



# 1. Python: Multiset Implementation



ALL

A *multiset* is the same as a set except that an element might occur more than once in a multiset.

1

Implement a multiset data structure in Python. Given a template for the *Multiset* class, implement 4 methods:

10



- 1 • *add(self, val)*: adds *val* to the multiset
- 2 • *remove(self, val)*: if *val* is in the multiset, removes *val* from the multiset; otherwise, do nothing
- 3 • *\_\_contains\_\_(self, val)*: returns True if *val* is in the multiset; otherwise, it returns False
- 4 • *\_\_len\_\_(self)*: returns the number of elements in the multiset

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27



Additional methods are allowed as necessary.

The implementations of the 4 required methods will be tested by a provided code stub on several input files. Each input file contains *several*/operations, each of one of the below types. Values returned by *query* and *size* operations are appended to a *result* list, which is printed as the output by the provided code stub.

- add val: calls *add(val)* on the *Multiset* instance
- remove val: calls *remove(val)* on the *Multiset* instance
- query val: appends the result of expression *val*/in *m*, where *m* is an instance of *Multiset*, and appends the value of that expression to the *result* list
- size: calls *len(m)*, where *m* is an instance of *Multiset*, and appends the returned value to the *result* list

Test

Results

Complete the class *Multiset* in the editor below with the 4 methods given above (*add*, *remove*,



Type here to search



1h 27m  
left

Complete the class `Multiset` in the editor below with the 4 methods given above (`add`, `remove`, `_contains_`, and `_len_`).

 Info

 Autocom



## Constraints

- $1 \leq$  number of operations in one test file  $\leq 10^5$
- if `val` is a parameter of operation, then `val` is an integer and  $1 \leq val \leq 10^9$

ALL



## ▼ Input Format Format for Custom Testing

1 In the first line, there is a single integer,  $q$ , denoting the number of queries.

2 Then,  $q$  lines follow. In the  $i^{th}$  of them, there is a string denoting an operation and optionally an integer denoting the parameter of the operation.

## ▼ Sample Case 0

### Sample Input

STDIN	Function
-----	-----
12	→ number of queries, $q = 12$
query 1	→ operations = ["query 1", "add 1", ..., "query 2", "size"]
add 1	
query 1	
remove 1	
query 1	
add 2	
add 2	
size	
query 2	
remove 2	
query 2	
size	

### Sample Output

Test

Results

1 > #

10

11

12

13

14

15

16

17

any

18

19

20

21

retu

22

23

24

25

26

27 > if \_

1h 26m  
left

query 2

size

Info

Autoc

## Sample Output

False

True

False

ALL

2

True

True

i

1

## Explanation

1

There are 12 operations to be performed. Start with an empty multiset:  $\text{multiset} = []$ .

2

1. The first operation asks if 1 is in the multiset. It is not, so False is appended to the result:  $\text{result} = [\text{False}]$ .
2. The second operation adds 1 to the multiset:  $\text{multiset} = [1]$ .
3. The third operation asks if 1 is in the multiset. It is now, so True is appended to the result:  $\text{result} = [\text{False}, \text{True}]$ .
4. The fourth operation removes 1 from the multiset:  $\text{multiset} = []$ .
5. The fifth operation asks if 1 is in the multiset. It is not, so False is appended to the result:  $\text{result} = [\text{False}, \text{True}, \text{False}]$ .
6. The sixth operation adds 2 to the multiset:  $\text{multiset} = [2]$ .
7. The seventh operation adds 2 to the multiset:  $\text{multiset} = [2, 2]$ .
8. The next operation asks what is the size of the multiset:  $\text{result} = [\text{False}, \text{True}, \text{False}, 2]$ .
9. The next operation asks if 2 is in the multiset. It is, so True is appended to the result:  $\text{result} = [\text{False}, \text{True}, \text{False}, 2, \text{True}]$ .
10. The next operation removes 2 from the multiset:  $\text{multiset} = [2]$
11. The next operation asks if 2 is in the multiset. It is, so True is appended to the result:  $\text{result} = [\text{False}, \text{True}, \text{False}, 2, \text{True}, \text{True}]$ .
12. Finally, the last operation asks for the size of the multiset and the length, 1, is appended to the result.  
 $\text{result} = [\text{False}, \text{True}, \text{False}, 2, \text{True}, \text{True}, 1]$

1 &gt;

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27 &gt; if

1h 26m  
left



ALL



1

### ▼ Sample Case 1

2

#### Sample Input

STDIN      Function

-----      -----

```
3      →    number of queries, q = 3
size    →    operations = ["size", "add 17", "size"]
add 17
size
```

#### Sample Output

```
0
1
```

#### Explanation

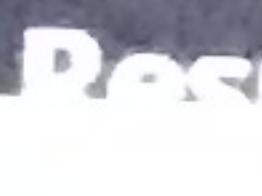
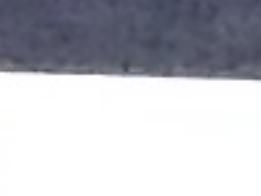
There are 3 operations to be performed. Start with the empty multiset: *multiset* = [].

1. The first asks what is the size of the multiset. Since the multiset is empty, 0 is appended to the result: *result* = [0].
2. The second operation adds 17 to the multiset: *multiset* = [17].
3. The third operation asks what is the size of the multiset. 1 is appended to the result: *result* = [0, 1].

Info Pyth  
Autocomplet  
1 > #!/bi  
10 class  
11  
12 de  
13  
14  
15  
16  
17 any  
18  
19  
20 def  
21 returns  
22  
23  
24 def  
25  
26  
27 > if \_\_nam



Type here to search



Test  
Results

to the

s 2 to the

at is the  
[False,

is in the  
ended  
True,

s 2 from

is in the  
ended  
True,

ks for  
the  
e result.

True,

ries,

]

Info Python 3 Autocomplete Ready



```
1 > #!/bin/python3...
10 class Multiset:
11
12     def add(self, val):
13         # adds one occurrence of val from the multiset, if any
14         pass
15
16     def remove(self, val):
17         # removes one occurrence of val from the multiset, if any
18         pass
19
20     def __contains__(self, val):
21         # returns True when val is in the multiset, else returns False
22         return False
23
24     def __len__(self):
25         # returns the number of elements in the multiset
26         return 0
27
28 if __name__ == '__main__':
29     def performOperations(operations):
30         m = Multiset()
31         result = []
32         for op_str in operations:
33             elems = op_str.split()
34             if elems[0] == 'size':
35                 result.append(len(m))
36             else:
37                 op, val = elems[0], int(elems[1])
38                 if op == 'query':
39                     result.append(val in m)
40                 elif op == 'add':
41                     ...
```

Line: 14

Test Results

Custom Input

Run Code

Run Tests

Sub



Mostly clear

ENG

1858  
07-04-2022

to the

s 2 to the

at is the

[False,

is in the

ended

True,

; 2 from

is in the

ended

True,

ks for

the

result.

True,

ries,

]

Info

Python 3

Autocomplete Ready



```
1 > #!/bin/python3...
10 class Multiset:
11
12     def add(self, val):
13         # adds one occurrence of val from the multiset, if any
14         pass
15
16     def remove(self, val):
17         # removes one occurrence of val from the multiset, if any
18         pass
19
20     def __contains__(self, val):
21         # returns True when val is in the multiset, else returns False
22         return False
23
24     def __len__(self):
25         # returns the number of elements in the multiset
26         return 0
27
28 if __name__ == '__main__':
29     def performOperations(operations):
30         m = Multiset()
31         result = []
32         for op_str in operations:
33             elems = op_str.split()
34             if elems[0] == 'size':
35                 result.append(len(m))
36             else:
37                 op, val = elems[0], int(elems[1])
38                 if op == 'query':
39                     result.append(val in m)
elif op == 'add':
    ...
```

Line: 14

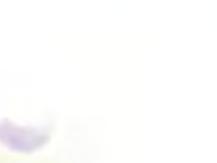
Test Results

Custom Input

Run Code

Run Tests

Sub



18:58  
Mostly clear ENG  
07-04-2022

Info

Python 3

Autocomplete Ready

```
44     def __len__(self):
45         # returns the number of elements in the multiset
46         return 0
47 
48     if __name__ == '__main__':
49         def performOperations(operations):
50             m = Multiset()
51             result = []
52             for op_str in operations:
53                 elems = op_str.split()
54                 if elems[0] == 'size':
55                     result.append(len(m))
56                 else:
57                     op, val = elems[0], int(elems[1])
58                     if op == 'query':
59                         result.append(val in m)
60                     elif op == 'add':
61                         m.add(val)
62                     elif op == 'remove':
63                         m.remove(val)
64             return result
65 
66     q = int(input())
67     operations = []
68     for _ in range(q):
69         operations.append(input())
70 
71     result = performOperations(operations)
72 
73     fptr = open(os.environ['OUTPUT_PATH'], 'w')
74     fptr.write('\n'.join(map(str, result)))
75     fptr.write('\n')
76     fptr.close()
```

Test Results

Custom Input

Run Code

Run Tests

f3j08q1

Python 2

● Autocomplete Ready ⓘ

```
1 >#!/bin/python...
10 class Multiset(object):
11     def __init__(self):
12         self.M=[]
13
14     def add(self, val):
15         # adds one occurrence of val from the multiset, if any
16         self.M.append(val)
17
18     def remove(self, val):
19         # removes one occurrence of val from the multiset, if any
20         if val in self.M:
21             self.M.remove(val)
22
23     def __contains__(self, val):
24         # returns True when val is in the multiset, else returns False
25         if val in self.M:
26             return True
27         return False
28
29     def __len__(self):
30         # returns the number of elements in the multiset
31         return len(self.M)
32
33 > if __name__ == '__main__':...
```

h

Info

Python 3

Autocomplete Ready



```
1 > #!/bin/python3...
10
11 class Multiset:
12     def __init__(self):
13         self.M=[]
14     def add(self, val):
15         # adds one occurrence of val from the multiset, if any
16         self.M.append(val)
17
18     def remove(self, val):
19         # removes one occurrence of val from the multiset, if any
20         if val in self.M :
21             self.M.remove(val)
22
23     def __contains__(self, val):
24         # returns True when val is in the multiset, else returns False
25         if val in self.M:
26             return True
27         return False
28
29     def __len__(self):
30         # returns the number of elements in the multiset
31         return len(self.M)
32
33 ~ if __name__ == '__main__':
34     def performOperations(operations):
35         m = Multiset()
36         result = []
37         for op_str in operations:
38             elems = op_str.split()
39             if elems[0] == 'size':
```

Info

Python 3



Autocomplete Ready



```
31         return len(self.M)
32
33 if __name__ == '__main__':
34     def performOperations(operations):
35         m = Multiset()
36         result = []
37         for op_str in operations:
38             elems = op_str.split()
39             if elems[0] == 'size':
40                 result.append(len(m))
41             else:
42                 op, val = elems[0], int(elems[1])
43                 if op == 'query':
44                     result.append(val in m)
45                 elif op == 'add':
46                     m.add(val)
47                 elif op == 'remove':
48                     m.remove(val)
49     return result
50
51 q = int(input())
52 operations = []
53 for _ in range(q):
54     operations.append(input())
55
56 result = performOperations(operations)
57
58 fptr = open(os.environ['OUTPUT_PATH'], 'w')
59 fptr.write('\n'.join(map(str, result)))
60 fptr.write('\n')
61 fptr.close()
```