

SR UNIVERSITY

AI ASSISTED CODING

NAME:B.HASINI

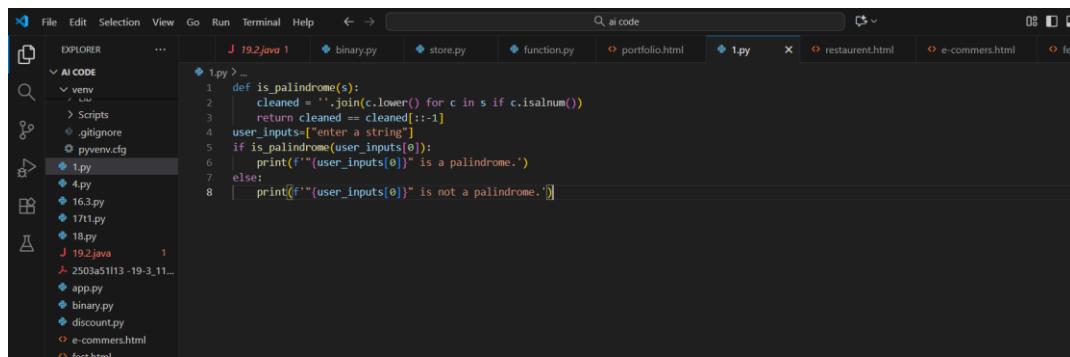
2503A51I13

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.

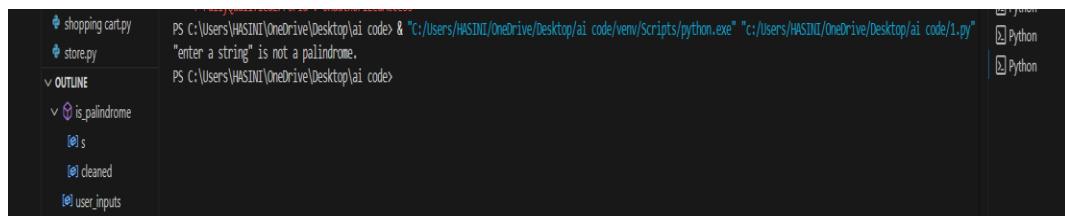
Code Generated:



```
File Edit Selection View Go Run Terminal Help ↻ → ai code
EXPLORER ... J 19.2.java 1 binary.py store.py function.py portfolio.html 1.py × restaurent.html e-commerce.html fe
AI CODE
venv .gitignore pyenv.cfg
1.py 4.py 16.3.py 17t1.py 18.py
J 19.2.java 1
2503a5113 -19-3.11...
app.py binary.py discount.py e-commerce.html fact.html

1.py
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.

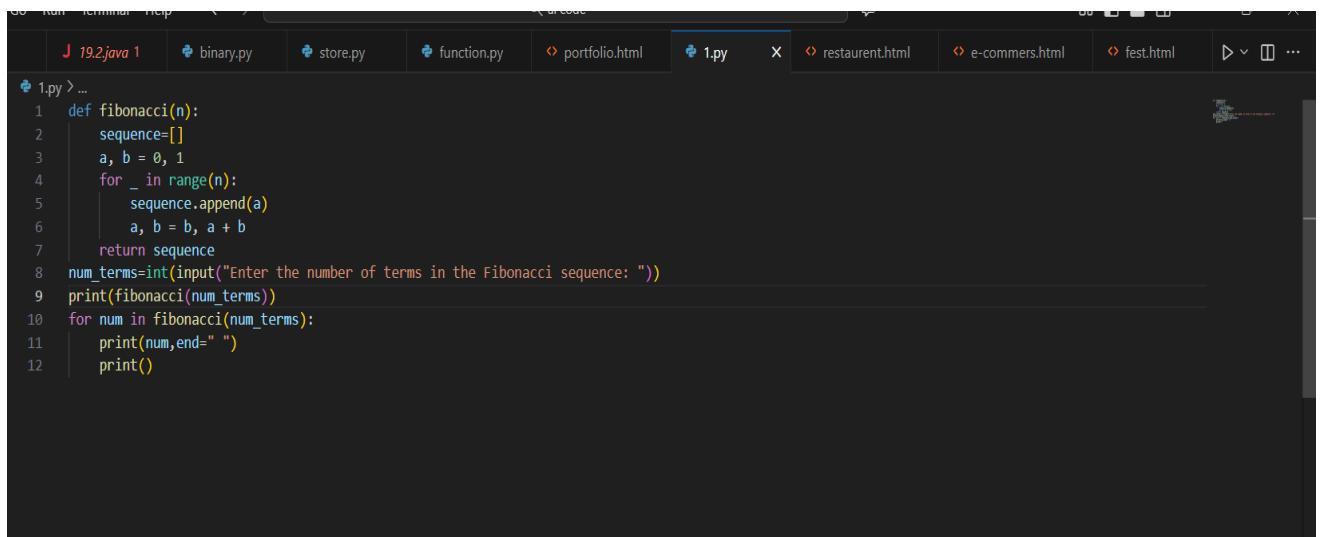
- Works for sentences and phrases (not just single words), as spaces, punctuation, and cases are ignored in the check.

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:

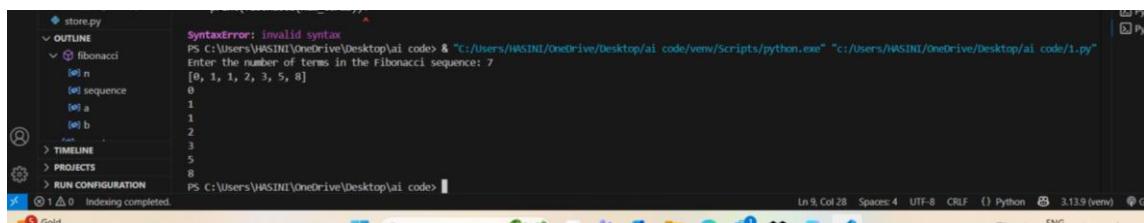


```

1.py > ...
1  def fibonacci(n):
2      sequence=[]
3      a, b = 0, 1
4      for _ in range(n):
5          sequence.append(a)
6          a, b = b, a + b
7      return sequence
8  num_terms=int(input("Enter the number of terms in the Fibonacci sequence: "))
9  print(fibonacci(num_terms))
10 for num in fibonacci(num_terms):
11     print(num,end=" ")
12     print()

```

OUTPUT:



```

store.py
fibonacci
n
sequence
a
b
TIMELINE
PROJECTS
RUN CONFIGURATION
Indexing completed.

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

```

Observations:

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

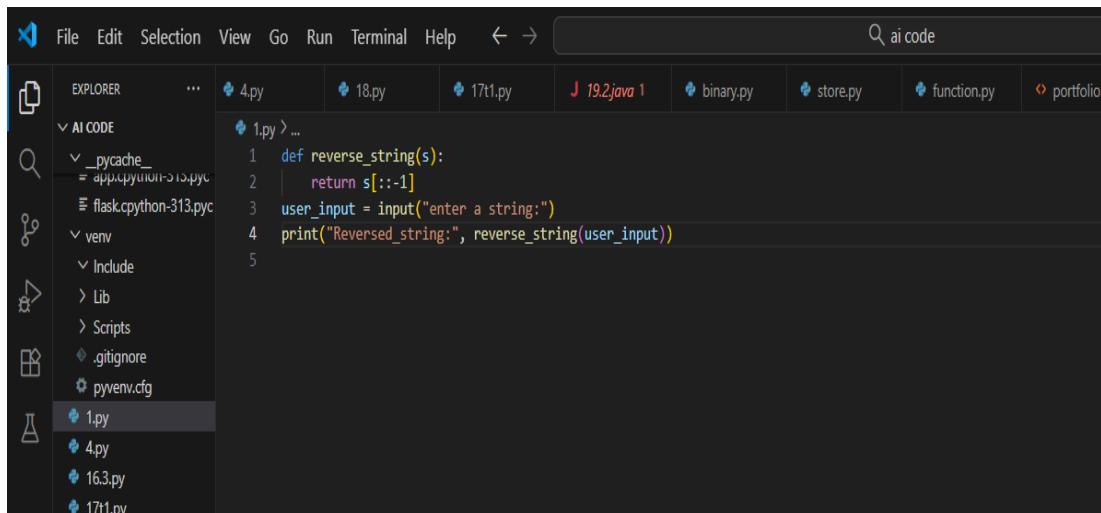
- The sequence is generated by initializing the first two terms ($a = 0$, $b = 1$) and iteratively updating them with $a, b = b, a + b$, ensuring each new term is the sum of its two immediate predecessors.
- Each generated term is stored in a list, which is then returned and printed.

TASK #3:

Prompt Used:

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

Code Generated:



```

File Edit Selection View Go Run Terminal Help ⌘ ⌘ ai code
EXPLORER AI CODE _pycache_ flask.appenv-313.pyc 1.py > ...
1 def reverse_string(s):
2     return s[::-1]
3 user_input = input("enter a string:")
4 print("Reversed_string:", reverse_string(user_input))
5

```

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar showing a directory structure with files like 1.py, 4.py, 18.py, 17t1.py, binary.py, store.py, function.py, and portfolio.py. The 1.py file is selected and its content is displayed in the main editor area. The code defines a function reverse_string that takes a string s and returns it reversed using slicing (s[::-1]). It then prompts the user for input and prints the reversed string.

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:

```
cache_.pyc
p_cpython-313.pyc
sk.cpython-313.pyc
y
clude
o
ipts
tignore
venv.cfg
r
.
py
.py
y
.java 1
Ba51I13 -19-3_11...
.py
try.py
ount.py
ommers.html
html
ction.py
pdf

• t.py > ...
1     # Simple calculator with 4 operations
2
3     def calculator(a, b, operation):
4         print('select operation:')
5         print('1. Addition')
6         print('2. Subtraction')
7         print('3. Multiplication')
8         print('4. Division')
9     choice = input('Enter choice (1/2/3/4): ')
10    num1=float(input("Enter first number: "))
11    num2=float(input("Enter second number: "))
12    if choice == '1':
13        | | print("result:", num1 + num2)
14    elif choice == '2':
15        | | print("result:", num1 - num2)
16    elif choice == '3':
17        | | print("result:", num1 * num2)
18    elif choice == '4':
19        | | if num2 != 0:
20            | | | print("result:", num1 / num2)
21        | | else:
22            | | | print("Error: Division by zero")
23    else:
24        | | | 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.