# SR UNIVERSITY

## AI ASSISSTED CODING

NAME:B.HASINI

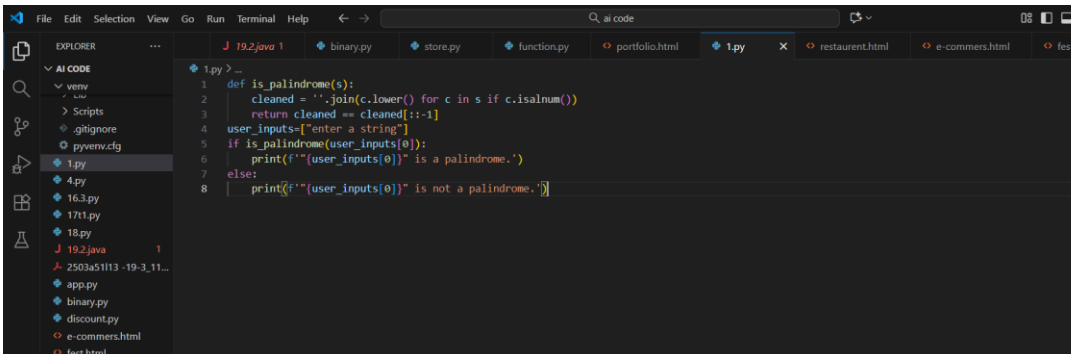
2503A51l13

### TASK #1:

### Prompt Used:

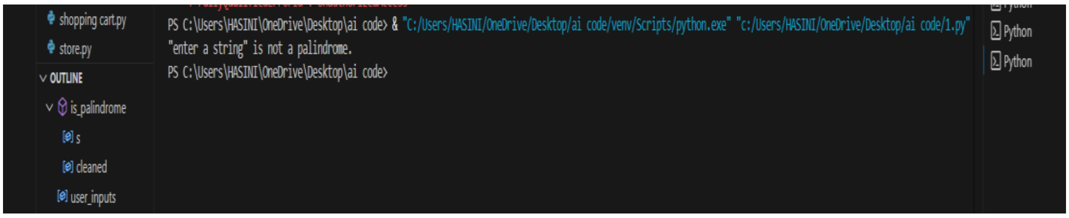
Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

### Code Generated:



```python
def is_palindrome(s):
    cleaned = ''.join(c.lower() for c in s if c.isalnum())
    return cleaned == cleaned[::-1]
user_inputs=["enter a string"]
if is_palindrome(user_inputs[0]):
    print(f'"{user_inputs[0]}" is a palindrome.')
else:
    print(f'"{user_inputs[0]}" is not a palindrome.')
```

### Output:



```
PS C:\Users\HASINI\OneDrive\Desktop\ai code> & "C:/Users/HASINI/OneDrive/Desktop/ai code/venv/Scripts/python.exe" "c:/Users/HASINI/OneDrive/Desktop/ai code/1.py"
"enter a string" is not a palindrome.
PS C:\Users\HASINI\OneDrive\Desktop\ai code>
```

### Observations:

- It cleans the input by removing all non-alphanumeric characters and converting everything to lowercase, ensuring accurate results regardless of punctuation, spacing, or case.

- Checks for palindrome by comparing the cleaned string to its reverse (cleaned[::-1]). If both match, the string is a palindrome.

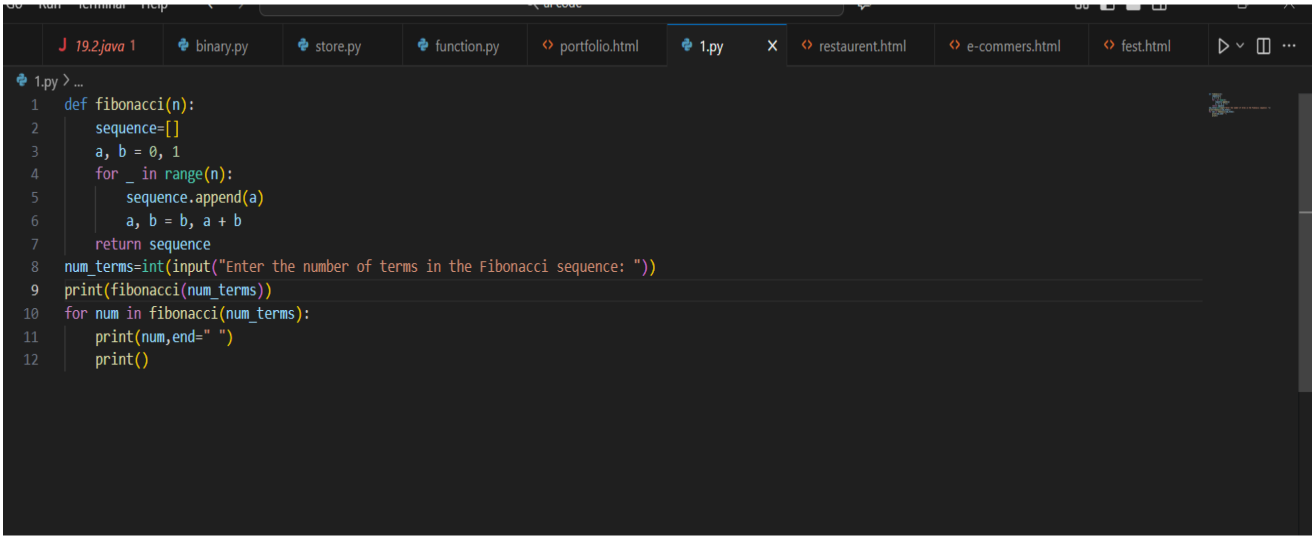
- Works for sentences and phrases (not just single words), as spaces, punctuation, and cases are ignored in the check.

## TASK #2:

### Prompt Used:

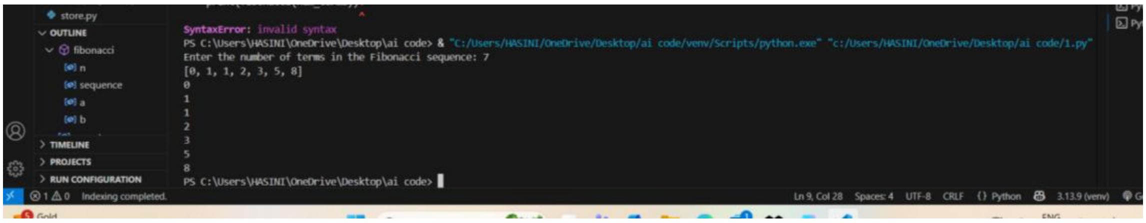
Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring.

### Code Generated:



```python
def fibonacci(n):
    sequence=[]
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence
num_terms=int(input("Enter the number of terms in the Fibonacci sequence: "))
print(fibonacci(num_terms))
for num in fibonacci(num_terms):
    print(num,end=" ")
    print()
```

### OUTPUT:


store.py
OUTLINE
fibonacci
n
sequence
a
b
TIMELINE
PROJECTS
RUN CONFIGURATION
⊗ 1 ⚠ 0 Indexing completed.
Ln 9, Col 28 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (venv)

```
SyntaxError: invalid syntax
PS C:\Users\HASINI\OneDrive\Desktop\ai code> & "C:/Users/HASINI/OneDrive/Desktop/ai code/venv/Scripts/python.exe" "c:/Users/HASINI/OneDrive/Desktop/ai code/1.py"
Enter the number of terms in the Fibonacci sequence: 7
[0, 1, 1, 2, 3, 5, 8]
0
1
1
2
3
5
8
PS C:\Users\HASINI\OneDrive\Desktop\ai code> 
```

### Observations:

- The code generates the Fibonacci sequence up to a user-specified number of terms using a simple iterative approach.

- The sequence is generated by initializing the first two terms (a = 0, b = 1) and iteratively updating them with a, b = b, a + b, ensuring each new term is the sum of its two immediate predecessors.

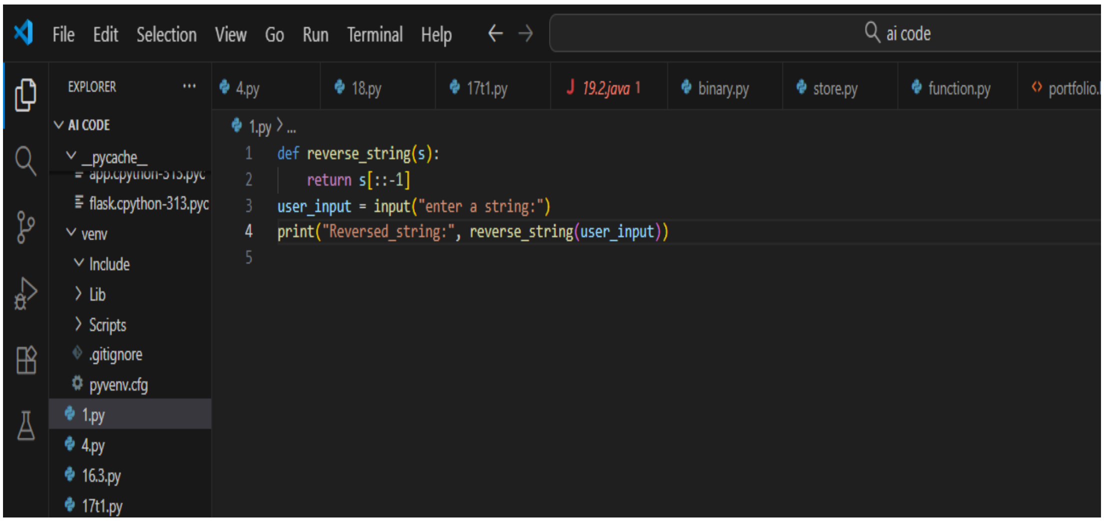
- Each generated term is stored in a list, which is then returned and printed.

**TASK #3:**

**Prompt Used:**

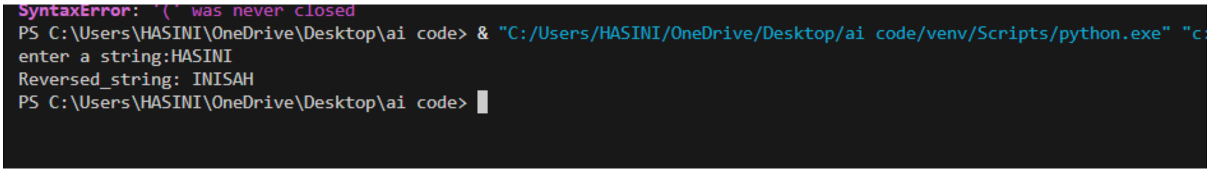
Write a comment like # Function to reverse a string and use Copilot to generate the function.

**Code Generated:**



```python
def reverse_string(s):
    return s[::-1]
user_input = input("enter a string:")
print("Reversed_string:", reverse_string(user_input))
```

**Output:**



```
SyntaxError: '(' was never closed
PS C:\Users\HASINI\OneDrive\Desktop\ai code> & "C:/Users/HASINI/OneDrive/Desktop/ai code/venv/Scripts/python.exe" "c:
enter a string:HASINI
Reversed_string: INISAH
PS C:\Users\HASINI\OneDrive\Desktop\ai code> 
```

**Observations:**

- The function uses Python's string slicing syntax s[::-1] to reverse the string.

- The slice step of -1 means characters are taken from end to start, effectively reversing the string.

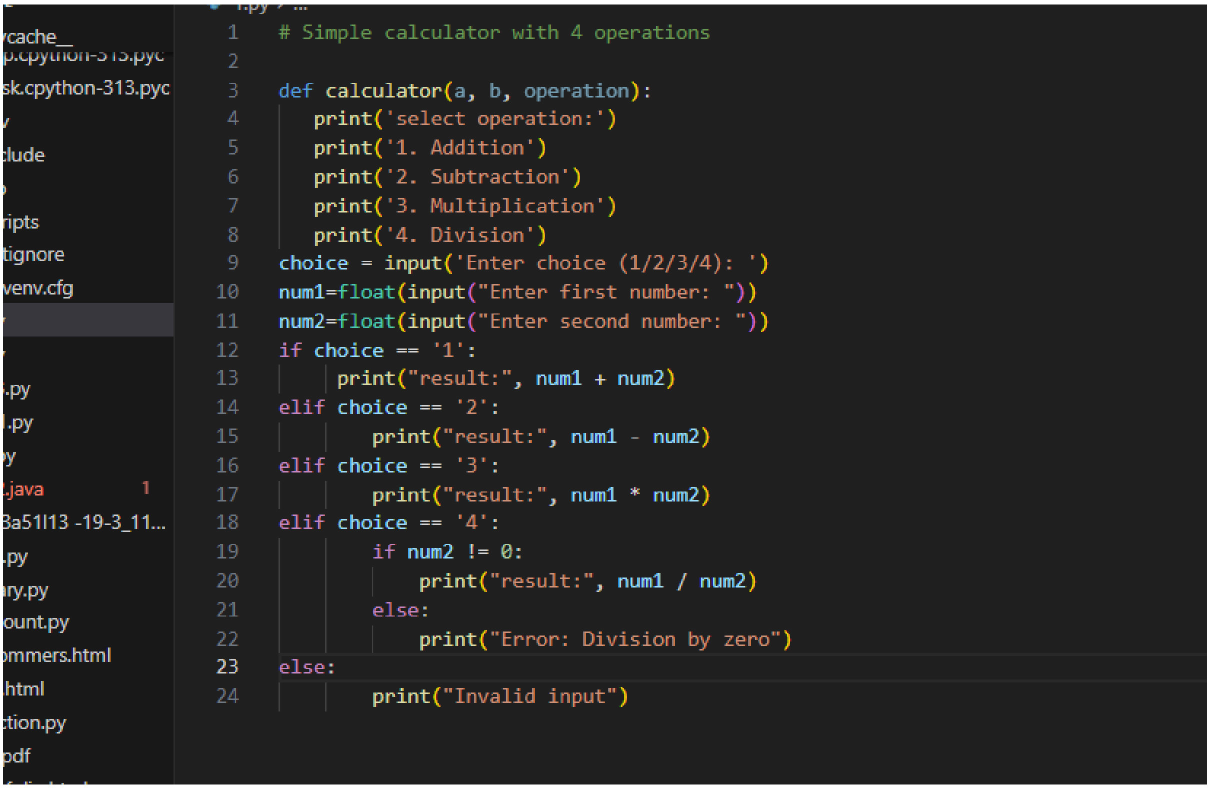
- User input is taken and passed to the function, with the reversed result printed.

**TASK #4:**

**Prompt Used:**

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.

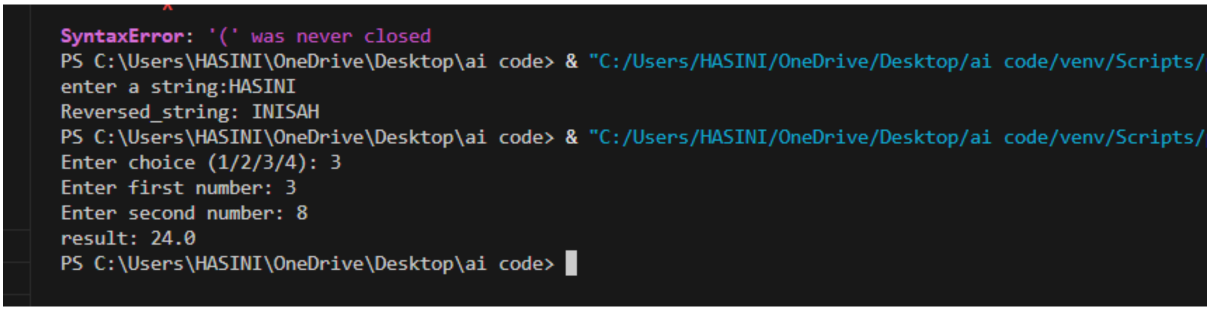
**CODE GENERATED:**



```
# Simple calculator with 4 operations

def calculator(a, b, operation):
    print('select operation:')
    print('1. Addition')
    print('2. Subtraction')
    print('3. Multiplication')
    print('4. Division')
choice = input('Enter choice (1/2/3/4): ')
num1=float(input("Enter first number: "))
num2=float(input("Enter second number: "))
if choice == '1':
     print("result:", num1 + num2)
elif choice == '2':
        print("result:", num1 - num2)
elif choice == '3':
        print("result:", num1 * num2)
elif choice == '4':
        if num2 != 0:
            print("result:", num1 / num2)
        else:
            print("Error: Division by zero")
else:
        print("Invalid input")
```

**OUTPUT:**



```
SyntaxError: '(' was never closed
PS C:\Users\HASINI\OneDrive\Desktop\ai code> & "C:/Users/HASINI/OneDrive/Desktop/ai code/venv/Scripts/
enter a string:HASINI
Reversed_string: INISAH
PS C:\Users\HASINI\OneDrive\Desktop\ai code> & "C:/Users/HASINI/OneDrive/Desktop/ai code/venv/Scripts/
Enter choice (1/2/3/4): 3
Enter first number: 3
Enter second number: 8
result: 24.0
PS C:\Users\HASINI\OneDrive\Desktop\ai code> 
```

Observations:

- The function uses Python's string slicing syntax s[::-1] to reverse the string.
- The slice step of -1 means characters are taken from end to start, effectively reversing the string.
- User input is taken and passed to the function, with the reversed result printed.