

ASSIGNMENT 0.1

The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for "palindrome 1.py", "reverse string 3.py", "calculator 4.py", and "sequence 2.py". Below the tabs is the code for "palindrome 1.py". The code defines a function `is_palindrome` that checks if a string is a palindrome by removing spaces, converting it to lowercase, and comparing it to its reverse. It includes example usage and prints "The string is a palindrome." if the input is "level".

```
# function to check if a string is a valid palindrome
def is_palindrome(s):
    # Remove spaces and convert to lowercase
    s = s.replace(" ", "").lower()
    # Check if the string is equal to its reverse
    return s == s[::-1]

# Example usage
string = input("Enter a string: ")
if is_palindrome(string):
    print("The string is a palindrome.")
else:
    print("The string is not a palindrome.")

PS C:\Users\nered\OneDrive\Desktop\wtml> & C:/Users/nered/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nered/OneDrive/Desktop/wtml/palindrome 1.py"
Enter a string: level
The string is a palindrome.
PS C:\Users\nered\OneDrive\Desktop\wtml>
```

A screenshot of a code editor window. At the top, there are tabs for five files: palindrome 1.py, reverse string 3.py, calculator 4.py, sequence 2.py, and sequence 2.py (the active tab). Below the tabs is the code for sequence 2.py:

```
palindrome 1.py • reverse string 3.py • calculator 4.py • sequence 2.py •
sequence 2.py > ...
1 # Function to return the Fibonacci sequence up to n terms
2 def fibonacci_sequence(n):
3     """
4         Returns a list containing the Fibonacci sequence up to n terms.
5     """
6     sequence = []
7     a, b = 0, 1
8     for _ in range(n):
9         sequence.append(a)
10        a, b = b, a + b
11    return sequence
12
13 # Example usage
14 num_terms = int(input("Enter the number of terms: "))
15 print("Fibonacci sequence:", fibonacci_sequence(num_terms))
```

The code defines a function `fibonacci_sequence` that takes an integer `n` and returns a list of the first `n` terms of the Fibonacci sequence. It uses a simple iterative approach with two variables `a` and `b`. The terminal below shows the script being run and outputting the sequence for 2 terms.

TERMINAL

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
PS C:\Users\nered\OneDrive\Desktop\wtml> & C:/Users/nered/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nered/OneDrive/Desktop/wtml/sequence 2.py"
Enter the number of terms: 2
Fibonacci sequence: [0, 1]
PS C:\Users\nered\OneDrive\Desktop\wtml>
```

A screenshot of a code editor window. At the top, there are tabs for five files: palindrome 1.py, reverse string 3.py, calculator 4.py, sequence 2.py, and reverse string 3.py (the active tab). Below the tabs is the code for reverse string 3.py:

```
palindrome 1.py • reverse string 3.py • calculator 4.py • sequence 2.py •
reverse string 3.py > ...
1 # Function to reverse a string
2 def reverse_string(s):
3     return s[::-1]
4
5 # Example usage
6 input_str = input("Enter a string: ")
7 print("reversed string:", reverse_string(input_str))
```

The code defines a function `reverse_string` that takes a string `s` and returns its reverse using slicing. It then prompts the user for a string and prints the reversed version. The terminal below shows the script being run and outputting the reversed string "olleh" for the input "hello".

TERMINAL

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
PS C:\Users\nered\OneDrive\Desktop\wtml> & C:/Users/nered/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nered/OneDrive/Desktop/wtml/reverse string 3.py"
Enter a string: hello
reversed string: olleh
PS C:\Users\nered\OneDrive\Desktop\wtml>
```

```
palindrome 1.py ● reverse string 3.py ● calculator 4.py ● sequence 2.py ●
calculator 4.py > ...
1  # Program to simulate a basic calculator
2
3  def add(x, y):
4      return x + y
5
6  def subtract(x, y):
7      return x - y
8
9  def multiply(x, y):
10     return x * y
11
12 def divide(x, y):
13     if y == 0:
14         return "Error! Division by zero."
15     return x / y
16
17 print("Select operation:")
18 print("1. Add")
19 print("2. Subtract")
20 print("3. Multiply")
21 print("4. Divide")
22
23 choice = input("Enter choice (1/2/3/4): ")
24
25 num1 = float(input("Enter first number: "))
26 num2 = float(input("Enter second number: "))
27
28 if choice == '1':
29     print("Result:", add(num1, num2))
30 elif choice == '2':
31     print("Result:", subtract(num1, num2))
32 elif choice == '3':
33     print("Result:", multiply(num1, num2))
34 elif choice == '4':
35     print("Result:", divide(num1, num2))
36 else:
37     print("invalid input")
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\nered\OneDrive\Desktop\wtml> & C:/Users/nered/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/nered/OneDrive/Desktop/wtml/calculator 4.py"
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 3
Enter first number: 2
Enter second number: 4
Result: 8.0
PS C:\Users\nered\OneDrive\Desktop\wtml>
```