

وظيفة برمجة الشبكات

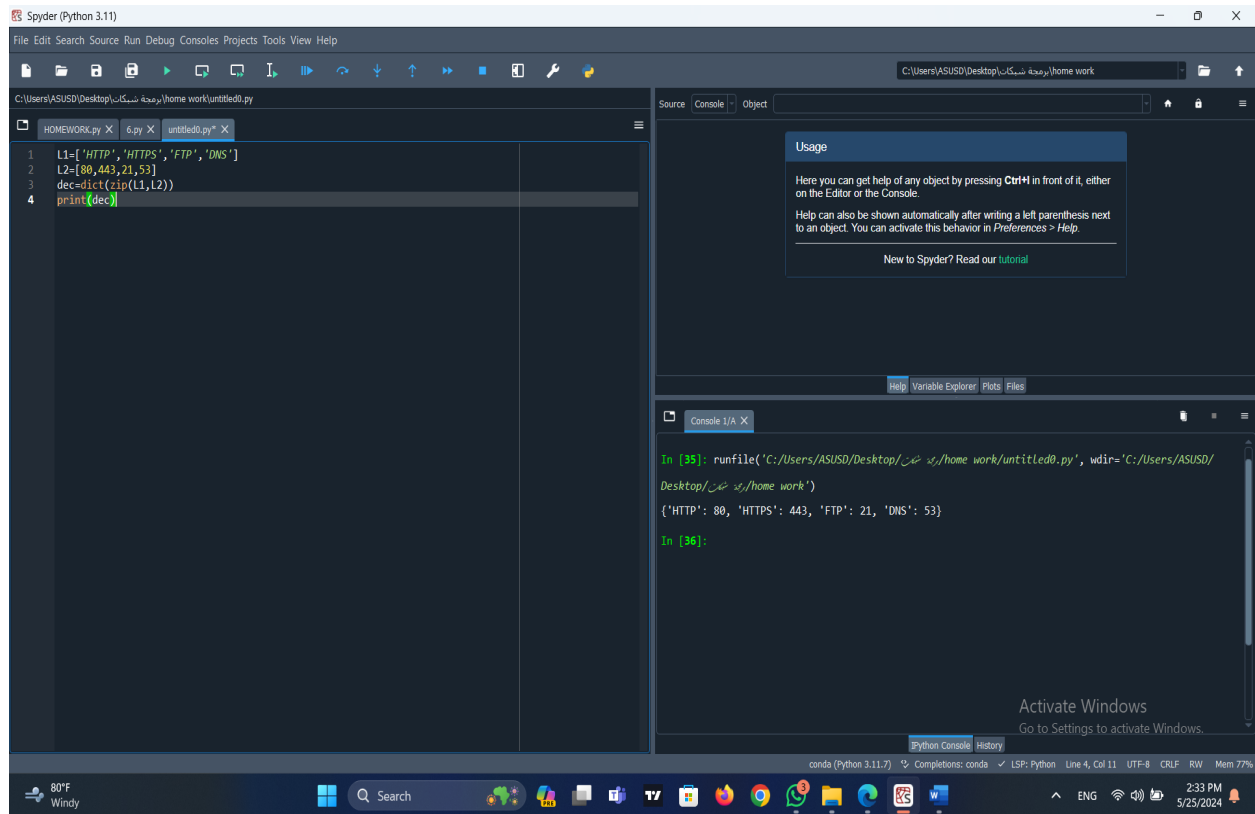
إعداد الطالب :

يوسف غسان خليل

2094

Q_1:

A



The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script in a file named 'untitled0.py'. The script contains the following code:

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

The right-hand pane is divided into two sections. The top section, labeled 'Console', shows the output of the script execution:

```
In [35]: runfile('C:/Users/ASUSD/Desktop/برمجة شبكات/home work/untitled0.py', wdir='C:/Users/ASUSD/Desktop/برمجة شبكات/home work')
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
In [36]:
```

The bottom section, labeled 'Object Explorer', shows the variable 'dec' and its value: {'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}.

شرح الكود :

عرفنا المصفوفة L2 , L1 و القاموس dec

قمنا بدمج المصفوفة L2 , L1 مع بعضهما عن طريق التابع zip حسب الفهارس و حولنا المصفوفة التي نتجت إلى قاموس عن طريق التابع dict() و الخرج في يمين الصورة من الأسفل .

B:

The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script for calculating the factorial of a number. The code is as follows:

```
1 def factorial(x):
2     if x==0:
3         return 1
4     else:
5         return x*factorial(x-1)
num= int(input("Enter a num > 0: "))
7 if num<0:
8     print("ERROR")
9 else:
10    print("factorial", num, "is",factorial(num))
```

The right-hand pane shows the IPython Console with the following output:

```
In [39]: runfile('C:/Users/ASUSD/Desktop/برمجة شمسكاتب/home work/untitled0.py', wdir='C:/Users/ASUSD/Desktop/برمجة شمسكاتب/home work')
Bio
In [40]: runfile('C:/Users/ASUSD/Desktop/برمجة شمسكاتب/home work/untitled0.py', wdir='C:/Users/ASUSD/Desktop/برمجة شمسكاتب/home work')
Enter a num > 0: 4
factorial 4 is 24
In [41]:
```

The bottom status bar indicates the current environment is conda (Python 3.11.7) and the file is LSP: Python, Line 10, Col 49, UTF-8, CRLF, RW, Mem 81%.

شرح الكود :

قمنا بتعريف تابع يحسب لنا عاملي عدد ما

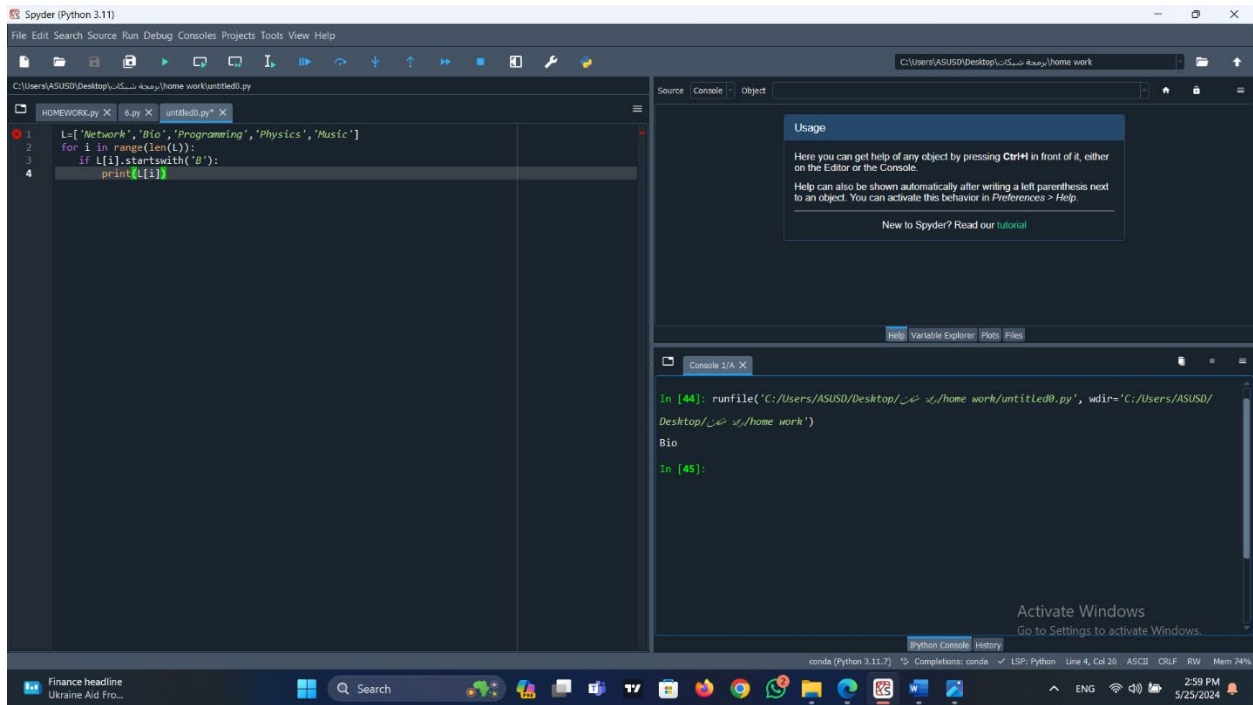
في حال كان العدد المدخل = 0 فإنه يعيد 1

و في حال أكبر من الصفر يعيد ناتج ضرب العدد الذي تم إدخاله ب

Factorial (x-1)

في الخرج أدخلنا العدد 4 فكان الناتج 24

C:

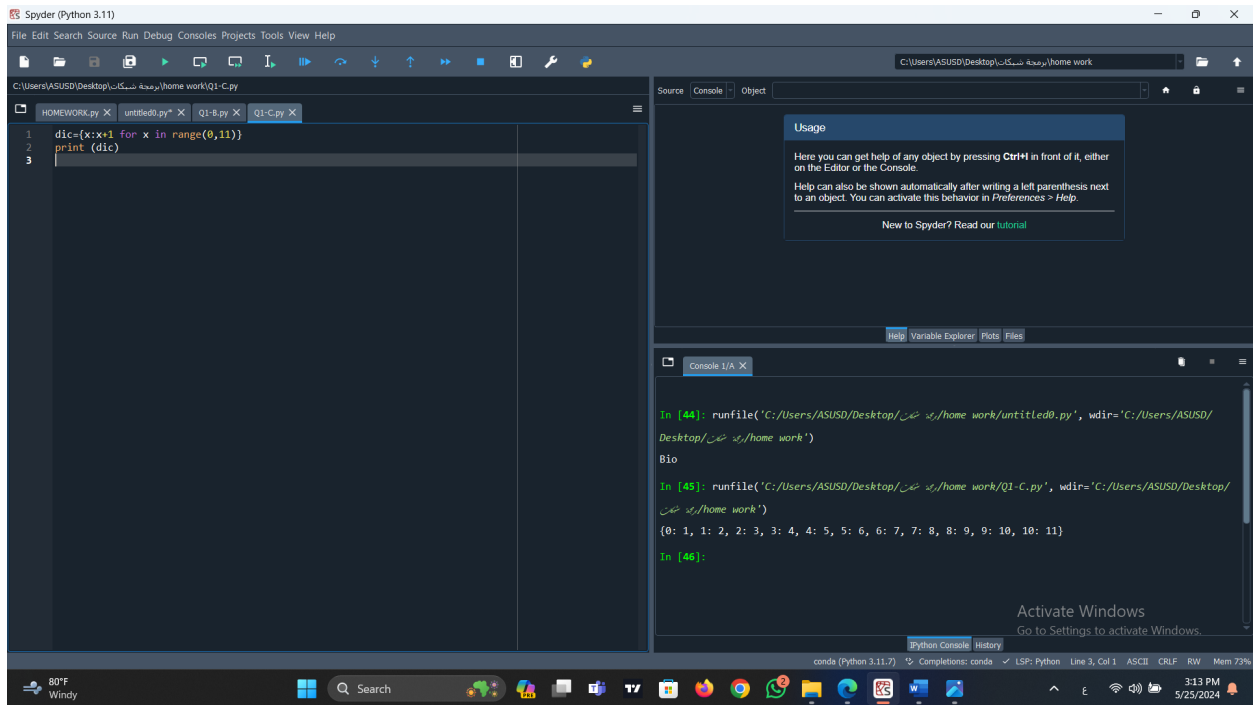


شرح الكود :

قمنا بتعريف مصفوفة من النوع string , ثم عرفنا حلقة for بمتحول i , و تم معرفة عدد عناصر المصفوفة من التابع len(L) , و وضعنا تعليمة طباعة لأول كلمة تبدأ بحرف 'B' باستخدام التابع startswith('B')

الخرج يكون: Bio

Q1_C:



The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script named `Q1-C.py` with the following code:

```
1 dic={x:x+1 for x in range(0,11)}
2 print (dic)
3
```

The right-hand pane shows the IPython Console with the following output:

```
In [44]: runfile('C:/Users/ASUSD/Desktop/مكتبات/home work/untitled0.py', wdir='C:/Users/ASUSD/Desktop/مكتبات/home work')
Bio
In [45]: runfile('C:/Users/ASUSD/Desktop/مكتبات/home work/Q1-C.py', wdir='C:/Users/ASUSD/Desktop/مكتبات/home work')
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
In [46]:
```

The bottom status bar indicates the environment is `conda (Python 3.11.7)` and the file is `Q1-C.py`.

شرح الكود:

قمنا بإنشاء قاموس `dic` من خلال حلقة `for` حيث `x` المفتاح و `(x-1)` هي قيمة هذا المفتاح .

Q2 :

The screenshot shows the Spyder Python IDE interface. The main editor displays a Python script named 'HOMEWORK.py' with the following code:

```
1 dNumber=[]
2 biNumber=[]
3 z=0
4 while True:
5     x=input("enter number")
6     for i in range(len(x)):
7         if x[i]!='0' or x[i]!='1':
8             a=int(x[i])
9             z=a*2**(len(x)-i-1)
10            biNumber.append(x[i])
11            dNumber.append(z)
12        else:
13            print('invaled input')
14            break
15    print("the number is:",x[:],sum(dNumber[:]))
16    x=input('pees any key to continue or f to finsh')
17    dNumber.clear()
18    if x=='f' or x=='f':
19        break
20 print('finsh')
```

The console output shows the execution of the script:

```
enter number1
the number is: 1 1
pees any key to continue or f to finsh
enter number0
the number is: 0 0
pees any key to continue or f to finsh
enter number4
invaled input
the number is: 4 0
pees any key to continue or f to finsh f
enter number5
invaled input
the number is: 5 0
pees any key to continue or f to finshf
finsh
```

The bottom status bar indicates the current environment is 'conda (Python 3.11.7)' and the file is 'HOMEWORK.py'.

شرح الكود :

* تم تعريف مصفوفتين واحدة للرقم العشري (dNumber) و الأخرى للرقم الثنائي (bNumber)

* متحول سنستفيد منه لاحقا

* حلقة صحيحة لا نهائية

* ندخل القيمة

* حلقة تمر على عناصر القيمة التي أدخلناها

* فحص العنصر $x[i]$ عند كل لفة في حلقة for

* تحويل القيمة إلى intger

* التحويل من ثنائي إلى عشري

* نضيف القيمة بعد التحويل إلى القائمة المعرفة في السطر 1

*من السطر 11 إلى 14 من أجل مراعاة القيمة المدخلة نقوم بطباعة عبارة تفيد بأن القيمة المدخلة غير صحيحة و عندها تخرج من الحلقة for من خلال تعليمة break

*15 إلى 17 نطبع القيمة بالنظام الثنائي ثم العشري

*بعد عملية الطباعة يتم مسح القائمة ذاتها من خلال إدخال المحرف fبحرف صغير أو بحرف كبير F ويطبع بعدها finish .

Q3 :

The screenshot shows the Spyder Python IDE interface. The main editor window displays a Python script for a quiz program. The script imports the json module, loads a JSON file named 'questions.txt', and iterates through a list of questions. For each question, it prompts the user for an answer and checks if it is correct. If correct, it increments the score and prints 'Correct'. If incorrect, it prints 'Wrong'. After all questions are answered, it writes the user's name and score to a file named 'score.txt' and prints the final score.

```
1 import json
2 questions = {
3     scores = 0
4     number=1
5     r = open("questions.txt", 'r')
6     questions = json.load(r)
7     r.close()
8     print("python quiz program")
9     print("Enter t for True or f for False")
10    name = input("Enter your full name: ")
11    for ques in questions.keys():
12        print("Question", number, ": ", ques)
13        ans = input("The answer is ")
14        #testing the result
15        if ans.upper() == questions[ques].upper():
16            scores = scores + 1
17            print("Correct ")
18        else:
19            print ("Wrong")
20            number = number + 1
21
22    #write the name and the score in a separate file
23    result={name:scores}
24    m = open("score.txt", 'w')
25    result = json.dump(result,m)
26    m.close()
```

The console window shows the output of the program. It displays the questions and the user's answers, along with the program's feedback (Correct or Wrong) and the final score.

```
Question 12 : Assembly is an app language.
The answer is
Wrong
Question 13 : MAC is address is 6 byte address.
The answer is t
Correct
Question 14 : IPv4 is a 128-bit address.
The answer is f
Correct
Question 15 : IP is a network Layer protocol.
The answer is f
Wrong
Question 16 : Eigrp is a Routing Protocol.
The answer is f
Wrong
Question 17 : Router can give guest a dynamic or static IP.
The answer is t
Correct
Question 18 : ICMP refers to Internet Control Message Protocol
The answer is
```

شرح الكود :

تم استدعاء مكتبة json

تم تعريف قاموس و هو ما سنعمل عليه

S : متحول لمعرفة النتيجة صحيحة أو خاطئة

Number : متحول يمر على جميع الأسئلة وقيمتها واحد أي السؤال الأول

تم استدعاء التابع open ليفتح الملف question.txt مع الخيار r من أجل القراءة

نحمل الملف من خلال التعليمة questions = json.load(r).close()

و في السطر السابع نقوم باغلاق الملف ثم نذهب لحلقة for ونمر على جميع المفاتيح و هي الأسئلة

في كل لفة نجد سؤال وتكون الإجابة T or F

و نخزن النتيجة في المتحول s .

Q4 :

Spyder (Python 3.11)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\ASUSD\Desktop\برمجة شبيكات\home work\untitled0.py

```
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         print(f"Deposited ${amount}. Current balance: ${self.balance:.2f}")
10
11    def withdraw(self, amount):
12        if self.balance >= amount:
13            self.balance -= amount
14            print(f"Withdrew ${amount}. Current balance: ${self.balance:.2f}")
15        else:
16            print("Insufficient funds!")
17
18    def get_balance(self):
19        return self.balance
20
21    class SavingsAccount(BankAccount):
22        def __init__(self, account_number, account_holder, interest_rate):
23            super().__init__(account_number, account_holder)
24            self.interest_rate = interest_rate
25
26        def apply_interest(self):
27            interest_amount = self.balance * (self.interest_rate / 100)
28            self.balance += interest_amount
29            print(f"Applied interest. Current balance: ${self.balance:.2f}")
30
31        def print(self):
32            print(f"Account holder: {self.account_holder}")
33            print(f"Account number: {self.account_number}")
34            print(f"Balance: ${self.balance:.2f}")
35            print(f"Interest rate: {self.interest_rate}%")
36
37    if __name__ == "__main__":
38        bank_account = BankAccount("554466", "Yousef khalil")
39        bank_account.deposit(15000)
40
```

Usage

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

New to Spyder? Read our [tutorial](#)

Help Variable Explorer Plots Files

Console 1/1 X

```
In [54]: runfile('C:/Users/ASUSD/Desktop/برمجة شبيكات/home work/untitled0.py', wdir='C:/Users/ASUSD/Desktop/برمجة شبيكات/home work')
Deposited $15000. Current balance: $15000.00
Withdrew $4000. Current balance: $11000.00
Deposited $2000. Current balance: $2000.00
Applied interest. Current balance: $2050.00
Account holder: Gafar Zayne
Account number: 778532
Balance: $2050.00
Interest rate: 2.5%

In [55]:
```

Activate Windows
Go to Settings to activate Windows.

Python Console History

conda (Python 3.11.7) Completions: conda LSP: Python Line 32, Col 56 UTF-8 CRLF RW Mem 81%

79°F Windy

Search

4:11 PM 5/25/2024

```
9 print(f"Deposited ${amount}. Current balance: ${self.balance:.2f}")
10
11 def withdraw(self, amount):
12     if self.balance >= amount:
13         self.balance -= amount
14         print(f"Withdrew ${amount}. Current balance: ${self.balance:.2f}")
15     else:
16         print("Insufficient funds!")
17
18 def get_balance(self):
19     return self.balance
20
21 class SavingsAccount(BankAccount):
22     def __init__(self, account_number, account_holder, interest_rate):
23         super().__init__(account_number, account_holder)
24         self.interest_rate = interest_rate
25
26     def apply_interest(self):
27         interest_amount = self.balance * (self.interest_rate / 100)
28         self.balance += interest_amount
29         print(f"Applied interest. Current balance: ${self.balance:.2f}")
30
31     def print(self):
32         print(f"Account holder: {self.account_holder}")
33         print(f"Account number: {self.account_number}")
34         print(f"Balance: ${self.balance:.2f}")
35         print(f"Interest rate: {self.interest_rate}%")
36
37 if __name__ == "__main__":
38
39     bank_account = BankAccount("554466", "Yousef khalil")
40     bank_account.deposit(15000)
41     bank_account.withdraw(4000)
42
43
44     savings_account = SavingsAccount("778532", "Gafar Zayne", 2.5)
45     savings_account.deposit(2000)
46     savings_account.apply_interest()
47     savings_account.print()
```

Usage

Here you can get help of any object by pressing **Ctrl+H** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Console 1/4 X

```
In [54]: runfile('C:/Users/ASUSD/Desktop/برمجة شبيكات/home work/untitled0.py', wdir='C:/Users/ASUSD/Desktop/برمجة شبيكات/home work')
Deposited $15000. Current balance: $15000.00
Withdrew $4000. Current balance: $11000.00
Deposited $2000. Current balance: $2000.00
Applied interest. Current balance: $2050.00
Account holder: Gafar Zayne
Account number: 778532
Balance: $2050.00
Interest rate: 2.5%

In [55]:
```

Activate Windows
Go to Settings to activate Windows.

Python Console History

conda (Python 3.11.7) | Completions: conda | LSP: Python | Line 32, Col 56 | UTF-8 | CRLF | RW | Mem 83%

79°F Windy | Search | ENG | 4:11 PM 5/25/2024

شرح الكود :

account_number: معرف للحساب.

account_holder: اسم صاحب الحساب.

الرصيد: الرصيد الحالي في الحساب (تمت تهيئته إلى 0.0).

ويوفر الطرق التالية:

الإيداع : يضيف المبلغ المحدد إلى رصيد الحساب.

السحب : يطرح المبلغ المحدد من رصيد الحساب في حالة توفر أموال كافية.

get_balance(): إرجاع رصيد الحساب الحالي.

حساب التوفير (يرث من حساب البنك):

يمثل حساب توفير بمعدل فائدة إضافي.

يأخذ كافة السمات والأساليب من BankAccount.

لديه ميزة إضافية وهي الفائدة :

معدل الفائدة السنوي بالنسبة المئوية لحساب التوفير.

و هناك طريقة إضافية:

Apply_interest(): حساب وإضافة الفائدة إلى رصيد الحساب بناءً على سعر الفائدة.

يمكن إنشاء حسابات لكلا الفئتين وتنفيذ بعض العمليات:

يمكن إنشاء حساب BankAccount لـ يوسف خليل بإيداع أولي قدره 15000 دولار.

يمكنك سحب 4000 دولار من هذا الحساب.

بعد ذلك، يمكنك إنشاء مثيل SavingsAccount لـ جعفر زيني بإيداع أولي قدره 2000 دولار وسعر فائدة قدره 2.5%.

تم بتطبيق الفائدة على حساب التوفير الخاص بجعفر

ثم نقوم بطباعة تفاصيل حساب التوفير الخاص بجعفر .
