

My HackerRank username: @n\_marthely

Before submitting my code:

HackerRank

PRACTICE

CERTIFICATION

COMPETE

JOBS

LEADERBOARD

Search

n\_marthely

Practice > Data Structures > Queues > Queue using Two Stacks

65 more points to get your next star!  
Rank: 1697079 | Points: 35/100

Queue using Two Stacks

Problem | Submissions | Leaderboard | Discussions | Editorial

A queue is an abstract data type that maintains the order in which elements were added to it, allowing the oldest elements to be removed from the front and new elements to be added to the rear. This is called a First-In-First-Out (FIFO) data structure because the first element added to the queue (i.e., the one that has been waiting the longest) is always the first one to be removed.

A basic queue has the following operations:

- Enqueue: add a new element to the end of the queue.
- Dequeue: remove the element from the front of the queue and return it.

In this challenge, you must first implement a queue using two stacks. Then process  $q$  queries, where each query is one of the following 3 types:

1. 1  $x$ : Enqueue element  $x$  into the end of the queue.
2. 2: Dequeue the element at the front of the queue.
3. 3: Print the element at the front of the queue.

**Input Format**

The first line contains a single integer,  $q$ , denoting the number of queries.

Each line  $i$  of the  $q$  subsequent lines contains a single query in the form described in the problem statement above. All three queries start with an integer denoting the query **type**, but only query 1 is followed by an additional space-separated value,  $x$ .

Author: saikiran9194

Difficulty: Medium

Max Score: 30

Submitted By: 47734

NEED HELP?

[View discussions](#)

[View editorial](#)

[View top submissions](#)

RATE THIS CHALLENGE

[Download problem statement](#)

[Download sample test cases](#)

[Suggest Edits](#)

Change Theme Python 3

```
1 class MyQueue(object):
2     def __init__(self):
3         self.q1 = []
4         self.q2 = []
5
6     def peek(self):
7         if not self.q2:
8             while self.q1:
9                 self.q2.append(self.q1.pop())
10
11         return self.q2[-1]
12
13     def pop(self):
14         if not self.q2:
15             while self.q1:
16                 self.q2.append(self.q1.pop())
17
18         return self.q2.pop()
19
20     def put(self, value):
21         self.q1.append(value)
22
23 queue = MyQueue()
24 t = int(input())
25 for line in range(t):
26     values = map(int, input().split())
27     values = list(values)
28     if values[0] == 1:
29         queue.put(values[1])
```

Line: 33 Col: 28

Upload Code as File

Test against custom input

Run Code

Submit Code

```
26 values = map(int, input().split())
27 values = list(values)
28 if values[0] == 1:
29     queue.put(values[1])
30 elif values[0] == 2:
31     queue.pop()
32 else:
33     print(queue.peek())
```

Line: 33 Col: 28

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

[Run Code](#) [Submit Code](#)

### Congratulations!


You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin) [Download](#)

```
1 10
2 1 42
3 2
4 1 14
5 3
6 1 28
7 3
8 1 60
9 1 78
10 2
11 2
```

After submitting my code :

 You have earned 30.00 points!  
You are now 35 points away from the 2nd star for your problem solving badge.


50% 65/100


### Congratulations


You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)


[Next Challenge](#)


✓ Test case 0


✓ Test case 1 

✓ Test case 2 

✓ Test case 3 

✓ Test case 4 

✓ Test case 5 

✓ Test case 6 


Compiler Message

Success

Input (stdin) [Download](#)

```
1 10
2 1 42
3 2
4 1 14
5 3
6 1 28
7 3
8 1 60
9 1 78
```

Windows taskbar: Type here to search, 3:31 PM, 2/6/2021



You have earned 30.00 points!  
You are now 35 points away from the 2nd star for your problem solving badge.

50%

65/100

### Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

[Next Challenge](#)

Test case 9

Test case 10

Test case 11

Test case 12

Test case 13

Test case 14

Test case 15

```

1  14
2  14
3  2
4  1 14
5  3
6  1 28
7  3
8  1 60
9  1 78
10 2
11 2

```

Expected Output

Download

```

1  14
2  14

```

HackerRank


PRACTICE CERTIFICATION COMPETE JOBS LEADERBOARD

n\_marthely

Practice > Data Structures > Queues > Queue using Two Stacks

35 more points to get your next star!

Rank: 1357480 | Points: 65/100



## Queue using Two Stacks

Your Queue using Two Stacks submission got 30.00 points.

[f](#)
[t](#)

You are now 35 points away from the 2nd star for your problem solving badge.

[Try the next challenge](#) | [Try a Random Challenge](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

A queue is an abstract data type that maintains the order in which elements were added to it, allowing the oldest elements to be removed from the front and new elements to be added to the rear. This is called a First-In-First-Out (FIFO) data structure because the first element added to the queue (i.e., the one that has been waiting the longest) is always the first one to be removed.

A basic queue has the following operations:

- Enqueue: add a new element to the end of the queue.
- Dequeue: remove the element from the front of the queue and return it.

In this challenge, you must first implement a queue using two stacks. Then process  $q$  queries, where each query is one of the following 3 types:

- 1 x: Enqueue element  $x$  into the end of the queue.

Author: saikiran9194

Difficulty: Medium

Max Score: 30

Submitted By: 47734

NEED HELP?

[View discussions](#)

[View editorial](#)

[View top submissions](#)

RATE THIS CHALLENGE

**HackerRank** PRACTICE CERTIFICATION COMPETE JOBS LEADERBOARD

Practice > Data Structures > Queues > Queue using Two Stacks

**Queue using Two Stacks** ☆

35 more points to get your next star!

Rank: 1357480 | Points: 65/100

Your Queue using Two Stacks submission got 30.00 points. [? Partner](#) [Tweet](#)

You are now 35 points away from the 2nd star for your problem solving badge.

[Try the next challenge](#) | [Try a Random Challenge](#)

Problem Submissions Leaderboard Discussions Editorial

You made this submission 2 minutes ago.  
**Score:** 30.00 **Status:** Accepted

People who solved Queue using Two Stacks attempted this next:

**Castle on the Grid**

Determine the number of steps to move a castle to the goal position on a given grid.

[Solve Challenge](#)

NEED HELP?

- [View discussions](#)
- [View editorial](#)
- [View top submissions](#)

Type here to search

3:24 PM 2/6/2021

My code:

```
class MyQueue(object):
    def __init__(self):
        self.q1 = []
        self.q2 = []

    def peek(self):
        if not self.q2:
            while self.q1:
                self.q2.append(self.q1.pop())

        return self.q2[-1]

    def pop(self):
        if not self.q2:
            while self.q1:
                self.q2.append(self.q1.pop())

        return self.q2.pop()
```

```
def put(self, value):
    self.q1.append(value)

queue = MyQueue()
t = int(input())
for line in range(t):
    values = map(int, input().split())
    values = list(values)
    if values[0] == 1:
        queue.put(values[1])
    elif values[0] == 2:
        queue.pop()
    else:
        print(queue.peek())
```