



الجمهورية العربية السورية
جامعة تشرين
كلية الهندسة
هندسة الاتصالات والالكترونيات
السنة الخامسة حل الوظيفة الاولى

الاسم : يوسف اسامة الشاعر
الرقم الجامعي : 2129

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 }

الحل :

ننشأ الـليست الأولى والثانية ونقوم بإنشاء قاموس فارغ لتخزين القيم فيه
ونقوم بإنشاء حلقة **for** بحيث تكون على عدد عناصر الـليست
ونقوم بـ أخذ المفاتيح من الـليست الأولى والقيم من الـليست الثانية
ونطبع النتيجة

```
+ Code + Text

#Q1 A
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- Write a Python program that calculates the factorial of a given number entered by user.

الحل :

نقوم بتعريف تابع يأخذ بارمتر وحيد هو الرقم المدخل

مداخل التابع :

نعرف متغير باسم `result` لنضع به النتيجة قيمته الابتدائية 1

ثم باستخدام حلقة `for` نقوم بالمرور على كل الارقام من العدد المدخل وحتى ال1 تنازلياً .

ونضرب الارقام سوياً ثم نحفظ النتيجة في المتغير `result`

في البرنامج الرئيسي نطلب ادخال رقم من المستخدم

`##` في حال كان الرقم يساوي صفر يعيد الكود قيمة 1

`##` في حال كان الرقم أكبر من صفر يذهب إلى التابع المعرف مسبقاً والذي يعيد القيمة المطلوبة .

✓
3s



```
#Q1 B
def factorial_calc(n):
    result = 1
    for i in range(n,1,-1):
        result *= i
    return result

num = int(input("Enter a number to calculat its factorial: "))
print("The factorial of " ,num, " is: " , factorial_calc(num))
```



```
Enter a number to calculat its factorial: 5
The factorial of 5 is: 120
```

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 ()'` , `startswith()` methods.

نقوم بتعريف القائمة المعطاة وننشئ حلقة `for` بحيث تمر على جميع عناصر القائمة ونستخدم

الخاصية `startswith` لايجاد العناصر التي تبدأ بحرف `B` ثم نقوم بطباعتها



0s



```
#Q1 C
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
length = len(L)
i = 0

while i < length:
    if L[i].startswith('B'):
        print(L[i])
    i += 1
```



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}

الحل :

ننشأ قاموس فارغ و ثم باستخدام حلقة while نقوم بانشاء المفاتيح والقيم المقابلة لها بحيث تكون القيم اكبر من المفاتيح بمقدار 1



```
#Q1 D
d = {}
i = 0
while i < 11:
    d[i] = i + 1
    i += 1
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 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.

الحل :

نقوم ب انشاء تابع مهمته تحويل العدد من ثنائي الى عشري وذلك بعد اختبار السلسلة المدخلة انها ثنائية ولا تحوي غير القيمتين 0 و 1 ونقوم بعدها بتحويل كل رمز الى رقم ومن ثم حساب الرقم العشري الموافق للترميز الثنائي وطباعة الناتج

```
#Q2
def bi2de(bi):
    dec = 0
    for i in range(len(bi)):
        if bi[i] != '0' and bi[i] != '1':
            print("Invalid input. Please enter a binary number.")
            return
        dec += int(bi[i]) * 2**(len(bi) - i - 1)
    print(f"The decimal equivalent of {bi} is: {dec}")

bi = input("Enter a binary number: ")
bi2de(bi)
```

Enter a binary number: 1011
The decimal equivalent of 1011 is: 11

Question 3: Working with Files" Quiz Program"

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.

الحل:

- انشأنا ملف اسمه y.text فيه 20 سؤال مع اجوبتهم وقمنا برفعه على الغوغل كولا

1- نعرف تابع يقوم بتحميل الأسئلة من ملف مدخل (البارمتر للتابع هو الملف المدخل) ثم يشكل قائمة بالأ سئلة مع أجوبتها للاستفادة منها .

ماداخل التابع :

- ننشئ قائمة فارغة لوضع الأسئلة .
- نحدد للتابع حالة قراءة الملف المدخل .
- نعرف متغير لعدد الأسطر لنتمكن لاحقا من المرور على كل الأسطر
- نستخدم حلقة للمرور على كل الأسطر التي تم تحديد عددها ويكون المرور بقفز ثنائي (لأن السطر اللاحق لكل سطر تمر عليه الحلقة هو إجابة - سطر سؤال ويليه سطر إجابة السؤال -)
- نقوم بتحديد الأسئلة من الاسطر التي تقفز لها الحلقة .
- نقوم بتحديد الأجوبة (السطر التالي لكل سؤال يمثل إجابة له)
- نضيف السؤال وإجابته إلى القائمة التي تم تعريفها مسبقا .

2- نعرف تابع يأخذ بارامترين (اسم المستخدم ونتيجته)

يقوم هذا التابع بإنشاء ملف لحفظ الاسم والنتيجة .

3- التابع الثالث يستخدم للمرور على الأسئلة وطرحها على المستخدم

(يمرر للتابع بارامتر وحيد هو قائمة الأسئلة التي تم إنشاؤها في التابع الأول)

ماداخل التابع :

- نعرف متغير النتيجة ونعطيه قيمة 0 قبل بدء طرح الأسئلة .
- باستخدام حلقة يتم المرور على كل الأسئلة وطباعتها للمستخدم كل على حدى
- يطلب من المستخدم إدخال إجابة عند كل سؤال
- في حال كانت إجابة المستخدم مطابقة لإجابة السؤال (إجابة السؤال ذي الفهرس 0 هي الفهرس 1 وه كذا) و يضاف 1 للنتيجة .
- بعد المرور على كل الأسئلة تكون النتيجة النهائية هي مجموع النقاط و هي ما يعيده التابع .

4- في التابع الرئيسي يتم تحميل الملف المطلوب

- يطلب من المستخدم إدخال اسمه

- يتم استدعاء تابع السئلة لمعرفة النتيجة

- تطبع النتيجة للمستخدم (عدد الإجابات الصحيحة من العدد الكلي للإجابات الصحيحة) ويطبع الاسم قبلها

- ترسل النتيجة والاسم لتابع النتيجة الذي يقوم بحفظهما في ملف .

```
✓ 1m #Q3
def load_questions(file_name):
    questions = []
    with open(file_name, 'r') as file:
        lines = file.readlines()
        for i in range(0, len(lines), 2):
            question = lines[i].strip()
            answer = lines[i+1].strip()
            questions.append((question, answer))
    return questions

def save_results(user_name, score):
    with open('results.csv', 'a') as file:
        file.write(f"{user_name},{score}\n")

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

def main():
    questions = load_questions('/content/y.txt')
```

✓
1m



```
return score
```

```
def main():  
    questions = load_questions('/content/y.txt')  
    user_name = input("Enter your name: ")  
    user_score = take_quiz(questions)  
    print(f"Your score:{user_score}/20")  
    save_results(user_name, user_score)  
if name == 'main':  
    main()
```



```
Enter your name: yousef  
1-what is the cpital of syria  
Your answer: damascus  
2-what is the capital of irak  
Your answer: bagdad  
3-what is the capital of jordon  
Your answer: amaan  
4-what is the capital of qatar  
Your answer: ppp  
5-what is 5 * 5  
Your answer: 25  
6-what is 6*6  
Your answer: 36  
7-what is 7*7  
Your answer: 49  
8-what is 2*2  
Your answer: 4  
9-what is 3*3
```

✓ 1m 33s completed at 10:00 PM


```
✓ 1m Your answer: 4
9-what is 3*3
Your answer: 9
10-what is 9*9
Your answer: 81
11-what is 1*9
Your answer: 9
12-what is 2*9
Your answer: 1
13-what is 3*1
Your answer: 3
14-what is 3*2
Your answer: 6
15-what is 3*5
Your answer: 15
16-what is 3*6
Your answer: 18
17-what is 2*1
Your answer: 2
18-what is 5*1
Your answer: 5
19-what is 6*1
Your answer: 6
20-what is 7*1
Your answer: 7
Your score:17/20
```

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 to the balance based on the interest rate.And **Override print()** method to print the current balance and rate.

- Create an instance of SavingsAccount , and call apply_interest() and print() functions.

الحل :

نعرف كلاس اول اسمه BankAccount حيث عند الانشاء ياخذ القيم رقم الحساب و اسم المستخدم ننشئ ميثود للايداع و ميثود للسحب و اخر لطباعة القيمة المالية المتبقية في الحساب

نشتق غرض من الكلاس BankAccount للمستخدم الذي اسمه Youssef ورقم حسابه 4444 ونقوم بعملية ايداع مبلغ \$1000 وسحب \$500 وثم طباعة العملية

ننشئ كلاس ثاني اسمه SavingsAccount يرث الكلاس الاول ويحتوي على توابع لحساب الفائدة وطباعة المبلغ بعد اضافة الفائدة

```
# السؤال الرابع
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
        print("Deposited $",amount, "Current balance: $",self.balance)

    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
            print("Withdrew $",amount," Current balance: $",self.balance)
        else:
            print("Insufficient balance.")

    def get_balance(self):
        return self.balance

account1 = BankAccount(account_number="4444", account_holder="youssef")
account1.deposit(1000)
account1.withdraw(500)
print("Final balance: $" , account1.get_balance())
```

+ Code + Text

✓
1s



```
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_amount = self.balance * (self.interest_rate / 100)
        self.balance += interest_amount
        print("Interest applied. Current balance: $",self.balance)

    def print(self):
        print("Balance: $",self.balance)
        print("Interest rate:",self.interest_rate,"%")

savings_account = SavingsAccount(account_number="2222", account_holder="noor", interest_rate=8)
savings_account.deposit(2000)
savings_account.apply_interest()
savings_account.print()
```



```
Deposited $ 1000 Current balance: $ 1000.0
Withdrew $ 500 Current balance: $ 500.0
Final balance: $ 500.0
Deposited $ 2000 Current balance: $ 2000.0
Interest applied. Current balance: $ 2160.0
Balance: $ 2160.0
Interest rate: 8 %
```

