III

Debugging

Finding and fixing errors in your code

✕ Syntax Errors

غالبًا يكون السبب خطأ مطبعي مثل نسيان النقطتين (:)، أو مسافات بادئة (indentation) غير صحيحة.

Algorithm 1 — Missing Colon in if/def

✕ Buggy:

```
def greet(name)
    print(name)
```

```
ERROR!Traceback (most recent call
last):  File "<main.py>", line 1
def greet(name)
^SyntaxError: expected ':'
```

"خطأ في الصياغة: كان من المفترض وجود النقطتين (:)"

Algorithm 2 — Wrong Indentation

✕ Buggy:

```
def add(a, b):
print(a, b)
   return a + b
```

```
ERROR!Traceback (most recent call
last):  File "<main.py>", line 2
print(a, b)
^IndentationError: expected an
indented block after function
definition on line 1
```

"خطأ في المسافات: كان من المتوقع وجود كتلة كود مُزاحة (indentation) بعد تعريف الدالة في السطر 1"

⚡ Runtime Errors

يبدو البرنامج صحيح لكنه يتعطل أثناء التنفيذ. مثال: القسمة على صفر أو استخدام متغير غير موجود.

Algorithm 1 — ZeroDivisionError

❌ **Buggy:**

```
x = 10
y = 0
result = x / y
```

```
ERROR!Traceback (most recent call last):  File "<main.py>", line 3, in <module>
ZeroDivisionError: division by zero
```

Algorithm 2 — NameError (undefined variable)

❌ **Buggy:**

```
def calc():
    total = price * qty
    return total
Calc()
```

```
ERROR!Traceback (most recent call last):  File "<main.py>", line 4, in <module>  File "<main.py>", line 2, in calcNameError: name 'price' is not defined. Did you mean: 'print'?
```

Algorithm 3 — IndexError (list out of range)

❌ **Buggy:**

```
Word= "Hello"
print(word[7])
```

```
ERROR!Traceback (most recent call last):  File "<main.py>", line 2, in <module>
IndexError: list index out of range
```

? Logic Errors

يعمل البرنامج بدون أن يتعطل، لكنه يعطي نتيجة خاطئة. هذا هو أصعب نوع لاكتشافه!

الحل: إضافة تعليمات **print()** مؤقتة لمعرفة القيم التي تأخذها المتغيرات في مراحل مختلفة من التنفيذ.

Algorithm 1 — Wrong Average Calculation

✗ **Buggy:**

```
def average(a, b):
    sum=a * b
    return sum/2
```

✓ **print() steps:**

```
print(sum)
```

✓ **solution:**

```
Sum = a+b
```

Algorithm 2 — Off-by-One in Loop

✗ **Buggy:**

```
for i in range(1, 5):
    total= total+i
```

✓ **print() steps:**

```
print(i)
```

✓ **solution:**

```
For i in range(1,6):
```

Algorithm 3 — Wrong Conditional Direction

✗ **Buggy:**

```
if score > 50:
    grade = "Fail"
else:
    grade = "Pass"
```

✓ **print() steps:**

```
print("score:", score)
```

```
print("grade assigned:", grade)
```

❗ *After debugging: remove or comment out your print() lines!*

Exercices:

SYNTAX : missing :

```
def calculate(width, height)
        area = width * height
        perimeter = width + height
        print("Area:", area)
        print("Perimeter:", perimeter)
calculate(5, 3)
```

LOGIC: should be 2*(w+h)

```
def get_price(age):
        if age < 12:
                return 500
        else:
                return 200

age = input("Enter age: ")
price = get_price(age)
print("Ticket price:" price)
```

LOGIC: condition is backwards

RUNTIME : input() returns string

SYNTAX : missing comma ,

Exercices:

```
# Countdown from 5 to 0
def countdown():
        for i in range(5, 0):
            print(i)
            print("Go!")
countdown(
```

LOGIC: final value = -1, step -1

SYNTAX : missing closing)