الاسم : محمد تيسير الراغب 2440

Q1 :

**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 }**

تحويل قائمتين الى قاموس

الكود

```
x = ['HTTP', 'HTTPS', 'FTP','DNS']
y = [80,443,21,53]
z= dict (zip(x,y))
print (z)
```

الخرج

```
IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/FX-tec/mm/untitled0.py', wdir='C:/Users/FX-tec/mm')
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}

In [2]:
```

**B: Write a Python program that calculates the factorial of a given number entered by user**

حساب عاملي رقم معين

الكود : باستخدام حلقة for يبدأ البرنامج بالقيمة ١ ثم يتكرر num

```
num = int (input("enter a number :"))
result=1
for i in range(1,num +1):
    result *= i
print (f"the factorial of {num} is {result}"
```

الخرج

```
In [1]: runfile('C:/Users/FX-tec/mm/untitled1.py', wdir='C:/Users/FX-tec/mm')

enter a number :4
the factorial of 4 is 24
```

**C : [‘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.**

الكود

```
7
8   L = ['NETWORK','BIO', 'PROGRAME','PHYSICS','MUSIC']
9   for x in L:
10      if x.startswith('B'):
11          print(x)
12
```

الخرج

```
In [2]: runfile('C:/Users/FX-tec/mm/untitled2.py', wdir='C:/Users/FX-tec/mm')
BIO

In [3]:
```

**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}**

الكود

```
6
7
8   d = {i: i +1 for i in range (11)}
9   print (d)
```

الخرج

```
In [3]: runfile('C:/Users/FX-tec/mm/untitled3.py', wdir='C:/Users/FX-tec/mm')
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}

In [4]:
```

**Q2 : 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**

```
7
8   b = list (input("input binary number :"))
9   x = 0
10  for i in range (len(b)):
11      digit = b.pop()
12     if digit == '1' :
13
14          x == x + pow(2, i)
15          print("the decimal value of number " , x)
16
17
18
```

**Q3: 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**

يتم تحديد الأسئلة والأجوبة مباشرة فيquestions القائمة داخلmain() الوظيفة.
تتكرر الوظيفةask_questions() على قائمة الأسئلة وتطرح على المستخدم كل سؤال. ثم يقوم بحساب درجة المستخدم بناءً على الإجابات المقدمة.
تظل الوظيفةsave_result() كما هي، حيث يتم حفظ اسم المستخدم ونتيجته وإجمالي الأسئلة التي تمت تجربتها في ملف CSV يسمى."results.csv"
يظل تدفق البرنامج مشابهًا للإصدار السابق، حيث يقوم المستخدم بإدخال اسمه، وإجراء الاختبار، ثم يتم عرض النتيجة وحفظها.
يلغي هذا الأسلوب الحاجة إلى ملفات خارجية ويبسط البرنامج عن طريق تحديد الأسئلة مباشرة داخل الكود.

```python
import csv

def ask_questions(questions):
    score = 0
    total_questions = len(questions)
    for i, (question, answer) in enumerate(questions, 1):
        print(f"Question {i}: {question}")
        user_answer = input("Your answer: ").strip()
        if user_answer.lower() == answer.lower():
            score += 1
    return score, total_questions

def save_result(user_name, score, total_questions, file_name):
    with open(file_name, 'a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow([user_name, score, total_questions])

def main():
    user_name = input("Enter your name: ")
    questions = [
        ("What is the largest country", "russia"),
        ("what color is the sea ?", "blue"),

    ]
    score, total_questions = ask_questions(questions)
    print(f"Your score: {score}/{total_questions}")
    save_result(user_name, score, total_questions, 'results.csv')

if __name__ == "__main__":
    main()
```

```
In [7]: runfile('C:/Users/FX-tec/mm/untitled5.py', wdir='C:/Users/FX-tec/mm')

Enter your name: mohammad
Question 1: What is the largest country

Your answer: russia
Question 2: what color is the sea ?

Your answer: blue
Your score: 2/2
```

Q4: **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**

```python

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"Deposited ${amount}. Current balance: ${self.balance}")

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            print(f"Withdrew ${amount}. Current balance: ${self.balance}")
        else:
            print("Insufficient funds.")

    def get_balance(self):
        return self.balance


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

    def apply_interest(self):
        interest = self.balance * (self.interest_rate / 100)
        self.balance += interest
        print(f"Interest applied. Current balance: ${self.balance}")

    def __str__(self):
        return f"Account holder: {self.account_holder}, Balance: ${self.balance}, Interest rate: {self.interest_rate}%"


bank_account = BankAccount("55555", "Mohammad")
bank_account.deposit(1000)
bank_account.withdraw(500)
print(f"Final balance: ${bank_account.get_balance()}")

savings_account = SavingsAccount("66666", "Ahmad", interest_rate=1.5)
savings_account.deposit(1000)
savings_account.apply_interest()
print(savings_account)

```

```
In [4]: runfile('C:/Users/FX-tec/mm/untitled8.py', wdir='C:/
Users/FX-tec/mm')
Deposited $1000. Current balance: $1000.0
Withdrew $500. Current balance: $500.0
Final balance: $500.0
Deposited $1000. Current balance: $1000.0
Interest applied. Current balance: $1015.0
Account holder: Ahmad, Balance: $1015.0, Interest rate: 1.5%
```