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

بشرى عقل سلامه 1519

السؤال 1:

A

```
L1 = ['HTTP', 'HTTPS', 'FTP', 'DNS']
L2 = [80, 443, 21, 53]
d = dict(zip(L1, L2))
print(d)
```

الخرج:

{HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53'}

B

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
while True:
    num = int(input("أدخل عددًا لحساب المضروب: "))
    print(f"مضروب {num} هو {factorial(num)}")
    s = input("Do you want to continue? (y/n) ").lower()
    if s == "n":
        break
```

الخرج:

```
أدخل عددًا لحساب المضروب: 5
مضروب 5 هو 120
Do you want to continue? (y/n) n

Process finished with exit code 0
```

C

```
L = ['Network', 'Bio', 'Programming', 'Physics', 'Music']
for i in L:
    if i.startswith('B'):
        print(i)
```

الخرج

```
Bio

Process finished with exit code 0
```

D

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

الخرج

{11 :10 ,10 :9 ,9 :8 ,8 :7 ,7 :6 ,6 :5 ,5 :4 ,4 :3 ,3 :2 ,2 :1 ,1 :0}

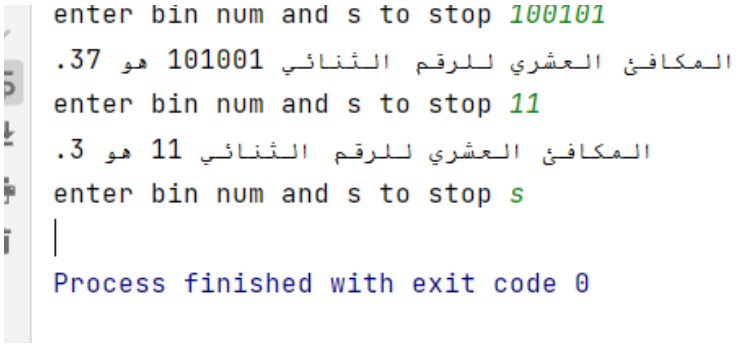
السؤال 2

```
while True:
    b = input("enter bin num and s to stop ").strip()
    if b=="s":
        break
    # صالح ثنائي رقم المدخلة السلسلة كانت إذا مما التحقق
    is_valid = True
    for char in b:
        if char not in ('0', '1'):
            is_valid = False
            break

    if is_valid:
        # عشري رقم إلى ثنائية سلسلة تحويل
        decimal_number = 0
        b = b[::-1]  # الحساب لتسهيل السلسلة عكس
        for i in range(len(b)):
            decimal_number += int(b[i]) * (2 ** i)

        print(f"الثنائي للرقم العشري المكافئ {b} هو {decimal_number}.")
    else:
        print("صالح غير إدخال يرجى .صالح ثنائي رقم إدخال (صحيح 0 و على فقط يحتوي
1).")
```

الخرج



السؤال 3

```python
import json
import os

questions_file = 'questions.json'
results_file = 'results.json'

# اقرأ الأسئلة والأجوبة من ملف JSON
with open(questions_file, 'r') as file:
    data = json.load(file)
    questions = [(item['question'], item['answer']) for item in data]

# اطرح الأسئلة واجمع الإجابات من المستخدم
username = input("أدخل اسم المستخدم: ").strip()
correct_answers = 0
for question, correct_answer in questions:
    user_answer = input(f"{question}? ").strip()
    if user_answer == correct_answer:
        correct_answers += 1
total_questions = len(questions)
print(f"{username}, لقد أجبت بشكل صحيح على {correct_answers} من أصل
{total_questions} سؤال.")
# احفظ اسم المستخدم والنتيجة في ملف بصيغة JSON
result = {
    "name": username,
    "score": correct_answers,
    "of": total_questions
}

if not os.path.isfile(results_file):
    with open(results_file, 'w') as file:
        json.dump([result], file)
else:
    try:
        with open(results_file, 'r+') as file:
            try:
                data = json.load(file)
            except json.JSONDecodeError:
                data = []
            data.append(result)
            file.seek(0)
            json.dump(data, file)
    except FileNotFoundError:
        with open(results_file, 'w') as file:
            json.dump([result], file)
```

الخرج

أدخل اسم المستخدم: bushra

What is 5 plus 3? 8

What is 7 minus 2? 5

What is 6 multiplied by 3? 18

What is 12 divided by 4? 3

What is 9 plus 1? 9

What is 8 minus 4? 2

What is 7 multiplied by 2? 14

What is 16 divided by 2? 8

What is 10 plus 5? 50

What is 11 minus 3? 8

What is 4 multiplied by 4? 16

What is 20 divided by 5?

What is 14 plus 6? 20

What is 18 minus 7? 11

What is 3 multiplied by 5? 15

What is 25 divided by 5? 6

What is 13 plus 7? 20

What is 19 minus 9? 10

What is 2 multiplied by 8? 16

What is 24 divided by 6? 4

لقد أجبت بشكل صحيح على 15 من أصل 20 سؤال ,bushra.

Process finished with exit code 0

results.json


```
[{"name": "bushra", "score": 15, "of": 20}]
```


questions.json


```
[
{"question": "What is 5 plus 3", "answer": "8"},
{"question": "What is 7 minus 2", "answer": "5"},
{"question": "What is 6 multiplied by 3", "answer": "18"},
{"question": "What is 12 divided by 4", "answer": "3"},
{"question": "What is 9 plus 1", "answer": "10"},
{"question": "What is 8 minus 4", "answer": "4"},
{"question": "What is 7 multiplied by 2", "answer": "14"},
{"question": "What is 16 divided by 2", "answer": "8"},
{"question": "What is 10 plus 5", "answer": "15"},
{"question": "What is 11 minus 3", "answer": "8"},
{"question": "What is 4 multiplied by 4", "answer": "16"},
{"question": "What is 20 divided by 5", "answer": "4"},
{"question": "What is 14 plus 6", "answer": "20"},
{"question": "What is 18 minus 7", "answer": "11"},
{"question": "What is 3 multiplied by 5", "answer": "15"},
{"question": "What is 25 divided by 5", "answer": "5"},
{"question": "What is 13 plus 7", "answer": "20"},
{"question": "What is 19 minus 9", "answer": "10"},
{"question": "What is 2 multiplied by 8", "answer": "16"},
{"question": "What is 24 divided by 6", "answer": "4"}
]
```

السؤال 4

```
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(f"Deposited: ${amount:.2f}")
        print(f"Balance: ${self.balance:.2f}")
    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient balance!")
        else:
            self.balance -= amount
            print(f"Withdrew: ${amount:.2f}")
            print(f"Current Balance: ${self.balance:.2f}")
    def get_balance(self):
        return self.balance
    def __str__(self):
        return f"Account Holder: {self.account_holder}, Balance:
${self.balance:.2f}"
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 / 100)
        self.balance += interest
        print(f"Interest: ${interest:.2f}")
        print(f"New Balance: ${self.balance:.2f}")

    def __str__(self):
        return f"Account Holder: {self.account_holder}, Balance:
${self.balance:.2f}, Interest Rate: {self.interest_rate}%"
bushra = BankAccount("1519", "bushra")
bushra.deposit(1000)
bushra.withdraw(500)
print(bushra)
b = SavingsAccount("1519", "bushra", 10)
b.deposit(150)
b.apply_interest()
print(b)
```

الخرج:

Deposited: $1000.00

Balance: $1000.00

Withdrew: $500.00

Current Balance: $500.00

Account Holder: bushra, Balance: $500.00

Deposited: $150.00

Balance: $150.00

Interest: $15.00

New Balance: $165.00

Account Holder: bushra, Balance: $165.00, Interest Rate: 10%

Process finished with exit code 0