Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No1

الجمهورية العربية السورية

اللاذقية -جامعــة تشريـــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة1 برمجة شبكات

زهراء رضوان حميدان ٢٩١٩

Question 1:

A:

```
In [1]: L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
        L2 = [80, 443, 21, 53]
        d = {L1[i]: L2[i] for i in range(len(L1))}
        print(d)
```

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

B:

```
In [5]: def f(a):
            b = 1
            for i in range(1, a + 1):
                b *= i
            return b

        n = int(input("Enter a number to calcuate The factorial: "))
        if n>=0:
            print(f(n))
        else:
            print("Negative number")
```

```
Enter a number to calcuate The factorial: 3
6
```

C:

```
In [6]: L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
        for item in L:
            if item.startswith('B'):
                print(item)
```

```
Bio
```

D:

In [7]:
```
d = {i: i + 1 for i 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}
```

Question 2:

```
def func(binary):
    l=[]
    dec=0
    for i in binary:
        l.append(int(i))
    l.reverse()
    for i in range(len (l)):
        dec+=l[i]*2**i
    return dec

while True:
    b=input("Enter binary and za to exit: ")
    if b=='za':
        print("zahraa")
        break
    if b.isalnum():
        if '1' not in b or "0" not in b:
            print("error input")
            continue
        else:
            print(func(b))
    else:
        print("error input")
```

```
Enter binary and za to exit: 1001
9
Enter binary and za to exit: 10001
17
Enter binary and za to exit: 222
error input
Enter binary and za to exit: za
zahraa
```

Question 3:

```
questions = []
try:
    with open('q.txt', 'r') as file:
        lines = file.readlines()
        for line in lines:
            question, answer = line.strip().split('=')
            questions.append((question.strip(), int(answer.strip())))
except Exception as e:
    print(f"Error reading file: {e}")

# الحصول على اسم المستخدم
username = input("Enter your name: ")

# حساب النتيجة
score = 0
for question, correct_answer in questions:
    try:
        user_answer = int(input(f"{question} = "))
        if user_answer == correct_answer:
            score += 1
    except ValueError:
        print("Invalid input. Please enter a valid number.")

# عرض النتيجة للمستخدم
print(f"{username}, your score is: {score}/20")

# حفظ النتيجة في ملف CSV
try:
    with open('النتيجة.csv', 'a') as file:
        file.write(f"{username},{score}\n")
except Exception as e:
    print(f"Error writing to file: {e}")
```

```
Enter your name: zahraa
2 + 2 = 4
8 - 3 = 5
7 * 5 = 35
50 / 5 = 10
11 + 8 = 19
20 - 10 = 10
10 * 2 = 20
16 / 4 = 1
9 + 3 = 1
21 - 11 = 1
4 * 6 = z
Invalid input. Please enter a valid number.
36 / 6 = 1
13 + 7 = 2
28 - 14 = 3
5 * 8 = 3
32 / 8 = 4
6 + 9 = 5
24 - 8 = 6
2 * 10 = 7
45 / 9 = 7
zahraa, your score is: 8/20
```

ملف النتيجة:

	A	B	C	D
1	zahraa	8		
2				
3				
4				

شرح:

نقوم بقراءة الأسئلة والإجابات من ملف نصي باسم "q.txt" باستخدام تابع open() مع استخدام with لضمان إغلاق الملف تلقائياً. نحن نقوم بقراءة جميع الاسطر من الملف باستخدام تابع readlines()، وبعد ذلك نقوم بفصل كل سطر إلى سؤال وإجابة صحيحة ونخزنهما في المتغير questions. نقوم بتكرار الأسئلة والاجوبة من القائمة questions وطلب إجابة المستخدم لكل سؤال. إذا كانت إجابة المستخدم صحيحة، نضيف درجة واحدة إلى النتيجة. إذا أدخل المستخدم إجابة غير صالحة، فسنطبع رسالة خطأ. نعرض النتيجة النهائية للمستخدم باستخدام تابع print(). ثم نحاول كتابة النتيجة في ملف CSV باسم "النتيجة.csv"

Question 4:

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

    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

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

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

    def __str__(self):
        return f"Current balance: ${self.balance:.2f}, Interest rate: {self.interest_rate * 100:.2f}%"

# إنشاء حساب بنك
account = BankAccount("2919", "zahraa")
account.deposit(1000)
print(f"Balance after deposit: ${account.get_balance():.2f}")
account.withdraw(500)
print(f"Balance after withdrawal: ${account.get_balance():.2f}")

# إنشاء حساب توفير
savings = SavingsAccount("2919", "zahraa", 0.05)
savings.deposit(1000)
savings.apply_interest()
print(savings)
```

```
Balance after deposit: $1000.00
Balance after withdrawal: $500.00
Current balance: $1050.00, Interest rate: 5.00%
```

نعرف كلاستين: BankAccount و SavingsAccount. BankAccount هي الكلاس الأب والتي تمثل حساب بنكي عادي، وتحتوي على خصائص مثل رقم الحساب والاسم وكذلك طرق للإيداع والسحب والحصول على الرصيد.

SavingsAccount هي الكلاس الابن المشتق من BankAccount والذي يمثل حساب توفير. هذا الكلاس له خاصية إضافية وهي معدل الفائدة، وطريقة apply_interest() التي تضيف الفائدة المستحقة إلى الرصيد.