FTP':21,'DNS':53 }.

B- Write a Python program that calculates the factorial of a given number entered by user.

C- L=['Network' , 'Bio' , 'Programming' , 'Physics' , 'Music'] In this exercise, you will implement a Python program that reads the items of the previous list and identifies the items that starts with 'B' letter, then print it on screen. Tips: using loop, 'len ()' , startswith() methods.

D- Using Dictionary comprehension, Generate this dictionary
d={0:1,1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11}

شرح كود السؤال الأول :

```
A1.py
C: > Users > ccl > Desktop > A1.py > ...
1  #Question 1 :
2
3  d={}
4  L1=['HTTP','HTTPS','FTP','DNS']
5  L2=[80,443,20,53]
6  for i , j in zip (L1,L2):
7      d[i]=j
8  print(d)
9
```

الطلب A :

يتم إنشاء قاموس فارغ " d "

يتم بعدها تعريف قائمتين " L1 " و

"L2" تحتويان على بروتوكولات الشبكة و أرقام المنفذ الخاصة بها على التوالي .

يتم استخدام حلقة " For " مع دالة " Zip " لتكرار القائمتين في وقت واحد و الرابط بين عناصر كل قائمة في القاموس " d " .

يتم طباعة القاموس " d " الذي تم إنشاؤه و الذي يحتوي الآن على بروتوكولات الشبكة كمفتاح و أرقام المنفذ الخاصة بها كقيم .

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python
PS C:\Users\ccl> & C:/Users/cc1/AppData/Local/Programs/Python/Python311/python.exe c:/Users/cc1/Desktop/A1.py
{'HTTP': 80, 'HTTPS': 443, 'FTP': 20, 'DNS': 53}
PS C:\Users\ccl>
```

الطلب B :

```
A1.py
C: > Users > ccl > Desktop > A1.py > ...
1 #Question 1/B :
2
3 def factorial(k):
4     if k == 0 or k == 1:
5         return 1
6     else:
7         t = k * factorial(k - 1)
8         return t
9
10 number = int(input("Enter integer: "))
11 if number > 0:
12     result = factorial(number)
13     print("factorial of: ", number, "is: ", result)
14 else:
15     print("Error")
16
```

يقوم الكود بإيجاد مضروب الرقم هو حاصل ضرب جميع الأرقام حتى هذا الرقم .

يحصل الكود أولاً على المدخلات من المستخدم من خلال وظيفة الإدخال و يحولها إلى عدد صحيح باستخدام وظيفة `int` .

ثم يتحقق مما إذا كان الرقم أكبر من 0 ، إذا كان الرقم موجبا فإنه يستدعي دالة المعامل مع الرقم كوسيط و يخزن النتيجة في متغير النتيجة .

دالة المعامل هي دالة عودة تحسب معامل الرقم عن طريق ضرب الرقم في معامل الرقم السابق .

إذا كان الرقم 0 أو 1 فسيتم إرجاع 1 ، إذا كان الرقم أكبر من 1 فإنه يقوم بضرب الرقم السابق و إرجاع النتيجة .

و أخيراً ، يطبع الكود نتيجة حساب العوامل . إذا كان الرقم غير موجب ، فإنه يطبع رسالة خطأ .

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python
PS C:\Users\ccl> & C:/Users/cc1/AppData/Local/Programs/Python/Python311/python.exe c:/Users/cc1/Desktop/A1.py
Enter integer: 20
factorial of: 20 is: 2432902008176640000
PS C:\Users\ccl> & C:/Users/cc1/AppData/Local/Programs/Python/Python311/python.exe c:/Users/cc1/Desktop/A1.py
Enter integer: -1
Error
PS C:\Users\ccl>
```

الطلب C :

يقدم هذا الكود برنامج يتكرر عبر قائمة من السلاسل و يطبع السلاسل التي تبدأ بحرف " M " .

يتم تخزين قائمة السلاسل في المتغير L و يتكرر البرنامج عبر قائمة باستخدام حلقة `for` .

يتم استخدام متغير الحلقة `i` للوصول إلى عناصر القائمة و يتم استخدام الدالة

`" Range (len (L)) "` للتكرار على مؤشرات القائمة .

بالنسبة لكل عنصر من عناصر القائمة ، يتحقق البرنامج مما إذا كان العنصر يبدأ بالحرف "M" باستخدام الدالة `" beginwith "` . إذا كان العنصر يبدأ بالحرف "M" يقوم البرنامج بطباعة العناصر باستخدام وظيفة الطباعة .

فيما يلي شرح تنفيذ الكود خطوة بخطوة :

```
A1.py X
C: > Users > cd > Desktop > A1.py > ...
1 #Question 1/C :
2
3 L=['Network','Bio','Programming','Physics','Music']
4 i = 0
5 for i in range(len(L)):
6     if L[i].startswith('M'):
7         print(L[i])
8
9
10
```

- يقوم البرنامج بتهيئة المتغير **L** إلى قائمة السلاسل النصية
- يقوم البرنامج بتهيئة المتغير **i** إلى 0
- يقوم البرنامج بإدخال حلقة **for** التي تتكرر فوق مؤشرات القائمة **L**

داخل الحلقة يتحقق البرنامج مما إذا كان عنصر القائمة في الفهرس **i** يبدأ بالحرف "M" باستخدام الدالة **"beginwith"**.

إذا كان العنصر يبدأ بالحرف "M" يقوم البرنامج بطباعة العنصر باستخدام وظيفة الطباعة .

تستمر الحلقة في التكرار على العناصر المتبقية من القائمة حتى يتم فحص جميع العناصر .

فيكون الخرج **Music** لأن العنصر الوحيد في القائمة الذي يبدأ بالحرف "M" هو **Music**.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python
PS C:\Users\cc1> & C:/Users/cc1/AppData/Local/Programs/Python/Python311/python.exe c:/Users/cc1/Desktop/A1.py
Music
PS C:\Users\cc1>
```

الطلب D :

يقوم الكود بفهم القاموس (هو طريقة مختصرة لإنشاء قاموس من كائن قابل للتكرار ، حيث يتم حساب المفاتيح و القيم من عناصر القابلة للتكرار) .

في هذه الحالة يكون التكرار هو نطاق الأرقام من 0 إلى 10 الذي تم إنشاؤه بواسطة دالة النطاق (11) . لكل رقم **h** في النطاق تتم إضافة زوج القيمة الرئيسية (**h** , **h+1**) إلى القاموس . هذا يعني أن المفتاح هو الرقم **h** و القيمة هي **h+1** .

يتم تخزين القاموس الناتج في المتغير **k** و يتم طباعته علة وحدة التحكم باستخدام وظيفة الطباعة .

فيما يلي شرح تنفيذ الكود خطوة بخطوة :

```
A1.py X
C: > Users > ccl > Desktop > A1.py > ...
1 k={h:h+1 for h in range(11)}
2 print(k)
3
4
5
6 |
```

- يقوم الكود بإنشاء فهم القاموس
- يتكرر فهم القاموس عبر نطاق الأرقام من 0 إلى 10 الذي تم إنشاؤه بواسطة دالة النطاق (11). لكل رقم h في النطاق تتم إضافة زوج القيمة الرئيسية ($h, h+1$) إلى القاموس. هذا يعني أن المفتاح هو الرقم h و القيمة هي $h+1$
- يتم تخزين القاموس الناتج في المتغير k .
- يتم طباعة القاموس على وحدة التحكم باستخدام وظيفة الطباعة.
- فتكون المخرجات كما في الكود.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python
PS C:\Users\ccl> & C:/Users/cc1/AppData/Local/Programs/Python/Python311/python.exe c:/Users/cc1/Desktop/A1.py
Music
PS C:\Users\ccl>
```

Question 2: Convert from Binary to Decimal :

Write a Python program that converts a Binary number into its equivalent Decimal number. The program should start reading the binary number from the user. Then the decimal equivalent number must be calculated. Finally, the program must display the equivalent decimal number on the screen

شرح الكود السؤال الثاني :

يقوم الكود بتحويل رقم ثنائي إلى معادلة بالعشري .

كيفية عمل الكود :

- يقوم بتهيئة **dec** إلى 0 و **m** إلى قائمة فارغة .
- يقوم بقراءة الرقم الثنائي كسلسلة من المستخدم و يقوم بتخزينها في **n** .
- يقوم بتكرار الرقم الثنائي و ألحق كل رقم بالقائمة **m** .
- ثم يعكس القائمة **m** بحيث تكون الأرقام بالترتيب الصحيح للتحويل .
- يقوم بتحويل كل رقم في **m** إلى عدد صحيح و ضربه في 2 مرفوعاً أس و موضوعه في القائمة المعكوسة ثم يضيف النتيجة إلى **dec**
- إذا تعذر تحويل أي من الأرقام الموجودة في **m** إلى عدد صحيح فسوف يطبع خطأ .
- بخلاف ذلك يقوم بطباعة قيمة **dec** .

```
Question2.py
C: > Users > ccl > Desktop > Question2.py > n
1 #Question 2 :
2
3 dec= 0
4 m=[]
5 n=input("enter binary number : ")
6 for i in n:
7     m.append(i)
8 m.reverse()
9 try:
10     for i in range(len(m)):
11         dec+=int(m[i]) * 2**i
12         print(dec)
13 except:
14     print("error")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ccl> & C:/Users/ccl/AppData/Local/Programs/Python/Python311/python.exe c:/Users/ccl/Desktop/Question2.py
enter binary number : 1101
13
PS C:\Users\ccl> 1100
1100
PS C:\Users\ccl> 
```

Question 3: Working with Files " Quiz Program " :

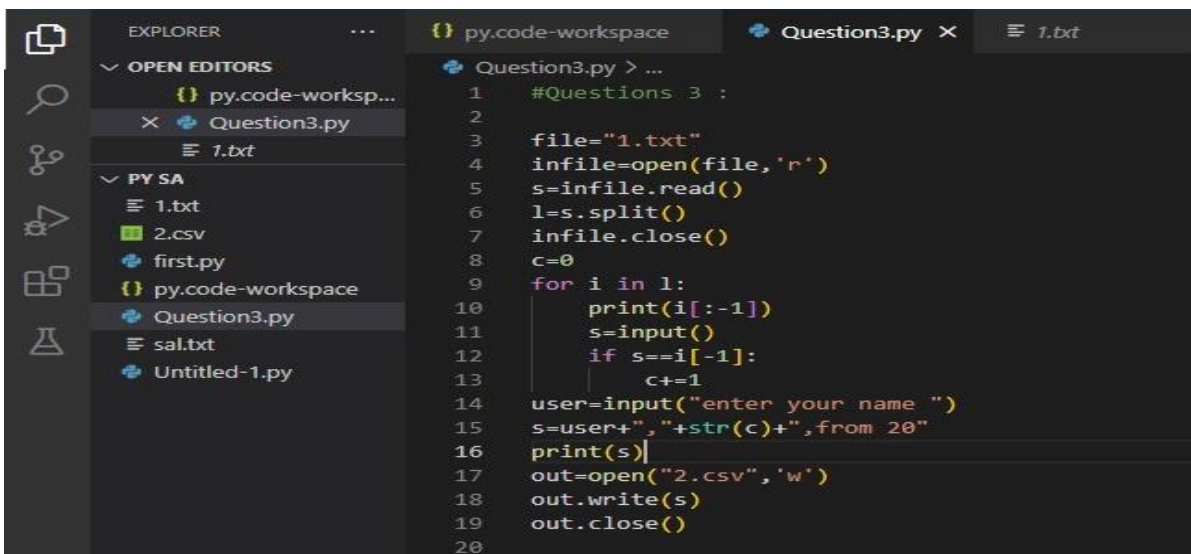
Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file csv or json file

شرح الكود السؤال الثالث :

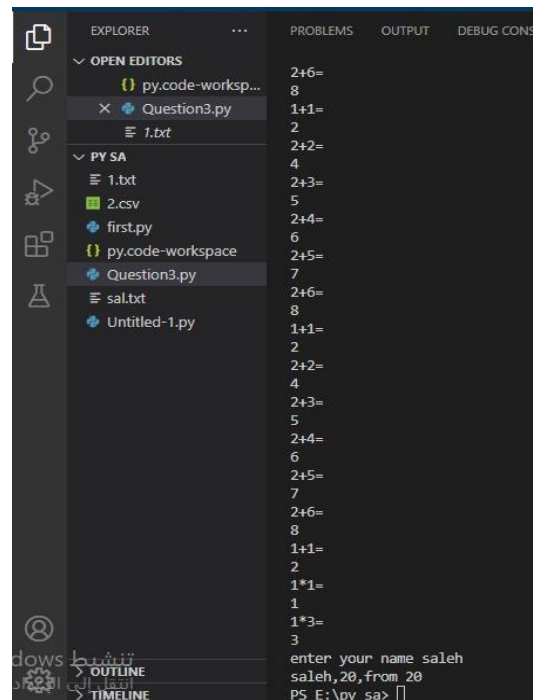
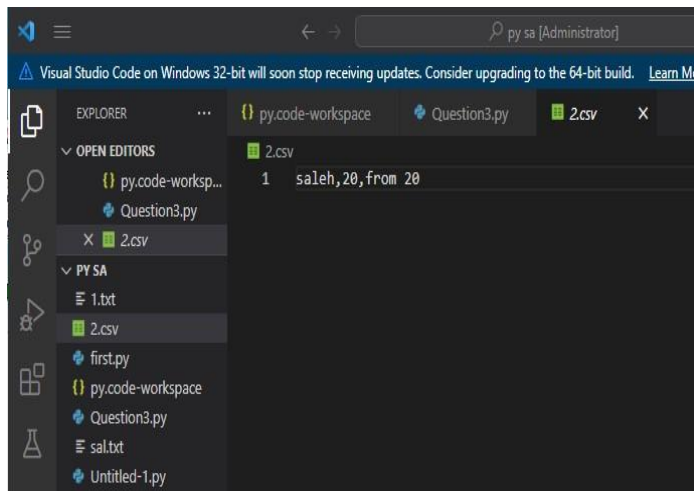
يقوم الكود بقراءة ملف نصي و يعالج محتواه و يكتب في ملف CSV

كيفية عمل الكود :

- يقوم بفتح ملف نصي **1.txt** لقراءة الملف و تخزينه في ملف المتغير .
- يقوم بقراءة محتويات الملف النصي في سلسلة باستخدام طريقة القراءة للملف .
- ثم يقوم بتقسيم السلسلة إلى قائمة كلمات باستخدام **Split()**
- ثم يقوم بتهيئة المتغير **C** إلى **0**
- ويكرر من خلال قائمة الكلمات **I**
- لكل كلمة **i** في القائمة ، يقوم بطباعة الكلمة بدون الحرف الأخير باستخدام عامل الشريحة - : **i[1]**
- ثم يطلب من المستخدم إدخال حرف **s** يساوي الحرف الأخير من الكلمة **i** ، و يزيد المتغير **c** بمقدار **1** .
- ويطلب من المستخدم إدخال اسمه و تخزينه في المستخدم المتغير .
- ثم يقوم بإنشاء سلسلة **s** عن طريق ربط اسم المستخدم و قيمة **c** و السلسلة من **20** .
- ثم يطبع السلسلة .
- و يفتح ملف **2.csv** لكتابة كائن الملف و تخزينه في المتغير **out** .
- ويكتب السلسلة في الملف **csv** باستخدام ال **write ()**
- ثم يغلق الملف .



```
1 #Questions 3 :
2
3 file="1.txt"
4 infile=open(file,'r')
5 s=infile.read()
6 l=s.split()
7 infile.close()
8 c=0
9 for i in l:
10     print(i[:-1])
11     s=input()
12     if s==i[-1]:
13         c+=1
14 user=input("enter your name ")
15 s=user+" "+str(c)+" from 20"
16 print(s)
17 out=open("2.csv",'w')
18 out.write(s)
19 out.close()
20
```

Question 4: Object-Oriented Programming - Bank Class

Define a class BankAccount with the following attributes and methods:

Attributes: account_number (string), account_holder (string), balance (float, initialized to 0.0)

Methods: deposit(amount), withdraw(amount) , get_balance()

- Create an instance of BankAccount .
- , - Perform a deposit of \$1000, - Perform a withdrawal of \$500. - Print the current balance after each operation.
- Define a subclass SavingsAccount that inherits from BankAccount and adds interest_rate Attribute and apply_interest() method that Applies interest to the balance based on the interest rate. And Override print() method to print the current balance and rate.
- Create an instance of SavingsAccount , and call apply_interest() and print() functions

شرح الكود السؤال الرابع :

❖ يوضح الكود مفاهيم البرمجة غرضية التوجه من خلال تحديد class اي فئتين و إظهار استخدامهما . فئة الحساب البنكي : تعمل فئة " BankAccount " كفئة أساسية تمثل حساباً مصرفياً عامًا. لديها الطرق التالية : (init:1) تهيئة كائن " BankAccount " جديد برقم الحساب واسم صاحب الحساب ورصيد أولي قدره (0.2) الإيداع : إيداع مبلغ محدد في الحساب ، مما يؤدي إلى زيادة الرصيد 3 . السحب: سحب مبلغ محدد من الحساب إذا كان الرصيد كافيًا . وإلا فإنه يطبع رسالة "الرصيد غير كاف". (get_balance:4) .

إرجاع رصيد الحساب الجاري .
(str:5) . يوفر تمثيل سلسلة لكائن " BankAccount " ، بما في ذلك رقم الحساب واسم صاحب الحساب والرصيد .

❖ فئة حساب التوفير: تعمل فئة " SavingsAccount " على توسيع فئة "BankAccount" ، مما يضيف وظائف خاصة بحسابات التوفير : (init:1) تهيئة كائن "SavingsAccount" جديد برقم الحساب واسم صاحب الحساب وسعر الفائدة . (Apply_interest:2) احتساب الفائدة وإضافتها إلى رصيد الحساب بناءً على سعر الفائدة المحدد .

❖ حساب مصرفي عادي برقم حساب "2910" واسم صاحب الحساب "صالح شهلة" . (Saving_account:2) حساب توفير برقم حساب "01111" واسم صاحب الحساب "صالح مروان" ، ونسبة فائدة 5% . ثم يقوم الكود بتنفيذ العمليات التالية:
إيداع مبلغ 1000 في الحساب البنكي وطباعة تفاصيله .
سحب مبلغ 500 من الحساب البنكي وطباعة تفاصيله .
يقوم بإيداع مبلغ 11000 في حساب التوفير .
تطبيق الفائدة على حساب التوفير .
طباعة تفاصيل حساب التوفير .

```

Question4.py
Question4.py > ...
1  #Question4 :
2
3  class BankAccount:
4      def __init__(self, account_number, account_holder):
5          self.account_number = account_number
6          self.account_holder = account_holder
7          self.balance = 0.0
8      def deposit(self, amount):
9          self.balance += amount
10     def withdraw(self, amount):
11         if self.balance >= amount:
12             self.balance -= amount
13         else:
14             print("Insufficient balance . ")
15     def get_balance(self):
16         return self.balance
17     def __str__(self):
18         return f"Account Number : {self.account_number}\n Username : {self.account_holder}\n Balance : {self.balance}"
19

```

```

Question4.py
Question4.py > ...
20 class SavingsAccount(BankAccount):
21     def __init__(self, account_number, account_holder, interest_rate):
22         super().__init__(account_number, account_holder)
23         self.interest_rate = interest_rate
24     def apply_interest(self):
25         interest = self.balance * (self.interest_rate / 100)
26         self.balance += interest
27     def __str__(self):
28         return f"Account Number :{self.account_number}\n Username : {self.account_holder}\n Balance : {self.balance}\n Inte
29
30 bank_account = BankAccount("2910", "Saleh Shhela")
31 bank_account.deposit(1000)
32 print(bank_account)
33 bank_account.withdraw(500)
34 print(bank_account)
35 savings_account = SavingsAccount("01111", "Saleh Marwan ", 5)
36 savings_account.deposit(11000)
37 savings_account.apply_interest()
38 print(savings_account)
39

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
> & C:/Users/cc1/AppData/Local/Programs/Python/Python311/python.e
xe "e:/py sa/Question4.py"
Account Number : 2910
Username : Saleh Shhela
Balance : 1000.0
Account Number : 2910
Username : Saleh Shhela
Balance : 500.0
Account Number : 01111
Username : Saleh Marwan
Balance : 11550.0
Interest : 5%
PS E:\py sa>
Ln 17, Col 5  Spaces: 4  UT

```