

AI assistant 1.2 assignment

Name:balaji

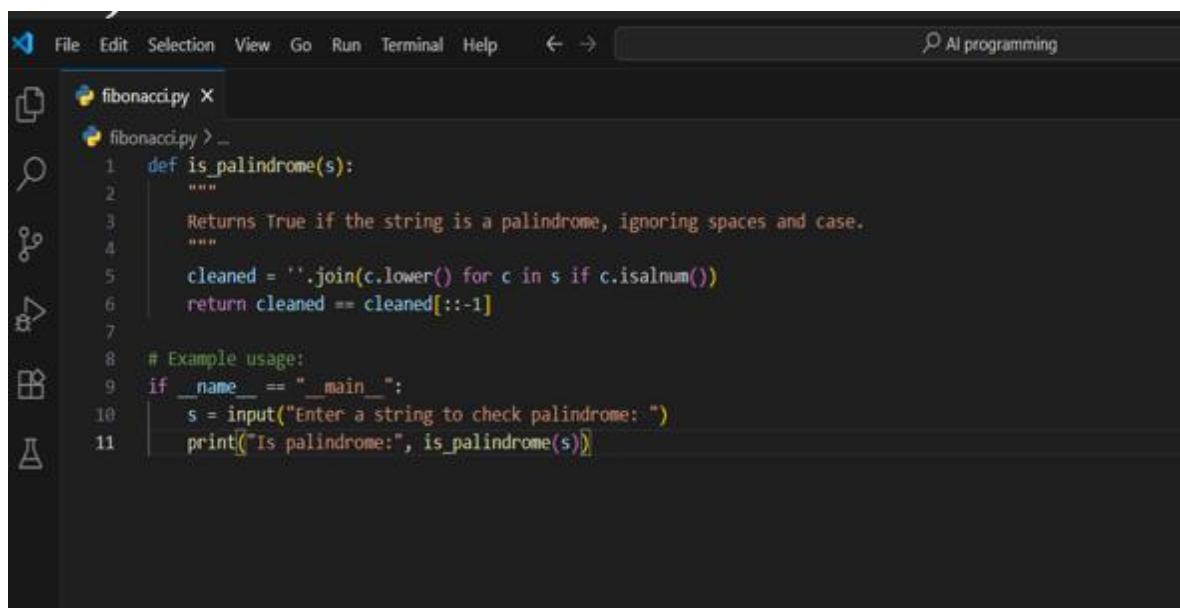
Roll no:2403A51102

Batch:06

task Description#1

- Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

Code:

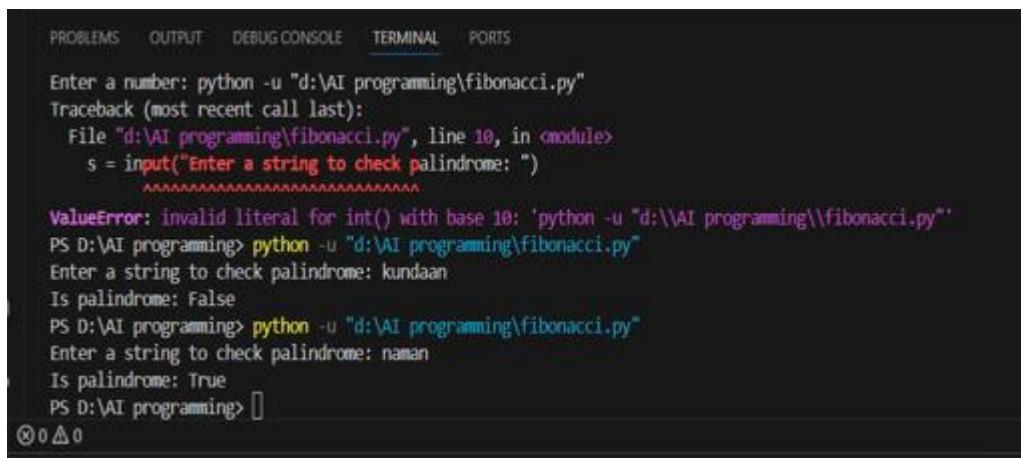


The screenshot shows a code editor window with the following Python code:

```
fibonaccipy X
fibonacci.py > _
1 def is_palindrome(s):
2     """
3         Returns True if the string is a palindrome, ignoring spaces and case.
4     """
5     cleaned = ''.join(c.lower() for c in s if c.isalnum())
6     return cleaned == cleaned[::-1]
7
8 # Example usage:
9 if __name__ == "__main__":
10     s = input("Enter a string to check palindrome: ")
11     print("Is palindrome:", is_palindrome(s))
```

The code defines a function `is_palindrome` that takes a string `s` and returns `True` if it is a palindrome, ignoring spaces and case. It uses a list comprehension to clean the string by removing non-alphanumeric characters and then compares it to its reverse. The code also includes a main block that prompts the user for input and prints the result of the palindrome check.

Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Enter a number: python -u "d:\AI programming\fibonacci.py"
Traceback (most recent call last):
  File "d:\AI programming\fibonacci.py", line 10, in <module>
    s = input("Enter a string to check palindrome: ")
    ~~~~~
ValueError: invalid literal for int() with base 10: 'python -u "d:\AI programming\fibonacci.py"'
PS D:\AI programming> python -u "d:\AI programming\fibonacci.py"
Enter a string to check palindrome: kundaan
Is palindrome: False
PS D:\AI programming> python -u "d:\AI programming\fibonacci.py"
Enter a string to check palindrome: naman
Is palindrome: True
PS D:\AI programming> []
```

Observation

The function first removes all spaces from the input string and converts it to lowercase to ensure the comparison is case-insensitive and whitespace-insensitive. It then compares the cleaned string to its reverse to determine if it's a palindrome.

Task:2

- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring

Prompt: Python function that returns the Fibonacci sequence up to n terms

Code:

```
fibonacipy.py
1 #!/usr/bin/python
2
3 def fibonacci_sequence(n):
4     """
5         Returns a list containing the Fibonacci sequence up to n terms.
6     """
7     sequence = []
8     a, b = 0, 1
9     for _ in range(n):
10        sequence.append(a)
11        a, b = b, a+b
12
13    return sequence
14
15 # Example usage:
16 if __name__ == "__main__":
17     n = int(input("Enter the number of terms: "))
18     print("Fibonacci sequence:", fibonacci_sequence(n))

fibonacci.py
1
2
3 def fibonacci_sequence(n):
4     """
5         Returns a list containing the Fibonacci sequence up to n terms.
6     """
7     sequence = []
8     a, b = 0, 1
9     for _ in range(n):
10        sequence.append(a)
11        a, b = b, a+b
12
13    return sequence
14
15 # Example usage:
16 if __name__ == "__main__":
17     n = int(input("Enter the number of terms: "))
18     print("Fibonacci sequence:", fibonacci_sequence(n))

PS D:\AI programming> python -m d:\AI programming\fibonacipy.py
Enter the number of terms: 9
Fibonacci sequence: [0, 1, 1, 2, 3, 5, 8, 13, 21]
PS D:\AI programming>
```

Output:

```
PS D:\AI programming> python -m d:\AI programming\fibonacipy.py
Enter the number of terms: 9
Fibonacci sequence: [0, 1, 1, 2, 3, 5, 8, 13, 21]
PS D:\AI programming>
```

Observation:

The Python program successfully generates the Fibonacci sequence up to a user-defined number of terms using an iterative approach.

- The function `fibonacci_sequence(n)` takes an integer input `n` and returns a list containing the first `n` terms of the Fibonacci sequence.

- The variables **a** and **b** are initialized to 0 and 1, representing the first two terms of the Fibonacci sequence.
- A for loop is used to generate the sequence:
 - In each iteration, the current value of **a** is appended to the list.
 - The values of **a** and **b** are then updated as **a = b** and **b = a + b**, ensuring the next term follows the Fibonacci pattern.
- The program uses **input()** to take the number of terms from the user, converts it to an integer using **int()**, and prints the resulting sequence using **print()**.

Task:3

- Write a comment like # Function to reverse a string and use Copilot to generate the function.

Prompt:write a python code to reverse the given string.make code simple

Code:

```
fibonacipy.py
1 def reverse_string(s):
2     """
3         Returns the reversed version of the input string.
4     """
5     return s[::-1]
6
7 # Example usage:
8 if __name__ == "__main__":
9     s = input("Enter a string to reverse: ")
10    print("Reversed string:", reverse_string(s))

fibonaccil.py
1 def fibonacil(n):
2     """
3         Returns the Fibonacci sequence up to n terms.
4     """
5     sequence = [0, 1]
6     for i in range(2, n):
7         sequence.append(sequence[-1] + sequence[-2])
8
9     return sequence
10
11 # Example usage:
12 if __name__ == "__main__":
13     n = int(input("Enter the number of terms: "))
14     print("Fibonacci sequence: ", fibonacil(n))

PS D:\VIT programming> python -u "D:\VIT programming\fibonacipy.py"
Enter the number of terms: 9
Fibonacci sequence: [0, 1, 1, 2, 3, 5, 8, 13, 21]
PS D:\VIT programming> python -u "D:\VIT programming\fibonaccil.py"
Enter a string to reverse: kundan
Reversed string: naduk
PS D:\VIT programming> 
```

Output:

```
PS D:\VIT programming> python -u "D:\VIT programming\fibonaccil.py"
Enter the number of terms: 9
Fibonacci sequence: [0, 1, 1, 2, 3, 5, 8, 13, 21]
PS D:\VIT programming> 
```

Observation:

Observation (Theory)

The Python program is designed to reverse a user-input string and display the result.

- The function `reverse_string(s)` takes a string `s` as input and returns its reversed form using Python's slicing technique:

- `return s[::-1]`
 - The slice `[::-1]` effectively reverses the string by starting from the end and stepping backwards.
- The program prompts the user to enter a string using the `input()` function.
- The entered string is passed to the `reverse_string()` function.
- The reversed string is then displayed using the `print()` function.

Task:4

- Generate a program that simulates a basic calculator (add, subtract, multiply, divide).
Write the comment: # Simple calculator with 4 operations and let AI complete it

Prompt:make a python code to make calculator that perform a basic calculator (add, subtract, multiply, divide).

Code:

```
File Edit Selection View Go Run Terminal Help < - > AI programming Fibonacci X Fibonacci ?_
25 if choice in ('1', '2', '3', '4'):
26     try:
27         num1 = float(input("Enter first number: "))
28         num2 = float(input("Enter second number: "))
29     except ValueError:
30         print("Invalid input! Please enter numbers.")
31     else:
32         if choice == '1':
33             print("Result:", add(num1, num2))
34         elif choice == '2':
35             print("Result:", subtract(num1, num2))
36         elif choice == '3':
37             print("Result:", multiply(num1, num2))
38         elif choice == '4':
39             print("Result:", divide(num1, num2))
40         else:
41             print("Invalid choice!")

# Basic calculator simulation
print("Basic calculator")
print("Select operation:")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")
choice = input("Enter choice (1/2/3/4):")
if choice in ('1', '2', '3', '4'):
    try:
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
    except ValueError:
        print("Invalid input! Please enter numbers.")
    else:
        if choice == '1':
            print("Result:", add(num1, num2))
        elif choice == '2':
            print("Result:", subtract(num1, num2))
        elif choice == '3':
            print("Result:", multiply(num1, num2))
        elif choice == '4':
            print("Result:", divide(num1, num2))
        else:
            print("Invalid choice!")

Similar code found with 1 license type - View matches
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POSIX

Reversed string: nadnuk
PS D:\AI programming> python -u "D:\AI programming\Fibonacci.py"
Basic calculator
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 4
Enter first number: 234
Enter second number:23
Result: 10.17993043478262
PS D:\AI programming>

Ln 41, Col 29 Spaces: 4 UTF-8 CR/LF C Python 3.13.6 Go Live Preview

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POSIX
```

```
Reversed string: nadnuk  
PS D:\AI programming> python -u "D:\AI programming\Fibonacci.py"  
Basic calculator  
Select operation:  
1. Add  
2. Subtract  
3. Multiply  
4. Divide  
Enter choice (1/2/3/4): 4  
Enter first number: 234  
Enter second number:23  
Result: 10.17993043478262  
PS D:\AI programming>
```

Ln 41, Col 29 Spaces: 4 UTF-8 CR/LF C Python 3.13.6 Go Live Preview

Observation:

The Python program functions as a basic calculator, allowing the user to perform arithmetic operations such as addition, subtraction, multiplication, and division.

Program Workflow:

1. User Interface:

- The program displays a menu listing four operations:
 - 1. Add
 - 2. Subtract
 - 3. Multiply
 - 4. Divide

2. User Input:

- The user is prompted to select an operation (choice) by entering a number between 1 and 4.
- If the input is valid (choice in ('1', '2', '3', '4')), the program proceeds to request two numbers as input.
- A try-except block is used to ensure both numbers are valid floating-point values. If not, it catches the error and prompts the user accordingly.

3. Operation Handling:

- Based on the user's choice:
 - 1: Calls add(num1, num2)

- 2: Calls subtract(num1, num2)
- 3: Calls multiply(num1, num2)
- 4: Calls divide(num1, num2)
- The result is then printed using print("Result:", ...).

4. Invalid Input Handling:

- If the user enters a choice outside of the range 1–4, the program prints "Invalid choice!".

● Task:5

Use a comment to instruct AI to write a function that reads a file and returns the number of lines..

Code:

```
def count_lines(filename):
    """Read a file and return the number of lines."""
    try:
        with open(filename, 'r') as file:
            lines = file.readlines()
            return len(lines)
    except FileNotFoundError:
        print(f"File '{filename}' not found.")
        return -1
    except Exception as e:
        print(f"Error reading file: {e}")
        return -1

def main():
    print("== File Line Counter ==")
    filename = input("Enter filename: ")

    line_count = count_lines(filename)

    if line_count >= 0:
        print(f"Number of lines: {line_count}")
    else:
        print("Could not count lines.")

if __name__ == "__main__":
    main()
```

Output:

**Hi I am Balaji.
I am from Warangal.**

```
PS D:\AI programming> py  
  
> python -u "d:\AI programming\line.py"  
== File Line Counter ===  
Enter filename: kund  
Number of lines: 2  
PS D:\AI programming>
```