



## Set .discard(), .remove() & .pop() ★

35 more points to get your gold badge!

Rank: 131666 | Points: 365/400

**Your Set .discard(), .remove() & .pop() submission got 10.00 points.**[Share](#)[Tweet](#)

You are now 35 points away from the gold level for your python badge.

[Try the next challenge](#) | [Try a Random Challenge](#)[Problem](#)[Submissions](#)[Leaderboard](#)[Editorial](#)

### .remove(x)

This operation removes element **x** from the set.

If element **x** does not exist, it raises a `KeyError`.

The `.remove(x)` operation returns `None`.

#### Example

```
>>> s = set([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> s.remove(5)
>>> print s
set([1, 2, 3, 4, 6, 7, 8, 9])
>>> print s.remove(4)
None
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
>>> s.remove(0)
KeyError: 0
```

### .discard(x)

This operation also removes element **x** from the set.

If element **x** does not exist, it **does not** raise a `KeyError`.

The `.discard(x)` operation returns `None`.

#### Example

```
>>> s = set([1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> s.discard(5)
>>> print s
set([1, 2, 3, 4, 6, 7, 8, 9])
>>> print s.discard(4)
None
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
>>> s.discard(0)
>>> print s
set([1, 2, 3, 6, 7, 8, 9])
```

### .pop()

This operation removes and return an arbitrary element from the set.

If there are no elements to remove, it raises a `KeyError`.

#### Example

```
>>> s = set([1])
>>> print s.pop()
1
>>> print s
set([])
>>> print s.pop()
KeyError: pop from an empty set
```

### Task

You have a non-empty set  $s$ , and you have to execute  $N$  commands given in  $N$  lines.

The commands will be pop, remove and discard.

### Input Format

The first line contains integer  $n$ , the number of elements in the set  $s$ .

The second line contains  $n$  space separated elements of set  $s$ . All of the elements are non-negative integers, less than or equal to 9.

The third line contains integer  $N$ , the number of commands.

The next  $N$  lines contains either pop, remove and/or discard commands followed by their associated value.

### Constraints

$$0 < n < 20$$

$$0 < N < 20$$

### Output Format

Print the sum of the elements of set  $s$  on a single line.

### Sample Input

```
9
1 2 3 4 5 6 7 8 9
10
pop
remove 9
discard 9
discard 8
remove 7
pop
discard 6
remove 5
pop
discard 5
```

### Sample Output

```
4
```

### Explanation

After completing these **10** operations on the set, we get set([4]). Hence, the sum is **4**.

**Note:** Convert the elements of set  $s$  to integers while you are assigning them. To ensure the proper input of the set, we have added the first two lines of code to the editor.

[Change Theme](#)

Python 3



```
1 def execute_commands(s, commands):
2     for c in commands:
3         parts = c.split()
4         command = parts[0]
```

```
5
6     if command == 'remove':
7         value = int(parts[1])
8
9         if value in s:
10            s.remove(value)
11    elif command == 'discard':
12        value = int(parts[1])
13
14        s.discard(value)
15    elif command == 'pop':
16        if len(s):
17            s.pop()
18
19    return sum(s)
20
21 if __name__ == '__main__':
22     n = int(input())
23     s = set(map(int, input().split()))
24
```

Line: 10 Col: 32

[Upload Code as File](#) ☐ [Test against custom input](#)[Run Code](#)[Submit Code](#)

You have earned 10.00 points!

You are now 35 points away from the gold level for your python badge.

81%

365/400



## Congratulations

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

[Next Challenge](#)

## Earn a certificate in Python

Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile

[Get Certified](#)

### Test case 0

[Test case 1](#)[Test case 2](#)[Test case 3](#)[Test case 4](#)[Test case 5](#)

Compiler Message

Success

Input (stdin)

```
1 9
2 1 2 3 4 5 6 7 8 9
3 10
4 pop
5 remove 9
6 discard 9
7 discard 8
8 remove 7
9 pop
```

[Download](#)