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

وظيفة رقم 1

اسم الطالبة دلامه ناجح عثمان

الرقم الجامعي: 2382

1-

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 }

```
File  Edit  Format  Run  Options  Window  Help
d={}
L1=['HTTP','HTTPS','FTP','DNS']
L2=[80,443,20,53]
for a,v in enumerate(L1):
    d[v]=L2[a]
print(d)
```

```
>>> {'HTTP': 80, 'HTTPS': 443, 'FTP': 20, 'DNS': 53}
```

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

تابع factorial يتعامل مع إدخال سالب، ويعيد 1 لـ 0، ويحسب العاملة للأعداد الموجبة باستخدام حلقة.

```
File  Edit  Format  Run  Options  Window  Help
def factorial(n):
  """يحسب عامل عدد صحيح غير سالب n."""
  if n < 0:
    return "العامل غير محدد للأعداد السالبة."
  elif n == 0:
    return 1
  else:
    result = 1
    for i in range(1, n + 1):
      result *= i
    return result

num = int(input("أدخل عددًا غير سالب: "))
fact_result = factorial(num)
print(fact_result)
```

OUTPUT:

عند ادخال عدد موجب 5 يكون 5 عاملة يساوي 120

```
أدخل عددًا غير سالب: 5
120
>>>
```

عند ادخال عدد سالب -5 يكون غير معرف

```
أدخل عددًا غير سالب: -5
العامل غير محدد للأعداد السالبة.
>>>
```

0 عاملة يساوي الواحد

```
أدخل عددًا غير سالب: 0
1
>>>
```

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.

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

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

تتكرر الحلقة من خلال L ، والوظيفة startswith('B') تتحقق مما إذا كان العنصر يبدأ بـ B

OUTPUT:

```
C.py
Bio
>>>
```

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}=

```
File  Edit  Format  Run  Options  Window  Help
d = {i: i + 1 for i in range(11)}
print(d)
```

OUTPUT:

```
D.py
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}
>>>
```

2- 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.

```
File  Edit  Format  Run  Options  Window  Help
def Func(b):
    dec=0
    l=[]
    try:
        for i in range(len(b)):
            t= int(b[i])
            dec += t * 2**(len(b)-i-1)
        return dec
    except Exception as e:
        print(e)
print("Bin To Dec Converter App")
while True:
    b=input("Input an binary number OR enter x to exit: ")
    if b=='x':
        print("End")
        break

    d=Func(b)
    if d:
        print(d)
```

يتيح هذا التابع تحويل أي رقم ثنائي تم إدخاله إلى معادله العشري. يعتمد التابع على دورة تتكرر عبر كل بت في الرقم الثنائي، مع مراعاة موقعه (قوة 2) لحساب القيمة العشرية. يتم جمع هذه القيم الجزئية تدريجيًا في متغير لتكوين القيمة العشرية النهائية.يضمن التابع معالجة الأخطاء بشكل أنيق في حال إدخال بيانات غير صالحة.

OUTPUT:

```
Bin To Dec Converter App
Input an binary number OR enter x to exit: 111
7
Input an binary number OR enter x to exit: 010
2
Input an binary number OR enter x to exit: 11110
30
Input an binary number OR enter x to exit: d
invalid literal for int() with base 10: 'd'
Input an binary number OR enter x to exit: x
End
```

3- Type python quiz program that takes a text or json or csv file as input for (20 (Questions, Answers)). It asks the questions and finally computes and prints user results and store user name and result in separate file csv or json file.

File Edit Format Run Options Window Help

```
def main():
#التابع الرئيسي
    print('answer with yes or no')
    user=input('enter username: ')
    quiz(user)
def quiz(name):
    #التابع الخاص بالاسئلة
    #قائمة فارغة لوضع الأسطر من الملف بداخلها من أجل المعالجة
    l=[]
    #عداد الاجابات الصحيحة
    count=0
    #فتح ملف الأسئلة للقراءة منه
    q=open('questions.csv','r')
    #فتح ملف لكتابة النتيجة داخله
    res=open('answers.csv','w')

    for i in q:
        #تحويل كل سؤال من سلسلة محرفية إلى قائمة بعنصرين هما السئوال والجواب
        ll=i.rstrip().split(',')
        l.append(ll)
   # طباعة السؤال والسماح للمستخدم بإدخال الإجابة
    for i in l:
        print(i[0])
        ans=input().lower()
        #مقارنة الإجابة بالإجابة الموجودة ضمن الملف
        if ans==i[1].lower():
            count+=1
            print('correct')
        else:
            print('incorrect')
    print(name)
    print('true answers is ',count,'from 20')
    #كتابة الاسم والنتيجة ضمن الملف
    res.write(name+'true answers is '+ str(count))
    q.close()
    res.close()
#استدعاء التابع الرئيسي
main()

```

التابع main()

يُعدّ هذا التابع نقطة البدء في البرنامج .فهو يطبع تعليمات للمستخدم للإجابة على الأسئلة بكلمة "true" أو ""falseثم يطلب من المستخدم إدخال اسم المستخدم الخاص به .يستدعي التابع quiz()للتعامل مع عملية الإجابة على الأسئلة الفعلية.

التابع quiz()

يدير هذا التابع تفاعل الاختبار مع المستخدم .فهو يقوم بإنشاء قائمة فارغة لتخزين الأسئلة والأجوبة المقروءة من الملف .كما يبادر بإنشاء عداد countلمتابعة عدد الإجابات الصحيحة .ثم يفتح ملفين questions.csv :لقراءة الأسئلة و answers.csvلكتابة النتائج.

معالجة الأسئلة والأجوبة:

يتكرر التابع quiz()على كل سطر في ملف questions.csv .يتم تقسيم كل سطر يقسمه إلى قائمة 1تحتوي على السؤال والإجابة الصحيحة (بحروف صغيرة) .ثم يلحق القائمة L1 بالقائمة L للتخزين.

تفاعل المستخدم وفحص الإجابة:

يتكرر التابع quiz()على القائمة L لعرض كل سؤال على المستخدم .لكل سؤال، يطبع السؤال ويطلب من المستخدم إدخال إجابته .ثم يحول إجابة المستخدم إلى أحرف صغيرة للمقارنة دون تمييز بين الأحرف الكبيرة والصغيرة .بعد ذلك، يقارن إجابة المستخدم بالإجابة الصحيحة (أيضًا بحروف صغيرة) .إذا تطابقت الإجابات، فإنه يزيد من countويطبع correct وإذا لم تتطابق الإجابات، فإنه يطبع incorrect.

عرض النتائج والكتابة إلى الملف:

يطبع التابع quiz()اسم المستخدم ودرجته (الإجابات الصحيحة من أصل 20) .كما يكتب اسم المستخدم ودرجته في ملف answers.csv.وأخيراً، يغلق كلا من ملف questions.csv و answers.csv.

questions.csv

	A	B
	Python code runs line by line as you write it.	TRUE
	You need curly braces ({}) for code blocks in Python.	FALSE
	You can directly add numbers to strings in Python.	FALSE
	Python offers ways to add comments to your code.	TRUE
	Python lists can hold various data types, even other lists.	TRUE
	You can't change the order of items in a Python list.	FALSE
	You can access elements in a Python list using their positic	TRUE
	The if statement lets you execute code based on a conditi	TRUE
	A function in Python is a reusable block of code you can ca	TRUE
0	You need separate Python programs to share code with m	FALSE
1	Python requires you to declare the data type of a variable	FALSE
2	Python is not suitable for creating games.	FALSE
3	Objects in Python can have properties (data) and methods	TRUE
4	Errors (mistakes) will always crash your Python program.	FALSE
5	There's no way to handle errors and keep your Python pro	FALSE
6	Python is an interpreted language.	TRUE
7	Python requires semicolons at the end of statements.	FALSE
8	Lists are mutable (can be changed) in Python.	TRUE
9	Tuples are ordered collections that can hold different data	TRUE
0	Python is a powerful and versatile language for various ap	TRUE
1		

OUTPUT:

```
File  Edit  Shell  Debug  Options  Window  Help
correct
A function in Python is a reusable block of code you can call.
true
correct
You need separate Python programs to share code with modules.
false
correct
Python requires you to declare the data type of a variable beforehand.
true
incorrect
Python is not suitable for creating games.
false
correct
Objects in Python can have properties (data) and methods (actions).
false
incorrect
Errors (mistakes) will always crash your Python program.
false
correct
There's no way to handle errors and keep your Python program running.
false
correct
Python is an interpreted language.
false
incorrect
Python requires semicolons at the end of statements.
false
correct
Lists are mutable (can be changed) in Python.
flase
incorrect
Tuples are ordered collections that can hold different data types.
false
incorrect
Python is a powerful and versatile language for various applications.
false
incorrect
Dolamah
true answers is  13 from 20
>>>
```

answers.csv

	A	B	C
1	Dolamah true answers is 13		
2			
3			
4			

4- Object-Oriented Programming - Bank Class

File Edit Format Run Options Window Help

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

  def deposit(self, amount):
    if amount > 0:
      self.balance += amount
      print(f"Deposited {amount}. New balance: {self.balance}")
    else:
      print("Invalid deposit amount. Please enter a positive value.")

  def withdraw(self, amount):
    if amount > 0 and self.balance >= amount:
      self.balance -= amount
      print(f"Withdrew {amount}. New balance: {self.balance}")
    else:
      print("Insufficient funds or invalid withdrawal amount.")

  def get_balance(self):
    return self.balance

  def __str__(self):
    return f"Account Number: {self.account_number}, Account Holder: {self.account_holder}, Balance: {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):
    interest = self.balance * self.interest_rate
    self.balance += interest
    print(f"Applied interest of {interest}. New balance: {self.balance}")

  def __str__(self):
    return super().__str__() + f"\nInterest Rate: {self.interest_rate}"

# Create a BankAccount
my_account = BankAccount("1234567890", "Dolamah Osman")

# Deposit $1000
my_account.deposit(1000)

# Withdraw $500
my_account.withdraw(500)

# Print current balance
print(my_account)  # This will use the overridden __str__ of BankAccount

# Create a SavingsAccount instance
savings_account = SavingsAccount("9876543210", "Ali ALi", 0.05)

# Apply interest
savings_account.apply_interest()

# Print account details with interest rate
print(savings_account)
```

1- تعريف BankAccount

التابع __init__

يُنشئ رقم الحساب، اسم صاحب الحساب، ورصيد الحساب (الافتراضي 0.0).

التابع deposit

يودع مبلغاً موجب في الرصيد ويطبع التحديث.

التابعwithdraw

يسحب مبلغ من الرصيد مع التحقق من رصيد كاف ويطبع التحديث.

التابع get_balance

يُرجع الرصيد الحالي.

التابع __str__

يوفر تمثيل نصي لتفاصيل الحساب.

1- تعريف SavingsAccount يرث من BankAccount

التابع:__init__

يستدعي مُنشئ التابع الأساسي ويضيف interest_rate

التابع apply_interest

يحسب الفائدة بناءً على السعر ويضيفها إلى الرصيد ويطبع التحديث.

التابع __str__

يُلغي طريقة التابع الأساسية لتضمين interest_rate في تمثيل النص.

OUTPUT:

```
Deposited 1000. New balance: 1000.0
Withdrew 500. New balance: 500.0
Account Number: 1234567890, Account Holder: Dolamah Osman, Balance: 500.0
Applied interest of 0.0. New balance: 0.0
Account Number: 9876543210, Account Holder: Ali ALi, Balance: 0.0
Interest Rate: 0.05
>>
```