



الجمهورية العربية السورية

اللاذقية_جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة : وظيفة 1 برمجة شبكات

إشراف الدكتور:

مهند عيسى

اسم الطالب : أدهم صلاح طه

الرقم الجامعي : 2602

Question 1 :

A :

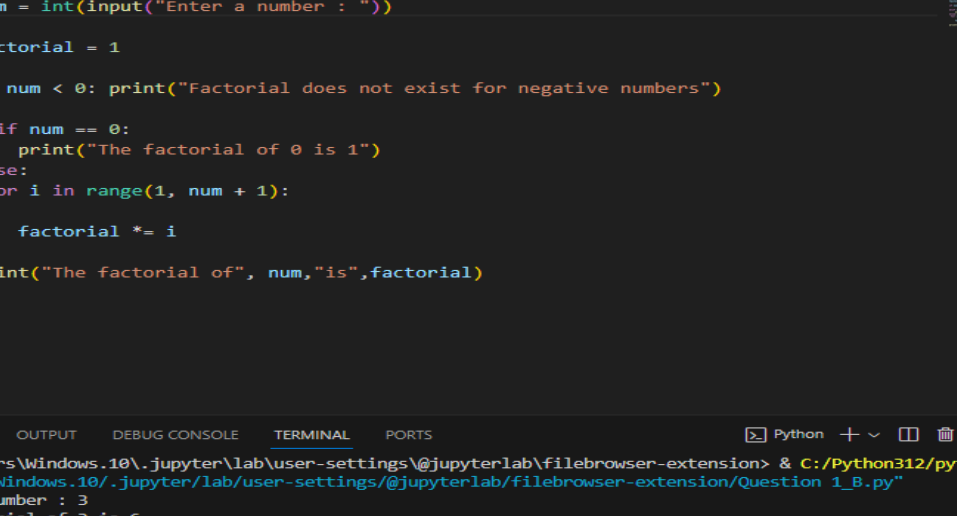
```
Questino 1_A.py X
filebrowser-extension > Questino 1_A.py > ...
1  L1=["HTTP","HTTPS","FTP","DNS"]
2  L2=[80,443,21,53]
3  D={}
4  for i in range(len(L1)):
5      D[L1[i]]=L2[i]
6  print(D)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Windows.10\jupyter\lab\user-settings\@jupyterlab\filebrowser-extension> & C:/Python312/python.exe "c:/Users/Windows.10/jupyter/1
ab/user-settings/@jupyterlab/filebrowser-extension/Questino 1_A.py"
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
PS C:\Users\Windows.10\jupyter\lab\user-settings\@jupyterlab\filebrowser-extension>
```

عرفنا dict فارغة ثم وضعنا حلقة بمجال هو طول ال list الموجودة ثم نضيف قيم L1 مفتاح لل D وقيم L2 قيمة المفتاح ثم قمنا بالطباعة.

Question 1 B :



The screenshot displays a JupyterLab environment with a dark theme. At the top, a file browser shows 'Question 1_B.py' as the active file. The main editor contains a Python script that calculates the factorial of a user-input number. The script includes comments for each step: taking input, initializing factorial, handling negative numbers, handling zero, and calculating the factorial for positive numbers using a loop. The terminal at the bottom shows the command to run the script, the user input '3', and the output 'The factorial of 3 is 6'.

```

1  num = int(input("Enter a number : "))
2
3  factorial = 1
4
5  if num < 0: print("Factorial does not exist for negative numbers")
6
7  elif num == 0:
8      print("The factorial of 0 is 1")
9  else:
10     for i in range(1, num + 1):
11
12         factorial *= i
13
14     print("The factorial of", num,"is",factorial)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension> & C:/Python312/python.exe "c:/Users/Windows.10/.jupyter/lab/user-settings/@jupyterlab/filebrowser-extension/Question 1_B.py"

Enter a number : 3

The factorial of 3 is 6

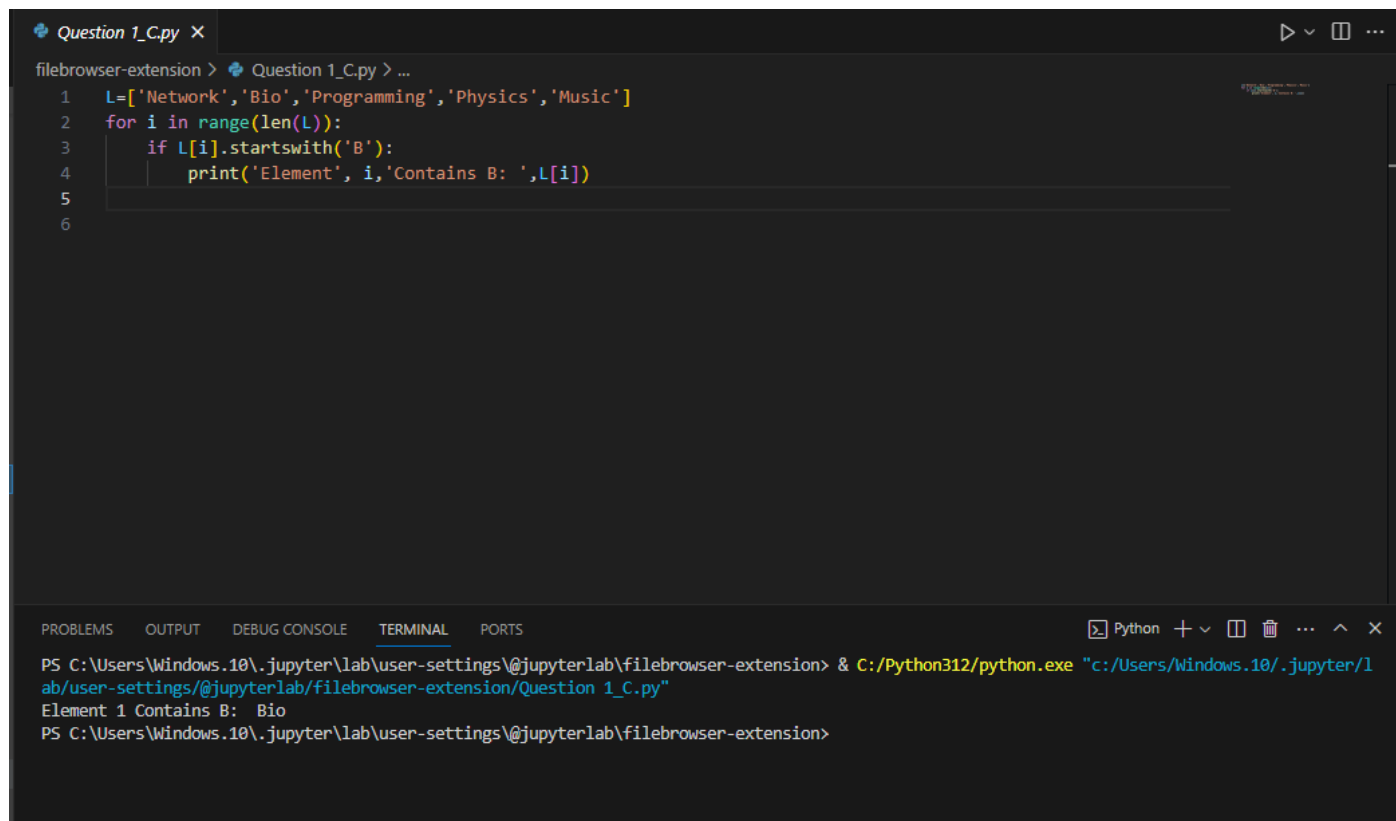
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension> |

:B

يطلب من المستخدم ادخال رقم ويضعه في المتغير num نعرف متغير factorial=1 لتخزين نتيجة الحساب اذا كان العدد سالب يعطي ان factorial لا يقبل ارقام سالبة واذا كان 0 يطبع ان نتيجته 1 والا يدخل في حلقة لها مجال هو بين 1 و num+1 ويزيد قيمة المتغير عن طريق الضرب وفي النهاية يطبع المتغير

Question 1 :

C:

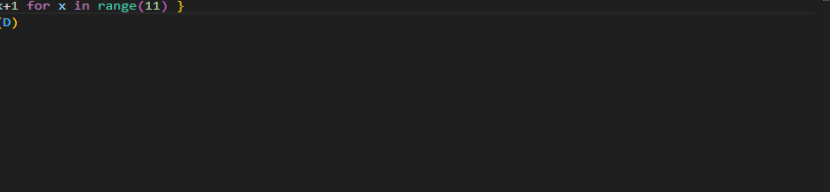


```
Question 1_C.py X
filebrowser-extension > Question 1_C.py > ...
1 L=['Network','Bio','Programming','Physics','Music']
2 for i in range(len(L)):
3     if L[i].startswith('B'):
4         print('Element', i,'Contains B: ',L[i])
5
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + v [icon] [icon] ... ^ X
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension> & C:/Python312/python.exe "c:/Users/Windows.10/.jupyter/lab/user-settings/@jupyterlab/filebrowser-extension/Question 1_C.py"
Element 1 Contains B: Bio
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension>
```

وضعنا حلقة بمجال هو طول ال L وقمنا بالمرور على عناصر L ووجدنا العنصر الذي يبدأ ب B ثم قمنا بطباعة هذا العنصر.

D:



The screenshot displays the JupyterLab environment. At the top, a file browser shows 'Question 1.D.py'. The main editor contains a Python script:

```
1 D=[x:x+1 for x in range(11)]
2 print(D)
3
```

Below the editor, the 'TERMINAL' tab is active, showing the command prompt and the execution of the script:

```
PS C:\Users\Windows.10\jupyter\lab\user-settings\@jupyterlab\filebrowser-extension> C:/Python312/python.exe "c:/Users/Windows.10/.jupyter/lab/user-settings/@jupyterlab/filebrowser-extension/Question 1.D.py"
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
```

The output of the script is a dictionary where keys are integers from 0 to 10 and values are integers from 1 to 11, respectively.

Question 2 :

```
Question 2.py X
filebrowser-extension > Question 2.py > x
1 x=int(input('Enter the number of digits in the binary number : '))
2 sum=0
3 z=0
4 for i in range(x):
5     print("The box",i)
6     a=int(input("Enter it : "))
7     if a==1:
8         sum=sum+2**z
9         z=z+1
10        continue
11    elif a==0:
12        sum=sum+0
13        z=z+1
14        continue
15    else:
16        print("invalid entry ")
17        break
18 if a==1 or a==0 :
19     print(sum)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension> & C:/Python312/python.exe "c:/Users/Windows.10/.jupyter/lab/user-settings/@jupyterlab/filebrowser-extension/Question 2.py"
Enter the number of digits in the binary number : 3
The box 0
Enter it : 1
The box 1
Enter it : 1
The box 2
Enter it : 0
3
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension>
```

أولاً طلبنا من المستخدم ادخال عدد خانات الرقم الثنائي ثم عرفنا متغير sum من اجل وضع الناتج ضمنه ومتغير Z

وضعنا حلقة بمجال عدد خانات الرقم الثنائي ومتغيراً

ثم طلبنا من المستخدم ان يدخل اول خانة

وضعنا شرط اذا كانت 1 يزيد sum بمقدار $2**z$

واذا كانت 0 لا يزيد sum ويزيد z

والا يطبع انه غير موجود

وهكذا بالنسبة لكل الخانات

وأخيرا يطبع الناتج العشري للرقم اذا كانا الادخال 1 او 0

Question 3 :

```
Question 3.py X
filebrowser-extension > Question 3.py > ...
1  import json
2  f=open('C:\\Quiz.json','r')
3  d= json.load(f)
4  f.close()
5
6  score = 0
7  Useranswers = []
8  x=dict(d['Que'])
9  z=list(d['Ans'])
10
11 print("Welcome to the quiz")
12 for i in x.values():
13     print(i)
14     useranswer = input("Your answer:")
15     Useranswers.append(useranswer)
16     for s in z:
17         if useranswer == s:
18             score=score+1
19
20 print("Your score:", score ,"out of",len(x))
21
22 username = input("Enter your name: ")
23 result = {}
24 result["name"]=username
25 result["score"]=score
26 result["total_questions"]=len(x)
27 result["Useranswers"]=Useranswers
28
29 f=open('C:\\Result.json','w')
30 json.dump(result,f)
31 f.close()
32 print("Thank you for taking the quiz")
```

اولا نقوم بتعيين ملف json ونضع فيه الاسئلة ك dict والاجوبة ك list
ثم نستدعي json ونقوم بفتح الملف باستخدام التابع open ونضع المسار والوضعية القراءة ('r') وتخزين المحتوى في
المغير d باستخدام التابع read().
نعرف متغير score الذي سيحوي عدد الاجوبة الصحيحة ونعرف القائمة Useranswers التي سوف تحوي اجوبة
المستخدم.

نضع الاسئلة الموجودة في ملف ال json في متغير X ونضع الاجوبة في متغير Z
ثم وضعنا حلقة تمر على الاسئلة بالترتيب ويدخل المستخدم اجابة كل سؤال ويخزنها في useranswers
ثم وضعنا حلقة داخل الاولى للمرور على عناصر ال z ومقارنتها مع ادخالات المستخدم اذا كانت نفسها يزداد ال score
بمقدار 1

وبعد الخروج من الحلقة الاولى يطبع العلامة.

*يطلب من المستخدم ادخال الاسم

*عرفنا قاموس فارغ result ووضعنا داخله الاسم والعلامة وعدد الاسئلة واجابات المستخدم

نقوم بفتح ملف json جديد الذي سنضع فيه القاموس result عن طريق التابع open ونضع مسار الملف والوضعية كتابة
'w' وتخزين result فيه عن طريق dump

Question 3.py X

filebrowser-extension > Question 3.py > ...

```
1  import json
2  f=open('C:\\Quiz.json','r')
3  d= json.load(f)
4  f.close()
5
6  score = 0
7  Useranswers = []
8  x=dict(d['Que'])
9  z=list(d['Ans'])
10
11 print("Welcome to the quiz")
12 for i in x.values():
13     print(i)
14     useranswer = input("Your answer:")
15     Useranswers.append(useranswer)
16     for s in z:
17         if useranswer == s:
18             score=score+1
19
20 print("Your score:", score ,"out of",len(x))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Your answer:21
19_what is the output 3*8 ?
Your answer:24
20_what is the output 3*9 ?
Your answer:27
21_ what is the output 3*10?
Your answer:30
Your score: 20 out of 21
Enter your name: adham
Thank you for taking the quiz
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension>
```

```
Question 3.py Quiz.json X
C: > {} Quiz.json > {} Que > q21
1 {
2   "Que":
3   [{"q1": "1_Are you a college student? ", "q2": "2_what is the output 2*1? ",
4     "q3": "3_what is the output 2*2 ? ", "q4": "4_what is the output 2*3 ? ",
5     "q5": "5_what is the output 2*4 ? ", "q6": "6_what is the output 2*5 ? ",
6     "q7": "7_what is the output 2*6? ", "q8": "8_what is the output 2*7 ? ",
7     "q9": "9_what is the output 2*8 ? ", "q10": "10_what is the output 2*9 ?",
8     "q11": "11_what is the output 2*10 ? ", "q12": "12_what is the output 3*1 ? ",
9     "q13": "13_what is the output 3*2 ? ", "q14": "14_what is the output 3*3 ? ",
10    "q15": "15_what is the output 3*4 ? ", "q16": "16_what is the output 3*5 ? ",
11    "q17": "17_what is the output 3*6 ? ", "q18": "18_what is the output 3*7 ? ",
12    "q19": "19_what is the output 3*8 ? ", "q20": "20_what is the output 3*9 ? ",
13    "q21": "21_ what is the output 3*10? "}],
14
15
16   "Ans":
17   ["yes", "2",
18     "4", "6",
19     "8", "10",
20     "12", "14",
21     "16", "18",
22     "20", "3",
23     "6", "9",
24     "12", "15",
25     "18", "21",
26     "24", "27",
27     "30"]
28
29 }

Question 3.py Result.json
C: > {} Result.json > [ ] Useranswers
1 [{"name": "adham", "score": 20, "total_questions": 21,
2   "Useranswers": ["yes", "2", "4", "6", "8", "10", "12", "14", "16", "18", "20", "22", "24", "26", "28", "30"]}]
```


Question 4 :

```
Question 4.py x
filebrowser-extension > Question 4.py > SavingsAccount > apply_interest

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         print("Deposited $",amount," into account ",self.account_number," New balance: $",self.balance)
10
11    def withdraw(self, amount):
12        if self.balance >= amount:
13            self.balance -= amount
14            print("Withdrew $",amount," from account ",self.account_number," New balance: $",self.balance)
15        else:
16            print("Insufficient funds in account ",self.account_number," Current balance: $",self.balance)
17
18    def get_balance(self):
19        return self.balance
20
21    def __str__(self):
22        return( "Account Number:" + str(self.account_number) + "Account Holder:" + str(self.account_holder) +
23
24
25 class SavingsAccount(BankAccount):
26     def __init__(self, account_number, account_holder, balance=0.0, interest_rate=0.01):
27         super().__init__(account_number, account_holder, balance)
28         self.interest_rate = interest_rate
29
30     def apply_interest(self):
31         interest_amount = self.balance * self.interest_rate
32         self.balance += interest_amount
33         print("Interest of $",interest_amount," applied to account", self.account_number,". New balance: $",s
34
35     def __str__(self):
36         return (super().__str__() + ", Interest Rate:" + str(self.interest_rate))
37
38 bank_account = BankAccount("26022602", "adham taha")
39 bank_account.deposit(1000.0)
40 bank_account.withdraw(500.0)
41 print(bank_account)
42 savings_account = SavingsAccount("12123434", "Jane Smith", 1000.0, 0.05)
43 savings_account.apply_interest()
44 print(savings_account)
```

```
Question 4.py X
filebrowser-extension > Question 4.py > SavingsAccount > apply_interest
1 class BankAccount:
20
21     def __str__(self):
22         return( "Account Number:" + str(self.account_number) + "Account Holder:" + str(self.account_holder) +
23
24 class SavingsAccount(BankAccount):
25     def __init__(self, account_number, account_holder, balance=0.0, interest_rate=0.01):
26         super().__init__(account_number, account_holder, balance)
27         self.interest_rate = interest_rate
28
29     def apply_interest(self):
30         interest_amount = self.balance * self.interest_rate
31         self.balance += interest_amount
32         print("Interest of $",interest_amount," applied to account", self.account_number,". New balance: $",s
33
34     def __str__(self):
35         return (super().__str__() + ", Interest Rate:" + str(self.interest_rate))
36
37 bank_account = BankAccount("26022602", "adham taha")
38 bank_account.deposit(1000.0)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + v [ ] [ ] ... ^ X
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension> & C:/Python312/python.exe "c:/Users/Windows.10/.jupyter/1
ab/user-settings/@jupyterlab/filebrowser-extension/Question 4.py"
Deposited $ 1000.0 into account 26022602 New balance: $ 1000.0
Withdrew $ 500.0 from account 26022602 New balance: $ 500.0
Account Number:26022602Account Holder:adham taha Balance: $500.0
Interest of $ 50.0 applied to account 12123434 . New balance: $ 1050.0
Account Number:12123434Account Holder:Jane Smith Balance: $1050.0, Interest Rate:0.05
PS C:\Users\Windows.10\.jupyter\lab\user-settings\@jupyterlab\filebrowser-extension>
```

1.تعريف فئة BankAccount :

: __init__(self, account_number, account_holder, balance=0.0)

هذه الدالة تقوم بإنشاء حساب بنكي جديد مع رقم الحساب صاحب الحساب، وكذلك الرصيد الافتتاحي الذي يكون 0.0 بشكل افتراضي

: deposit(self, amount)

هذه الدالة تقوم بإضافة المبلغ المحدد إلى رصيد الحساب، وطباعة رسالة تظهر المبلغ المودع والرصيد الجديد

: withdraw(self, amount)

هذه الدالة تقوم بخصم المبلغ المحدد من رصيد الحساب، شريطة أن يكون الرصيد كاف. إذا كان الرصيد غير كاف فسيتم طباعة رسالة تشير إلى عدم كفاية الأموال

get_balance(self) : هذه الدالة تقوم بإرجاع الرصيد الحالي للحساب

`_str_(self)` : هذه الدالة تقوم بإرجاع تمثيل سلسلة نصية للحساب، بما في ذلك رقم الحساب، صاحب الحساب والرصيد

2. تعريف فئة `SavingsAccount` :

`__init__(self, account_number, account_holder, balance=0.0, interest_rate=0.01)` :

هذه الدالة تقوم بإنشاء حساب توفير جديد مع رقم الحساب، صاحب الحساب الرصيد الافتتاحي، ومعدل الفائدة الذي يكون 0.01 بشكل افتراضي .

`apply_interest(self)` : هذه الدالة تقوم بحساب الفائدة المكتسبة على رصيد الحساب بناءً على معدل الفائدة وإضافتها إلى الرصيد، ثم طباعة رسالة تظهر الرصيد الجديد

`_str_(self)` : هذه الدالة تقوم من الفئة `_str_` بتجاوز طريقة لتضمين معدل الفائدة في التمثيل السلسلي للحساب الأصلي

3. إنشاء الكائنات وتنفيذ العمليات :

يتم إنشاء كائن من فئة `BankAccount` باستخدام:

```
bank_account = BankAccount("26022602", "adham taha")
```

يتم إيداع مبلغ 1000.0 دولار في الحساب باستخدام `bank_account.deposit(100 0.0)`

يتم سحب مبلغ 500.0 دولار من الحساب باستخدام `bank_account.withdraw(50 0.0)`

يتم طباعة تمثيل سلسلة نصية للحساب باستخدام `print(bank_account)`

يتم إنشاء كائن من فئة :

`SavingsAccount` باستخدام:

```
savings_account = SavingsAccount("12123434 ", "adham taha", 1000.0, 0.05)
```

يتم تطبيق الفائدة على الحساب باستخدام `savings_account.apply_in terest()`

يتم طباعة تمثيل سلسلة نصية للحساب باستخدام `print(savings_account)`