Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No1

الجمهورية العربية السورية

اللاذقية -جامعـــة تشريـــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: وظيفة1 برمجة شبكات

رنام نضال المهلوبي ٢٩٢٧

Question 1:

A:

```
In [1]: L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
        L2 = [80, 443, 21, 53]
        d = dict(zip(L1, L2))
        print(d)

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

B:

```
In [2]: def f(x):
            result = 1
            for i in range(1, x + 1):
                result *= i
            return result

        number = int(input("Enter a number: "))
        print(f(number))

        Enter a number: 5
        120
```

C:

```
In [3]: L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
        for item in L:
            if item.startswith('B'):
                print(item)

        Bio
```

D:

```
n [4]: 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:

```python
def b(x):
    try:
        d = int(x, 2)
        return d
    except:
        return "wrong binary number"

a = input("Enter binary number ")
result = b(a)
print(result)
```

```
Enter binary number 1001
9
```

Question 3:

يبدأ البرنامج بطلب اسم المستخدم باستخدام تابع input(). ثم يحاول البرنامج قراءة محتوى ملف quat.txt ويخزن المحتوى في قائمة quiz_data. كل عنصر في القائمة هو زوج من السؤال والإجابة الصحيحة.

إذا كان هناك خطأ في قراءة الملف، يطبع البرنامج رسالة خطأ. بعد ذلك يبدأ البرنامج في طرح الأسئلة على المستخدم واحتساب النتيجة النهائية. النتيجة النهائية هي عدد الإجابات الصحيحة للمستخدم.

في الخرج، نرى أن المستخدم أدخل اسمه "رئام" وأجاب على 16 سؤالاً بشكل صحيح من أصل 20 سؤال. ثم يكتب البرنامج النتيجة النهائية في ملف resault.csv.

```python
user_name = input("أدخل اسمك: ")

quiz_data = []
try:
    quiz_file = open('quat.txt', 'r')
    content_lines = quiz_file.readlines()
    for content_line in content_lines:
        quiz_question, quiz_answer = content_line.strip().split('=')
        quiz_data.append((quiz_question.strip(), int(quiz_answer.strip())
except:
    print("خطأ في قراءة الملف")


# حساب النتيجة
total_score = 0
for quiz_question, correct_response in quiz_data:
    user_response = int(input(f"{quiz_question} = "))
    if user_response == correct_response:
        total_score += 1

final_message = str(user_name)+", "+str(total_score)
print(final_message)

result_file = open('resault.csv', 'w')
result_file.write(final_message)
result_file.close()
```

```
أدخل اسمك: رئام
9 = 3 * 3
9 = 3 / 27
8 = 4 + 4
2 = 1+1
10 = 10 - 20
20 = 2 * 10
4 = 4 / 16
12 = 3 + 9
9 = 3 * 3
9 = 3 / 27
8 = 4 + 4
2 = 1+1
10 = 10 - 20
20 = 2 * 10
0 = 4 / 16
12 = 3 + 9
3 = 3 * 3
9 = 3 / 27
5 = 4 + 4
5 = 1+1
رئام, 16
```

الملف الذي سيتم فيه كتابة النتائج:

	A	B	C
1	رئام	16	
2			

Question 4:

تم إنشاء كائن من نوع BankAccount باستخدام رقم الحساب "2927" واسم صاحب الحساب "رئام". تم إجراء عملية إيداع بمبلغ 1000 دولار، وطباعة الرصيد الحالي. بعد ذلك، تم إجراء عملية سحب بمبلغ 500 دولار، وطباعة الرصيد الحالي.

SavingsAccount هو 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

تم إنشاء كائن من نوع SavingsAccount باستخدام رقم الحساب "787878"، اسم صاحب الحساب "رئام المهلوبي"، ومعدل الفائدة 0.25. تم تطبيق الفائدة على الرصيد باستخدام طريقة apply_interest()، وطباعة تفاصيل الحساب باستخدام طريقة __str__().

```python
# Class BankAccount
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 self.balance >= amount:
            self.balance -= amount
        else:
            print("Insufficient funds.")

    def get_balance(self):
        return self.balance

# Create an instance of BankAccount
bank_account = BankAccount("2927", "رئام")

# Perform a deposit of $1000
bank_account.deposit(1000)
print(f"Balance after deposit: ${bank_account.get_balance():.2f}")

# Perform a withdrawal of $500
bank_account.withdraw(500)
print(f"Balance after withdrawal: ${bank_account.get_balance():.2f}")
```

```python
# Subclass SavingsAccount
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"Account Number: {self.account_number}\nAccount Holder: {self.account

# Create an instance of SavingsAccount
savings_account = SavingsAccount("787878", "0.25 ,"رئام الملوبي)

# Apply interest
savings_account.apply_interest()
print(savings_account)
```

```
Balance after deposit: $1000.00
Balance after withdrawal: $500.00
Account Number: 787878
Account Holder: رئام الملوبي
Balance: $0.00
Interest Rate: 25.00%
```