

Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and  
electrical engineering

5<sup>a</sup>, Network Programming : Homework  
No1



الجمهورية العربية السورية

اللاذقية جامعة تشرين

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والإلكترونيات

السنة الخامسة: وظيفة ١ برمجة شبكات

Name: Nour Badour , Number:2632 , Submitted To GitHub:@ NourBadour

## First Network Programming Homework

### Question 1: Python Basics?

A-if you have two lists, L1=['HTTP','HTTPS','FTP','DNS'] L2=[80,443,20,53], convert it to generate this dictionary d={'HTTP':80,'HTTPS':443,'FTP':20,'DNS':53}

```
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
L2 = [80, 443, 21, 53]

d = {}
for i in range(len(L1)):
    d[L1[i]] = L2[i]

print(d)
```

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

السؤال الأول الطلب الأول:

إنشاء قاموس من اثنين من القوائم. القائمة الأولى تحتوي على البروتوكولات، بينما القائمة الثانية تحتوي على الأرقام المنافذ لهذه البروتوكولات. يُستخدم مؤشر للوصول إلى نفس الموقع في القوائم الاثنتين أثناء إنشاء أزواج المفتاح والقيمة في القاموس. في النهاية، يُطبع القاموس الذي تم إنشاؤه.

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

```
def factorial_iterative(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
    return result  
  
number = int(input("Enter a number: "))  
  
print(f"The factorial of {number} is {factorial_iterative(number)}")
```

```
Enter a number: 10  
The factorial of 10 is 3628800
```

السؤال الأول الطلب الثاني:

يتم تهيئة متغير بقيمة الواحد، ثم يتم استخدام حلقة تكرارية لحساب العامل. في كل دورة من الحلقة، يتم ضرب الناتج الحالي بالعدد الذي يأتي بعده. تبدأ الحلقة من 1 وتنتهي عند العدد الذي يدخله المستخدم. في النهاية، يُطبع الناتج النهائي الذي تم حسابه.

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.

```
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']  
items_starting_with_B = [item for item in L if item.startswith('B')]  
for item in items_starting_with_B:  
    print(item)
```

```
Bio
```

السؤال الأول طلب ثالث:

يتم استخدام قائمة التكرار (List comprehension) لبناء قائمة جديدة تحتوي على العناصر التي تبدأ بالحرف 'B'. ثم يتم طباعة كل عنصر في القائمة الجديدة، والتي تمثل العناصر التي تبدأ بالحرف 'B'.

D: Using Dictionary comprehension, Generate this dictionary

d={1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11}

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

```
>>> {0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
```

السؤال الأول الطلب الرابع:

يتم تكرار الأعداد من 0 إلى 10 باستخدام range(10)، ولكل عدد يتم إنشاء مفتاح وقيمة في القاموس. القيمة تكون مجرد العدد زائد واحد. في النهاية، يتم طباعة القاموس الناتج، الذي يحتوي على أزواج المفاتيح والقيمة.

### 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.

```
def binary_to_decimal(binary):  
    decimal = 0  
    power = len(binary) - 1  
    for bit in binary:  
        if bit == '1':  
            decimal += 2 ** power  
        elif bit != '0':  
            return None # Invalid binary digit  
        power -= 1  
    return decimal  
  
def main():  
    binary = input("Enter a binary number: ")  
    decimal = binary_to_decimal(binary)  
    if decimal is not None:  
        print(f"The decimal equivalent of {binary} is {decimal}.")  
    else:  
        print("Conversion failed. Please check the input.")  
  
if __name__ == "__main__":  
    main()
```

```
>>> Enter a binary number: 1111  
The decimal equivalent of 1111 is 15.
```

## السؤال الثاني:

تحويل الأعداد الثنائية إلى العشرية يتم من خلال استخدام دالة تقوم بالتحويل التدريجي لكل رقم في الرقم الثنائي المُدخل. يتم ضرب كل رقم بالقوة المناسبة لمكانه في الرقم الثنائي ليتم تحويله إلى عدد عشري. في حالة وجود أي رقم غير صحيح في الرقم الثنائي، ستقوم الدالة بإرجاع قيمة "None" للتحذير من وجود خطأ. بعد ذلك، يُطلب من المستخدم إدخال الرقم الثنائي، ثم يتم استدعاء الدالة لتنفيذ التحويل. إذا كان الرقم الثنائي صالحًا، يتم طباعة النتيجة، وإلا يتم طباعة رسالة تحذيرية تطلب من المستخدم التحقق من صحة الرقم الثنائي المُدخل.

### 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.

```
1 .....
2 .....
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
19 .....
20 .....
21 .....
22 .....
23 .....
24 .....
25 .....
```

```

import json
def load_questions_from_file(file_path):
    with open(file_path, 'r') as file:
        data = json.load(file)
    return data['questions']
def evaluate_quiz(questions):
    score = 0
    for question_info in questions:
        question_text = question_info['question']
        correct_answer = question_info['answer']
        user_response = input(f"{question_text} ")
        if user_response.strip().lower() == correct_answer.strip().lower():
            score += 1
    return score
def prompt_username():
    return input("Enter your name: ")
def record_user_score(username, score, file_path):
    with open(file_path, 'a') as file:
        file.write(f"{username},{score}\n")
def execute_quiz_program():
    quiz_file = 'Q.json'
    results_file = 'results.csv'
    questions = load_questions_from_file(quiz_file)
    score = evaluate_quiz(questions)
    username = prompt_username()
    record_user_score(username, score, results_file)
if __name__ == "__main__":
    execute_quiz_program()

```

HTTP stands for HyperText Transfer Protocol. true  
 FTP stands for File Transfer Protocol. TRUE  
 The default port for HTTPS is 443. TRUE  
 The OSI model has 6 layers. TRUE  
 IPv6 addresses are 128 bits long. TRUE  
 The TCP protocol is connectionless. TRUE  
 A router operates at the Network layer of the OSI model. TRUE  
 SMTP is used to send emails. TRUE  
 The default port for HTTP is 80. TRUE  
 DNS translates domain names to IP addresses. TRUE  
 A MAC address is a unique identifier assigned to network interfaces. TRUE  
 The maximum length of a CAT6 Ethernet cable is 100 meters. TRUE  
 SSL stands for Secure Socket Layer. TRUE  
 A switch operates at the Data Link layer of the OSI model. TRUE  
 UDP is more reliable than TCP. TRUE  
 The loopback IP address is 127.0.0.1. TRUE  
 The default subnet mask for a Class B network is 255.255.0.0. TRUE  
 Wi-Fi stands for Wireless Fidelity. TRUE  
 The ICMP protocol is used for error reporting and diagnostics. TRUE  
 The port number for SSH is 22. TRUE  
 Enter your name: nour

	A	B	C	D
1	nour,17			
2				

### السؤال الثالث:

في هذا الكود، يتم استخدام مكتبة JSON لفتح وقراءة محتوى ملف يحتوي على أسئلة. بعد ذلك، تستخدم البيانات المحملة لاسترداد الأسئلة. يتم تقييم أداء المستخدم في الاختبار بالاستناد إلى الأسئلة والإجابات الصحيحة، حيث يتم طرح الأسئلة واستلام إجابات المستخدم ومقارنتها بالإجابات الصحيحة لحساب النقاط. بعد ذلك، يُطلب من المستخدم إدخال اسمه لتسجيل النتيجة. وأخيراً، تُسجل نتيجة المستخدم في ملف النتائج، مع اسم المستخدم وعدد النقاط التي حصل عليها.

#### 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.

```
class BankAccount:
    def __init__(self, account_number, account_holder, balance=0.0):
        self.account_number = account_number
        self.account_holder = account_holder
        self.balance = balance

    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited {amount}. Current balance: {self.balance}")
        else:
            print("Deposit amount must be positive.")

    def withdraw(self, amount):
        if 0 < amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew {amount}. Current balance: {self.balance}")
        else:
            print("Insufficient funds or invalid withdrawal amount.")

    def get_balance(self):
        return self.balance

    def __str__(self):
        return f"Account Number: {self.account_number}, Account Holder: {self.account_holder}, Balance: {self.balance}"

# Create an instance of BankAccount
account = BankAccount("123456789", "batool")
account.deposit(1000)
account.withdraw(500)
```

```

class SavingsAccount(BankAccount):
    def __init__(self, account_number, account_holder, interest_rate, balance=0.0):
        super().__init__(account_number, account_holder, balance)
        self.interest_rate = interest_rate

    def apply_interest(self):
        if self.interest_rate > 0:
            interest = self.balance * self.interest_rate / 100
            self.balance += interest
            print(f"Applied interest: {interest}. New balance: {self.balance}")
        else:
            print("Interest rate must be positive.")

    def __str__(self):
        return (f"Account Number: {self.account_number}, Account Holder: {self.account_holder}, "
                f"Balance: {self.balance}, Interest Rate: {self.interest_rate}%")

# Create an instance of SavingsAccount
savings_account = SavingsAccount("987654321", "aya", 5)
savings_account.deposit(1000)
savings_account.apply_interest()
print(savings_account)

```

```

Deposited 1000. Current balance: 1000.0
Withdrew 500. Current balance: 500.0
Deposited 1000. Current balance: 1000.0
Applied interest: 50.0. New balance: 1050.0
Account Number: 987654321, Account Holder: aya, Balance: 1050.0, Interest Rate:
5%
>>>

```

#### السؤال الرابع:

الصف `BankAccount` تمثل حساباً بنكياً عادياً، حيث تحتوي على خصائص مثل رقم الحساب واسم صاحب الحساب والرصيد الحالي. توفر هذه الفئة مجموعة من الوظائف مثل `deposit` و `withdraw` التي تمكن المستخدم من إيداع الأموال في الحساب وسحبها على التوالي، بالإضافة إلى وظيفة `get_balance` التي تُستخدم لاسترجاع الرصيد الحالي للحساب. وتُعرف دالة `__str__` لتمثيل الحساب بشكل نصي.

صنف `SavingsAccount` يمثل حساب بنكي للادخار، وهي تشق من صنف `BankAccount`، مما يعني أنها ترث جميع الخصائص والوظائف من الفئة الأساسية. ولكنها تحتوي أيضاً على خاصية إضافية وهي معدل الفائدة. توفر هذه الفئة وظيفة إضافية تُسمى `apply_interest` التي تحسب وتطبق الفائدة على الرصيد الحالي للحساب، بناءً على معدل الفائدة المحدد. وتُعرف دالة `__str__` لإعادة تمثيل الحساب بشكل مخصص.

في النهاية، يتم إنشاء حسابين، واحد من كل فئة، وتُستخدم الوظائف المتاحة في كل منها، مما يمثل العمليات الأساسية لإدارة الحسابات المصرفية وتطبيق الفوائد.