



الجمهورية العربية السورية

وزارة التعليم العالي والبحث العلمي

جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الإلكترونيات والاتصالات

السنة الخامسة: وظيفة 1 برمجة شبكات

الاسم: زين العابدين وائل عثمان

الرقم: 2884

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].

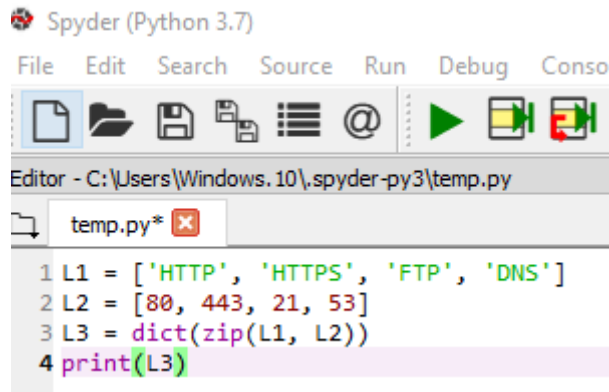
حل السؤال الأول A:

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().



```
1 L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
2 L2 = [80, 443, 21, 53]
3 L3 = dict(zip(L1, L2))
4 print(L3)
```

وستكون النتيجة:

{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53'}

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

حل السؤال الأول B:

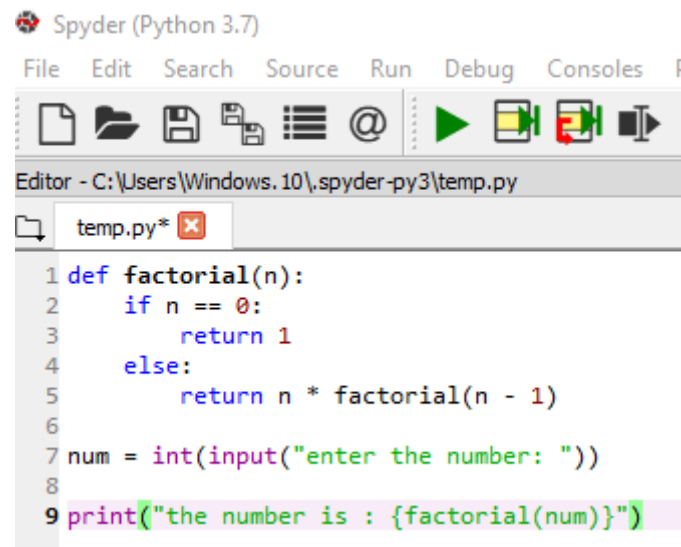
نعرف دالة **factorial()** تأخذ رقم **n** كمدخل.

داخل الدالة، نقوم بفحص إذا كان **n** مساوياً لـ ٠ ، وإذا كان كذلك نعيد القيمة ١ .

إذا لم يكن **n** مساوياً لـ ٠ ، نعيد ناتج **n** مضروباً في نتيجة استدعاء الدالة **factorial()** مع **n - 1**.

نطلب من المستخدم إدخال الرقم المراد حساب عامله التكراري.

نطبع النتيجة بإستدعاء الدالة **factorial()** مع الرقم المدخل.



```
1 def factorial(n):
2     if n == 0:
3         return 1
4     else:
5         return n * factorial(n - 1)
6
7 num = int(input("enter the number: "))
8
9 print("the number is : {factorial(num)}")
```

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.

حل السؤال الأول C:

نبدأ بتعريف القائمة L المطلوبة.

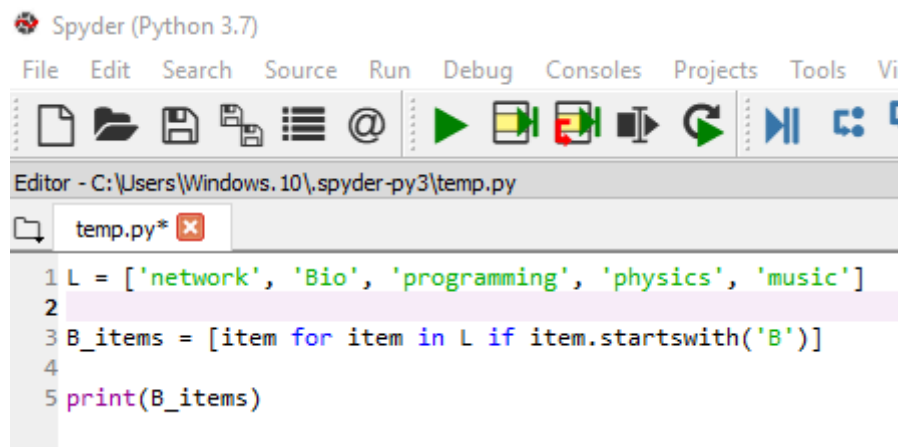
نستخدم قائمة تمهيدية لإنشاء قائمة جديدة B_items تحتوي على العناصر التي تبدأ بالحرف 'B'.

القائمة التمهيدية تتكون من:

[item for item in L]: يتم التكرار على جميع العناصر في القائمة L.

if item.startswith('B'): يتم تضمين العنصر في القائمة الجديدة فقط إذا بدأ بالحرف 'B'.

نقوم بطباعة القائمة الجديدة B_items التي تحتوي على العناصر التي تبدأ بالحرف 'B'.



```
1 L = ['network', 'Bio', 'programming', 'physics', 'music']
2
3 B_items = [item for item in L if item.startswith('B')]
4
5 print(B_items)
```

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}

حل السؤال الأول D:

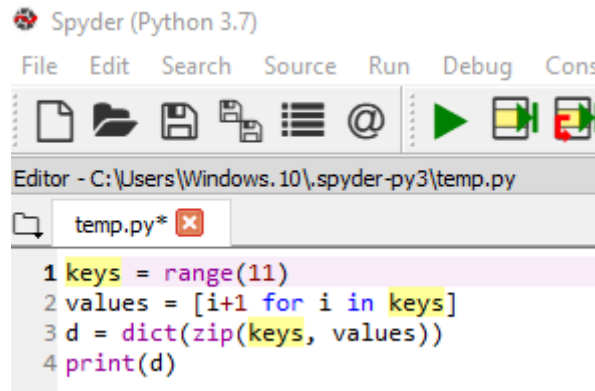
ستستخدم **range(11)** لإنشاء قائمة المفاتيح من 0 إلى 10.

نستخدم قائمة تمهيدية لإنشاء قائمة القيم من 1 إلى 11.

نستخدم **zip()** لإنشاء زوج من المفاتيح والقيم.

نستخدم **dict()** لتحويل الزوج إلى قاموس **d**.

نقوم بطباعة القاموس **d** النهائي.



```
1 keys = range(11)
2 values = [i+1 for i in keys]
3 d = dict(zip(keys, values))
4 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.

حل السؤال الثاني:

ستستخدم دالة `input()` لقراءة الرقم الثنائي من المستخدم وتخزينه في متغير `binary_number`.

نبدأ بتعيين `decimal_number` إلى ٠ .

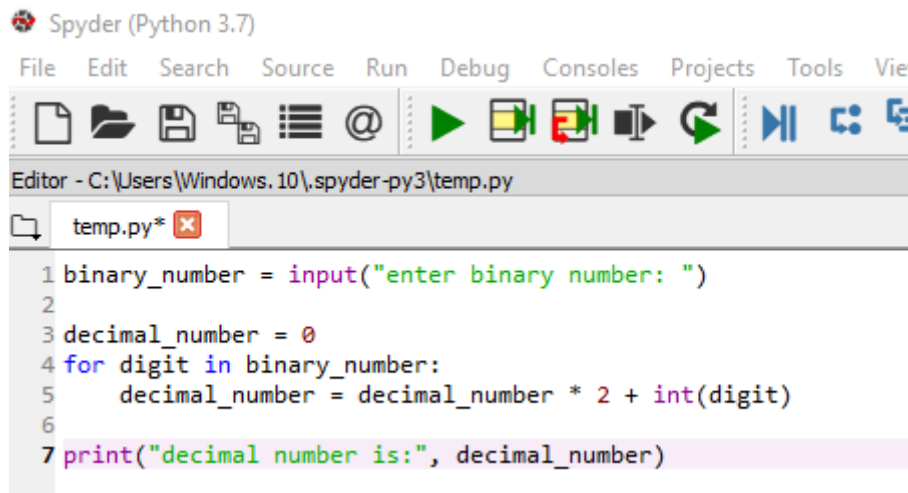
نستخدم حلقة `for` لتكرار كل رقم في `binary_number`:

في كل تكرار، نضاعف `decimal_number` في ٢ لتحريك الرقم إلى اليسار.

ثم نضيف قيمة الرقم الحالي (الذي تم تحويله إلى عدد صحيح باستخدام `int()`) إلى `decimal_number`.

بعد انتهاء الحلقة، decimal_number سيحتوي على الرقم العشري المكافئ.

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



```
1 binary_number = input("enter binary number: ")
2
3 decimal_number = 0
4 for digit in binary_number:
5     decimal_number = decimal_number * 2 + int(digit)
6
7 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.

حل السؤال الثالث:

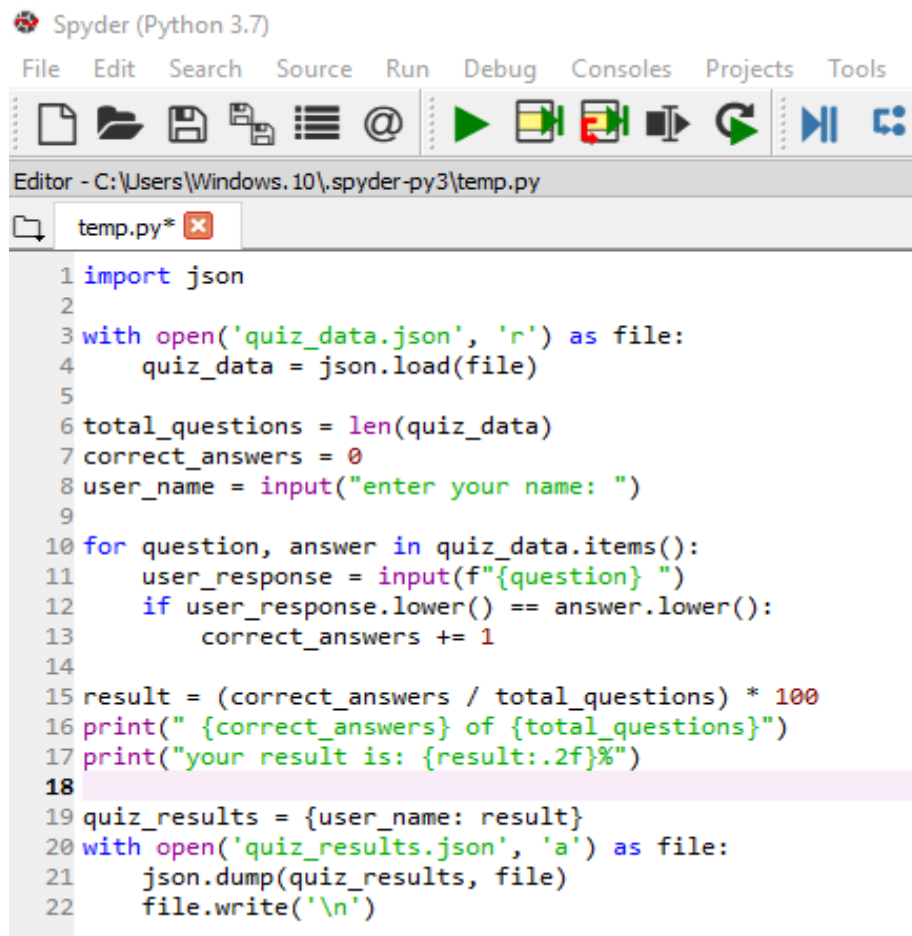
ستستخدم json لقراءة أسئلة وإجابات الاختبار من ملف JSON وتخزينها في متغير quiz_data.

نسأل المستخدم عن اسمه وندرجه في متغير user_name.

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

نحسب النتيجة النهائية للمستخدم ونطبعها.

نستخدم json لإضافة اسم المستخدم والنتيجة إلى ملف JSON، مع كتابة كل نتيجة في سطر جديد.



```
1 import json
2
3 with open('quiz_data.json', 'r') as file:
4     quiz_data = json.load(file)
5
6 total_questions = len(quiz_data)
7 correct_answers = 0
8 user_name = input("enter your name: ")
9
10 for question, answer in quiz_data.items():
11     user_response = input(f"{question} ")
12     if user_response.lower() == answer.lower():
13         correct_answers += 1
14
15 result = (correct_answers / total_questions) * 100
16 print(" {correct_answers} of {total_questions}")
17 print("your result is: {result:.2f}%")
18
19 quiz_results = {user_name: result}
20 with open('quiz_results.json', 'a') as file:
21     json.dump(quiz_results, file)
22     file.write('\n')
```

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.

حل السؤال الرابع:

ستكون فئة BankAccount بقيمة افتراضية تبلغ 0.0 لمعلمة الرصيد في المنشئ.

تحتوي فئة SavingsAccount أيضاً على قيمة افتراضية تبلغ 0.0 لمعلمة الرصيد في المنشئ وتضيف معلمة الفائدة.

في قسم الاختبار، يختلف إنشاء مثيلات BankAccount و SavingsAccount قليلاً. يتم إنشاء مثيل SavingsAccount برصيد أولي قدره 1000 ومعدل فائدة قدره 0.05.

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Windows.10

Editor - C:\Users\Windows.10\spyder-py3\temp.py

temp.py*

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