Collections.deque()

by DOSHI

Problem Submissions Leaderboard Discussions Editorial

collections.deque()

A *deque* is a double-ended queue. It can be used to add or remove elements from both ends.

Deques support thread safe, memory efficient appends and pops from either side of the deque with approximately the same $O(1)$ performance in either direction.

Click on the link to learn more about **deque() methods**.
Click on the link to learn more about various approaches to working with deques: **Deque Recipes**.

Example

Code

```
>>> from collections import deque
>>> d = deque()
>>> d.append(1)
>>> print d
deque([1])
>>> d.appendleft(2)
>>> print d
deque([2, 1])
>>> d.clear()
>>> print d
deque([])
>>> d.extend('1')
>>> print d
deque(['1'])
>>> d.extendleft('234')
>>> print d
deque(['4', '3', '2', '1'])
>>> d.count('1')
1
>>> d.pop()
'1'
>>> print d
deque(['4', '3', '2'])
>>> d.popleft()
'4'
>>> print d
deque(['3', '2'])
>>> d.extend('7896')
>>> print d
deque(['3', '2', '7', '8', '9', '6'])
>>> d.remove('2')
>>> print d
deque(['3', '7', '8', '9', '6'])
>>> d.reverse()
>>> print d
deque(['6', '9', '8', '7', '3'])
>>> d.rotate(3)
>>> print d
deque(['8', '7', '3', '6', '9'])
```

Task

Perform *append*, *pop*, *popleft* and *appendleft* methods on an empty deque d.

Input Format

The first line contains an integer N, the number of operations.
The next N lines contains the space separated names of methods and their values.

Constraints

$0 < N \leq 100$

Output Format

Print the space separated elements of deque d.

Sample Input

```
6
append 1
append 2
append 3
appendleft 4
pop
popleft
```

Sample Output

```
1 2
```


Current Buffer (saved locally, editable) Python 3

```python
from collections import deque

n=int(input().strip())
dq = deque()

for i in range(n):
    items = input().strip().split(' ')
    if len(items) == 2:
        action,num = items
    else:
        action = items[0]

    # num = int(num)
    if action =='append':
        dq.append(num)
    elif action == 'appendleft':
        dq.appendleft(num)
    elif action == 'pop':
        dq.pop()
    elif action == 'popleft':
        dq.popleft()

[print(x,end=' ') for x in dq]
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
6
append 1
append 2
append 3
appendleft 4
pop
popleft
```

Your Output (stdout)

```
1 2
```

Expected Output

```
1 2
```