

The screenshot shows a dark-themed interface of the Visual Studio Code (VS Code) code editor. At the top, there are tabs for several files: 'palindrome 1.py' (which is the active tab), 'reverse string 3.py', 'calculator 4.py', and 'sequence 2.py'. Below the tabs, the code for 'palindrome 1.py' is displayed:

```
1 # function to check if a string is a valid palindrome
2 def is_palindrome(s):
3     # Remove spaces and convert to lowercase
4     s = s.replace(" ", "").lower()
5     # Check if the string is equal to its reverse
6     return s == s[::-1]
7
8 # Example usage
9 string = input("Enter a string: ")
10 if is_palindrome(string):
11     print("The string is a palindrome.")
12 else:
13     print("The string is not a palindrome.")
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined, indicating it is active), and PORTS. To the right of the navigation bar is a terminal window showing the execution of the script:

```
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 interface, likely Visual Studio Code, showing a Python script named `sequence 2.py`. The script defines a function `fibonacci_sequence(n)` that generates a list of Fibonacci numbers up to `n` terms. It uses a loop to calculate each term by summing the previous two. An example usage is demonstrated in the terminal below, where the user enters the number of terms and the script prints the resulting sequence.

```
# Function to return the Fibonacci sequence up to n terms
def fibonacci_sequence(n):
    """
    Returns a list containing the Fibonacci sequence up to n terms.
    """
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence

# Example usage
num_terms = int(input("Enter the number of terms: "))
print("Fibonacci sequence:", fibonacci_sequence(num_terms))
```

TERMINAL

```
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 interface, likely Visual Studio Code, showing a Python script named `reverse string 3.py`. The script defines a function `reverse_string(s)` that reverses a given string using slicing. An example usage is demonstrated in the terminal below, where the user enters a string and the script prints its reverse.

```
# Function to reverse a string
def reverse_string(s):
    return s[::-1]

# Example usage
input_str = input("Enter a string: ")
print("reversed string:", reverse_string(input_str))
```

TERMINAL

```
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 Python + ▾ ⌂ ⌚ ⌚ ×
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>
```