جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الاتصالات

إعداد الطالبة : ساره عبد الحليم نجار

الرقم الجامعي :2974

بإشراف الدكتور:

مهند عيسى

Question 1 (A)

```
L1=['HTTP','HTTPS','FTP','DNS']
L2=[80,443,21,53]
d=dict(zip(L1,L2))
print(d)
```

Output:

```
Run    Question 1 (A)
D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\Question 1 (A).py"
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}

Process finished with exit code 0
C: > Users > moham > Desktop > برمجة شبكات > lec4 codes > Question 1 (A).py
```

This code snippet will output the dictionary d as follows:
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
The zip() function combines the elements of L1 and L2 into pairs and then the dict() function converts these pairs into a dictionary.

Question 1 (B)

```
def factorial (n):
    if n ==0:
        return 1
    else:
        return n * factorial(n-1)
num = int(input("Enter a number to calculate its
factorial :"))
if num<0:
    print("Factorial is not defined for negative
numbers.")
elif num ==0:
    print("The factorial of 0 is 1")
else:
    result=factorial(num)
    print (f"The factorial of {num} is {result}")
```

Output:

```
Run    Question 1 (B)
D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\Question 1 (B).py"
Enter a number to calculate its factorial :65
The factorial of 65 is 8247650592082470666723170306785496252186258551345437492922123134388955774976000000000000000

Process finished with exit code 0
```

In this program: 1. We define a recursive function factorial(n) that calculates the factorial of a given number n.

2. We take input from the user for the number whose factorial needs to be calculated.

3. We check if the input number is negative, in which case we inform the user that factorial is not defined for negative numbers.

4. If the input number is non-negative, we calculate its factorial using the factorial() function and display the result.

Question 1 (C)

```
# Define the list
L = ['Network', 'Bio', 'Programing', 'Physics',
'Music']
# Loop through each item in the list
for item in L:
    # Check if the length of the items is greater than
0 and if it starts with the letter "B"
    if len(item) > 0 and item.startswith('B'):
        print(item)
```

Output:

```
Run    Question 1 (C)
D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\Question 1 (C).py"
Bio

Process finished with exit code 0
```

In this program: 1. We define the list L containing the items.

2. We loop through each item in the list using a for loop.

3. For each item, we use the startswith() method to check if it starts with the letter 'B'.

4. If an item starts with 'B', we print that item on the screen.

Question 1 (D)

```
#Generate the dictionary using dictionary comprehension
d={i: i+1 for i in range (11)}
#print the generated dictionary
print(d)
```

Output:

```
Run    Question 1 (D)
D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\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}

Process finished with exit code 0
```

In this program: 1. We define the list L containing the items.

2. We loop through each item in the list using a for loop.

3. For each item, we check if the length of the item is greater than 0 and if it starts with the letter 'B'.

4. If both conditions are met, we print that item on the screen.

Question 2

```python
while True:
    try:
        #Read binary number from the user
        binary=input("Enter a binary number : ")
        #Convert binary to decimal using int() function
        decimal=int(binary , 2)
        #Display the equivalent decimal number
        print("Decimal equivalent :",decimal)
        break #Exit the loop if inputs is valid
    except ValueError :
        print("Invalid input. \n  "
        "please enter a valid binary number .")
```

Output:

```
Run    Question 2
D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\Question 2.py"
Enter a binary number : 1111011001
Decimal equivalent : 985

Process finished with exit code 0
```

```
Run    Question 2
D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\Question 2.py"
Enter a binary number : 15265
Invalid input.
  please enter a valid binary number .
Enter a binary number : 110001
Decimal equivalent : 49

Process finished with exit code 0
```

In this program:

- We use a while loop to repeatedly prompt the user for input until a valid binary number is entered. - Inside the loop, we use a try

-except block to catch any ValueError that may occur during the conversion.
- The input() function is used to read the binary number from the user.

- The int() function is used to convert the binary number to its decimal equivalent. The second argument 2 specifies that the input is in base 2 (binary).

- If the conversion is successful, we display the equivalent decimal number using the print() function.

- If an invalid input is entered, a ValueError is raised and caught by the except block. An error message is displayed, and the loop continues to prompt for a valid input

Question 3

```python
import json


def load_questions_from_json(file_path):
    with open(file_path, 'r') as file:
        questions = json.load(file)
    return questions


def save_results_to_json(user_name, score):
    results = {user_name: score}
    with open('results.json', 'w') as file:
        json.dump(results, file)


def quiz(questions):
    score = 0
    for question, answer in questions.items():
        user_answer = input(question + ' ')
        if user_answer.lower() == answer.lower():
            score += 1
    return score


def main():
    file_path = "./questions.json"
    questions = load_questions_from_json(file_path)

    user_name = input("Enter your name: ")

    user_score = quiz(questions)

    print(f"Your score: {user_score}/{len(questions)}")

    save_results_to_json(user_name, user_score)


if __name__ == "__main__":
    main()
```

Output:

```
Run    Question 3
D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\Question 3 .py"
Enter your name: Sarah
Who are you? sese
what is your father name? abd alhallem
what is your sister name ? leen
what is your favourite colour? pink
what is your hoppy? shopping
How are you? fine
what do you do for living? I'm a communications engineer
what is your bestie name? Hanan
what is the weather like ? its sunny
Are you happy? yes
what are you doing? doing my network homework
what are you study? Communications engineer
Where do you live? in lattakia
what is your favourite city? Lattakia
How old are you? I'm 22
Where are you from? Syria , lattakia
what is your favourite animal? Dog
do you have a pet? No
What is the boiling point of water in Celsius? 100
Your score: 1/19

Process finished with exit code 0
C: > Users > moham > Desktop > برمجة شبكات > lec4 codes > Question 3 .py
```

In this program: 1. The load_questions_from_json function reads questions and answers from a JSON file.

2. The save_results_to_json function stores the user's name and result in a JSON file.

3. The quiz function prompts the user with the questions and calculates the user's score.

Question 4

```
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):
        self.balance += amount
        return self.balance

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            return self.balance
        else:
            return "Insufficient funds"

    def get_balance(self):
        return self.balance


class SavingsAccount(BankAccount):
    def __init__(self, account_number, account_holder,
balance=0.0, interest_rate=0.02):
        super().__init__(account_number,
account_holder, balance)
        self.interest_rate = interest_rate

    def apply_interest(self):
        interest_amount = self.balance *
self.interest_rate
        self.balance += interest_amount
        return interest_amount

    def print_info(self):
        print(f"Current Balance: {self.balance}")
        print(f"Interest Rate: {self.interest_rate}")
```

```python
# Create an instance of BankAccount
bank_acc = BankAccount("123456789", "John Doe")
print("Bank Account Operations:")
print("Initial Balance:", bank_acc.get_balance())
print("After depositing $1000:",
bank_acc.deposit(1000))
print("After withdrawing $500:",
bank_acc.withdraw(500))
print()

# Create an instance of SavingsAccount
savings_acc = SavingsAccount("987654321", "Jane Smith")
print("Savings Account Operations:")
print("Initial Balance:", savings_acc.get_balance())
savings_acc.deposit(2000)
print("After applying interest:",
savings_acc.apply_interest())
savings_acc.print_info()
```

Output:

```
Run    Question 4

D:\bb\pythonProject1\.venv\Scripts\python.exe "C:\Users\moham\Desktop\برمجة شبكات\lec4 codes\Question 4 .py"
Bank Account Operations:
Initial Balance: 0.0
After depositing $1000: 1000.0
After withdrawing $500: 500.0

Savings Account Operations:
Initial Balance: 0.0
After applying interest: 40.0
Current Balance: 2040.0
Interest Rate: 0.02

Process finished with exit code 0
```

In this code:

- The BankAccount class defines the basic attributes and methods for a bank account.

- The SavingsAccount subclass inherits from BankAccount and adds an interest_rate attribute and an apply_interest method to apply interest to the balance.

- We create an instance of BankAccount and perform deposit and withdrawal operations, printing the balance after each operation.

- We also create an instance of SavingsAccount, deposit some money, apply interest, and print the current balance and interest rate.