PRACTICE COMPETE JOBS LEADERBOARD Search punnammani1

Practice > Python > Sets > Set Mutations

Set Mutations ☆

77/115 challenges solved

Rank: 3480 | Points: 1535

Python ★★★★★

Your Set Mutations submission got 10.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions 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)
```

Author	DOSHI
Difficulty	Easy
Max Score	10
Submitted By	18354

NEED HELP?

- View discussions
- View editorial
- View top submissions

RATE THIS CHALLENGE

☆ ☆ ☆ ☆ ☆

MORE DETAILS

- Download problem statement
- Download sample test cases
- Suggest Edits

```
>>> print H
set(['c', 'e', 'H', 'n', 'r', 'R'])
```

TASK

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

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

Input Format

The first line contains the number of elements in set A.
The second line contains the space separated list of elements in set A.
The third line contains integer 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 $A = \mathit{set}([1, 2, 3, 4, 5, 6, 7, 8, 9, 11])$

After the second operation, (*update operation*), we get:
set $A = \mathit{set}([1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 55, 66])$

After the third operation, (*symmetric_difference_update operation*), we get:
set $A = \mathit{set}([1, 2, 3, 4, 5, 6, 8, 9, 11, 22, 35, 55, 58, 62, 66])$

After the fourth operation, (*difference_update operation*), we get:
set $A = \mathit{set}([1, 2, 3, 4, 5, 6, 8, 9])$

The sum of elements in set A after these operations is 38.

Current Buffer (saved locally, editable)

Python 3

```
temp=input()
s={i for i in input().split()}
n=int(input())
l=[input() for i in range(2*n)]
for i in range(0,2*n,2):
  if("symmetric_difference_update" in l[i]):
    temp=set(l[i+1].split())
    s^=temp
  elif("difference_update" in l[i]):
    temp=set(l[i+1].split())
    s-=temp
  elif("intersection_update" in l[i]):
    temp=set(l[i+1].split())
    s&=temp
  elif("update" in l[i]):
    temp=set(l[i+1].split())
    s|=temp
ans=0
for i in s:
  ans=ans+int(i)
print(ans)
```

Line: 21 Col: 6

Upload Code as File

☐ Test against custom input

Run Code

Submit Code

Python ★★★★★

You have earned 10.00 points!

77/115 challenges solved.

67%

Congratulations

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

Next Challenge

✓ Testcase 0 ✓ Testcase 1 ✓ Testcase 2 ✓ Testcase 3

6 Testcases

Input (stdin) Download

```
16
1 2 3 4 5 6 7 8 9 10 11 12
 13 14 24 52
4
```

Expected Output Download

```
38
```

Compiler Message

```
Success
```

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