Name: Nguyen Sy Thanh Trung – Student ID: 104224529

Task 1:

I need to list all of the incorrect uses of arithmetic operators to detect the testing case of ANY possible incorrect use.

1. First case: the operator “-” is incorrect and the operator “*” is correct.

a) C = (A + B) * 2
b) C = (A * B) * 2
c) C = (A / B) * 2

2. Second case: The operator “-” is correct and the operator “*” is incorrect

a) C = (A – B) + 2
b) C = (A – B) – 2
c) C = (A – B) / 2

3. Third case: both operators “-“ and “*” are incorrect, leading to both statements A = A – B and C = A * 2 being wrong.

a) C = (A + B) + 2
b) C = (A + B) – 2
c) C = (A + B) / 2
d) C = (A * B) + 2
e) C = (A * B) – 2
f) C = (A * B) / 2
g) C = (A / B) + 2
h) C = (A / B) – 2
i) C = (A / B) / 2

There are a total of 15 incorrect uses of arithmetic operators.

Task 2:

A=3, B=1

1. The correct usage: A = A – B = 3 – 1 = 2

The incorrect usage:
a) A = A + B = 3 + 1 = 4 => C = A * 2 = 4 * 2 = 8
b) A = A * B = 3 * 1 = 3 => C = A * 2 = 3 * 2 = 6
c) A = A / B = 3 / 1 = 3 => C = A * 2 = 3 * 2 = 6

From the first test case, the output from the correct statement is different from the incorrect statements

2. The correct usage: C = A * 2 = 2 * 2 = 4

a) C = A + 2 = 2 + 2 = 4
b) C = A – 2 = 2 – 2 = 0
c) C = A / 2 = 2 / 2 = 1

From the second test case, one incorrect statement has the same result as the correct statement, so the given values of A and B cannot achieve the required objective.

Task 3:

In this task, I will choose A = 10 and B = 5, and I will explain and justify my test case.

The correct statements: A = A – B and C = A * 2

Expected results: A = A – B = 10 – 5 = 5; C = A * 2 = 10.

Test case 1.1: the incorrect statements of the operator “–“

A = A + B = 10 + 5 = 15 => C = A * 2 = 15 * 2 = 30
A = A * B = 10 * 5 = 50 => C = A * 2 = 50 * 2 = 100
A = A / B = 10 / 5 = 2 => C = A * 2 = 2 * 2 = 4

Test case 1.2: the incorrect statements of operator “*”

C = A + 2 = 5 + 2 = 7
C = A – 2 = 5 – 2 = 3
C = A / 2 = 5 / 2 = 2.5

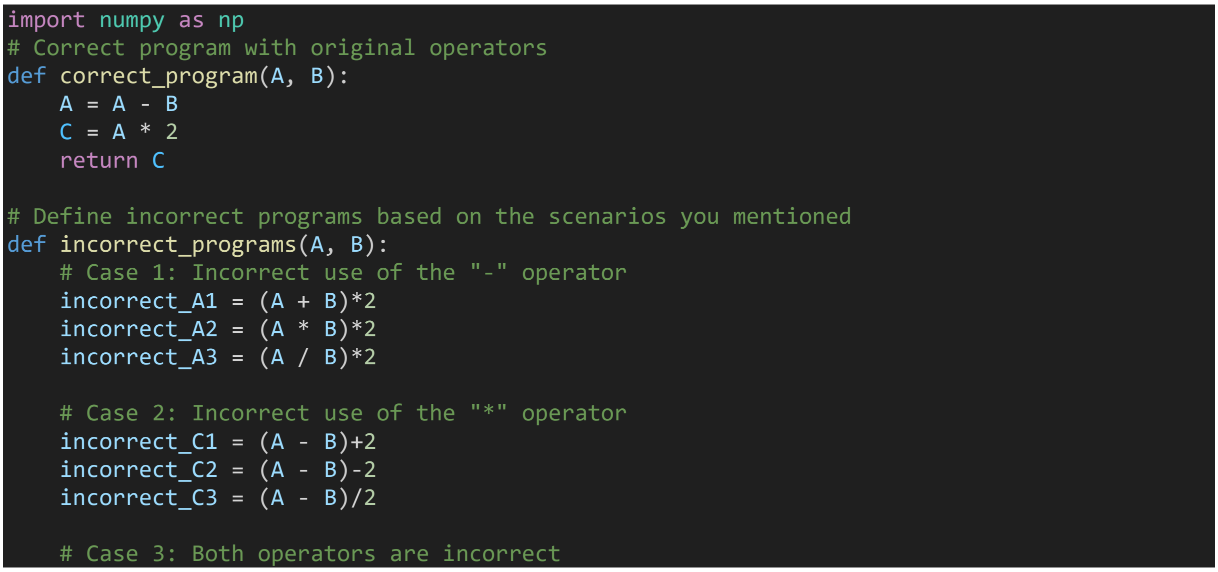
Test case 1.3: the incorrect statements of both operator

C = (A + B) + 2 = (10 + 5) + 2 = 17
C = (A + B) – 2 = (10 + 5) – 2 = 13
C = (A + B) / 2 = (10 + 5) / 2 = 7.5
C = (A * B) + 2 = (10 * 5) + 2 = 52
C = (A * B) – 2 = (10 * 5) – 2 = 48
C = (A * B) / 2 = (10 * 5) / 2 = 25
C = (A / B) + 2 = (10 / 5) + 2 = 4
C = (A / B) – 2 = (10 / 5) – 2 =0
C = (A / B) / 2 = (10 / 5) / 2 = 1

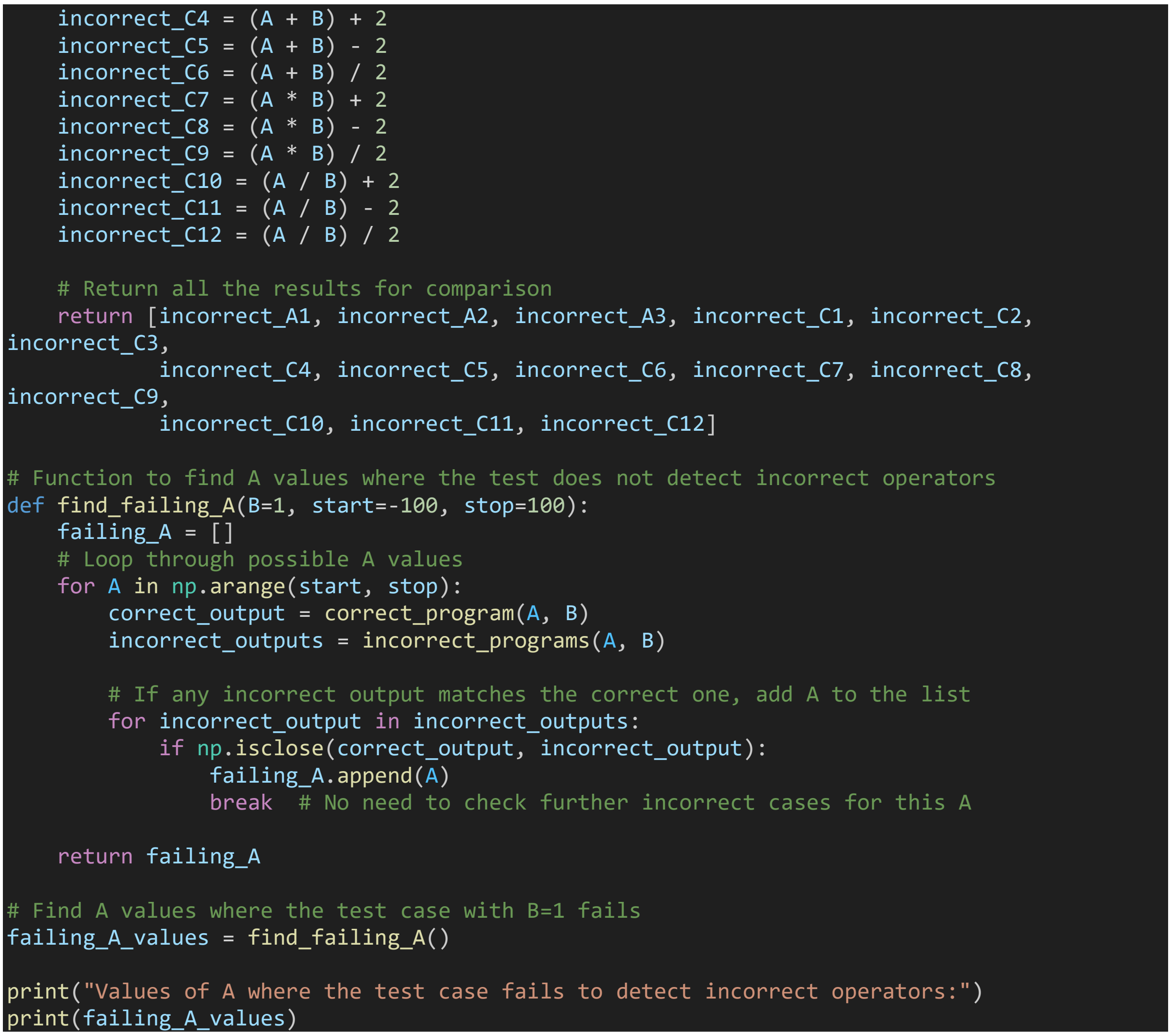
These test cases cover all possible operator errors by comparing correct outputs with those from incorrect operators, ensuring any mistakes are caught.

Task 4:

In this task, I developed a program to find A automatically.

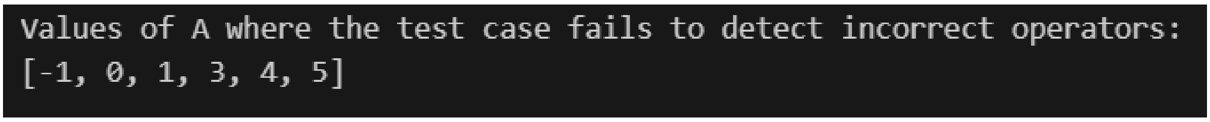


```
import numpy as np
# Correct program with original operators
def correct_program(A, B):
    A = A - B
    C = A * 2
    return C

# Define incorrect programs based on the scenarios you mentioned
def incorrect_programs(A, B):
    # Case 1: Incorrect use of the "-" operator
    incorrect_A1 = (A + B)*2
    incorrect_A2 = (A * B)*2
    incorrect_A3 = (A / B)*2

    # Case 2: Incorrect use of the "*" operator
    incorrect_C1 = (A - B)+2
    incorrect_C2 = (A - B)-2
    incorrect_C3 = (A - B)/2

    # Case 3: Both operators are incorrect
```



```
    incorrect_C4 = (A + B) + 2
    incorrect_C5 = (A + B) - 2
    incorrect_C6 = (A + B) / 2
    incorrect_C7 = (A * B) + 2
    incorrect_C8 = (A * B) - 2
    incorrect_C9 = (A * B) / 2
    incorrect_C10 = (A / B) + 2
    incorrect_C11 = (A / B) - 2
    incorrect_C12 = (A / B) / 2

    # Return all the results for comparison
    return [incorrect_A1, incorrect_A2, incorrect_A3, incorrect_C1, incorrect_C2,
incorrect_C3,
            incorrect_C4, incorrect_C5, incorrect_C6, incorrect_C7, incorrect_C8,
incorrect_C9,
            incorrect_C10, incorrect_C11, incorrect_C12]

# Function to find A values where the test does not detect incorrect operators
def find_failing_A(B=1, start=-100, stop=100):
    failing_A = []
    # Loop through possible A values
    for A in np.arange(start, stop):
        correct_output = correct_program(A, B)
        incorrect_outputs = incorrect_programs(A, B)

        # If any incorrect output matches the correct one, add A to the list
        for incorrect_output in incorrect_outputs:
            if np.isclose(correct_output, incorrect_output):
                failing_A.append(A)
                break  # No need to check further incorrect cases for this A

    return failing_A

# Find A values where the test case with B=1 fails
failing_A_values = find_failing_A()

print("Values of A where the test case fails to detect incorrect operators:")
print(failing_A_values)
```

Here are the outputs that the program found



```
Values of A where the test case fails to detect incorrect operators:
[-1, 0, 1, 3, 4, 5]
```

Figure 1

The script will print the values of A where the test case fails to detect the incorrect operators for B = 1.