

Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and
electrical engineering

5th , Network Programming : Homework
No1



الجمهورية العربية السورية

اللاذقية - جامعة تشرين

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة I برمجة شبكات

رشا علان علي 2729

السؤال الأول

A

```
code1.py - C:\Users\LCT\Desktop\New folder (2)\code1.py (3.12.3)
File Edit Format Run Options Window Help
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
L2 = [80, 443, 21, 53]
d = dict(zip(L1, L2))
print(d) # Output: {'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
```

النتيجة :

```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\LCT\Desktop\New folder (2)\code1.py
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
>>>
```

B

```
code2.py - C:/Users/LCT/Desktop/New folder (2)/code2.py (3.12.3)
File Edit Format Run Options Window Help
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

number = int(input("Enter a number: "))
print(f"The factorial of {number} is {factorial(number)}")
```

النتيجة :

```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/LCT/Desktop/New folder (2)/code2.py
Enter a number: 3
The factorial of 3 is 6
>>>
```

C

```
code3.py - C:/Users/LCT/Desktop/New folder (2)/code3.py (3.12.3)
File Edit Format Run Options Window Help
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']

for item in L:
    if item.startswith('B'):
        print(item) # Output: Bio
```

النتيجة :

```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/LCT/Desktop/New folder (2)/code3.py =====
>>> Bio
>>> |
```

D

```
code4.py - C:/Users/LCT/Desktop/New folder (2)/code4.py (3.12.3)
File Edit Format Run Options Window Help
d = {i: i for i in range(12)}
print(d) # Output: {0: 0, 1: 1, 2: 2, ..., 11: 11}
```

: النتيجة

```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/LCT/Desktop/New folder (2)/code4.py
{0: 0, 1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 6, 7: 7, 8: 8, 9: 9, 10: 10, 11: 11}
>>> |
```

السؤال الثاني :

File Edit Format Run Options Window Help

```
def is_valid_binary(binary_str):  
    for char in binary_str:  
        if char not in ('0', '1'):  
            return False  
    return True  
def binary_to_decimal(binary_str):  
    # تحقق من أن الإدخال يتكون فقط من 0 و 1  
    if not all(char in '01' for char in binary_str):  
        return None # إذا كان الإدخال غير صالح، نعيد None  
    # تحويل السلسلة الثنائية إلى عدد عشري  
    decimal_number = int(binary_str, 2)  
    return decimal_number  
def main():  
    while True:  
        binary_str = input("أدخل رقماً ثنائياً: ").strip()  
        result = binary_to_decimal(binary_str)  
        if result is not None:  
            print(f"{result} هو {binary_str} العدد العشري المكافئ للعدد الثنائي").  
            break  
        else:  
            print("رقماً ثنائياً صالحاً يتكون فقط من 0 و 1. يرجى المحاولة مرة أخرى")  
if __name__ == "__main__":  
    main()
```

طلبنا من المستخدم ادخال عدد ثنائي و الا يظهر رسالة خطأ ويعيد طلب ادخال العدد الثنائي

النتيجة :

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/LCT/Desktop/New folder (2)/code5.py
أدخل رقماً ثنائياً: 10011
العدد العشري المكافئ للعدد الثنائي 10011 هو 19
>>>
===== RESTART: C:/Users/LCT/Downloads/Telegram Desktop/121212121.py =====
jj: أدخل رقماً ثنائياً
خطأ: أدخل رقماً ثنائياً صالحاً يتكون فقط من 0 و 1. يرجى المحاولة مرة أخرى
65: أدخل رقماً ثنائياً
خطأ: أدخل رقماً ثنائياً صالحاً يتكون فقط من 0 و 1. يرجى المحاولة مرة أخرى
1100110: أدخل رقماً ثنائياً
العدد العشري المكافئ للعدد الثنائي 1100110 هو 102
>>>
```

السؤال الثالث :

تم إنشاء ملف نصي يحوي 20 سؤال وجواب كل منها على سطر فيتم قراءة السطر الذي يحوي السؤال وإدخال الإجابة من الطالب ومن ثم مقارنة الجواب المدخل مع الجواب الصحيح في حال كانت الاجابة صحيحة تزداد علامة الطالب وفي النهاية يتم تخزين اسم الطالب وعلامته في ملف جديد

```
q3.py - C:/Users/LCT/Desktop/q3.py (3.12.3)
File Edit Format Run Options Window Help
name=input("enter your name")
ff=open('c://qw.txt','r')
deg=0
for x in range (20):
    s=ff.readline()
    print(s)
    an=input()
    an=an+"\n"
    s=ff.readline()
    if (an==s):
        deg+=1
out=open("c:\\an.txt",'w')
out.write(name)
out.write(str(deg))
ff.close()
out.close()
```

```
File Edit Shell Debug Options Window Help
8
c
9
c
10
c
11
cc
12
c
13
c
14
c
15
c
16
c
17
c
18
c
19
c
20
```

ملف النتيجة:

ملف الأسئلة:

```
an - Notepad
File Edit Format View Help
rasha10
```

```
qw - Notepad
File Edit Format View Help
1
a
2
a
3
a
4
b
5
b
6
b
7
b
8
c
9
c
10
c
11
c
12
c
13
c
14
c
```

السؤال الرابع:

إنشاء صنف لحساب بنك مع إمكانية إضافة وسحب مبلغ مالي منه وإنشاء صنف حساب آخر يأخذ قيمة فائدة على المبلغ الموجود في الحساب

q4.py - C:/Users/LCT/Desktop/q4.py (3.12.3)

File Edit Format Run Options Window Help

```
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
    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
        else:
            print("Insufficient funds")
    def get_balance(self):
        return self.balance
    def prent(self):
        print(f"Account Number: {self.account_number}")
        print(f"Account Holder: {self.account_holder}")
        print(f"Balance: {self.balance}")

class SavingsAccount(BankAccount):
    def __init__(self, account_number, account_holder, balance=0.0, rate=0.0):
        super().__init__(account_number, account_holder, balance)
        self.rate = rate
    def apply_interest(self):
        if self.rate > 0:
            self.balance += self.balance * self.rate
        else:
            print("Interest rate must be positive")
    def prent(self):
        print(f"Interest Rate: {self.rate}")
    def get_balance(self):
        return self.balance

# Create instances and perform operations
account = BankAccount("88888", "rasha")
account.deposit(1000)
account.withdraw(500)
print(account.get_balance())
account.prent()

savings = SavingsAccount("9999", "rasha", 0.0, 0.2)
savings.deposit(1000)
savings.apply_interest()
print(savings.get_balance())
savings.prent()
```

```
500.0
Account Number: 88888
Account Holder: rasha
Balance: 500.0
1200.0
Interest Rate: 0.2
```