https://github.com/majidsh97/dsss_homework_2

Task 3 :

```
PS D:\Python Projects\DSSS\dsss_homework_2> git merge code_cleanup
Updating b8c40a3..3fd4b8f
Fast-forward
 .vscode/settings.json        | 14 ++++++++++
 math_quiz/math_quiz.py       | 56 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++-------
 math_quiz/tests_math_quiz.py | 23 +++++++++++++++++++++++-----
 3 files changed, 78 insertions(+), 15 deletions(-)
 create mode 100644 .vscode/settings.json
```

```python
import random

def get_random_num(min, max):
    """
    this function gets two numbers and returns one random number
    :param
    `min`= minimum number,
    `max`= maximuim number
    :return
    Random integer.
    """
    try:
        r = random.randint(min, max)
    except:
        r = random.randint(int(min), int(max))
    return r


def get_random_operation():

    """
    @param: no Args
    :)
    ------------------------------------
    @return: a random char
    """
    return random.choice(['+', '-', '*'])


def operation(n1, n2, o):
    """

    do the operation
    """
    p = f"{n1} {o} {n2}"
    if o == '+': a = n1 + n2
    elif o == '-': a = n1 - n2
    else: a = n1 * n2
    return p, a
```

Task 5:

```
PS D:\Python Projects\DSSS\dsss_homework_2> python -m pip install git+https://github.com/majidsh97/dsss_homework_2.git
WARNING: Ignoring invalid distribution -ip (d:\python\python3.9\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (d:\python\python3.9\lib\site-packages)
Collecting git+https://github.com/majidsh97/dsss_homework_2.git
  Cloning https://github.com/majidsh97/dsss_homework_2.git to c:\users\majid\appdata\local\temp\pip-req-build-cddkor3a
  Running command git clone --filter=blob:none --quiet https://github.com/majidsh97/dsss_homework_2.git 'C:\Users\Majid\AppData\Local\Temp\pip-req-build-cddkor3a'
  Resolved https://github.com/majidsh97/dsss_homework_2.git to commit 25326320f7082c95b7301ca10020c6270be44e66
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: math-quiz
  Building wheel for math-quiz (setup.py) ... done
  Created wheel for math-quiz: filename=math_quiz-1.0.0-py3-none-any.whl size=7081 sha256=351397f2cefdd8255d7a3bea1acef4ac6a57f8032e62faf2ca21b0c656fd2bb4
  Stored in directory: C:\Users\Majid\AppData\Local\Temp\pip-ephem-wheel-cache-5tqbw130\wheels\c4\22\93\2634f7060d85c20282efd3e86cfd011453f776169f2936ad78
Successfully built math-quiz
WARNING: Ignoring invalid distribution -ip (d:\python\python3.9\lib\site-packages)
Installing collected packages: math-quiz
WARNING: Ignoring invalid distribution -ip (d:\python\python3.9\lib\site-packages)
Successfully installed math-quiz-1.0.0
```

```python
def math_quiz():
    """
    Executes a math quiz game by presenting math problems and evaluating user answers.

    The user is presented with a set number of questions determined by the value of `t_q`.

    Parameters:
    None

    Returns:
    None
    """


    s = 0
    t_q = 3 #3.14159265359

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

    for _ in range(t_q):
        n1 = get_random_num(1, 10); n2 = get_random_num(1, 5); o = get_random_operation()

        PROBLEM, ANSWER = operation(n1, n2, o)
        print(f"\nQuestion: {PROBLEM}")
        useranswer = input("Your answer: ")
        try :
            useranswer = int(useranswer)

            if useranswer == ANSWER:
                print("Correct! You earned a point.")
                s += -(-1)
            else:
                print(f"Wrong answer. The correct answer is {ANSWER}.")
        except :
            print('Error!')

    print(f"\nGame over! Your score is: {s}/{t_q}")

if __name__ == "__main__":
    math_quiz()
```

Task 4:

```python
import unittest
from math_quiz import get_random_num, get_random_operation, operation


class TestMathGame(unittest.TestCase):

    def test_get_random_num(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 = get_random_num(min_val, max_val)
            self.assertTrue(min_val <= rand_num <= max_val)

    def  test_get_random_operation(self):
        # TODO
        # test if it returns all of the set { +, - , *}
        for _ in range(1000):  # Test a large number of random values
            o = get_random_operation()
            self.assertTrue(o in ['+','-','*'])
        pass

    def test_operation(self):
            test_cases = [
                (5, 2, '+', '5 + 2', 7),
                (3 , 1 , '-' ,'3 - 1',2),
                (5 , 1 , '*' ,'5 * 1',5),
                (0 , 0 , '-' ,'0 - 0',0),
                (0 , 5 , '*' ,'0 * 5',0)
            ]

            for num1, num2, operator, expected_problem, expected_answer in test_cases:
                # TODO
                p,a = operation(num1,num2,operator)
                self.assertTrue(p==expected_problem)
                self.assertTrue(a==expected_answer)


                pass

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

Majid Sharghi Foroushani, M.N. 23193769, ix05ogym