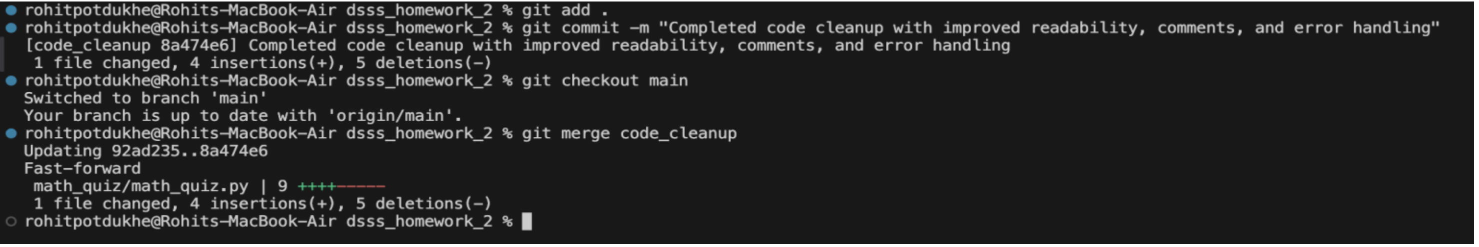
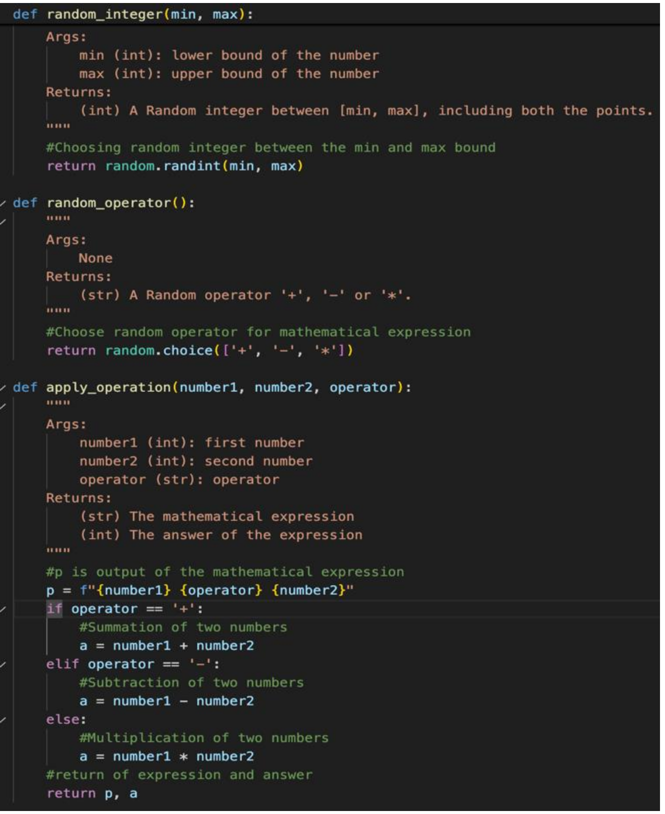
Task 1: https://github.com/rohitpotdukhe01/dsss_homework_2

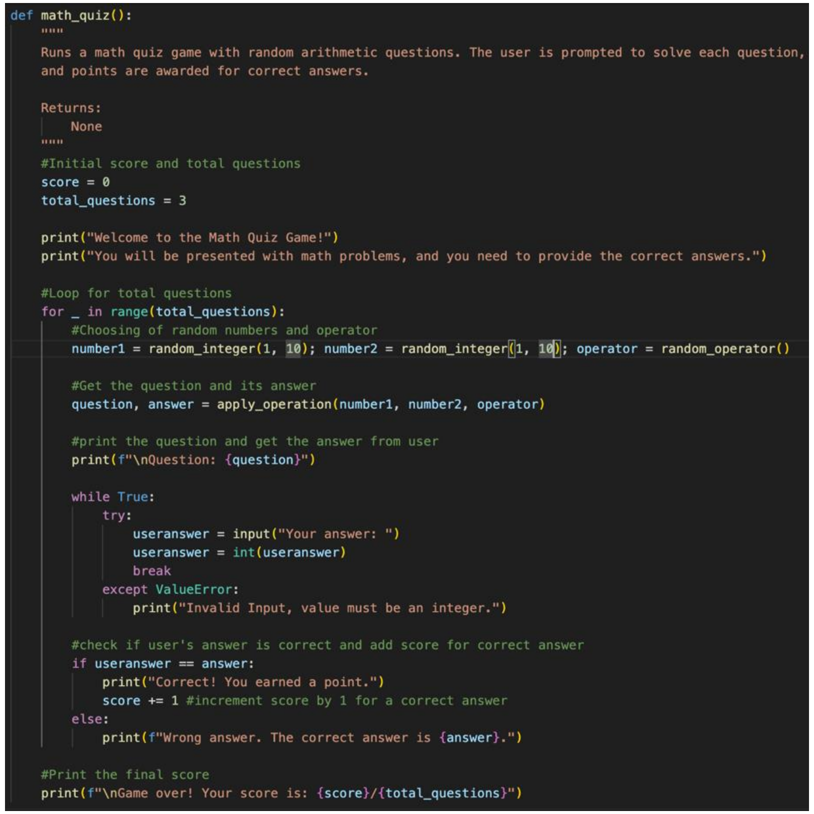
Task 3:



```
● rohitpotdukhe@Rohits-MacBook-Air dsss_homework_2 % git add .
● rohitpotdukhe@Rohits-MacBook-Air dsss_homework_2 % git commit -m "Completed code cleanup with improved readability, comments, and error handling"
  [code_cleanup 8a474e6] Completed code cleanup with improved readability, comments, and error handling
   1 file changed, 4 insertions(+), 5 deletions(-)
● rohitpotdukhe@Rohits-MacBook-Air dsss_homework_2 % git checkout main
  Switched to branch 'main'
  Your branch is up to date with 'origin/main'.
● rohitpotdukhe@Rohits-MacBook-Air dsss_homework_2 % git merge code_cleanup
  Updating 92ad235..8a474e6
  Fast-forward
   math_quiz/math_quiz.py | 9 ++++-----
   1 file changed, 4 insertions(+), 5 deletions(-)
○ rohitpotdukhe@Rohits-MacBook-Air dsss_homework_2 % █
```

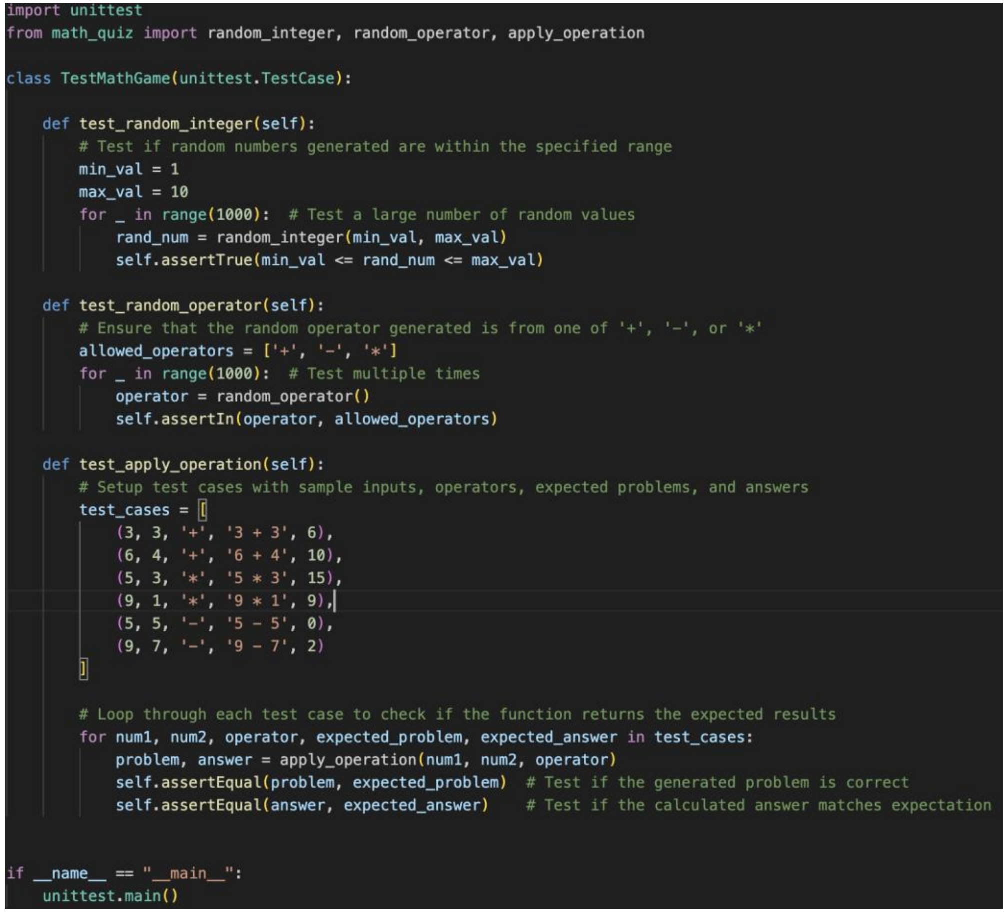


```python
def random_integer(min, max):
    Args:
        min (int): lower bound of the number
        max (int): upper bound of the number
    Returns:
        (int) A Random integer between [min, max], including both the points.
    """
    #Choosing random integer between the min and max bound
    return random.randint(min, max)

def random_operator():
    """
    Args:
        None
    Returns:
        (str) A Random operator '+', '-' or '*'.
    """
    #Choose random operator for mathematical expression
    return random.choice(['+', '-', '*'])

def apply_operation(number1, number2, operator):
    """
    Args:
        number1 (int): first number
        number2 (int): second number
        operator (str): operator
    Returns:
        (str) The mathematical expression
        (int) The answer of the expression
    """
    #p is output of the mathematical expression
    p = f"{number1} {operator} {number2}"
    if operator == '+':
        #Summation of two numbers
        a = number1 + number2
    elif operator == '-':
        #Subtraction of two numbers
        a = number1 - number2
    else:
        #Multiplication of two numbers
        a = number1 * number2
    #return of expression and answer
    return p, a
```



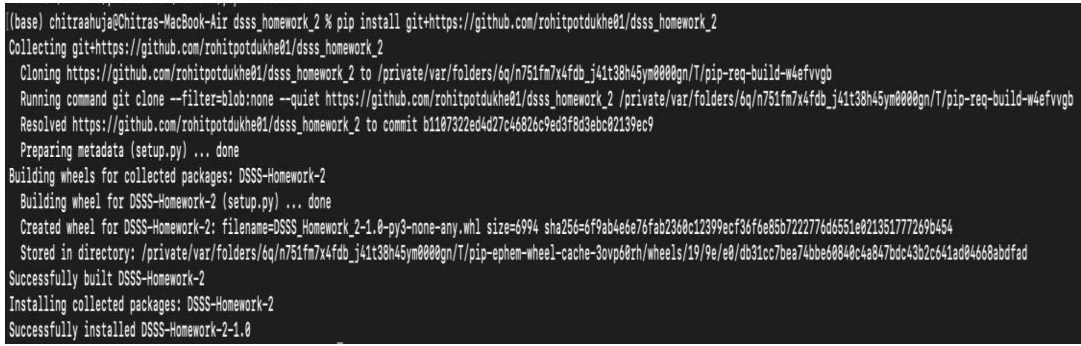
```python
def math_quiz():
    """
    Runs a math quiz game with random arithmetic questions. The user is prompted to solve each question,
    and points are awarded for correct answers.

    Returns:
        None
    """
    #Initial score and total questions
    score = 0
    total_questions = 3

    print("Welcome to the Math Quiz Game!")
    print("You will be presented with math problems, and you need to provide the correct answers.")

    #Loop for total questions
    for _ in range(total_questions):
        #Choosing of random numbers and operator
        number1 = random_integer(1, 10); number2 = random_integer(1, 10); operator = random_operator()

        #Get the question and its answer
        question, answer = apply_operation(number1, number2, operator)

        #print the question and get the answer from user
        print(f"\nQuestion: {question}")

        while True:
            try:
                useranswer = input("Your answer: ")
                useranswer = int(useranswer)
                break
            except ValueError:
                print("Invalid Input, value must be an integer.")

        #check if user's answer is correct and add score for correct answer
        if useranswer == answer:
            print("Correct! You earned a point.")
            score += 1 #increment score by 1 for a correct answer
        else:
            print(f"Wrong answer. The correct answer is {answer}.")

    #Print the final score
    print(f"\nGame over! Your score is: {score}/{total_questions}")
```

Task 4:



```python
import unittest
from math_quiz import random_integer, random_operator, apply_operation

class TestMathGame(unittest.TestCase):

    def test_random_integer(self):
        # Test if random numbers generated are within the specified range
        min_val = 1
        max_val = 10
        for _ in range(1000):  # Test a large number of random values
            rand_num = random_integer(min_val, max_val)
            self.assertTrue(min_val <= rand_num <= max_val)

    def test_random_operator(self):
        # Ensure that the random operator generated is from one of '+', '-', or '*'
        allowed_operators = ['+', '-', '*']
        for _ in range(1000):  # Test multiple times
            operator = random_operator()
            self.assertIn(operator, allowed_operators)

    def test_apply_operation(self):
        # Setup test cases with sample inputs, operators, expected problems, and answers
        test_cases = [
            (3, 3, '+', '3 + 3', 6),
            (6, 4, '+', '6 + 4', 10),
            (5, 3, '*', '5 * 3', 15),
            (9, 1, '*', '9 * 1', 9),
            (5, 5, '-', '5 - 5', 0),
            (9, 7, '-', '9 - 7', 2)
        ]

        # Loop through each test case to check if the function returns the expected results
        for num1, num2, operator, expected_problem, expected_answer in test_cases:
            problem, answer = apply_operation(num1, num2, operator)
            self.assertEqual(problem, expected_problem)  # Test if the generated problem is correct
            self.assertEqual(answer, expected_answer)    # Test if the calculated answer matches expectation


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

Task 5:



```
(base) chitraahuja@Chitras-MacBook-Air dsss_homework_2 % pip install git+https://github.com/rohitpotdukhe01/dsss_homework_2
Collecting git+https://github.com/rohitpotdukhe01/dsss_homework_2
  Cloning https://github.com/rohitpotdukhe01/dsss_homework_2 to /private/var/folders/6q/n751fm7x4fdb_j41t38h45ym0000gn/T/pip-req-build-w4efvvgb
  Running command git clone --filter=blob:none --quiet https://github.com/rohitpotdukhe01/dsss_homework_2 /private/var/folders/6q/n751fm7x4fdb_j41t38h45ym0000gn/T/pip-req-build-w4efvvgb
  Resolved https://github.com/rohitpotdukhe01/dsss_homework_2 to commit b1107322ed4d27c46826c9ed3f8d3ebc02139ec9
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: DSSS-Homework-2
  Building wheel for DSSS-Homework-2 (setup.py) ... done
  Created wheel for DSSS-Homework-2: filename=DSSS_Homework_2-1.0-py3-none-any.whl size=6994 sha256=6f9ab4e6e76fab2360c12399ecf36f6e85b7222776d6551e021351777269b454
  Stored in directory: /private/var/folders/6q/n751fm7x4fdb_j41t38h45ym0000gn/T/pip-ephem-wheel-cache-3ovp60rh/wheels/19/9e/e0/db31cc7bea74bbe60840c4a847bdc43b2c641ad04668abdfad
Successfully built DSSS-Homework-2
Installing collected packages: DSSS-Homework-2
Successfully installed DSSS-Homework-2-1.0
```

Rohit Potdukhe, M. N. 22985091, idM yx49uxym