



ابراهيم وائل يوسف 1874

Question 1:

A:

```
In [1]: 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 [3]: def func1(n):
        if n == 0:
            return 1
        else:
            return n * func1(n - 1)

        number = int(input("input a number: "))
        print(func1(number))

input a number: 3
6
```

C:

```
In [4]: L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
        for item in L:
            if item.startswith('B'):
                print(item)
```

Bio

D:

```
In [5]: 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:

```
l=[]  
while True:  
    b=input('enter binary: ')  
    if b.isdigit():  
        d=0  
        for i in b:  
            l.append(int(i))  
        l.reverse()  
        for i in range(len(l)):  
            d+=l[i]*2**i  
        print(d)  
    else:  
        print("input binary number")  
    s=input('enter s to stop: ')  
    if s!='s':  
        l=[]  
    else:  
        break
```

```
enter binary: 1010  
10  
enter s to stop: 1  
enter binary: 10  
2  
enter s to stop: i  
enter binary: i  
input binary number  
enter s to stop: s
```

Question 3:

قمنا بقراءة الأسئلة والأجوبة من ملف نصي باسم "qa.txt". نقوم بذلك باستخدام تابع open () لفتح الملف وتابع readlines () لقراءة جميع الخطوط من الملف. ثم نقوم بتحليل كل خط ليفصل السؤال عن الإجابة الصحيحة ونخزن هذه المعلومات في قائمة تدعى questions.

بعد ذلك نطلب من المستخدم إدخال اسمه باستخدام تابع input ().

ثم نبدأ في طرح الأسئلة على المستخدم. نقوم بذلك بتكرار قائمة questions وطلب إجابة المستخدم لكل سؤال. إذا كانت إجابة المستخدم صحيحة، نزيد درجة المستخدم بواحد.

في النهاية نعرض النتيجة النهائية للمستخدم في رسالة تتكون من اسم المستخدم والنتيجة. ثم نقوم بكتابة هذه المعلومات في ملف "r.csv" باستخدام تابع open () و write ().

المستخدم أدخل اسمه "ابراهيم يوسف" وأجاب على 15 سؤالاً بشكل صحيح من أصل 20 سؤال.

```
# قراءة الأسئلة والأجوبة من الملف النصي
questions = []
try:
    file=open('qa.txt', 'r')
    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"خطأ في قراءة الملف: {e}")

# الحصول على اسم المستخدم
username = input("أدخل اسمك: ")

# حساب النتيجة
score = 0
for question, correct_answer in questions:
    user_answer = int(input(f"{question} = "))
    if user_answer == correct_answer:
        score += 1

# عرض النتيجة النهائية للمستخدم
a=f"{username}، نتيجتك هي {score}"
print(a)
outfile=open('r.csv', 'w')
outfile.write(a)
outfile.close()
```

أدخل اسمك: ابراهيم يوسف

$$10 = 7 + 3$$

$$11 = 7 - 18$$

$$32 = 8 * 4$$

$$8 = 8 / 64$$

$$17 = 5 + 12$$

$$15 = 20 - 35$$

$$18 = 9 * 2$$

$$9 = 3 / 27$$

$$20 = 15 + 5$$

$$20 = 25 - 40$$

$$18 = 6 * 3$$

$$9 = 6 / 54$$

$$21 = 6 + 14$$

$$12 = 19 - 33$$

$$28 = 4 * 7$$

$$9 = 8 / 72$$

$$20 = 11 + 9$$

$$11 = 7 - 22$$

$$18 = 8 * 3$$

$$8 = 6 / 48$$

ابراهيم يوسف، نتيجتك هي: 15

A	B
ابراهيم يوسف، نتيجتك هي: 17	

Question 4:

```

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

# إنشاء حساب بنك
account = BankAccount("1874", "ibraheem")
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("12345", "Ahmed", 0.05)
savings.deposit(1000)
savings.apply_interest()
savings.print()

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

```

BankAccount هو الكلاس الأب والذي يمثل حساب بنكي. هذا الكلاس له خصائص مثل رقم الحساب واسم صاحب الحساب وكذلك طرق للإيداع والسحب والحصول على الرصيد الحالي.

SavingsAccount الكلاس الابن من BankAccount، والذي يمثل حساب توفير. هذا الكلاس له خاصية إضافية وهي معدل الفائدة، وطريقة apply_interest() التي تضيف الفائدة المستحقة إلى الرصيد. تم استخدام طريقة __init().__super__() لاستدعاء المُنشئ الخاص بكلاس الأب BankAccount.