

CODETANTRA

Home

tejas.mahurkar.batch2025@sitnagpur.siu.edu.in

Support

Logout

4.1.1. Set Operations

11:17

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

setoperat...

Submit

Debugger

```
1 set_a = set(map(int,input("Set A: ").split()))
2 set_b = set(map(int,input("Set B: ").split()))
3 union_set = set_a | set_b
4 intersection_set = set_a & set_b
5 difference_set = set_a - set_b
6 print("Union:", union_set)
7 print("Intersection:", intersection_set)
8 print("Difference:", difference_set)
9
```

Average time

0.018 s

18.25 ms

Maximum time

0.023 s

23.00 ms

2 out of 2 shown test case(s) passed

2 out of 2 hidden test case(s) passed

Test case 1

23 ms

Debug

Expected output

Actual output

Set A: {0, 2, 4, 5, 8}

Set A: {0, 2, 4, 5, 8}

Set B: {1, 2, 3, 4, 5}

Set B: {1, 2, 3, 4, 5}

Union: {0, 1, 2, 3, 4, 5, 8}

Union: {0, 1, 2, 3, 4, 5, 8}

Intersection: {2, 4, 5}

Intersection: {2, 4, 5}

Difference: {0, 8}

Difference: {0, 8}

Terminal

Test cases

Algorithm: Set Operations (Union, Intersection, Difference)

1. Start
2. Prompt the user to enter elements of Set A as space-separated integers.
3. Read the input and convert it into a set `set_a`.
4. Prompt the user to enter elements of Set B as space-separated integers.
5. Read the input and convert it into a set `set_b`.
6. Find the union of `set_a` and `set_b` using the union operation.
7. Find the intersection of `set_a` and `set_b` using the intersection operation.
8. Find the difference of `set_a` and `set_b` (elements present in Set A but not in Set B).
9. Display the union result.
10. Display the intersection result.
11. Display the difference result.
12. Stop

