Set Mutations ★



Q

```
X
                                 Your Set Mutations submission got 10.00 points.
                                                                                                     Tweet
                                            Try the next challenge | Try a Random Challenge
    Problem
                    Submissions
                                         Leaderboard
                                                               Editorial 🖰
We have seen the applications of union, intersection, difference and symmetric difference operations, but these operations do not make any changes or mutations
to the set.
We can use the following operations to create mutations to a set:
.update() or |=
Update the set by adding elements from an iterable/another set.
  >>> H = set("Hacker")
  >>> R = set("Rank")
  >>> H.update(R)
  >>> print H
  set(['a', 'c', 'e', 'H', 'k', 'n', 'r', 'R'])
.intersection_update() or &=
Update the set by keeping only the elements found in it and an iterable/another set.
  >>> H = set("Hacker")
  >>> R = set("Rank")
  >>> H.intersection_update(R)
  >>> print H
  set(['a', 'k'])
.difference_update() or -=
Update the set by removing elements found in an iterable/another set.
  >>> H = set("Hacker")
  >>> R = set("Rank")
  >>> H.difference_update(R)
  >>> print H
  set(['c', 'e', 'H', 'r'])
.symmetric_difference_update() or ^=
Update the set by only keeping the elements found in either set, but not in both.
  >>> H = set("Hacker")
  >>> R = set("Rank")
  >>> H.symmetric_difference_update(R)
  set(['c', 'e', 'H', 'n', 'r', 'R'])
TASK
```

Your task is to execute those operations and print the sum of elements from set $m{A}$.

You are given a set \pmb{A} and \pmb{N} number of other sets. These \pmb{N} number of sets have to perform some specific mutation operations on set \pmb{A} .

Input Format

The first line contains the number of elements in set $oldsymbol{A}$.

The second line contains the space separated list of elements in set $m{A}$.

The third line contains integer $oldsymbol{N}$, the number of other sets.

The next 2 * N lines are divided into N parts containing two lines each.

The first line of each part contains the space separated entries of the operation name and the length of the other set.

The second line of each part contains space separated list of elements in the other set.

```
0 < len(set(A)) < 1000
```

0 < len(otherSets) < 100

0 < N < 100

Output Format

Output the sum of elements in set A.

Sample Input

```
16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 24 52 4
intersection_update 10
2 3 5 6 8 9 1 4 7 11
update 2
55 66
symmetric_difference_update 5
22 7 35 62 58
difference_update 7
11 22 35 55 58 62 66
```

Sample Output

38

Explanation

After the first operation, (intersection_update operation), we get:

```
set A = set([1, 2, 3, 4, 5, 6, 7, 8, 9, 11])
```

After the second operation, (update operation), we get:

set
$$A = set([1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 55, 66])$$

After the third operation, (symmetric_difference_update operation), we get:

set
$$A = set([1, 2, 3, 4, 5, 6, 8, 9, 11, 22, 35, 55, 58, 62, 66])$$

After the fourth operation, (difference_update operation), we get:

$${\rm set}\, A = set([1,2,3,4,5,6,8,9])$$

The sum of elements in set $oldsymbol{A}$ after these operations is $oldsymbol{38}$.

```
Change Theme
                                                                               Python 3
1
   def apply_commands(a, operations):
2
       for o in operations:
3
           command = o[0]
4
5
           if command == 'update':
6
               a.update(o[2])
7
           elif command == 'intersection_update':
8
               a.intersection_update(o[2])
           elif command == 'difference_update':
```

```
a.difference_update(o[2])
elif command == 'symmetric_difference_update':
 11
                  a.symmetric_difference_update(o[2])
 12
 13
 14
          return sum(a)
 15
      if __name__ == '__main__':
 16
 17
          m = int(input())
          a = set(map(int, input().split()))
 18
 19
          n = int(input())
 20
 21
 22
          operations = []
 23
 24
          for _ in range(n):
                                                                                                      Line: 33 Col: 1
Run Code
                                                                                                      Submit Code
```

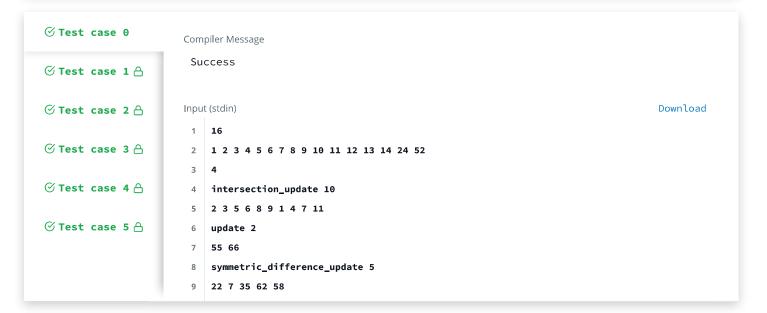
You have earned 10.00 points!

37/115 challenges solved.

32%



Congratulations You solved this challenge. Would you like to challenge your friends? Next Challenge



Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature