

الجمهورية العربية السورية

Syrian Arab Republic

اللاذقية-جامعة تشرين

Lattakia - Tishreen University

كلية الهندسة الكهربائية والميكانيكية قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة برمجة شبكات

Name: _____ الياس هيثم ملحم _____, Number: _____ 2086 _____,

إعادة عملي

Submitted To GitHub _____ : <https://github.com/eliasmelhem1/assignment-1> _____

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 }

```
1.A.py ×
1.A.py > ...
1 L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
2 L2 = [80, 443, 21, 53]
3
4 d = {}
5 for i in range(len(L1)):
6     d[L1[i]] = L2[i]
7
8 print(d)
```

هذا الكود يقوم بإنشاء قاموس (dictionary) يربط بين بروتوكولات الشبكة وأرقام المنافذ الخاصة بها. نقوم بإنشاء قاموس فارغ اسمه d حيث سنضع فيه الأزواج (البروتوكول، المنفذ). نستخدم حلقة for للتكرار عبر جميع عناصر القائمة L1 في كل تكرار:

- i يمثل الفهرس الحالي في الحلقة.
- L1[i] يعطينا البروتوكول الحالي.
- L2[i] يعطينا رقم المنفذ المقابل للبروتوكول الحالي.
- نقوم بربط البروتوكول برقم المنفذ في القاموس d

الخرج:

```
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}  
PS D:\Elias Melhem>
```

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

```
1.B.py > factorial  
1 def factorial(n):  
2     result = 1  
3     for i in range(1, n + 1):  
4         result *= i  
5     return result  
6  
7 # Get input from the user  
8 number = int(input("Enter a number: "))  
9  
10 # Calculate the factorial  
11 result = factorial(number)  
12  
13 # Print the result  
14 print(f"The factorial of {number} is {result}")
```

تعريف دالة: factorial

- الدالة تأخذ عددًا n
- تبدأ بـ result = 1
- تضرب result في كل عدد من 1 إلى n
- ترجع قيمة result التي تمثل العامل (factorial) للعدد n

الخرج:

```
Enter a number: 12  
The factorial of 12 is 479001600  
PS D:\Elias Melhem>
```

```
Enter a number: 5  
The factorial of 5 is 120  
PS D:\Elias Melhem>
```

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.

```
1.C.py > ...
1  L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
2
3  # Iterate through the list using a loop
4  for i in range(len(L)):
5      # Check if the item starts with 'B'
6      if L[i].startswith('B'):
7          # Print the item
8          print(L[i])
```

تحتوي على مجموعة من الكلمات. ['Network', 'Bio', 'Programming', 'Physics', 'Music'] :

نستخدم حلقة for للتكرار عبر كل عنصر في القائمة L باستخدام الفهارس داخل الحلقة، نتحقق إذا كانت الكلمة تبدأ بالحرف 'B' باستخدام الدالة startswith.

إذا كانت الكلمة تبدأ بالحرف 'B' ، نقوم بطباعتها.

الخرج:

```
Bio
PS D:\Elias Melhem>
```

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}

```
1.D.py > [?] d
1  d = {i: i + 1 for i in range(11)}
2  print(d)
```

يستخدم تعبير توليد القواميس (dictionary comprehension) لإنشاء قاموس. range(11) ينشئ تسلسلاً من الأعداد من 0 إلى 10.

لكل قيمة i في التسلسل، يتم إنشاء زوج (مفتاح: قيمة) حيث المفتاح هو i والقيمة هي i + 1.

الخرج:

```
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
PS D:\Elias Melhem>
```

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.

Tips: solve input errors.

```
2.py > ...
1 while(True):
2     def binary_to_decimal(binary):
3         decimal = 0
4         power = len(binary) - 1
5
6         for digit in binary:
7             if digit == '1':
8                 decimal += 2 ** power
9             elif digit != '0':
10                return None # Return None for invalid input
11            power -= 1
12
13        return decimal
14
15    # Get binary number input from the user
16    binary_number = input("Enter a binary number: ")
17
18    # Check if the input contains only 0s and 1s
19    if all(bit in '01' for bit in binary_number):
20        # Convert binary to decimal
21        decimal_number = binary_to_decimal(binary_number)
22
23        if decimal_number is not None:
24            # Print the equivalent decimal number
25            print("Equivalent decimal number:", decimal_number)
26            break
27        else:
28            print("Invalid binary number!")
29    else:
30        print("Invalid input! Please enter a binary number containing only 0s and 1s.")
31
```

حلقات لا نهائية تستمر الحلقة إلى أن يتم كسرها باستخدام break.

دالة تحويل من ثنائي إلى عشري binary_to_decimal :

- تأخذ سلسلة نصية تمثل عددًا ثنائيًا.
- تحسب قيمته العشرية عن طريق جمع القوى المناسبة للعدد 2.
- يطلب من المستخدم إدخال عدد ثنائي.

إذا كان المدخل صحيحًا، يتم تحويله إلى عدد عشري باستخدام الدالة binary_to_decimal. إذا كانت نتيجة التحويل ليست None، يتم طباعة العدد العشري وكسر الحلقة إذا كان المدخل غير صحيح، أو التحويل فشل، يتم إعلام المستخدم وإعادة المحاولة

```
Enter a binary number: -154
Invalid input! Please enter a binary number containing only 0s and 1s.
Enter a binary number: 110001
Equivalent decimal number: 49
PS D:\Elias Melhem>
```

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.

الكود:

```

3.py x {} quiz_questions.json
3.py > main
1 import json
2
3 def load_questions(file_name):
4     with open(file_name, 'r') as file:
5         data = json.load(file)
6         return data['questions']
7
8 def save_result(user_name, score):
9     result = {"user_name": user_name, "score": score}
10    with open("quiz_results.json", 'a') as file:
11        json.dump(result, file)
12        file.write('\n')
13
14    def take_quiz(questions):
15        score = 0
16        total_questions = len(questions)
17        for i, question_data in enumerate(questions, start=1):
18            print(f"Question {i}: {question_data['question']}")
19            user_answer = input("Your answer: ").strip()
20            if user_answer.lower() == question_data['answer'].lower():
21                print("Correct!")
22                score += 1
23            else:
24                print("Incorrect!")
25        return score
26
27    def main():
28        user_name = input("Enter your name: ")
29        questions = load_questions("quiz_questions.json")
30        score = take_quiz(questions)
31        print(f"\n{user_name}, your score is: {score}/{len(questions)}")
32        save_result(user_name, score)
33
34    if __name__ == "__main__":
35        main()

```

ملف الأسئلة:

```
3.py  {} quiz_questions.json X
{} quiz_questions.json > ...
1  {
2      "questions": [
3          {
4              "question": "Q1",
5              "answer": "A1"
6          },
7          {
8              "question": "Q2",
9              "answer": "A2"
10         },
11         {
12             "question": "Q3",
13             "answer": "A3"
14         },
15         {
16             "question": "Q4",
17             "answer": "A4"
18         },
19         {
20             "question": "Q5",
21             "answer": "A5"
22         },
23         {
24             "question": "Q6",
25             "answer": "A6"
26         },
27         {
28             "question": "Q7",
29             "answer": "A7"
30         },
31         {
32             "question": "Q8",
33             "answer": "A8"
34         },
35         {
36             "question": "Q9",
37             "answer": "A9"
```

استيراد مكتبة json:

load_questions تقرأ ملف JSON يحتوي على أسئلة الاختبار و ترجع قائمة بالأسئلة.

save_result تحفظ اسم المستخدم ونقاطه في ملف JSON و تفتح الملف (append) لإضافة النتائج دون مسح البيانات القديمة.

take_quiz تعرض الأسئلة على المستخدم وتتحقق من صحة الإجابات وتحسب النقاط بناءً على الإجابات الصحيحة.

Main تأخذ اسم المستخدم. تحمل الأسئلة من ملف JSON تدير عملية الاختبار وتحسب النقاط وتعرض النتيجة وتحفظها.

الكود يقوم بتحميل الأسئلة من ملف JSON ، يعرضها على المستخدم ليجيب عنها، يحسب النقاط، ويعرض النتيجة النهائية. ثم يحفظ النتيجة في ملف JSON.

الخرج:

```
Enter your name: elias
Question 1: Q1
Your answer: a1
Correct!
Question 2: Q2
Your answer: a2
Correct!
Question 3: Q3
Your answer: a
Incorrect!
```

```
Incorrect!
Question 20: Q20
Your answer:
Incorrect!

elias, your score is: 4/20
PS D:\Elias Melhem>
```

الملف حيث يتم حفظ النتيجة:

```
3.py  {} quiz_results.json X  {} quiz_ques
{} quiz_results.json > ...
1  [{"user_name": "elias", "score": 4}]
2
```

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.

3.py

4.py

X

4.py > BankAccount > withdraw

```
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 amount <= self.balance:
12            self.balance -= amount
13        else:
14            print("Insufficient funds!")
15
16    def get_balance(self):
17        return self.balance
18
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
24     def apply_interest(self):
25         interest_amount = self.balance * (self.interest_rate / 100)
26         self.balance += interest_amount
27
28     def print(self):
29         print(f"Current balance: ${self.balance}, Interest Rate: {self.interest_rate}%")
30
31
32 # Create an instance of BankAccount
33 bank_account = BankAccount("123456789", "John Doe")
34
35 # Perform a deposit of $1000
36 bank_account.deposit(1000)
37 print("After deposit of $1000, current balance:", bank_account.get_balance())
```

```
38
39 # Perform a withdrawal of $500
40 bank_account.withdraw(500)
41 print("After withdrawal of $500, current balance:", bank_account.get_balance())
42
43 # Create an instance of SavingsAccount
44 savings_account = SavingsAccount("987654321", "Jane Doe", 5) # 5% interest rate
45
46 # Perform a deposit of $2000
47 savings_account.deposit(2000)
48 print("\nAfter deposit of $2000, current balance and interest rate:")
49 savings_account.print()
50
51 # Apply interest
52 savings_account.apply_interest()
53 print("\nAfter applying interest, current balance and interest rate:")
54 savings_account.print()
```

1. تعريف الفئة: BankAccount

__init__ يهيئ الحساب برقم حساب، اسم صاحب الحساب، ورصيد يبدأ من 0.

Deposit تضيف مبلغًا إلى الرصيد.

Withdraw تسحب مبلغًا من الرصيد إذا كان كافيًا، وإلا تطبع رسالة.

get_balance تعيد الرصيد الحالي.

2. تعريف SavingsAccount ترث من BankAccount

__init__ يهيئ الحساب بنفس الخصائص السابقة بالإضافة إلى معدل الفائدة.

apply_interest تطبق الفائدة على الرصيد وتضيفها.

Print تطبع الرصيد الحالي ومعدل الفائدة.

3. استخدام BankAccount

- إنشاء حساب جديد.
- إجراء إيداع بقيمة 1000 دولار.
- طباعة الرصيد بعد الإيداع.
- إجراء سحب بقيمة 500 دولار.
- طباعة الرصيد بعد السحب.

4. استخدام SavingsAccount

- إنشاء حساب توفير جديد بنسبة فائدة 5%.
- إجراء إيداع بقيمة 2000 دولار.
- طباعة الرصيد ومعدل الفائدة بعد الإيداع.
- تطبيق الفائدة.
- طباعة الرصيد ومعدل الفائدة بعد تطبيق الفائدة.

الخرج:

```
After deposit of $1000, current balance: 1000.0
After withdrawal of $500, current balance: 500.0

After deposit of $2000, current balance and interest rate:
Current balance: $2000.0, Interest Rate: 5%

After applying interest, current balance and interest rate:
Current balance: $2100.0, Interest Rate: 5%
PS D:\Elias Melhem>
```