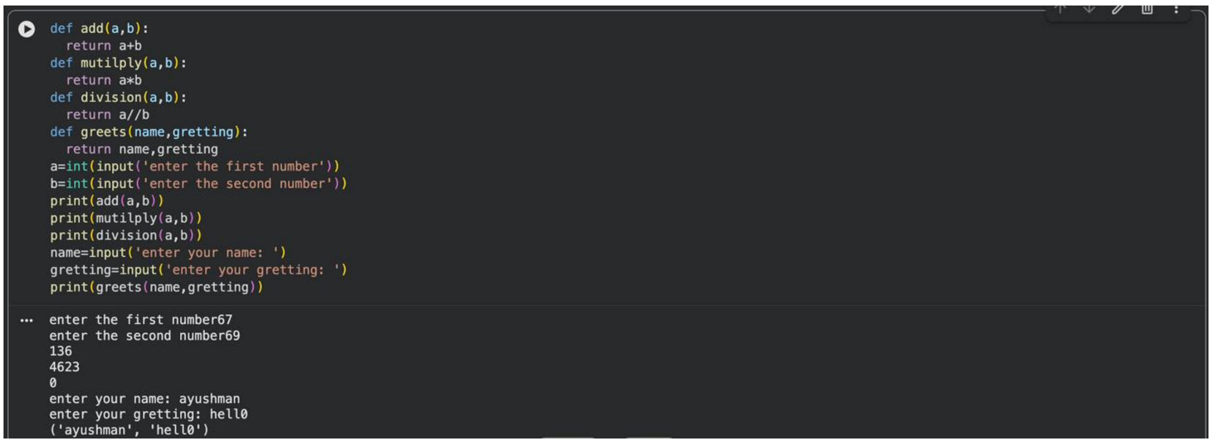
Assisgnment-9.4

P.Ayushman

2303a52294

Task-1:

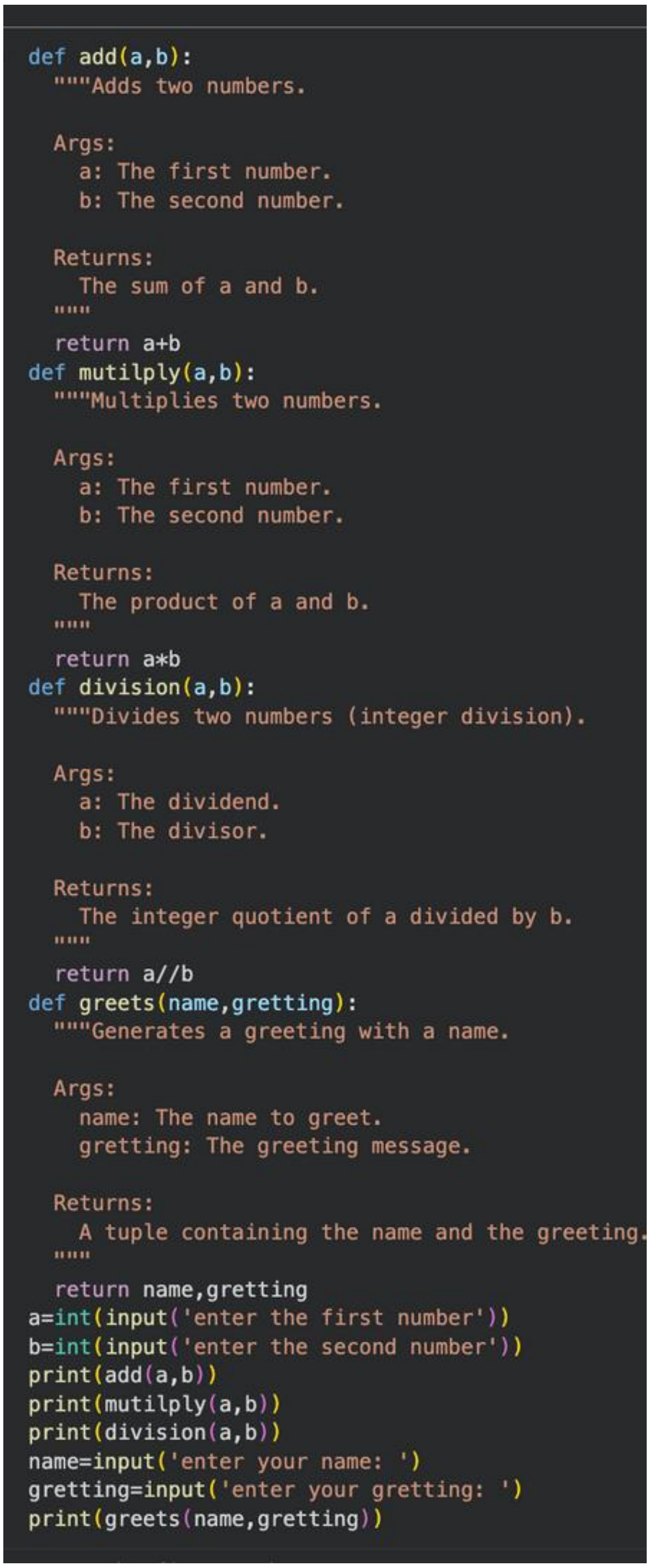
Colab generated file without docstirng:



```
def add(a,b):
  return a+b
def mutilply(a,b):
  return a*b
def division(a,b):
  return a//b
def greets(name,gretting):
  return name,gretting
a=int(input('enter the first number'))
b=int(input('enter the second number'))
print(add(a,b))
print(mutilply(a,b))
print(division(a,b))
name=input('enter your name: ')
gretting=input('enter your gretting: ')
print(greets(name,gretting))
```

```
enter the first number67
enter the second number69
136
4623
0
enter your name: ayushman
enter your gretting: hell0
('ayushman', 'hell0')
```

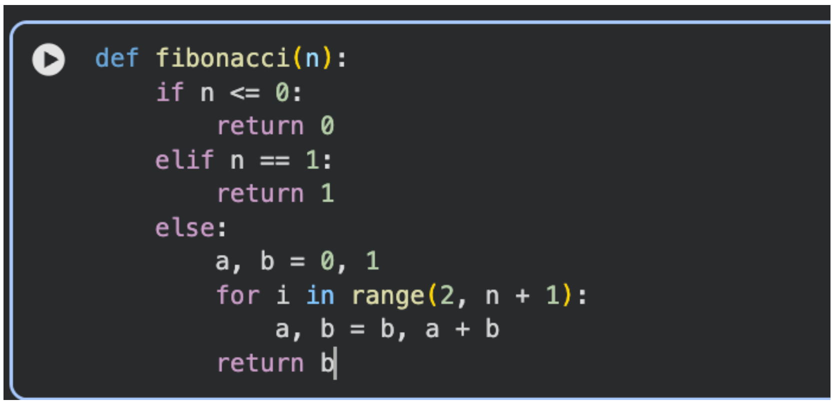
With Docstring:



```
def add(a,b):
  """Adds two numbers.

  Args:
    a: The first number.
    b: The second number.

  Returns:
    The sum of a and b.
  """
  return a+b
def mutilply(a,b):
  """Multiplies two numbers.

  Args:
    a: The first number.
    b: The second number.

  Returns:
    The product of a and b.
  """
  return a*b
def division(a,b):
  """Divides two numbers (integer division).

  Args:
    a: The dividend.
    b: The divisor.

  Returns:
    The integer quotient of a divided by b.
  """
  return a//b
def greets(name,gretting):
  """Generates a greeting with a name.

  Args:
    name: The name to greet.
    gretting: The greeting message.

  Returns:
    A tuple containing the name and the greeting.
  """
  return name,gretting
a=int(input('enter the first number'))
b=int(input('enter the second number'))
print(add(a,b))
print(mutilply(a,b))
print(division(a,b))
name=input('enter your name: ')
gretting=input('enter your gretting: ')
print(greets(name,gretting))
```

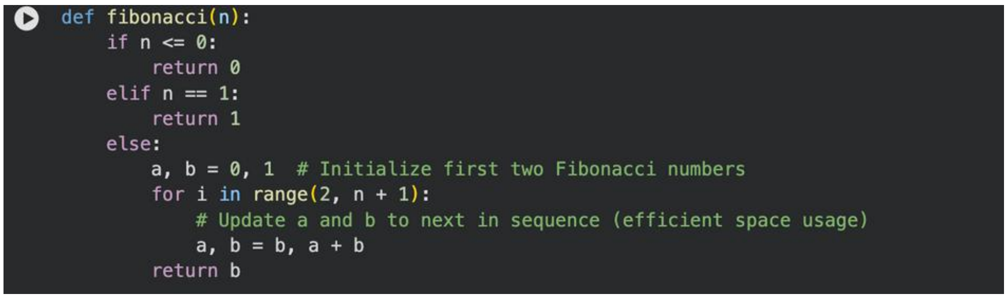
Task -2:

Colab generated code without inline comments:



```python
def fibonacci(n):
    if n <= 0:
        return 0
    elif n == 1:
        return 1
    else:
        a, b = 0, 1
        for i in range(2, n + 1):
            a, b = b, a + b
        return b
```

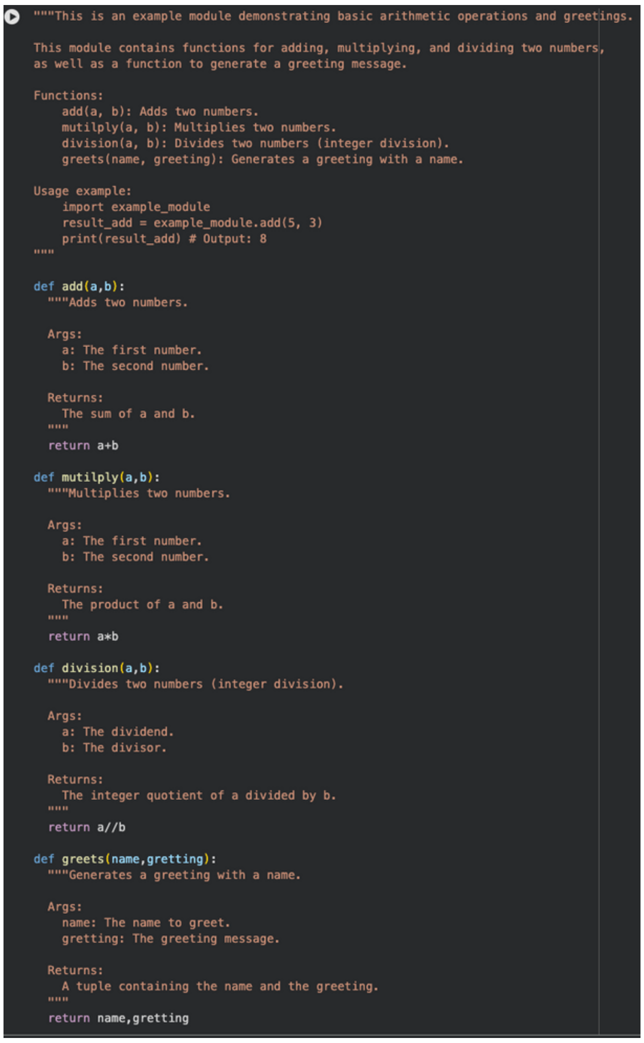
Colab generated inline comments:



```python
def fibonacci(n):
    if n <= 0:
        return 0
    elif n == 1:
        return 1
    else:
        a, b = 0, 1  # Initialize first two Fibonacci numbers
        for i in range(2, n + 1):
            # Update a and b to next in sequence (efficient space usage)
            a, b = b, a + b
        return b
```

Task-3:

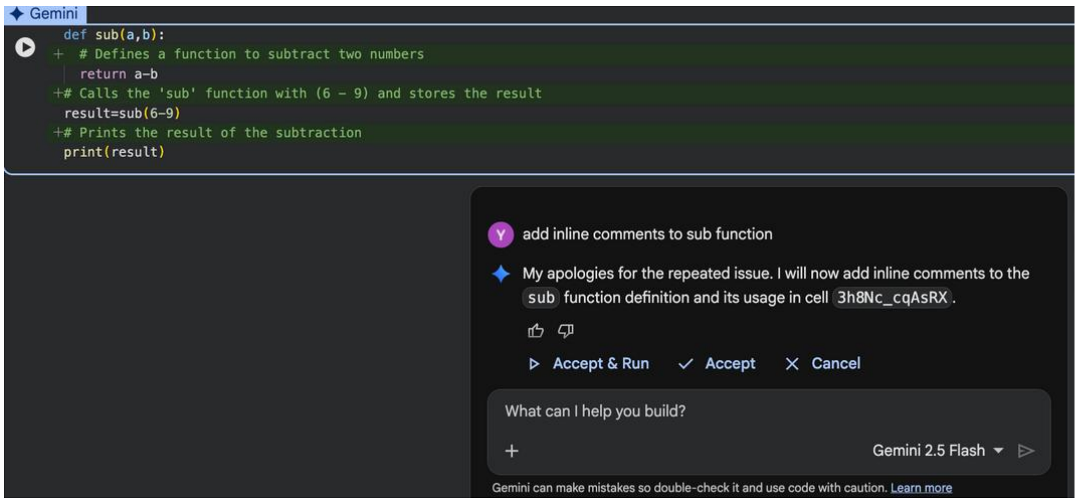
Colab generated module-Level Documentation for above code:



```python
"""This is an example module demonstrating basic arithmetic operations and greetings.

This module contains functions for adding, multiplying, and dividing two numbers,
as well as a function to generate a greeting message.

Functions:
    add(a, b): Adds two numbers.
    mutilply(a, b): Multiplies two numbers.
    division(a, b): Divides two numbers (integer division).
    greets(name, greeting): Generates a greeting with a name.

Usage example:
    import example_module
    result_add = example_module.add(5, 3)
    print(result_add) # Output: 8
"""

def add(a,b):
  """Adds two numbers.

  Args:
    a: The first number.
    b: The second number.

  Returns:
    The sum of a and b.
  """
  return a+b

def mutilply(a,b):
  """Multiplies two numbers.

  Args:
    a: The first number.
    b: The second number.

  Returns:
    The product of a and b.
  """
  return a*b

def division(a,b):
  """Divides two numbers (integer division).

  Args:
    a: The dividend.
    b: The divisor.

  Returns:
    The integer quotient of a divided by b.
  """
  return a//b

def greets(name,gretting):
  """Generates a greeting with a name.

  Args:
    name: The name to greet.
    gretting: The greeting message.

  Returns:
    A tuple containing the name and the greeting.
  """
  return name,gretting
```

Task -4:

Colab Generated Inline comments



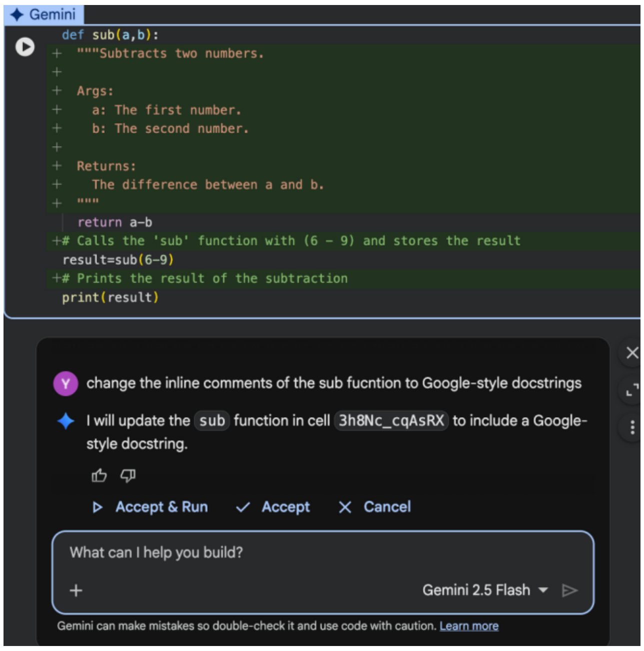
```python
def sub(a,b):
+  # Defines a function to subtract two numbers
   return a-b
+# Calls the 'sub' function with (6 - 9) and stores the result
result=sub(6-9)
+# Prints the result of the subtraction
print(result)
```

add inline comments to sub function

My apologies for the repeated issue. I will now add inline comments to the `sub` function definition and its usage in cell `3h8Nc_cqAsRX`.

Gemini can make mistakes so double-check it and use code with caution. Learn more

Colab converted inline to docstrings:



```
def sub(a,b):
+  """Subtracts two numbers.
+
+  Args:
+    a: The first number.
+    b: The second number.
+
+  Returns:
+    The difference between a and b.
+  """
   return a-b
+# Calls the 'sub' function with (6 - 9) and stores the result
result=sub(6-9)
+# Prints the result of the subtraction
print(result)
```

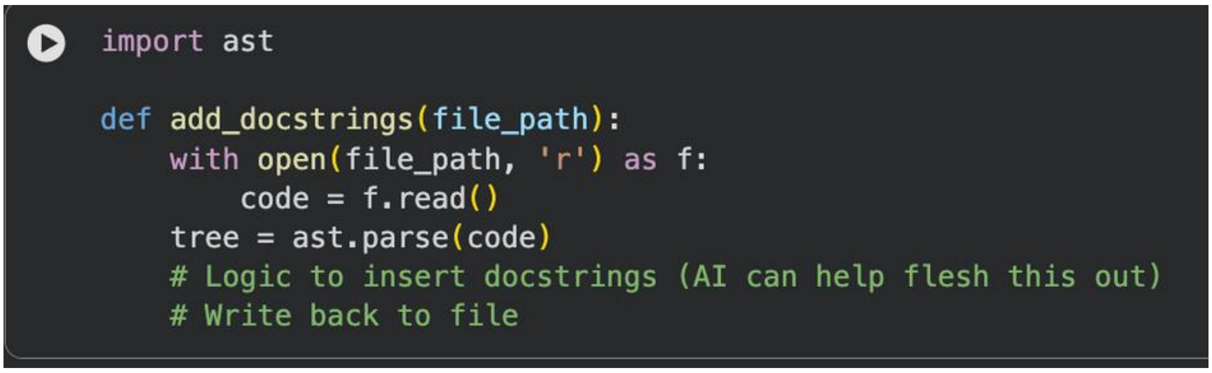
change the inline comments of the sub fucntion to Google-style docstrings

I will update the `sub` function in cell `3h8Nc_cqAsRX` to include a Google-style docstring.

Gemini
Accept & Run
Accept
Cancel
What can I help you build?
Gemini 2.5 Flash
Gemini can make mistakes so double-check it and use code with caution. Learn more

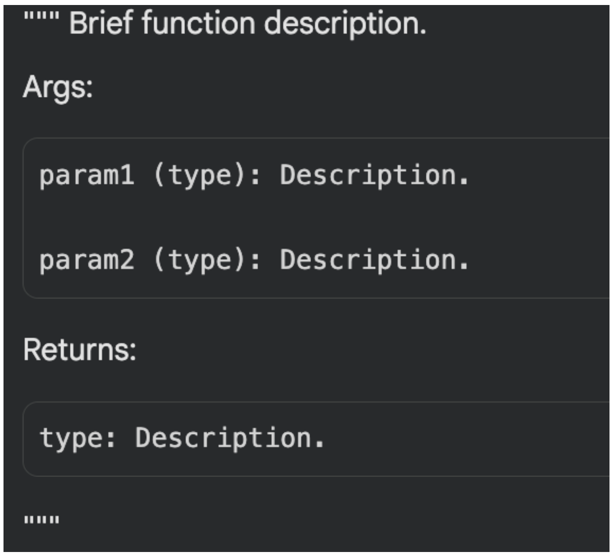
Task-5:

Code for Mini Automatic Documentation Generator:



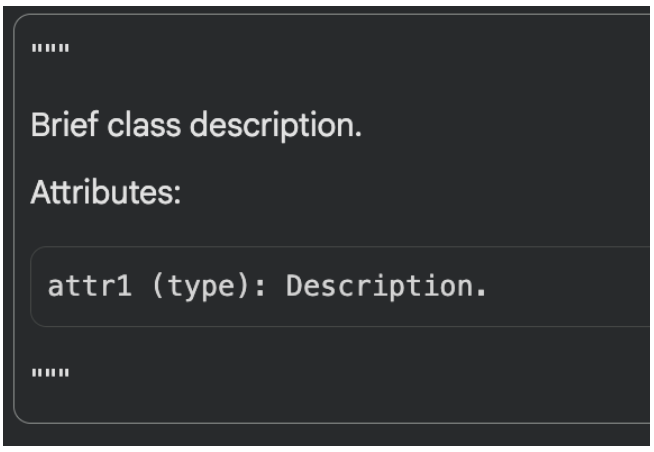
```
import ast

def add_docstrings(file_path):
    with open(file_path, 'r') as f:
        code = f.read()
    tree = ast.parse(code)
    # Logic to insert docstrings (AI can help flesh this out)
    # Write back to file
```

For Functions:



```
""" Brief function description.

Args:

    param1 (type): Description.

    param2 (type): Description.

Returns:

    type: Description.

"""
```

For Classes:



```
"""

Brief class description.

Attributes:

    attr1 (type): Description.

"""
```