LAB ASSIGNMENT – 7.3

NAME : M.SHIVA

HALL.NO : 2403A52377

BATCH.NO : AI 14

PLATFORM USED : GOOGLE COLLAB AND perplexity.ai

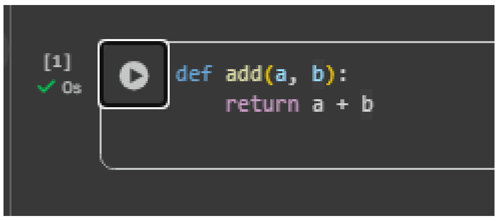
PROMPT 01 :

def add(a, b):

    return a + b

correct this code by fixing the syntax :

OUTPUT :



```
[1]
✓ 0s
def add(a, b):
    return a + b
```

CODE CORRECTION EXPLANATION :

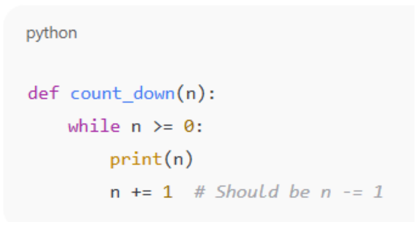
- Add a colon after the function header to start the code block: def add(a, b): is the correct header.
- Indent the next line to form the function body; the return statement must be inside the block.
- Final corrected code:

python

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

- Missing the colon causes a SyntaxError; wrong or missing indentation triggers an IndentationError.

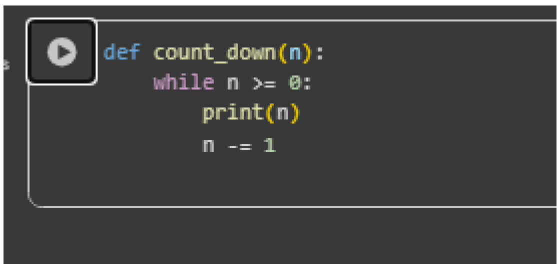
PROMPT 02 :



```
python

def count_down(n):
    while n >= 0:
        print(n)
        n += 1  # Should be n -= 1
```

Identify and fix a logic error in a loop that causes infinite iteration.

OUTPUT :

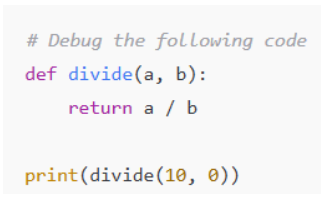


```
def count_down(n):
    while n >= 0:
        print(n)
        n -= 1
```

CODE CORRECTION EXPLANATION :

- The loop condition is n >= 0, so n must move downward each iteration to eventually become less than 0 and stop.
- The original code used n += 1, which increases n and keeps the condition true, causing an infinite loop.
- Replace n += 1 with n -= 1 so n decreases toward −1 and the loop terminates correctly.

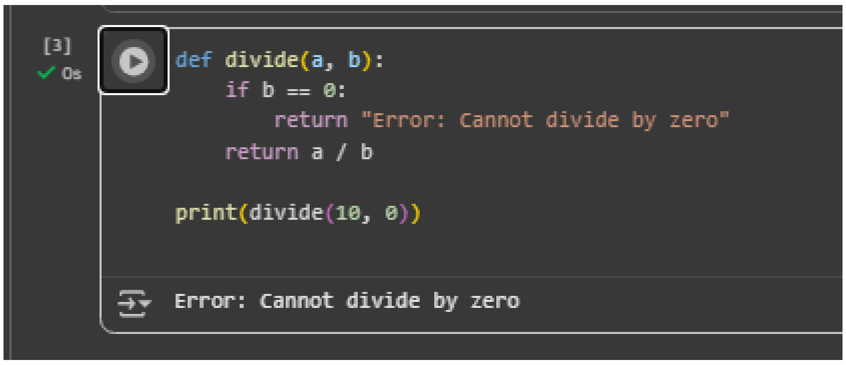
PROMPT 03 :



```
# Debug the following code
def divide(a, b):
    return a / b

print(divide(10, 0))
```

fix the Debug a runtime error caused by division by zero

OUTPUT :



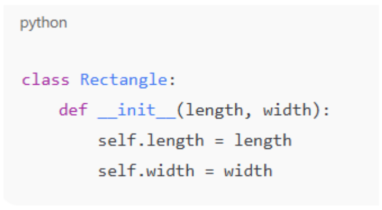
```
[3]
 0s
def divide(a, b):
    if b == 0:
        return "Error: Cannot divide by zero"
    return a / b

print(divide(10, 0))
```

```
Error: Cannot divide by zero
```

CODE CORRECTION EXPLANATION :

- Division by zero is undefined in Python and raises ZeroDivisionError at runtime.
- Fix by validating input first: if b == 0, handle it and skip division.
- Alternative: wrap the operation in try/except to catch ZeroDivisionError and recover gracefully.

PROMPT 04 :

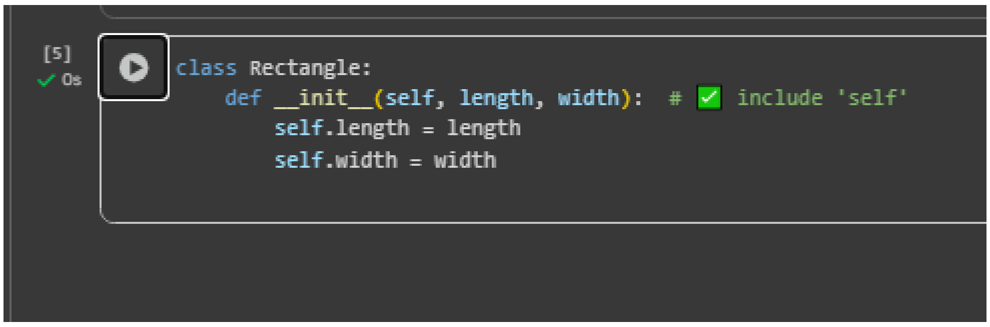


python

```python
class Rectangle:
    def __init__(length, width):
        self.length = length
        self.width = width
```

YOU NEED TO Provide a faulty class definition (missing self in parameters)

OUTPUT :

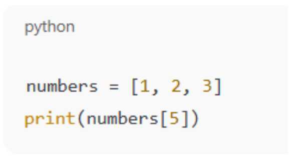


```
[5]
✓ 0s
class Rectangle:
    def __init__(self, length, width):  # ✅ include 'self'
        self.length = length
        self.width = width
```

CODE CORRECTION EXPLANATION :

- The constructor was missing the first parameter self, which is the reference to the instance Python automatically passes to instance methods.
- Without self, calling Rectangle(5, 3) raises “TypeError: init() missing 1 required positional argument: 'self'”.
- Fix by adding self as the first parameter and using it to set attributes:

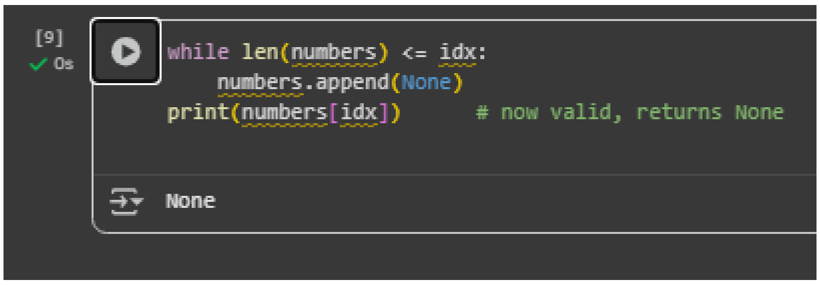
PROMPT 05 :



```python
numbers = [1, 2, 3]
print(numbers[5])
```

resolve the Index Error Access an invalid list index.

OUTPUT :



```
[9]
✓ 0s
while len(numbers) <= idx:
    numbers.append(None)
print(numbers[idx])      # now valid, returns None
```

```
None
```

CODE CORRECTION EXPLANATION :

- Valid indices for are 0, 1, 2; 5 is out of range.
- Fix by choosing an existing index or by checking bounds with len before access.
- When adding elements, use append or insert instead of assigning to a non-existent index.