

الجمهورية العربية السورية
وزارة التعليم العالي والبحث العلمي
جامعة تشرين
هندسة الاتصالات والإلكترونيات

برمجة الشبكات
إعداد الطالبة:

سدره زين تنزكلي

(2816)

بإشراف الدكتور مهند عيسى

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 }

```
1 L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
2 L2 = [80, 443, 21, 53]
3 d = {key: value for key, value in zip(L1, L2)}
4 print(d)
```

!

```
In [12]: runfile('C:/Users/windows/Desktop/sidra tnzkle/السؤال الاول A.py', wdir='C:/
Users/windows/Desktop/sidra tnzkle')
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
```

شرح الكود :

وضعت في السطر رقم 3 قيم المصفوفة L 1 كمفاتيح (key) وقيم L2 كقيم (value)
طريقة dictionary comprehension.

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

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

num = int(input("Enter a number: "))
result = factorial(num)
print(f"The factorial of {num} is {result}")
```

```
Users/windows/Desktop/sidra tnzkle')
Enter a number: 8
The factorial of 8 is 40320

In [14]: runfile('C:/Users/windows/Desktop/sidra tnzkle/السؤال الأول B.py', wdir='C:/
Users/windows/Desktop/sidra tnzkle')
Enter a number: 9
The factorial of 9 is 362880
```

شرح الكود: تم بناء التابع factorial لحساب عاملي العدد المدخل

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)
```

```
In [1]: runfile('C:/Users/windows/Desktop/sidra tnzkle/السؤال الأول C.py', wdir='C:/Users/
windows/Desktop/sidra tnzkle')
Bio

In [2]:
```

شرح الكود : يبحث عن عنصر ضمن القائمة يضم الحرف B اذا وجد يتم طباعته

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

```
1 d = {x:x+1 for x in range(11)}
2 print(d)
3
```

```
In [2]: runfile('C:/Users/windows/Desktop/sidra tnzkle/السؤال الأول D.py', wdir='C:/Users/
windows/Desktop/sidra tnzkle')
{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.

```
1 binary=input("Enter Binary number : ")
2 try:
3     dicemal = int(binary , 2)
4     print ("the dicemal of", binary,"is",dicemal)
5 except ValueError:
6     print("invalid input the number must be binary number")
7
```

```
Enter Binary number : 11001  
the dicemal of 11001 is 25
```

شرح الكود: يبدأ البرنامج بمطالبة المستخدم بإدخال رقم ثنائي ثم يقوم بتحويل الرقم الثنائي إلى مكافئته العشرية باستخدام التابع `int`، إذا لم يكن الإدخال رقم ثنائي يقوم التابع بطباعة رسالة خطأ تشير إلى أن الإدخال غير صالح. إذا كان الإدخال صالحاً، يقوم البرنامج بطباعة المكافئ العشري للرقم الثنائي

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.

```
C:\Users\windows\.spyder-py3\السؤال الثالث.py  
السؤال الاول C.py X السؤال الاول D.py X السؤال الثاني.py X السؤال الرابع.py X السؤال الثالث.py X  
1 import json  
2  
3 filename = input("Enter your name of the json file with question and answer:  
4 grad=0  
5 with open ("questions.json","r") as file:  
6     z=json.load(file)  
7     name = input("enter your name: ")  
8     for i in range (1,21):  
9         user_answer = input(f"{i} : {z[str(i)][0]} : ")  
10        if user_answer == z[str(i)][1]:  
11            grad+=1  
12    print(f"the deserved grad is {grad}")  
13    try:  
14        with open('result.json','a',newline='') as file:  
15            file.write(user_answer+": "+ str(grad)+"\n")  
16    except IOError:  
17        print("Error: could not write to result.json file.")  
18  
19  
20
```

الخرج

```
windows/.spyder-py3')
Enter your name of the json file with question and answer: questions.json
enter your name: sidra
1 : layered architecturehelps in reduceing networks design complexity : true
2 : TCP allows the sending process to deliver data as a stearm of byte : true
3 : STP was developed to solve layer 3 loops : false
4 : ports frome 0 to 1023 can be freely used as temporary or prive port number :
false
5 : repeaters are basedband devices : true
6 : ARP resolve IP to MAC address : true
7 : ARP request is unicast : false
8 : ARP replay is unicast : true
9 : vlans allow us to implement logical grouping of devices by function rather than
by geographic location : true

IPython Console History
conda: base (Python 3.10.9)  Completions: conda(base)  Line 1, Col 1  ASCII  CRLF  RW  Mem 75%
```

ملف الاسئلة

```
1 {
2   "1" : ["layered architecturehelps in reduceing networks design complexity","true"],
3   "2" : ["TCP allows the sending process to deliver data as a stearm of byte","true"],
4   "3" : ["STP was developed to solve layer 3 loops","false"],
5   "4" : ["ports frome 0 to 1023 can be freely used as temporary or prive port number","false"],
6   "5" : ["repeaters are basedband devices","true"],
7   "6" : ["ARP resolve IP to MAC address","true"],
8   "7" : ["ARP request is unicast","false"],
9   "8" : ["ARP replay is unicast","true"],
10  "9" : ["vlans allow us to implement logical grouping of devices by function rather than by geographic location","true"],
11  "10" : ["in vlans the switch maintains a separate MAC table for each vlan","false"],
12  "11" : ["in pure aloha the station can send any time","true"],
13  "12" : ["in TCP the ISN is a random number","true"],
14  "13" : ["leading bits of class C ara 110","true"],
15  "14" : ["vlsm reduces the number of lost address compared flsm","true"],
16  "15" : ["port number helps us to find witch process we are communicate with","true"],
17  "16" : ["IEEE 802.1Q headeris larger than ISL header","false"],
18  "17" : ["the ICMP replay sends pack the first 8 bytes of received packet in its payload","false"],
19  "18" : ["BPDPU is bridgeprotocol data unit","true"],
20  "19" : ["what is the answer of 3+61","64"],
21  "20" : ["what is the answer of 5+9","14"]
22 }
23
```

شرح الكود: نقوم بتحميل مكتبة JSON ثم نطلب من المستخدم إدخال اسم ملف JSON الذي يحتوي على الاسئلة والاجابات ثم نقوم بفتح ملف الـ JSON ونطلب من المستخدم إدخال اسمه ونستخدم حلقة لتكرار عملية طرح الاسئلة وتلقي إجابات المستخدم على الاسئلة و يقوم البرنامج بالتحقق مما إذا كانت الاجابة التي قدمها المستخدم صحيحة أم لا، ويزيد درجة المستخدم إذا كانت الاجابة صحيحة ثم يقوم البرنامج بطباعة الدرجة النهائية التي حصل عليها المستخدم ثم نستخدم

العبارة try و except للتحقق من وجود ملف "results.json" وإذا كان موجود
نقوم بكتابة الاجابة المعطاة من قبل المستخدم والدرجة التي حصل عليها في
نهاية الاختبار في الملف وإذا لم يكن الملف يطبع العبارة .

Error: Could not write to results.json file

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.


```
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 self.balance >= amount:
            self.balance -= amount
        else:
            print("Insufficient funds")

    def get_balance(self):
        return self.balance

bank_account = BankAccount("123456789", "sidra")
bank_account.deposit(1000)
print("Balance after deposit: $", bank_account.get_balance())
bank_account.withdraw(500)
print("Balance after withdrawal: $", bank_account.get_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

    def print(self):
        print("Balance: $", self.get_balance())
        print("Interest Rate: ", self.interest_rate)
```

```
def withdraw(self, amount):
    if self.balance >= amount:
        self.balance -= amount
    else:
        print("Insufficient funds")

    def get_balance(self):
        return self.balance

bank_account = BankAccount("123456789", "sidra")
bank_account.deposit(1000)
print("Balance after deposit: $", bank_account.get_balance())
bank_account.withdraw(500)
print("Balance after withdrawal: $", bank_account.get_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

    def print(self):
        print("Balance: $", self.get_balance())
        print("Interest Rate: ", self.interest_rate)

savings_account = SavingsAccount("987654321", "sidra", 0.05)
savings_account.deposit(2000)
print("Balance before applying interest: $", savings_account.get_balance())
savings_account.apply_interest()
print("Balance after applying interest: $", savings_account.get_balance())
savings_account.print()
```

```
In [2]: runfile('C:/Users/windows/Desktop/sidra tnzkle/untitled0.py', wdir='C:/Users/windows/Desktop/sidra tnzkle')
```

```
Balance after deposit: $ 1000.0
```

```
Balance after withdrawal: $ 500.0
```

```
Balance before applying interest: $ 2000.0
```

```
Balance after applying interest: $ 2100.0
```

```
Balance: $ 2100.0
```

```
Interest Rate: 0.05
```

```
In [3]:
```

IPython Console History