



Name: زينب سامر منى , Number 2739

First Network Programming Homework

Question 1: Python Basics?

A:

```
In [7]: 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}
```

B:

```
In [1]: def func(x):
        if x == 0:
            return 1
        else:
            return x * func(x-1)
        while True:
            try:
                number = int(input("Enter a number ## Enter -1 to stop: "))
                if number == -1:
                    break
                if number < 0:
                    print("Error Input")
                    continue
                result = func(number)
                print(f"The factorial of {number} is {result}")
            except:
                print("Error Input")
```

```
print("Error Input")
```

```
Enter a number ## Enter -1 to stop: 3
The factorial of 3 is 6
Enter a number ## Enter -1 to stop: 11
The factorial of 11 is 39916800
Enter a number ## Enter -1 to stop: 10
The factorial of 10 is 3628800
Enter a number ## Enter -1 to stop: 9
The factorial of 9 is 362880
Enter a number ## Enter -1 to stop: 8
The factorial of 8 is 40320
Enter a number ## Enter -1 to stop: 7
The factorial of 7 is 5040
Enter a number ## Enter -1 to stop: 6
The factorial of 6 is 720
Enter a number ## Enter -1 to stop: 5
The factorial of 5 is 120
Enter a number ## Enter -1 to stop: 4
The factorial of 4 is 24
Enter a number ## Enter -1 to stop: 3
The factorial of 3 is 6
Enter a number ## Enter -1 to stop: 2
The factorial of 2 is 2
Enter a number ## Enter -1 to stop: 1
The factorial of 1 is 1
Enter a number ## Enter -1 to stop: 0
The factorial of 0 is 1
Enter a number ## Enter -1 to stop: aa
Error Input
Enter a number ## Enter -1 to stop: -5
Error Input
Enter a number ## Enter -1 to stop: -1
```

C:

```
In [3]: L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']

for item in L:
    if item.startswith('B'):
        print(item)

Bio
```

D:

```
1 [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: Convert from Binary to Decimal

```
In [2]: def binary_to_decimal(binary_str):
        try:
            decimal = int(binary_str, 2)
            return decimal
        except ValueError:
            print(ValueError)
            return "Invalid binary number"
    while True:
        binary_number = input("Enter a binary number #### s to stop: ")
        if binary_number=="s":
            print("Thanks")
            break
        decimal_number = binary_to_decimal(binary_number)
        print(f"The decimal equivalent of {binary_number} is {decimal_number}")
```

```
Enter a binary number #### s to stop: 111
The decimal equivalent of 111 is 7
Enter a binary number #### s to stop: 2
<class 'ValueError'>
The decimal equivalent of 2 is Invalid binary number
Enter a binary number #### s to stop: 00
The decimal equivalent of 00 is 0
Enter a binary number #### s to stop: 0100
The decimal equivalent of 0100 is 4
Enter a binary number #### s to stop: 10001
The decimal equivalent of 10001 is 17
Enter a binary number #### s to stop: s
Thanks
```

الشرح:

تعريف التابع `binary_to_decimal`: يبدأ التابع بكلمة `def` متبوعة باسم التابع `binary_to_decimal`، يأخذ بارامتر واحد `binary_str`، الذي يجب أن يكون سلسلة من الأرقام الثنائية (0 و 1).

تحويل العدد:

`try`: يتم محاولة تنفيذ الكود داخل هذا البلوك.

`decimal = int(binary_str, 2)`: يتم تحويل السلسلة الثنائية إلى عدد عشري باستخدام التابع `int` مع تحديد القاعدة 2 (الثنائية).

`return decimal`: إذا نجحت عملية التحويل، يتم إرجاع العدد العشري.

`except ValueError`: إذا حدث خطأ أثناء التحويل (مثلاً إذا كانت السلسلة تحتوي على أحرف غير صالحة)، يتم التقاط هذا الخطأ.

`print(ValueError)`: يتم طباعة رسالة الخطأ.

`return "Invalid binary number"`: يتم إرجاع رسالة "Invalid binary number" للإشارة إلى أن البرامتر غير صالح.

الحلقة while True

الحلقة اللانهائية: يبدأ الكود بحلقة while True، وهي حلقة لانهاية تستمر في التكرار حتى يتم كسرها يدوياً.

إدخال عدد ثنائي من المستخدم. إذا أدخل المستخدم الحرف "s"، سيتم خروج من الحلقة. يتم طلب

كسر الحلقة باستخدام break. إذا كان المدخل هو الحرف "s"، تتم طباعة "Thanks" ويتم

decimal_number = binary_to_decimal(binary_number): يتم استدعاء التابع binary_to_decimal لتحويل العدد الثنائي إلى عشري.

print(f"The decimal equivalent of {binary_number} is {decimal_number}") : يتم طباعة النتيجة على الشاشة.

Question 3: Working with Files” Quiz Program”

```
In [4]: import json

def read_questions(file_path):
    questions = []
    with open(file_path, 'r') as f:
        lines = f.readlines()
        for i in range(0, len(lines), 2):
            question = lines[i].strip()
            answer = lines[i+1].strip()
            questions.append({"question": question, "answer": answer})
    return questions

def quiz(questions):
    score = 0
    for q in questions:
        user_answer = input(q["question"] + " ")
        if user_answer.lower() == q["answer"].lower():
            score += 1
    return score

def save_results(file_path, user_name, score):
    try:
        with open(file_path, 'r') as f:
            results = json.load(f)
    except FileNotFoundError:
        results = []
    results.append({"name": user_name, "score": score})
```

```

with open(file_path, 'w') as f:
    json.dump(results, f, indent=4)

def main():
    questions_file = 'questions.txt'
    results_file = 'results.json'

    questions = read_questions(questions_file)
    score = quiz(questions)

    user_name = input("Enter your name: ")
    print(f"{user_name}, your score is: {score}/{len(questions)}")

    save_results(results_file, user_name, score)

main()

```

```

What is the capital of France? paris
What is 2 + 2? 4
What is the largest planet in our solar system? jupiter
What is the square root of 64? 8
What is 7 * 6? 42
What is the capital of France? paris
What is 2 + 2? 4
What is the largest planet in our solar system? s
What is the square root of 64? s
What is 7 * 6? s
What is the capital of France? s
What is 2 + 2? s
What is the largest planet in our solar system? jupiter
What is the square root of 64? 8
What is 7 * 6? 42
Enter your name: ZAINAB
ZAINAB, your score is: 10/15

```

شرح البرنامج

نبدأ بتعريف تابع `read_questions` الذي يقرأ الأسئلة والأجوبة من ملف نصي. يقوم هذا التابع بفتح الملف وقراءة الأسطر فيه، ثم يقوم بتجميع كل سؤال مع إجابته في قائمة من المتغيرات نوع `dictionary`، حيث يحتوي كل `dictionary` على مفتاحين: `"question"` و `"answer"`.

بعد ذلك نعرف تابع `quiz` الذي يتلقى قائمة الأسئلة ويطرحها على المستخدم. يقوم هذا التابع بحساب النتيجة عن طريق مقارنة إجابة المستخدم مع الإجابة الصحيحة، وذلك بتحويل كلا الإجابتين إلى أحرف صغيرة للتأكد من التطابق بغض النظر عن حالة الأحرف.

تابع save_results يستخدم لحفظ نتائج المستخدم في ملف JSON. يحاول التابع أولاً قراءة الملف إذا كان موجوداً وإذا لم يكن موجود يقوم بإنشاء قائمة جديدة. ثم يضيف التابع نتيجة المستخدم إلى القائمة ويكتبها مرة أخرى في الملف بصيغة JSON.

تابع main هو بداية تنفيذ البرنامج ويقوم بتحديد أسماء الملفات المستخدمة للأسئلة والنتائج، ويستدعي توابع قراءة الأسئلة وتنفيذ الاختبار وحفظ النتائج.

بعد انتهاء الاختبار: يطلب التابع من المستخدم إدخال اسمه ويطبع النتيجة النهائية قبل حفظها في ملف JSON.

Question 4: Object-Oriented Programming - Bank Class

```
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
        print(f"Deposited {amount}. Current balance: {self.balance}")

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            print(f"Withdrawn {amount}. Current balance: {self.balance}")
        else:
            print("Insufficient funds.")

    def get_balance(self):
        return self.balance

class SavingsAccount(BankAccount):
    def __init__(self, account_number, account_holder, balance=0.0, interest_rate=0.01):
        super().__init__(account_number, account_holder, balance)
        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}, Interest rate: {self.interest_rate}"
```

```
# إنشاء حساب بنكي
Zainab=BankAccount("2379","Zainab muna")
Zainab.deposit(1000)
Zainab.withdraw(500)

Z=SavingsAccount("2379","Zainab muna",interest_rate=0.12)
Z.deposit(700)
Z.apply_interest()

print(Z)
```

Output:

```
Deposited 1000. Current balance: 1000.0
Withdrawn 500. Current balance: 500.0
Deposited 700. Current balance: 700.0
Current balance: 784.0, Interest rate: 0.12
```

شرح البرنامج

الكلاس BankAccount يمثل حساب بنكي عادي، ويبدأ بتعريف تابع الإنشاء __init__ الذي يُستخدم لإنشاء كائن جديد من الكلاس. هذا التابع يأخذ رقم الحساب واسم صاحب الحساب ورصيد ابتدائي (افتراضياً صفر)، ويتم تخزين هذه القيم كخصائص للكائن. يحتوي الكلاس أيضاً على ثلاثة توابع أخرى: تابع deposit يُستخدم لإيداع مبلغ معين في الحساب بزيادة الرصيد وطباعة رسالة توضح المبلغ المودع والرصيد الحالي، وتابع withdraw يُستخدم لسحب مبلغ معين من الحساب بعد التحقق من أن المبلغ المسحوب لا يتجاوز الرصيد الحالي، ويطبع رسالة توضح المبلغ المسحوب والرصيد الحالي، وإذا كان المبلغ المطلوب أكبر من الرصيد، يطبع رسالة تفيد بعدم كفاية الرصيد. أما تابع get_balance فيُستخدم لإرجاع الرصيد الحالي للحساب. الكلاس SavingsAccount يرث من الكلاس BankAccount ويضيف ميزات جديدة تتعلق بالحسابات الادخارية. يبدأ تابع الإنشاء __init__ في الكلاس SavingsAccount باستدعاء تابع الإنشاء للكلاس الأب باستخدام super()، ثم يضيف خاصية جديدة وهي معدل الفائدة الافتراضي (0.01). يحتوي الكلاس على تابعين إضافيين هما: تابع apply_interest الذي يُستخدم لتطبيق الفائدة على الرصيد الحالي عن طريق زيادة الرصيد بمقدار الفائدة المحسوبة بناءً على معدل الفائدة، وتابع __str__ الذي يُستخدم لتحويل الكائن إلى سلسلة نصية، ويُرجع سلسلة نصية تحتوي على الرصيد الحالي ومعدل الفائدة. في نهاية البرنامج، يتم إنشاء كائن من الكلاس BankAccount باسم "Zainab" برقم حساب "2379" هو رقمي الجامعي، ويتم تنفيذ عمليات إيداع وسحب عليه. بعد ذلك، يتم إنشاء كائن آخر من الكلاس SavingsAccount بنفس رقم الحساب والاسم ومعدل فائدة 0.12، ويتم تنفيذ عمليات إيداع وتطبيق الفائدة عليه. أخيراً، نطبع الكائن SavingsAccount مما يستدعي التابع __str__ ويعرض الرصيد الحالي ومعدل الفائدة.