ASSIGNMENT-3.4

2303A51546

Batch-10

TASK-1:

Prompt:

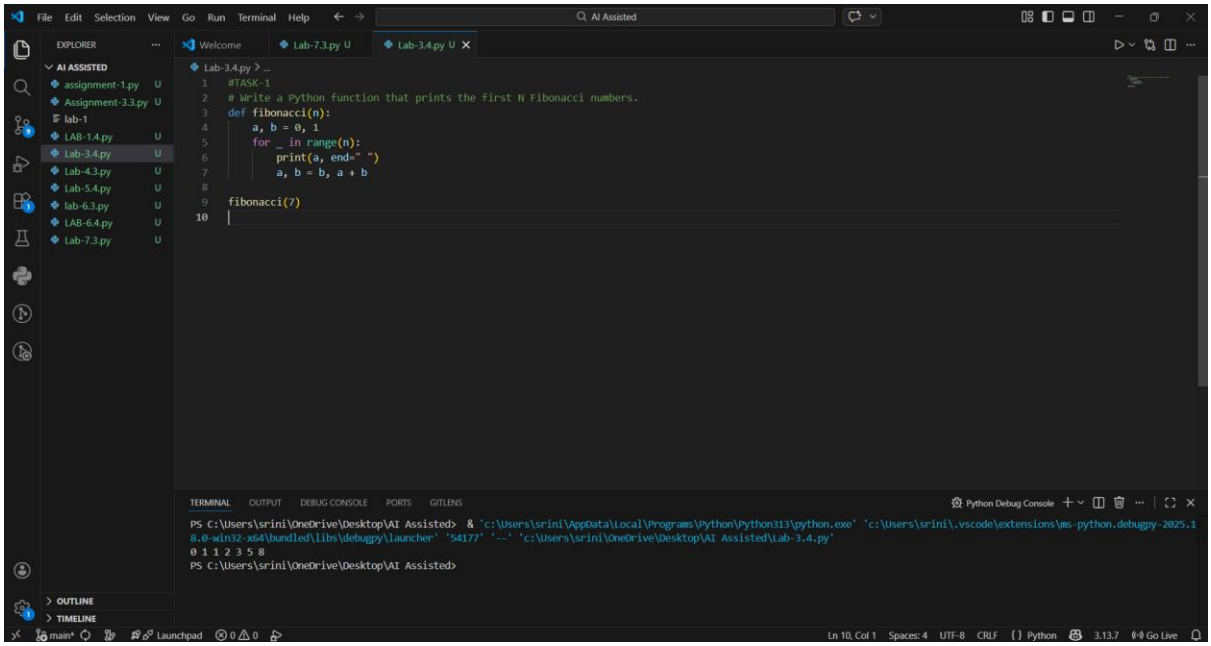
Write a Python function that prints the first N Fibonacci numbers.

CODE:

```
def fibonacci(n):
    a, b = 0, 1
    for _ in range(n):
        print(a, end=" ")
        a, b = b, a + b


fibonacci(7)
```

Output:



Analysis:

The function fibonacci(n) prints the first N Fibonacci numbers using a loop. It starts with 0 and 1, then keeps updating values using tuple swapping.

TASK-2:

Prompt:

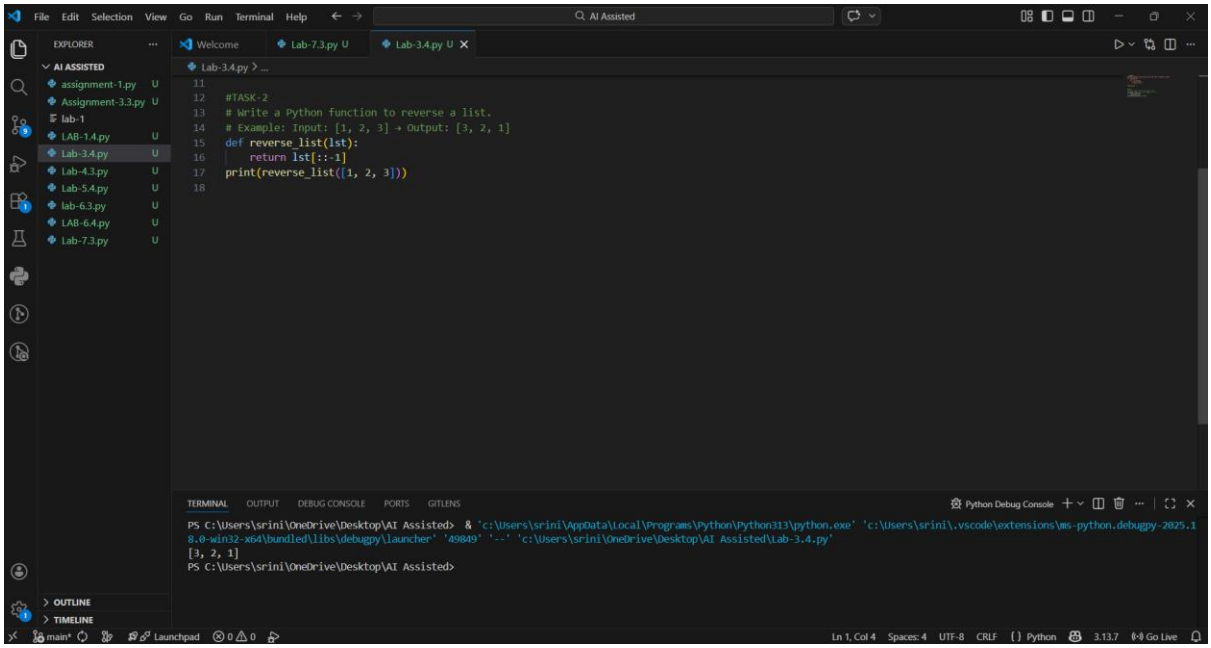
Write a Python function to reverse a list.

Example: Input: [1, 2, 3] → Output: [3, 2, 1]

CODE:

```
def reverse_list(lst):
   return lst[::-1]
print(reverse_list([1, 2, 3]))
```

Output:



Analysis:

The function reverse_list(lst) reverses a list using Python slicing.

TASK-3:

Prompt:

Write a function is_valid(s) that returns True if a string starts with a capital letter and ends with a period.

Examples:

"Hello." → True

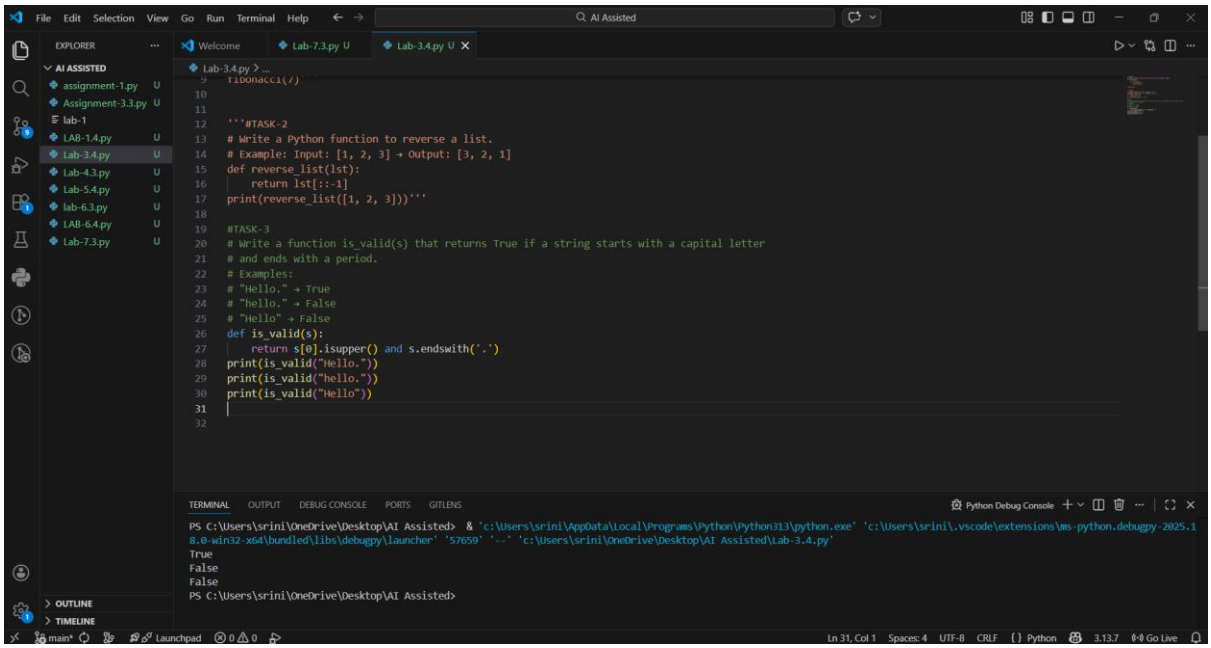
"hello." → False

"Hello" → False

CODE:

```
def is_valid(s):
    return s[0].isupper() and s.endswith('.')
print(is_valid("Hello."))
print(is_valid("hello."))
print(is_valid("Hello"))
```

Output:



Analysis:

The function checks two conditions:

First letter is uppercase → s[0].isupper()

String ends with a period → s.endswith('.')

TASK-4:

Prompt:

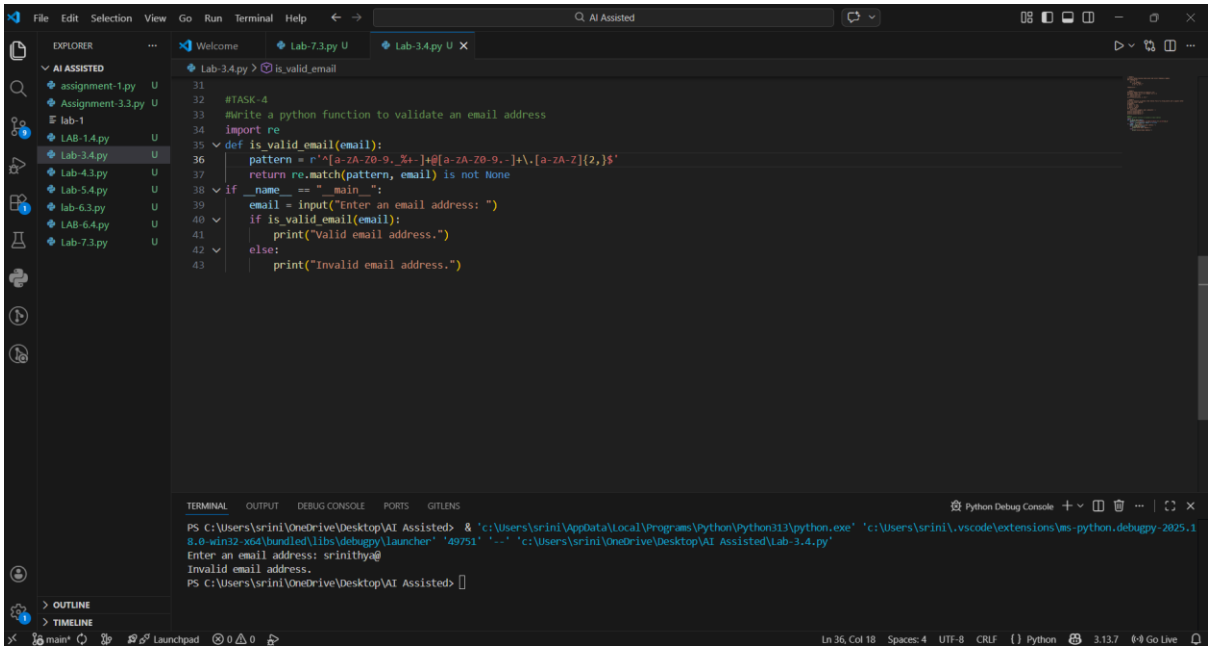
Write a python function to validate an email address

CODE:

```
import re
def is_valid_email(email):
```

```
    pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
    return re.match(pattern, email) is not None
if __name__ == "__main__":
    email = input("Enter an email address: ")
    if is_valid_email(email):
        print("Valid email address.")
    else:
        print("Invalid email address.")
```

Output:



Analysis:

The function uses regular expressions (regex) to validate email format.

TASK-5:

Prompt:

Write a function to return sum of digits of a number

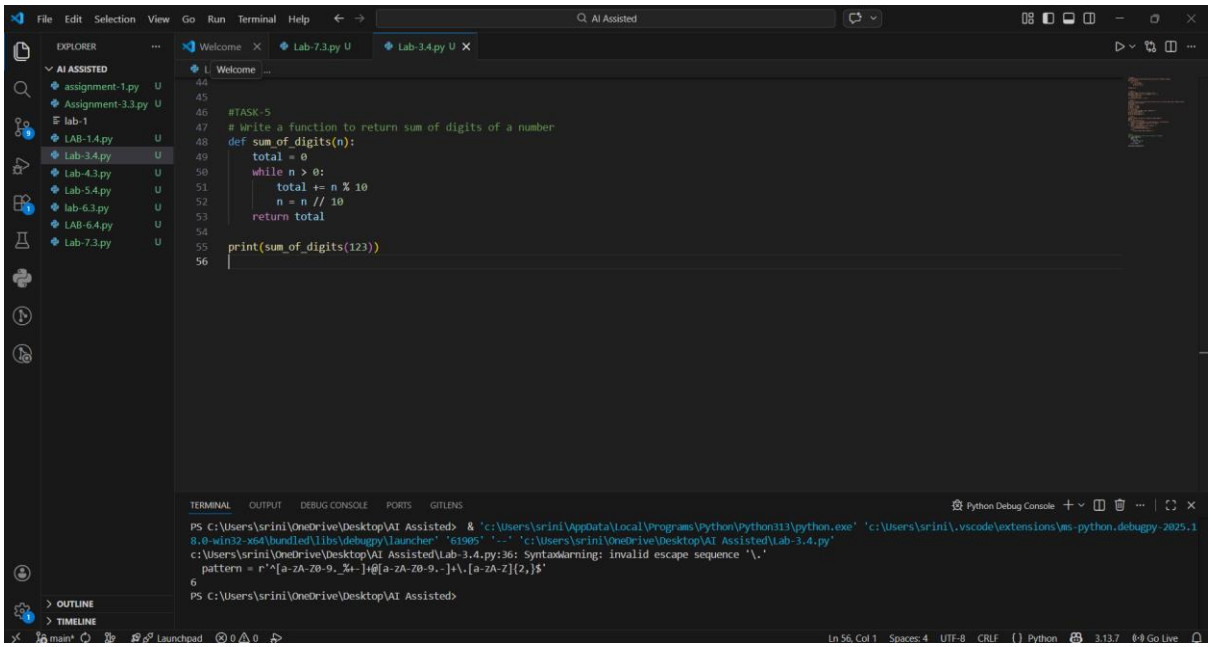
CODE:

```
def sum_of_digits(n):
    total = 0
    while n > 0:
```

```
    total += n % 10
    n = n // 10
  return total


print(sum_of_digits(123))
```

Output:



Analysis:

The function extracts digits using modulus (% 10) and division (// 10) until the number becomes 0.