



الجمهورية العربية السورية

وزارة التعليم العالي والبحث العلمي

جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الالكترونيات والاتصالات

السنة الخامسة: وظيفة 1 برمجة شبكات

الاسم: ناره هيثم الغرير

الرقم: 2906

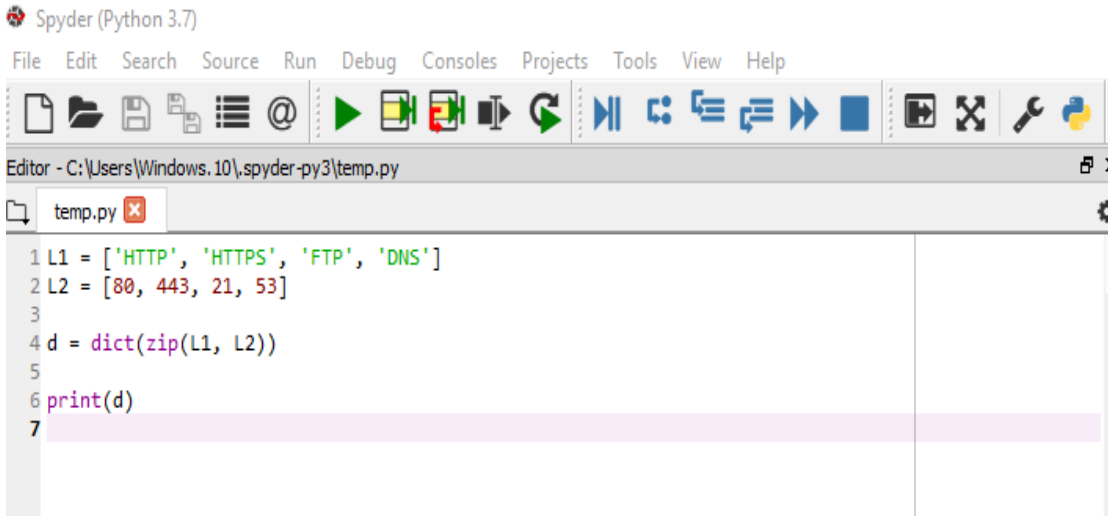
### Question 1: python basics?

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]

الحل :

سنقوم بإنشاء القائمتين L1 و L2.  
ثم سوف نستخدم دالة zip() لإنشاء قائمة من الأزواج  
بين العناصر المتناظرة في L1 و L2.  
و تحويل هذه القائمة المحكمة إلى قاموس باستخدام دالة  
dict().

## وطباعة القاموس النهائي الذي يحتوي على النتيجة المطلوبة.



```
1 L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
2 L2 = [80, 443, 21, 53]
3
4 d = dict(zip(L1, L2))
5
6 print(d)
7
```

**B- write a python program that calculates the factorial of a given number entered by user.**

**الحل :**

سوف نطلب من المستخدم إدخال الرقم الذي يريد حساب عامله التكراري.

و ثم نقوم بتهيئة متغير factorial بالقيمة ١ كنقطة بداية.

نستخدم حلقة for للتكرار من ١ إلى الرقم المدخل (بما في ذلك الرقم المدخل).

في كل تكرار، نضرب قيمة factorial في الرقم الحالي .i

بعد انتهاء الحلقة، نطبع قيمة factorial كنتيجة.

A screenshot of the Spyder Python IDE interface. The title bar reads 'Spyder (Python 3.7)'. The menu bar includes 'File', 'Edit', 'Search', 'Source', 'Run', 'Debug', 'Consoles', 'Projects', 'Tools', 'View', and 'Help'. Below the menu is a toolbar with icons for file operations (new, open, save, save as, print, close all), editing (undo, redo, cut, copy, paste), and execution (run, step through, step over, step under, break, toggle breakpoint, clear all breakpoints). The editor window shows a file named 'temp.py' with the following Python code:

```
1 num = int(input("inter the number: "))
2
3 factorial = 1
4
5 for i in range(1, num + 1):
6     factorial = factorial * i
7     print("factorial")
```

**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.

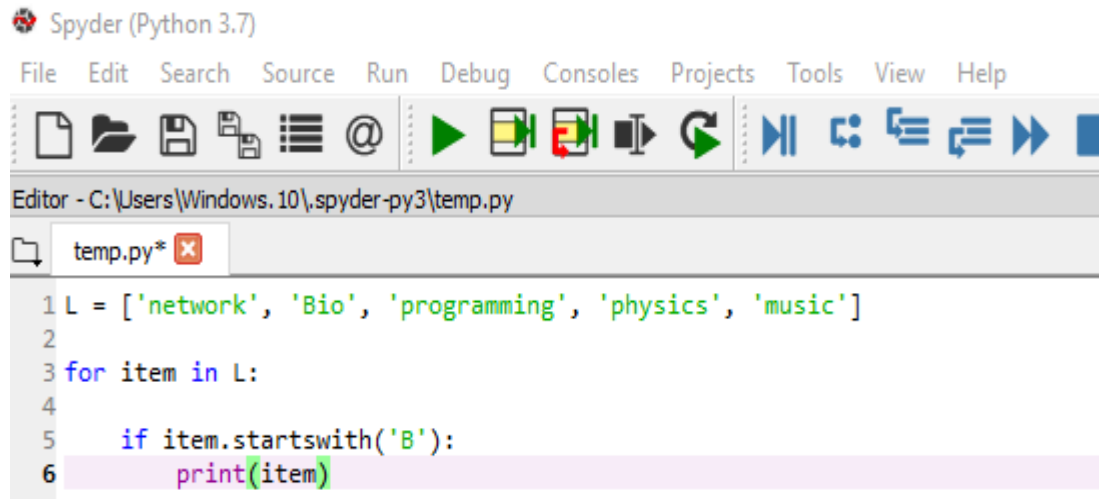
**الحل:**

نبدأ بتعريف القائمة L المطلوبة.

نستخدم حلقة for لمعالجة كل عنصر في القائمة L.

بالنسبة لكل عنصر، نستخدم الدالة startswith() لفحص ما إذا كان بداية العنصر هو الحرف 'B' .

إذا كان العنصر يبدأ بالحرف 'B'، فإننا نقوم بطباعته على الشاشة.



```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
temp.py*
1 L = ['network', 'Bio', 'programming', 'physics', 'music']
2
3 for item in L:
4
5     if item.startswith('B'):
6         print(item)
```

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}`

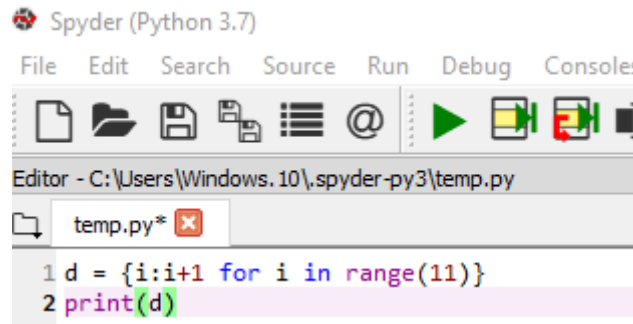
الحل:

سوف نستخدم تركيبة قاموس لإنشاء القاموس d.

داخل تركيبة القاموس، نستخدم حلقة for لتكرار الأرقام من ٠ إلى ١٠.

بالنسبة لكل رقم i، نقوم بإضافة قيمة i+1 إلى القاموس مع المفتاح i.

و نقوم بطباعة القاموس d النهائي.

The image shows a screenshot of the Spyder Python IDE interface. The title bar reads 'Spyder (Python 3.7)'. The menu bar includes 'File', 'Edit', 'Search', 'Source', 'Run', 'Debug', and 'Console'. Below the menu bar is a toolbar with icons for file operations (new, open, save, save as, print), editing (undo, redo, cut, copy, paste), and execution (run, debug, console). The editor window shows a file named 'temp.py' with the following Python code:

```
1 d = {i:i+1 for i in range(11)}  
2 print(d)
```

## Question 2 : **convert from binary to decimal**

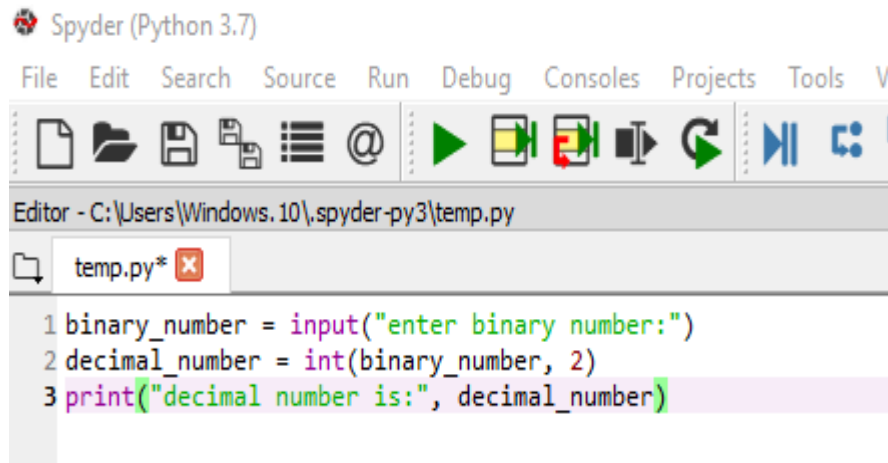
write a python program that converts a binary number into its equivalent decimal number . the program should start reading the binary number from the user .then the decimal equivalent number must be calculated . finally , the program must display the equivalent decimal number on the screen.

الحل: سوف نستخدم دالة `input()` لقراءة الرقم الثنائي من المستخدم وتخزينه في متغير `binary_number`.

ثم نستخدم دالة `int()` لتحويل الرقم الثنائي إلى رقم عشري. يأخذ دالة `int()` اثنين من المعطيات: النص الذي يمثل الرقم والقاعدة.

نقوم بتخزين الرقم العشري المكافئ في متغير `decimal_number`.

وبعدها نستخدم دالة `print()` لإظهار الرقم العشري المكافئ على الشاشة.



The screenshot shows the Spyder Python IDE interface. The title bar reads 'Spyder (Python 3.7)'. The menu bar includes 'File', 'Edit', 'Search', 'Source', 'Run', 'Debug', 'Consoles', 'Projects', and 'Tools'. The toolbar contains icons for file operations and execution. The editor window is titled 'Editor - C:\Users\Windows.10\spyder-py3\temp.py' and shows a Python script in 'temp.py\*'. The script consists of three lines: a prompt for a binary number, conversion to a decimal integer, and a print statement.

```
1 binary_number = input("enter binary number:")
2 decimal_number = int(binary_number, 2)
3 print("decimal number is:", decimal_number)
```

### Question 3 : working with files "Quiz program".

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

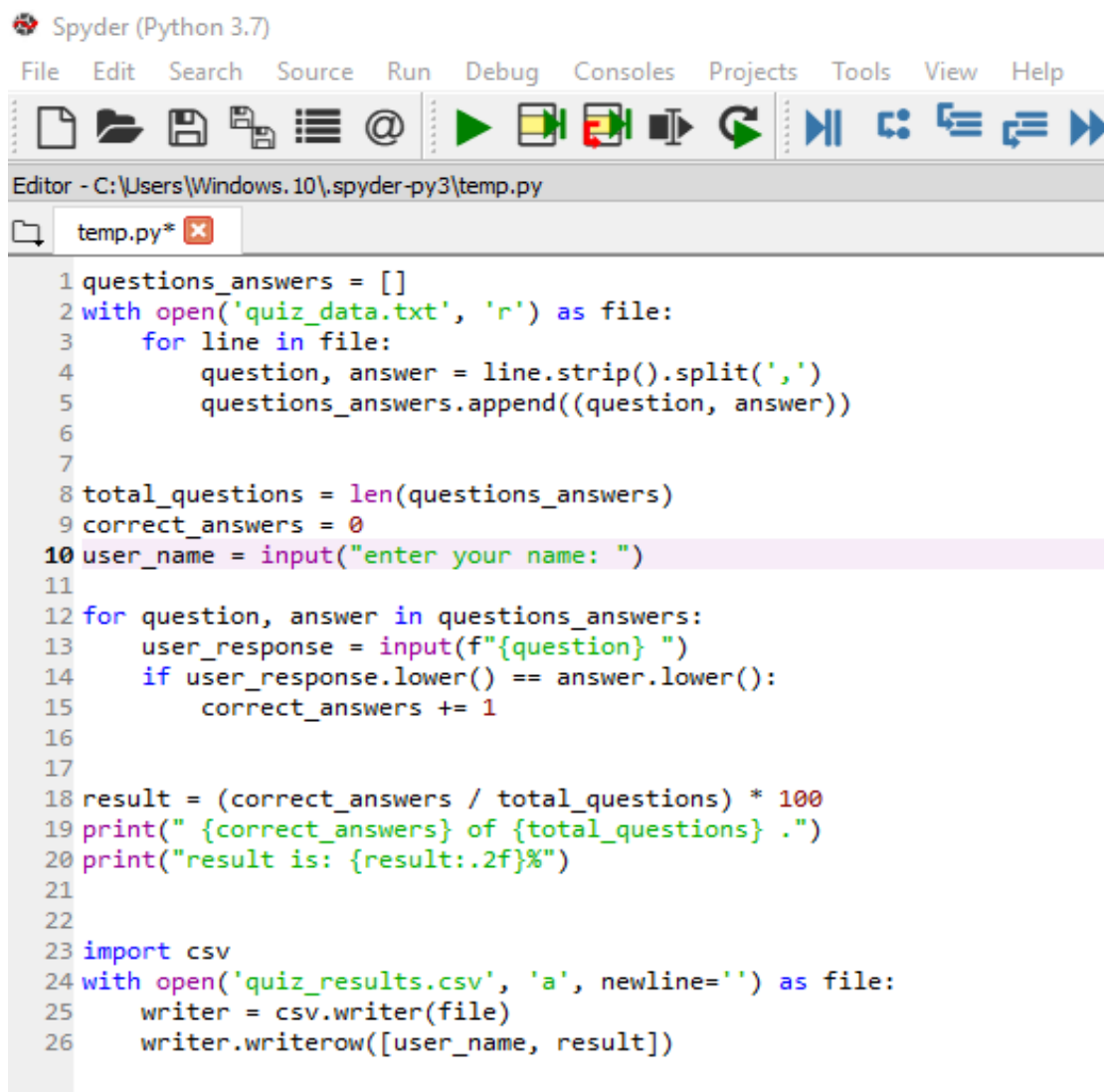
الحل: نقرأ أسئلة وإجابات الاختبار من ملف نص باستخدام حلقة for وندرجها في قائمة `questions_answers`.

نسأل المستخدم عن اسمه وندرجه في متغير `user_name`.

نمر على كل سؤال وإجابة في `questions_answers` وندرج إجابة المستخدم. إذا كانت الإجابة صحيحة، نزيد عدد الإجابات الصحيحة بواحد.

نحسب النتيجة النهائية للمستخدم ونطبعها.

نستخدم csv لإضافة اسم المستخدم والنتيجة إلى ملف CSV.



The image shows the Spyder Python IDE interface. The title bar reads "Spyder (Python 3.7)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. The toolbar contains icons for file operations (new, open, save, save as, print, close all), editing (undo, redo, cut, copy, paste, delete), and execution (run, debug, step through, step over, step under, break, continue). The editor window title is "Editor - C:\Users\Windows.10\spyder-py3\temp.py". The tab bar shows "temp.py\*" with a close button. The code in the editor is as follows:

```
1 questions_answers = []
2 with open('quiz_data.txt', 'r') as file:
3     for line in file:
4         question, answer = line.strip().split(',')
5         questions_answers.append((question, answer))
6
7
8 total_questions = len(questions_answers)
9 correct_answers = 0
10 user_name = input("enter your name: ")
11
12 for question, answer in questions_answers:
13     user_response = input(f"{question} ")
14     if user_response.lower() == answer.lower():
15         correct_answers += 1
16
17
18 result = (correct_answers / total_questions) * 100
19 print(" {correct_answers} of {total_questions} .")
20 print("result is: {result:.2f}%")
21
22
23 import csv
24 with open('quiz_results.csv', 'a', newline='') as file:
25     writer = csv.writer(file)
26     writer.writerow([user_name, result])
```

## 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 , intialized  
to 0.0).

methods: deposit(amount), withdraw(amount) , get\_balance().

-creat an instance of bankaccount, - perform a deposit of \$1000 , -perform a withdrawal of \$500.

- print the current balance after each operation.

- define a subclass savingAcvount 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 balace and rate.

- create an instance of savingsaccount , and call apply\_interest() and print() functions.

الحل: تحتوي فئة BankAccount على السمات المطلوبة  
(رقم\_الحساب، وحامل\_الحساب، والرصيد) والطرق (الإيداع،  
والسحب، والحصول على\_الرصيد). تم تجاوز الأسلوب  
\_\_str\_\_ لتوفير تمثيل سلسلة منسقة للحساب.

ترث فئة SavingsAccount من BankAccount وتضيف  
سمة سعر الفائدة وطريقة Apply\_interest. تم تجاوز الطريقة  
\_\_str\_\_ لتشمل سعر الفائدة.



في قسم الاختبار، نقوم بإنشاء مثيل **BankAccount**، وإجراء الإيداع والسحب، وطباعة الرصيد.

بعد ذلك، نقوم بإنشاء مثيل **SavingsAccount**، وإيداع مبلغ، وتطبيق الفائدة، وطباعة الرصيد مع سعر الفائدة.

```
1 class BankAccount:
2     def __init__(self, account_number, account_holder):
3         self.account_number = account_number
4         self.account_holder = account_holder
5         self.balance = 0.0
6
7     def deposit(self, amount):
8         self.balance += amount
9
10    def withdraw(self, amount):
11        if self.balance >= amount:
12            self.balance -= amount
13        else:
14            print("Insufficient funds.")
15    def get_balance(self):
16        return self.balance
17    def __str__(self):
18        return f"Account Number: {self.account_number}\nAccount Holder: {self.account_holder}\nBalance: {self.balance:.2f}"
19 class SavingsAccount(BankAccount):
20     def __init__(self, account_number, account_holder, interest_rate):
21         super().__init__(account_number, account_holder)
22         self.interest_rate = interest_rate
23     def apply_interest(self):
24         self.balance += self.balance * self.interest_rate
25     def __str__(self):
26         return super().__str__() + f"\nInterest Rate: {self.interest_rate * 100:.2f}%"
27
28 bank_account = BankAccount("123456789", "John Doe")
29 bank_account.deposit(1000)
30 print(bank_account)
31 bank_account.withdraw(500)
32 print(bank_account)
33
34 savings_account = SavingsAccount("987654321", "Jane Smith", 0.05)
35 savings_account.deposit(1000)
36 savings_account.apply_interest()
37 print(savings_account)
```