4.1.1 Set Operations:

Algorithm:

Step 2: Input elements of Set A

Step 3: Input elements of Set B

Step 4: Display Union

set_a | set_b

Step 5: Display Intersection

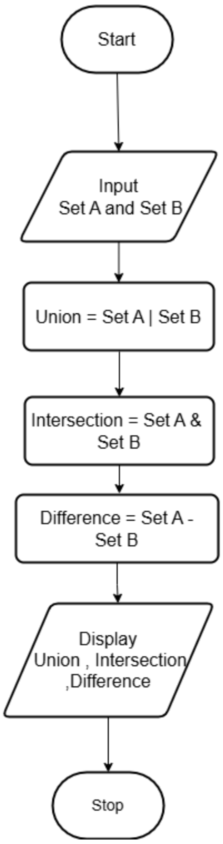
set_a & set_b

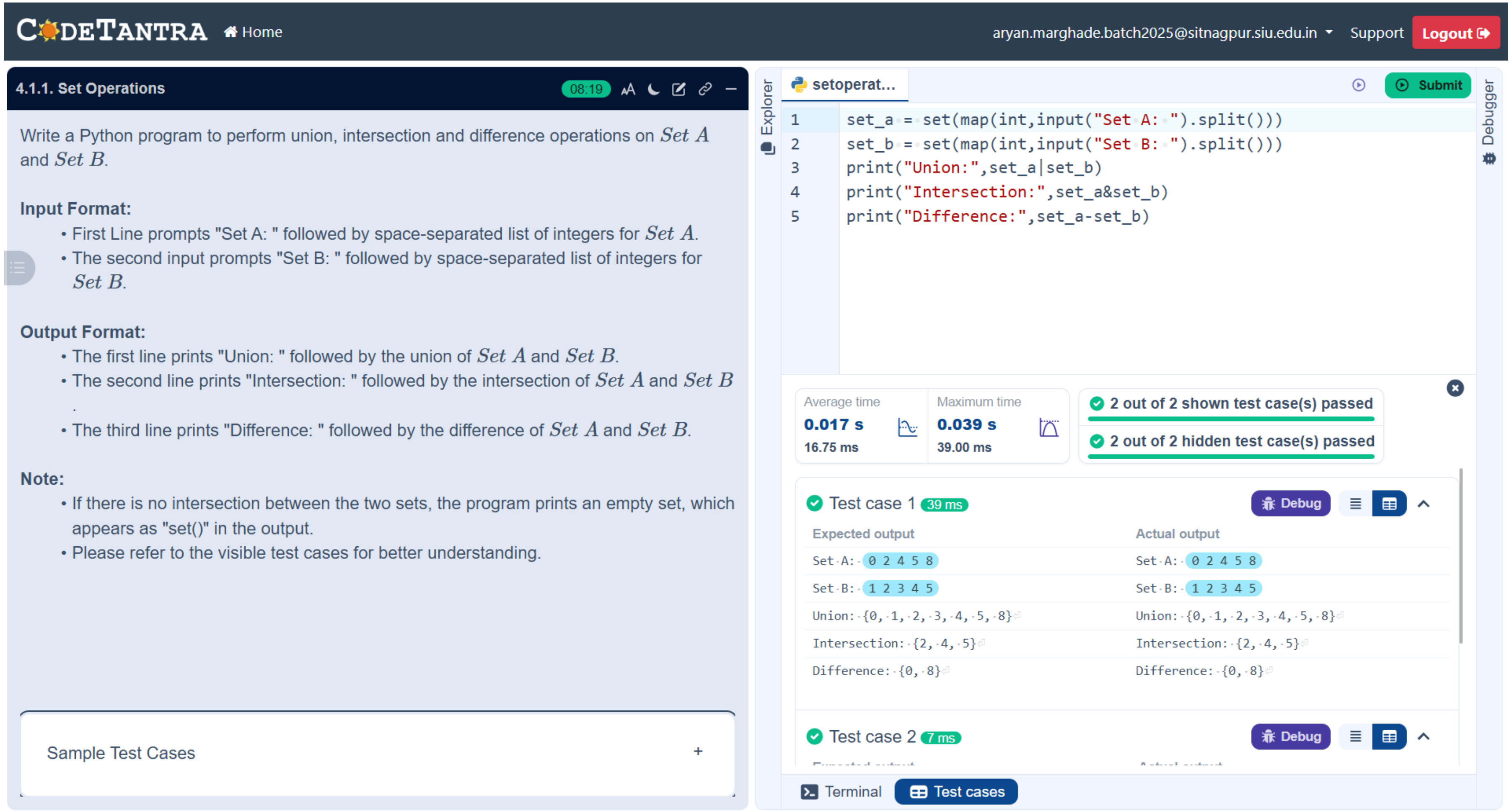
Step 6: Display Difference

set_a - set_b

Step 7: Stop

Flowchart:





4.1.1. Set Operations

Write a Python program to perform union, intersection and difference operations on $Set\ A$ and $Set\ B$.

Input Format:

- First Line prompts "Set A: " followed by space-separated list of integers for $Set\ A$.
- The second input prompts "Set B: " followed by space-separated list of integers for $Set\ B$.

Output Format:

- The first line prints "Union: " followed by the union of $Set\ A$ and $Set\ B$.
- The second line prints "Intersection: " followed by the intersection of $Set\ A$ and $Set\ B$.
- The third line prints "Difference: " followed by the difference of $Set\ A$ and $Set\ B$.

Note:

- If there is no intersection between the two sets, the program prints an empty set, which appears as "set()" in the output.
- Please refer to the visible test cases for better understanding.

Sample Test Cases +

```python
set_a = set(map(int,input("Set A: ").split()))
set_b = set(map(int,input("Set B: ").split()))
print("Union:",set_a|set_b)
print("Intersection:",set_a&set_b)
print("Difference:",set_a-set_b)
```

Average time 0.017 s 16.75 ms

Maximum time 0.039 s 39.00 ms

2 out of 2 shown test case(s) passed

2 out of 2 hidden test case(s) passed

Test case 1 39 ms

Expected output

```
Set A: 0 2 4 5 8
Set B: 1 2 3 4 5
Union: {0, 1, 2, 3, 4, 5, 8}
Intersection: {2, 4, 5}
Difference: {0, 8}
```

Actual output

```
Set A: 0 2 4 5 8
Set B: 1 2 3 4 5
Union: {0, 1, 2, 3, 4, 5, 8}
Intersection: {2, 4, 5}
Difference: {0, 8}
```

Test case 2 7 ms