Syrian Arab Republic

Lattakia -Tishreen University

Department Of Communication AndElectrical Engineering

الجمهورية العربية السورية

اللاذقية- جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الاتصالات والالكترونيات

5th ,Net Work Programming : Homework No 1

السنة الخامسة: الوظيفة ١ برمجة شبكات

Name: سوزان عماد الدين شكوحي **, Number:** 2228 **, Submitted To GetHub :**

First Network Programming Homework

Question 1: Python Basics ?

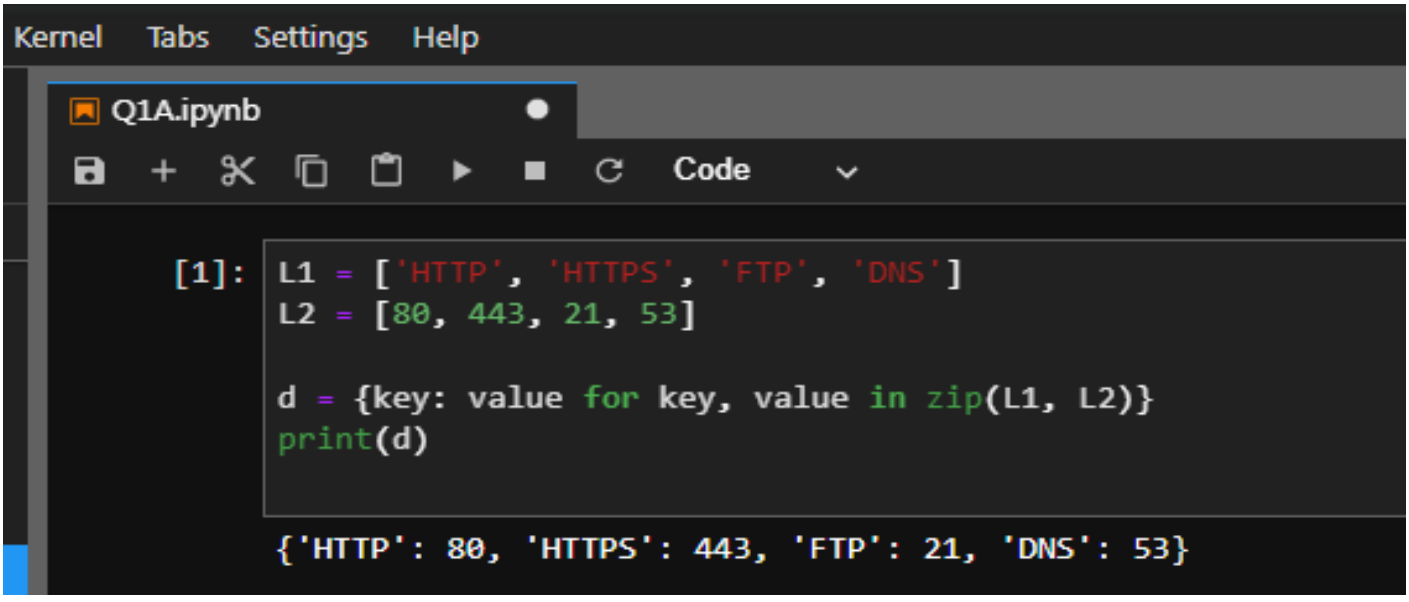
A-If you have two lists, L1=['HTTP','HTTPS','FTP','DNS'] L2=[80,443,21,53] ,Convert it to generate this dictionary d={'HTTP':80,'HTTPS':443,'FTP':21,'DNS':53} .

لتحويل القائمتين L1=['HTTP','HTTPS','FTP','DNS'] و L2=[80,443,21,53]

الى القاموس d={'HTTP':80,'HTTPS':443,'FTP':21,'DNS':53}

قمنا باستخدام التابع zip لربط عناصر القائمتين وتحويلها للقاموس المطلوب .

```
L1=['HTTP','HTTPS','FTP','DNS']
L2=[80,443,21,52]
d={key:value for key , value in zip(L1,L2)}
print(d)
```



B- Write a Python program that calculates the factorial of a given number entered by user .

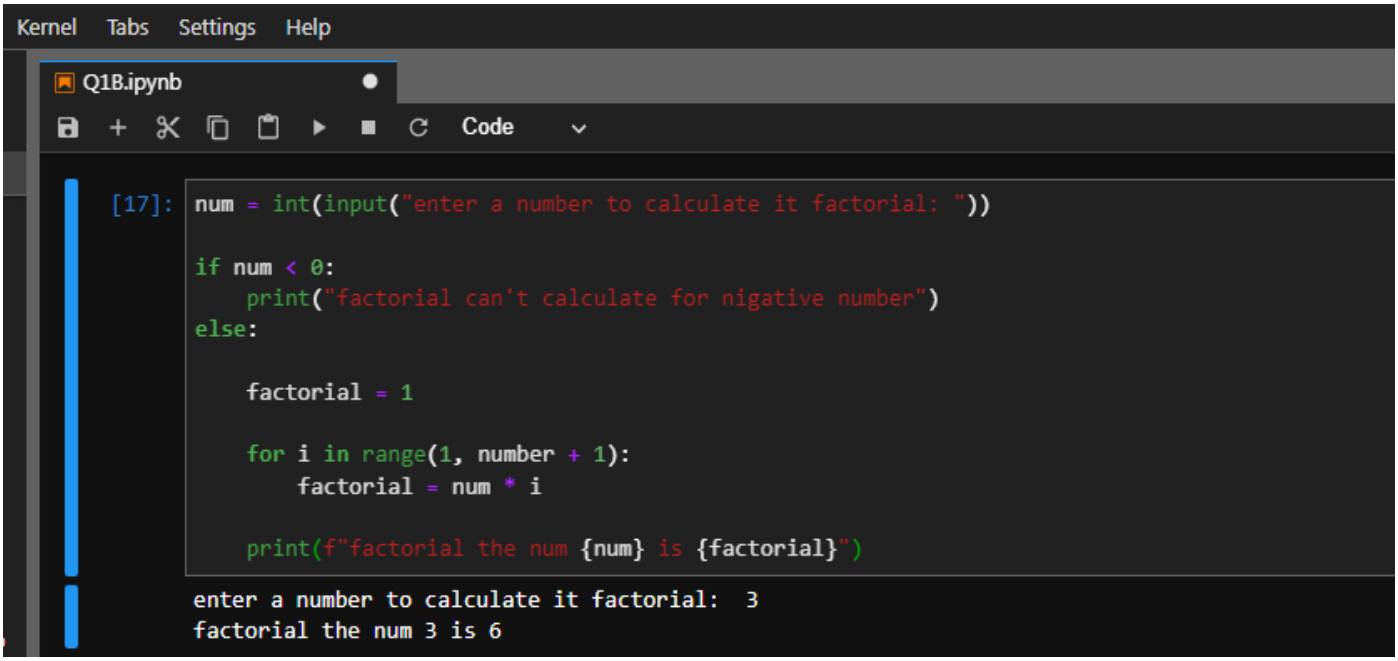
لحساب العاملي لأي رقم يتم ادخاله نقوم اولا بطلب ادخال رقم من قبل المستخدم ومن ثم نقوم بالتحقق من ان الرقم المدخل رقم صحيح غير سالب .

القيمة الاولية للعاملي هي ال١ .

نستخدم حلقة for التكرارية لحساب العاملي .

ونطبع النتيجة .

```
num = int(input("enter a number to calculate it factorial: "))

if num < 0:
    print("factorial can't calculate for nigative number")
else:

    factorial = 1

    for i in range(1, number + 1):
        factorial = num * i

    print(f"factorial the num {num} is {factorial}")
```



c-L=['Network','Bio','Programming','Physics','Music']

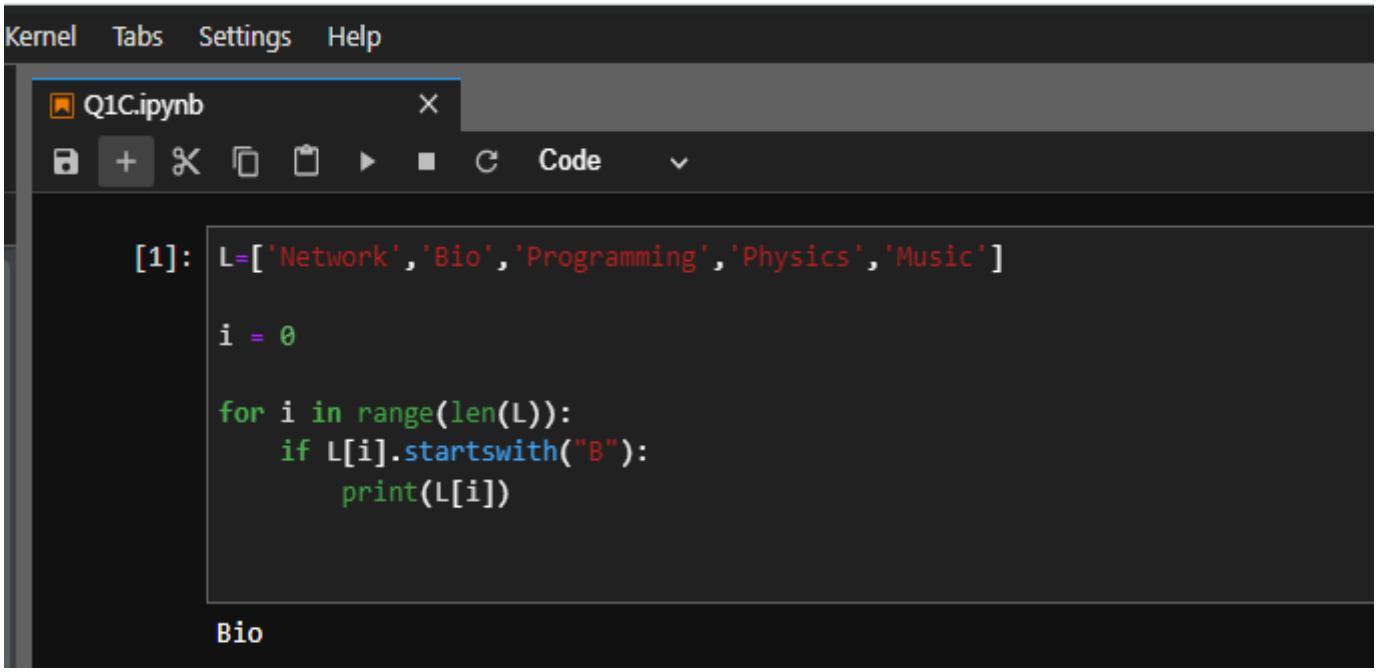
In this exercise , you will implement a Python program that reads the items of the previous list and identifies the **items that starts with 'B'** letter , then print it on screen .

Tips: using loop , 'len()' , startwith()methods .

نقوم بانشاء القائمة المطلوبة بالعناصر المطلوبة .

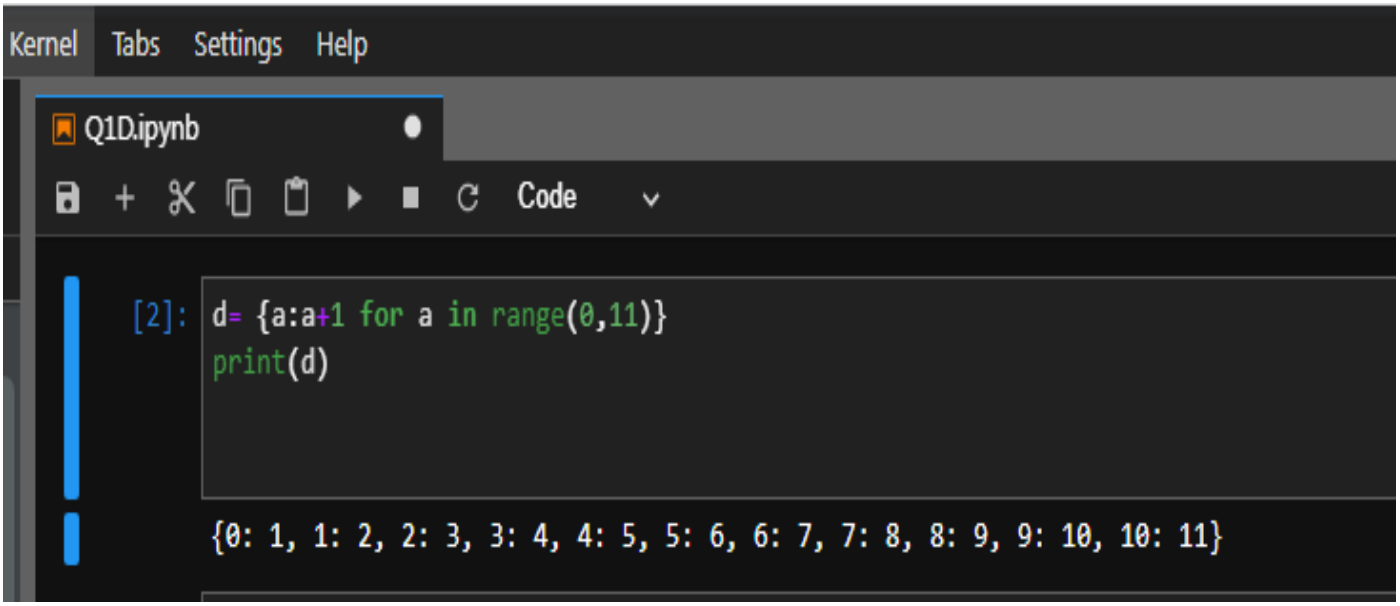
ومن ثم البحث عن الكلمات التي تبدا بالحرفB.

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

i = 0

for i in range(len(L)):
    if L[i].startswith("B"):
        print(L[i])
```



D-Using dictionary comprehension ,Generate this dictionary d={0:1,1:2,2:3,3:4,4:5,5:6,6:7,7:8,8:9,9:10,10:11} .

```
d= {a:a+1 for a in range(0,11)}
print(d)
```



Question 2: Convert from Binary to Decimal

Write a Python program that converts a Binary number into its equivalent Decimal number .

The program should start reading the binary number from the user . Then the decimal equivalent number must be calculated . finally ,the program must display the equivalent decimal number on the screen .

Tips: solve input errors .

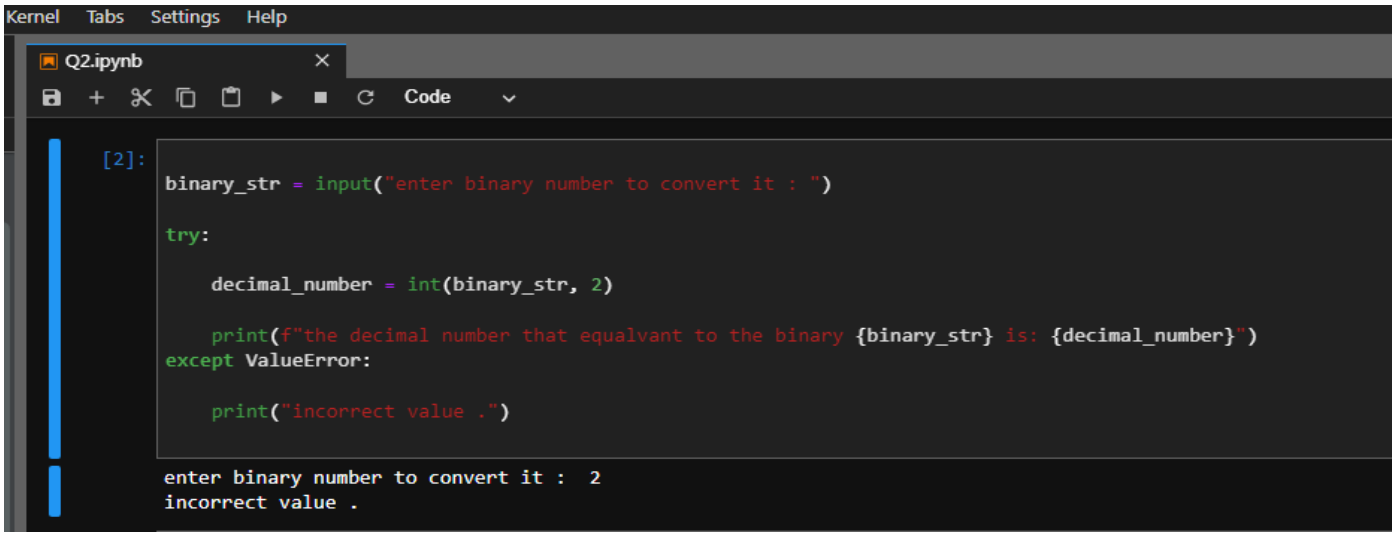
نطلب من المستخدم إدخال الرقم الثنائي .

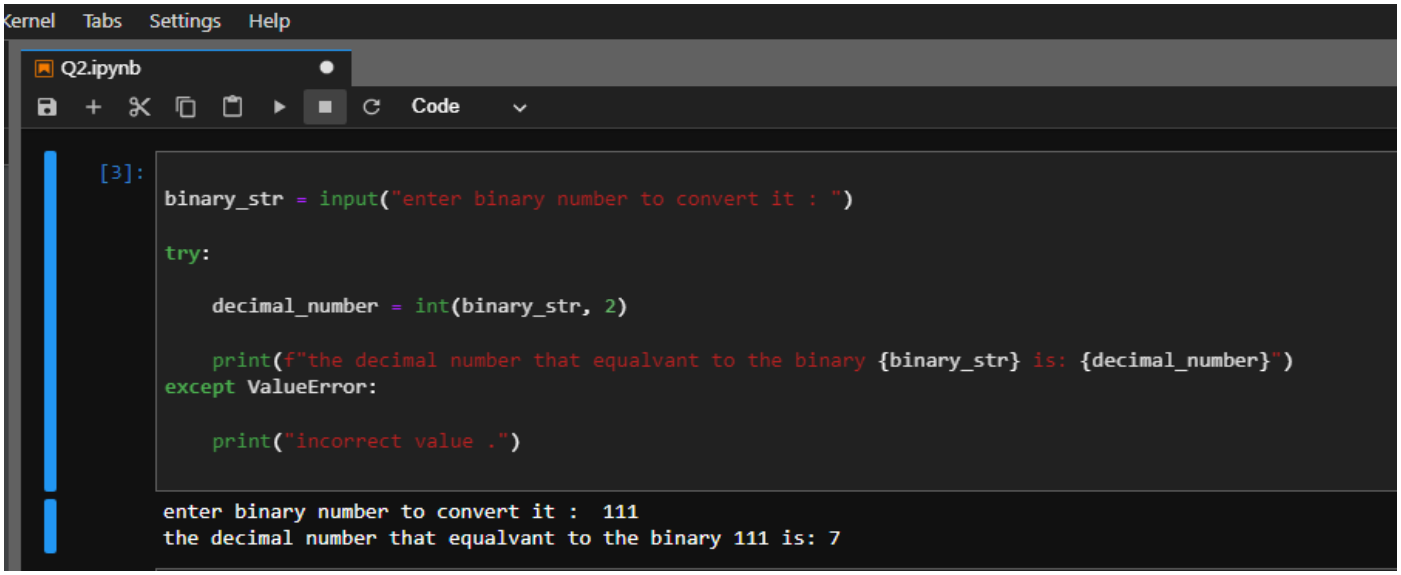
نحول الرقم الثنائي المدخل إلى عشري .

نطبع الرقم العشري الموافق .

ونطبع رسالة خطأ في حال كان الادخال غير صحيح .

```
binary_str = input("enter binary number to convert it : ")

try:

    decimal_number = int(binary_str, 2)

    print(f"the decimal number that equalvant to the binary {binary_str} is: {decimal_number}")
except ValueError:

    print("incorrect value .")
```





Question 3: Working With Files "Quiz Program"

Type python quiz program that takes a text or json or csv file as input for (20(Question , Answer)). It asks the question and finally computes and prints user name and result in separate file csv or json file .

نقوم بإنشاء ملف يحوي على اختبار ونتائجه ومن ثم نقوم بتصدير ملف Json .

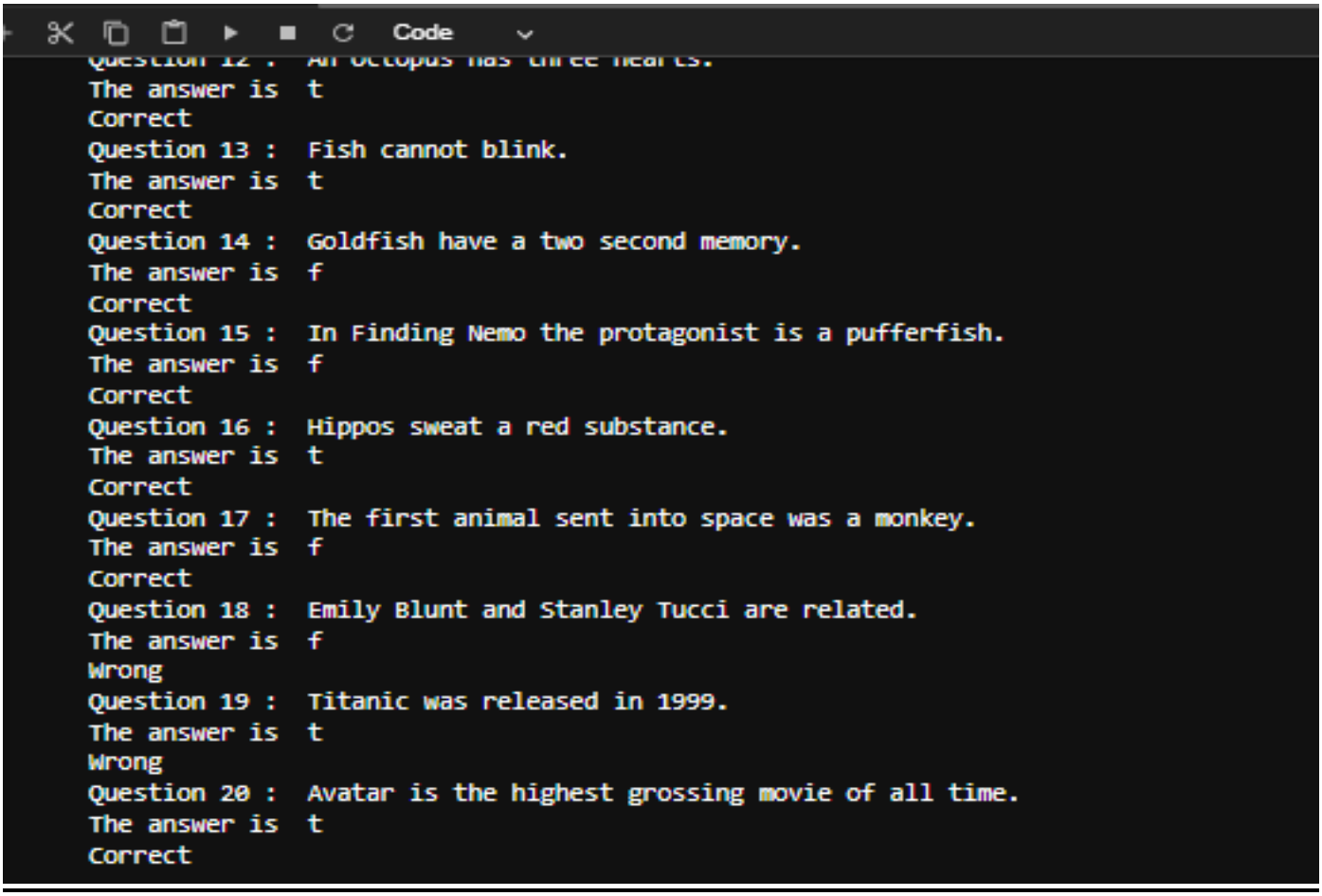
ومن ثم نقوم بتعيين قيمة ابتدائية لملف الخرج الذي سيحوي النتيجة ومن ثم نقوم بتعيين درجة كل سؤال

ومن ثم سنقوم بفتح ملف الاسئلة ضمن البرنامج وتحديد طريقة الاجابة وادخال الاسم الكامل من ثم نقوم بإظهار الاسئلة واختبار النتيجة وفي النهاية نقوم بكتابة الاسم والنتيجة في ملف منفصل .

```
import json
questions = { }
scores = 0
number=1
f = open("c:\questions.txt",'r')
questions = json.load(f)
f.close()
print("python quiz programm")
print("Enter t for True or f for False")
name = input("Enter your full name: ")
for ques in questions.keys():
    print("Question",number,": ", ques)
    ans = input("The answer is ")
    if ans.upper() == questions[ques].upper():
        scores = scores + 1
        print("Correct ")
    else:
        print ("Wrong")
    number = number + 1
result={name:scores}
m = open("score.txt",'w')
result = json.dump(result,m)
m.close()
```

```
[1]: import json
     questions = { }
     scores = 0
     number=1
     f = open("c:\questions.txt",'r')
     questions = json.load(f)
     f.close()
     print("python quiz programm")
     print("Enter t for True or f for False")
     name = input("Enter your full name: ")
     for ques in questions.keys():
         print("Question",number,": ", ques)
         ans = input("The answer is ")
         if ans.upper() == questions[ques].upper():
             scores = scores + 1
             print("Correct ")
         else:
             print ("Wrong")
         number = number + 1
     result={name:scores}
     m = open("score.txt",'w')
     result = json.dump(result,m)
     m.close()
```

```
python quiz programm
Enter t for True or f for False
Enter your full name:  suzan shakohi
Question 1 :  'A' is the most common letter used in the English language.
The answer is  f
Correct
Question 2 :  ASOS stands for As Seen On Screen.
The answer is  t
Correct
Question 3 :  The Battle Of Hastings took place in 1066.
The answer is  t
Correct
Question 4 :  H&M stands for Hennes & Mauritz.
The answer is  t
Correct
Question 5 :  K is worth four points in Scrabble.
The answer is  t
Correct
Question 6 :  In a deck of cards, the king has a moustache.
The answer is  f
Correct
Question 7 :  When the two numbers on opposite sides of a dice are added together it always equals 7.
The answer is  t
Correct
Question 8 :  In the English language there is no word that rhymes with orange.
The answer is  t
Correct
Question 9 :  English is the most spoken language in the world.
The answer is  f
Correct
Question 10 :  A lion's roar can be heard up to eight kilometres away.
The answer is  t
Correct
Question 11 :  Canis lupur is the scientific name for a wolf.
The answer is  t
Wrong
```



Question 4: Object – Oriented Programming – Bank Class

Define a class BankAccount with the following attributes and methods :

Attributes : account_number (string) , account_holder (string) , balance (float , initialized to 0.0)

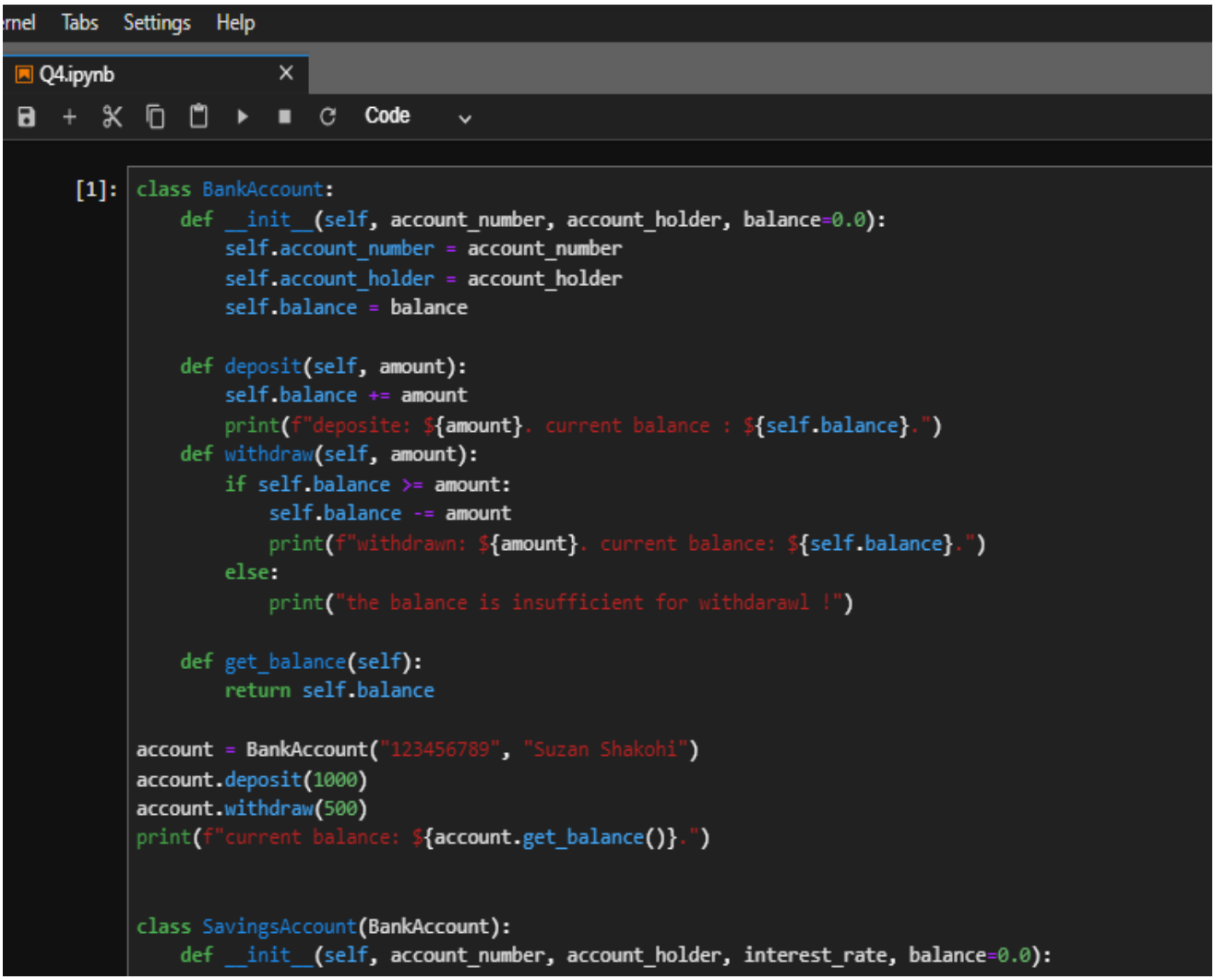
Methods : deposit (amount) , withdraw(amount) , get_balance()

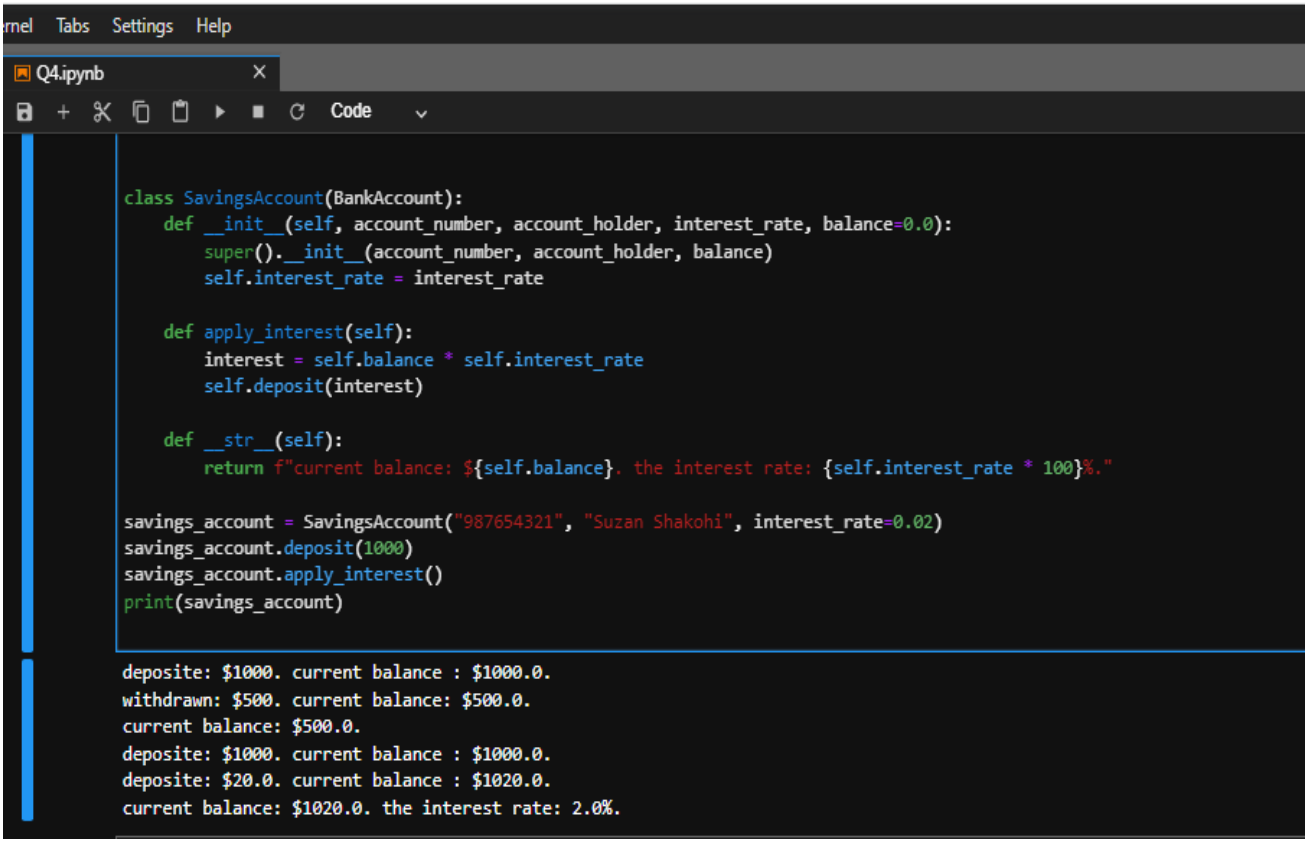
- Create an instance of BankAccount, -Perform a deposit of 1000$,
- Perform a withdrawal of 500$.
- Print the current balance after each operation .
- Define a subclass SavingsAccount that inherits from BankAccount and adds interest_rate Attribute and apply_interest() method that Applies interest rate .
- And Override print() method to print the current balance and rate .
- Create an instance of SavingsAccount , and call apply_interest() and print() function .

- لدينا الصف Bank Account الذي يمثل حساب بنكي يحتوي على :
- __init__ الذي يمثل ال contractor للصف وتستخدم لإنشاء كائنات جديدة من Bank Account الذي يقبل رقم الحساب واسم صاحب الحساب والرصيد الافتتاحي الذي تم تعيين قيمته الابتدائية ٠
- Deposit تقوم بإضافة مبلغ للرصيد الحالي وتقوم بطباعة الرصيد المودع والرصيد الجديد
- Withdraw يستخدم هذا التابع لسحب مبلغ من الحساب اذا كان كافيا
- get_balance ترجع الرصيد الحالي في الحساب
- القسم الثاني من الكود هو الصف SavingAccount وهو يورث الصف BankAccount اي انه يتضمن جميع الخصائص والطرق الموجودة لديه
- Interest_rate تمثل معدل الفائدة
- apply_interest تحسب الفائدة عن طريق ضرب معدل الفائدة في الرصيد الحالي واضافتها للرصيد
- __str__ تعريف لطريقة print()

```
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"deposite: ${amount}. current balance :
${self.balance}.")
    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
            print(f"withdrawn: ${amount}. current balance:
${self.balance}.")
        else:
            print("the balance is insufficient for withdarawl
!")

    def get_balance(self):
        return self.balance

account = BankAccount("123456789", "Suzan Shakohi")
account.deposit(1000)
account.withdraw(500)
print(f"current balance: ${account.get_balance()}.")
```

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

    def apply_interest(self):
        interest = self.balance * self.interest_rate
        self.deposit(interest)

    def __str__(self):
        return f"current balance: ${self.balance}. the
interest rate: {self.interest_rate * 100}%."

savings_account = SavingsAccount("987654321", "Suzan Shakohi",
interest_rate=0.02)
savings_account.deposit(1000)
savings_account.apply_interest()
print(savings_account)
```




rnel Tabs Settings Help
Q4.ipynb
Code
class SavingsAccount(BankAccount):
def __init__(self, account_number, account_holder, interest_rate, balance=0.0):
super().__init__(account_number, account_holder, balance)
self.interest_rate = interest_rate
def apply_interest(self):
interest = self.balance * self.interest_rate
self.deposit(interest)
def __str__(self):
return f"current balance: ${self.balance}. the interest rate: {self.interest_rate * 100}%."
savings_account = SavingsAccount("987654321", "Suzan Shakohi", interest_rate=0.02)
savings_account.deposit(1000)
savings_account.apply_interest()
print(savings_account)
deposite: $1000. current balance : $1000.0.
withdrawn: $500. current balance: $500.0.
current balance: $500.0.
deposite: $1000. current balance : $1000.0.
deposite: $20.0. current balance : $1020.0.
current balance: $1020.0. the interest rate: 2.0%.