

سالي محمد علي رزق

2466

Homework 1

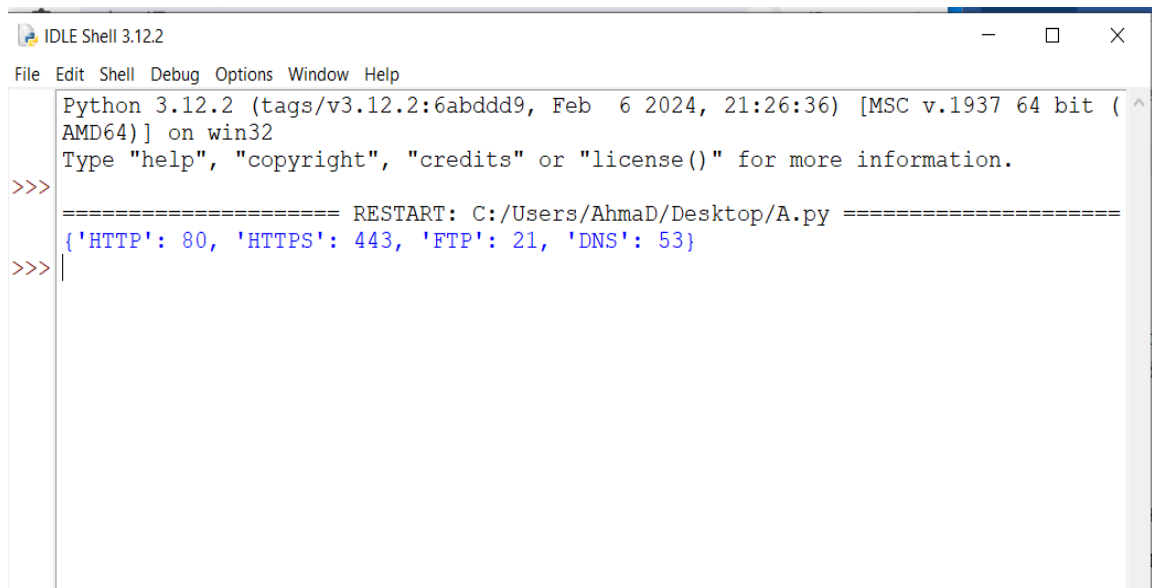
Q1:

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)

الكود:

```
L1 = ["HTTP", "HTTPS", "FTP", "DNS"]
L2 = [80, 443, 21, 53]
d = {L1[i]: L2[i] for i in range(len(L1))}
print(d)
```

الخرج:



```
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/AhmaD/Desktop/A.py =====
{'HTTP': 80, 'HTTPS': 443, 'FTP': 21, 'DNS': 53}
>>>
```

الشرح:

ندخل قائمتين L1 و L2 وننشئ القاموس d

نستخدم دالة range(len(L1)) لتوليد مؤشرات تتراوح من 0 إلى عدد العناصر في القائمة L1 لكل مؤشر i، نجعل L1[i] مفتاحاً في القاموس، و L2[i] القيمة المقابلة لهذا المفتاح.

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

الكود:

```
def calculate_factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * calculate_factorial(n-1)  
  
number = eval(input("أدخل عدداً لحساب العاملّي الخاص به "))  
f = calculate_factorial(number)  
print(f"{f} هو {number} العاملّي للعدد")
```

الخرج:

```
IDLE Shell 3.12.2  
Edit Shell Debug Options Window Help  
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
> = RESTART: C:/Users/Ahmad/Desktop/B.py  
أدخل عدداً لحساب العاملّي الخاص به : 0  
العاملّي للعدد 0 هو 1  
> ===== RESTART: C:/Users/Ahmad/Desktop/B.py =====  
أدخل عدداً لحساب العاملّي الخاص به : 1  
العاملّي للعدد 1 هو 1  
> ===== RESTART: C:/Users/Ahmad/Desktop/B.py =====  
أدخل عدداً لحساب العاملّي الخاص به : 4  
العاملّي للعدد 4 هو 24  
>
```

الشرح:

نعرف دالة calculate_factorial التي تأخذ n كعدد يدخله المستخدم
نستخدم الحالة الأساسية: إذا كان n يساوي 0 أو 1، نرجع 1 (لأن العاملّي لهما هو 1)

إذا كان n أكبر من 1، نستخدم الاستدعاء الذاتي للدالة مع $n-1$ ونضرب النتيجة في n للحصول على
العاملي نطلب من المستخدم إدخال عدد عبر استخدام دالة `input` ثم نطبع النتيجة.

```
C-L-["Network", "Bio", "Programming", "Physics", "Music1"]
```

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:

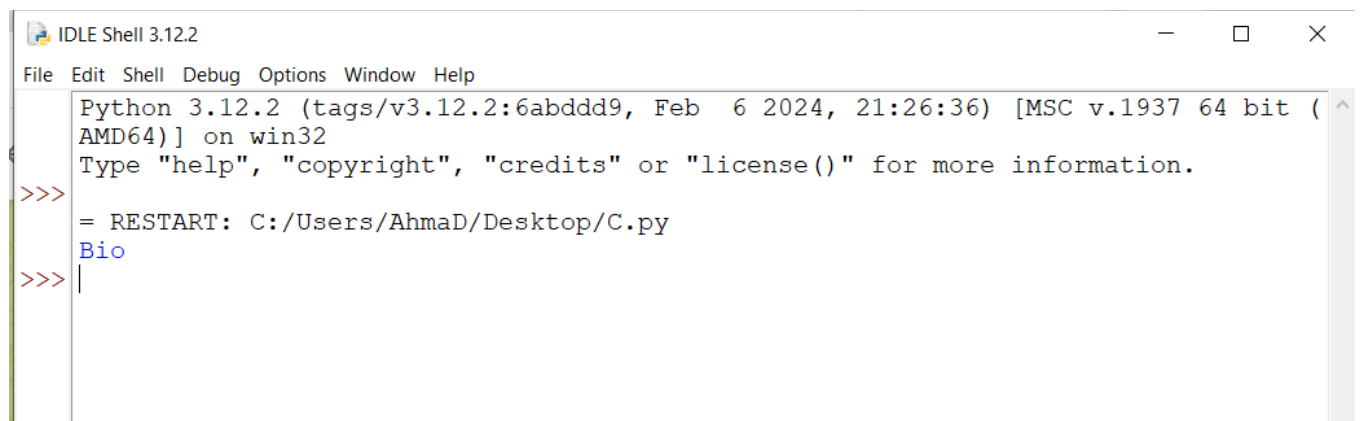
```
L = ["Network", "Bio", "Programming", "Physics", "Music"]
```

```
for item in L:
```

```
    if item.startswith('B'):
```

```
        print(item)
```

الخرج:



الشرح:

نستخدم حلقة `for` للتكرار عبر كل العناصر في القائمة مباشرةً، ونتحقق من كل عنصر باستخدام دالة `startswith()` لمعرفة إذا كان يبدأ بحرف 'B' أم لا.

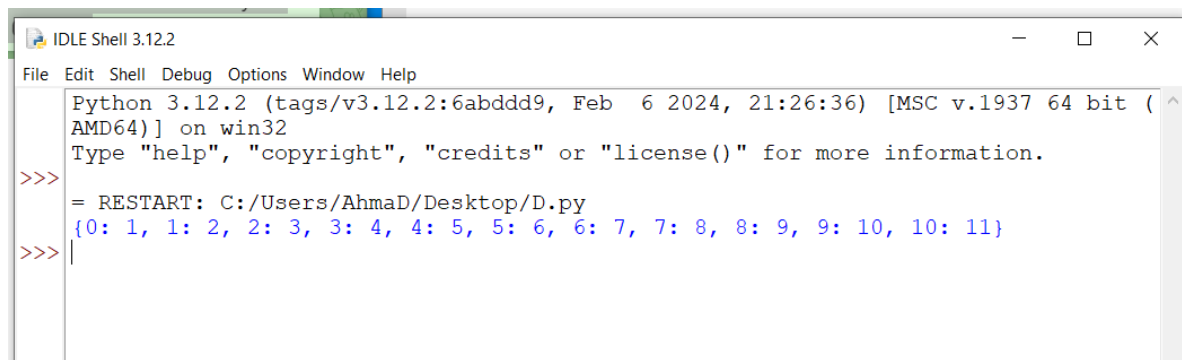
لم نستخدم دالة `len()` في هذا الكود لأننا لا نحتاج إلى معرفة طول القائمة لتنفيذ العملية المطلوبة.

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:

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

الخرج:



```
IDLE Shell 3.12.2  
File Edit Shell Debug Options Window Help  
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: C:/Users/Ahmad/Desktop/D.py  
{0: 1, 1: 2, 2: 3, 3: 4, 4: 5, 5: 6, 6: 7, 7: 8, 8: 9, 9: 10, 10: 11}  
>>>
```

الشرح:

كل مفتاح في القاموس `d` يمثل رقمًا من الصفر إلى العشرة، وكل قيمة مقابلة للمفتاح هي الرقم التالي له.

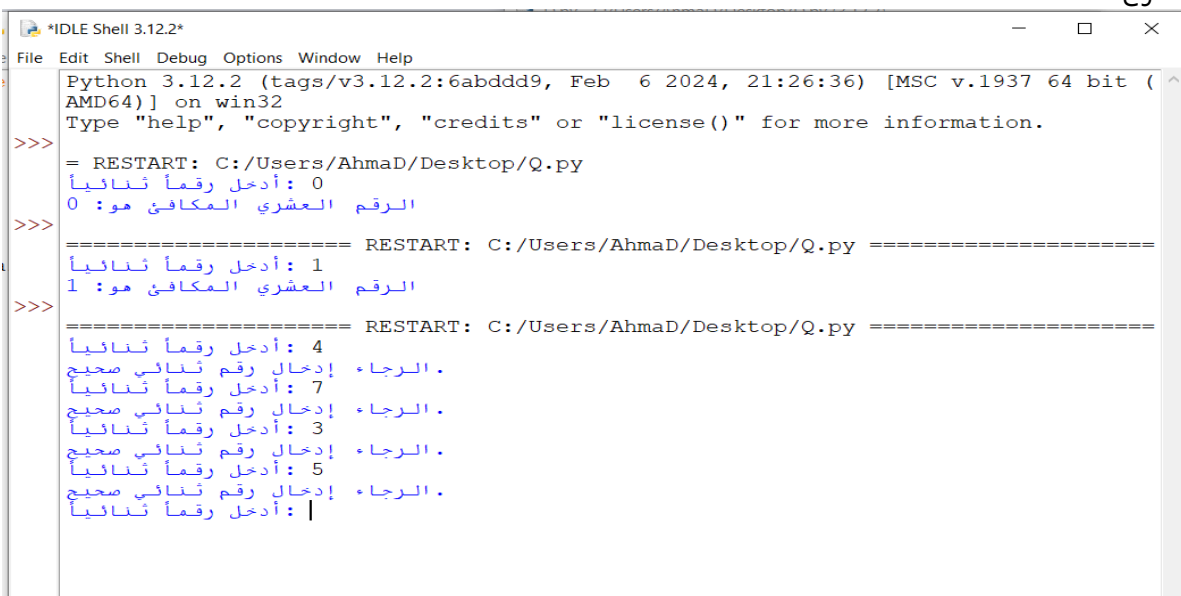
Q2:

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. Tip: solve input errors.

الكود:

```
def main():  
  
    while True:  
  
        binary_num = input("أدخل رقماً ثنائياً: ")  
  
        try:  
  
            decimal_num = int(binary_num, 2)  
  
            print(f"الرقم العشري المكافئ هو {decimal_num}")  
  
            break  
  
        except ValueError:  
  
            print("الرجاء إدخال رقم ثنائي صحيح.")  
  
main()
```

الخرج:



```
*IDLE Shell 3.12.2*  
File Edit Shell Debug Options Window Help  
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: C:/Users/Ahmad/Desktop/Q.py  
أدخل رقماً ثنائياً: 0  
الرقم العشري المكافئ هو: 0  
>>> ===== RESTART: C:/Users/Ahmad/Desktop/Q.py =====  
أدخل رقماً ثنائياً: 1  
الرقم العشري المكافئ هو: 1  
>>> ===== RESTART: C:/Users/Ahmad/Desktop/Q.py =====  
أدخل رقماً ثنائياً: 4  
الرجاء إدخال رقم ثنائي صحيح.  
أدخل رقماً ثنائياً: 7  
الرجاء إدخال رقم ثنائي صحيح.  
أدخل رقماً ثنائياً: 3  
الرجاء إدخال رقم ثنائي صحيح.  
أدخل رقماً ثنائياً: 5  
الرجاء إدخال رقم ثنائي صحيح.  
أدخل رقماً ثنائياً: 7  
الرجاء إدخال رقم ثنائي صحيح.  
أدخل رقماً ثنائياً: 7
```

الشرح:

نستخدم حلقة while True لتكرار قراءة الرقم الثنائي من المستخدم حتى يتم إدخال قيمة صحيحة .
ونستخدم دالة int() مع القاعدة 2 لتحويل الرقم الثنائي الذي تم إدخاله إلى رقم عشري
إذا كان الإدخال غير صالح (أي يحتوي على أحرف غير 0 و 1)، سيتم إظهار رسالة خطأ ValueError.
نستخدم try...except لمعالجة الاستثناء (الخطأ) وإعلام المستخدم بأن الإدخال غير صالح وطلب
إدخال جديد .
وعندما يتم إدخال رقم ثنائي صحيح، نطبع الرقم العشري المكافئ ونخرج من الحلقة.

Q3:

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.

الكود:

```
questions = []
with open('questions.txt') as f:
    lines = f.read()
    lines = lines.split("\n")

for line in lines:
    questions.append(line.split('='))

name = input('Please, Enter Your name: ')
mark = 0

print('Please, Ask for this questions..')

for q in questions:
    answer = input('{} = '.format(q[0]))
    if answer==q[1]:
        mark+=1

print("Thank you, your mark is: {}".format(mark))

with open('results.txt', 'a') as f:
    f.write('{}: {}\n'.format(name,mark))
```

الخرج:

```
Q3.py - C:\Users\AhmaD\Desktop\MM\Q3.py (3.12.2)
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\AhmaD\Desktop\MM\Q3.py =====
Please, Enter Your name: SALY
Please, Ask for this questions..
10*1 = 1
10*2 = 2
10*3 = 3
10*4 = 4
10*5 = 5
10*6 = 6
10*7 = 7
10*8 = 8
10*9 = 9
10*10 = 100
0*1 = 0
0*2 = 0
0*3 = 0
0*4 = 0
0*5 = 0
0*6 = 0
0*7 = 0
0*8 = 00
0*9 = 0
0*10 = 0
Thank you, your mark is: 10
>>>
```

المفكرة - results

ملف تحرير تنسيق عرض تعليمات

SALY: 10

المفكرة - questions

ملف تحرير تنسيق عرض تعليمات

10=1*10
20=2*10
30=3*10
40=4*10
50=5*10
60=6*10
70=7*10
80=8*10
90=9*10
100=10*10
0=1*0
0=2*0
0=3*0
0=4*0
0=5*0
0=6*0
0=7*0
0=8*0
0=9*0
0=10*0

الشرح:

تم تخزين الأسئلة في ملف نصي على الشكل التالي:

كل سطر فيه السؤال مع الإجابة يفصل بينهما إشارة =

يتم قراءة هذا الملف ثم تقسيم المحتوى إلى أسطر عن طريق التابع split

ثم يتم تقسيم كل سطر إلى جزئين : السؤال والإجابة الصحيحة وتخزينهم في قائمة هي questions

يتم إدخال اسم الطالب وعرض الأسئلة عليه واحد تلو الآخر باستخدام حلقة

وفي كل مرة نطابق إجابة الطالب مع الإجابة الصحيحة في حال التطابق يكسب الطالب علامة

أخيرا نقوم بإضافة اسم الطالب وعلامته إلى نهاية الملف results.txt

Q4:

Object-Oriented Programing 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_halance() 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 Savings Account that inherits from Bank Account and adds interesta inte 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 Savings Account, and call apply_interest() and print() functions

الكود:

```
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}. New balance: ${self.balance}.")

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

def get_balance(self):
    return self.balance

def __str__(self):
    return f"Account holder: {self.account_holder}, Account number: {self.account_number}, Balance: ${self.balance}"

#Subclass SavingsAccount that inherits from BankAccount
class SavingsAccount(BankAccount):
    def __init__(self, account_number, account_holder, interest_rate, balance=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.deposit(interest)

```

```
def __str__(self):  
    return super().__str__() + f", Interest rate: {self.interest_rate}"%
```

#Create an instance of BankAccount

```
bank_account = BankAccount("0000000000", "SALY REZK")
```

```
bank_account.deposit(1000)
```

```
bank_account.withdraw(500)
```

```
print(bank_account)
```

#Create an instance of SavingsAccount

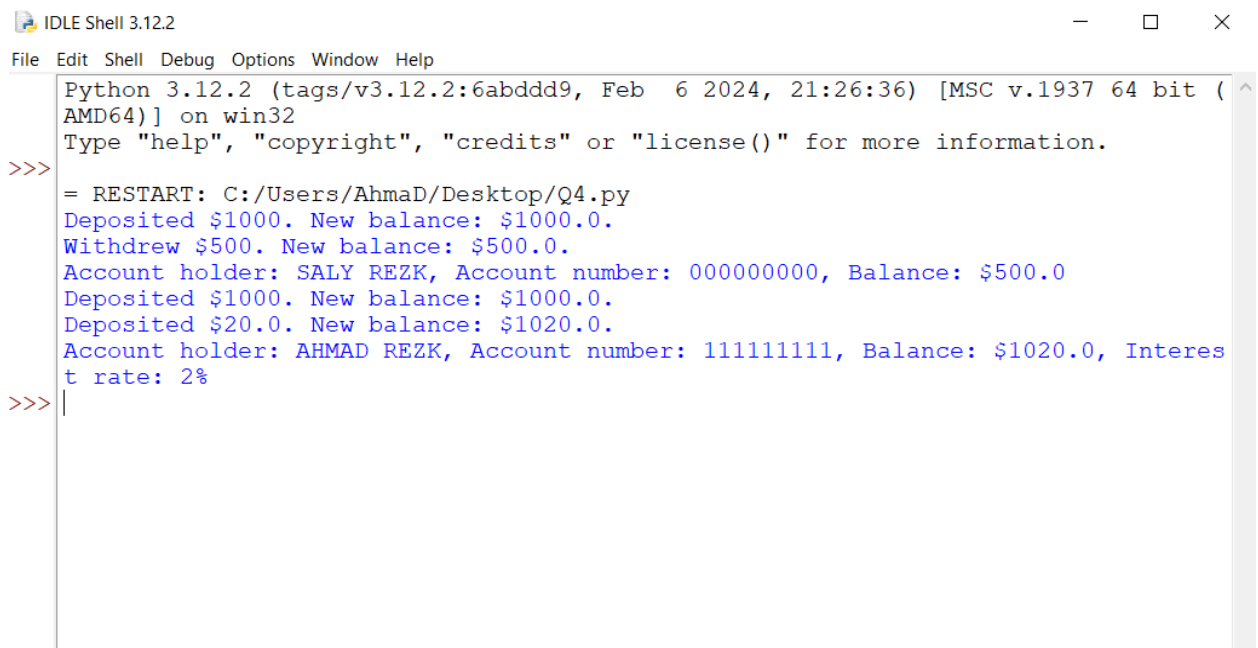
```
savings_account = SavingsAccount("1111111111", "AHMAD REZK", interest_rate=2)
```

```
savings_account.deposit(1000)
```

```
savings_account.apply_interest()
```

```
print(savings_account)
```

الخرج:



```
IDLE Shell 3.12.2  
File Edit Shell Debug Options Window Help  
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> = RESTART: C:/Users/Ahmad/Desktop/Q4.py  
Deposited $1000. New balance: $1000.0.  
Withdrew $500. New balance: $500.0.  
Account holder: SALY REZK, Account number: 0000000000, Balance: $500.0  
Deposited $1000. New balance: $1000.0.  
Deposited $20.0. New balance: $1020.0.  
Account holder: AHMAD REZK, Account number: 1111111111, Balance: $1020.0, Interest rate: 2%  
>>> |
```

الشرح:

نعرف صنف `BankAccount` بالخصائص والطرق المطلوبة

نعرف صنف `SavingsAccount` الذي يرث من `BankAccount` ويضيف خاصية `interest_rate` وطريقة `apply_interest`.

نعدل طريقة `__str__` في كلا الصنفين لتوفير تمثيل نصي للحساب ثم ننشئ مثلاً على `BankAccount`، نُؤدي عمليات إيداع وسحب، ونطبع رصيد الحساب ثم ننشئ مثلاً على `SavingsAccount`، نُؤدي عملية إيداع، نطبق الفائدة، ونطبع تفاصيل الحساب بما في ذلك نسبة الفائدة

عند تشغيل هذا الكود، سيتم محاكاة عمليات الإيداع والسحب على كائن `BankAccount` وتطبيق الفائدة على كائن `SavingsAccount`، وطباعة النتائج إلى وحدة التحكم.