

HackerrankName: Modester Mwangi

Practice > Data Structures > Queues > Queue using Two Stacks

Queue using Two Stacks ☆

34 more points to get your next star!



Rank: 1342651 | Points: 66/100



Your Queue using Two Stacks submission got 30.00 points.



You are now 34 points away from the 2nd star for your problem solving badge.

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



You have earned 30.00 points!

You are now 34 points away from the 2nd star for your problem solving badge.

51%

66/100

Congratulations



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

[Next Challenge](#)

Earn a certificate in Problem Solving

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

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
```

Congratulations

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

[Next Challenge](#)

✓ Test case 7

✓ Test case 8

✓ Test case 9

✓ Test case 10

✓ Test case 11

✓ Test case 12

✓ Test case 13

✓ Test case 14

```
4 1 42
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
```

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 class stack:
3     def __init__(self):
4         self.elements = []
5
6     def push(self,item):
7         return self.elements.append(item)
8
9     def pop(self):
10         return self.elements.pop()
11
12     def __len__(self):
13         return len(self.elements)
14
15     def top(self):
16         if self.elements:
17             return self.elements[-1]
18         return None
19
20 class Queue:
21     def __init__(self):
22         self.stack_add = stack()
23         self.stack_delete = stack()
24     def Enqueue(self,item):
25         self.stack_delete.push(item)
```

Line: 38 Col: 33

Hackerrank Code:

```
class stack:
    def __init__(self):
        self.elements = []

    def push(self, item):
        return self.elements.append(item)

    def pop(self):
        return self.elements.pop()

    def __len__(self):
        return len(self.elements)

    def top(self):
        if self.elements:
            return self.elements[-1]
        return None

class Queue:
    def __init__(self):
        self.stack_front = stack()
        self.stack_rear = stack()

    #Enqueue
    def Enqueue(self, item):
        self.stack_rear.push(item)

    #Dequeue
    def Dequeue(self):
        if self.stack_front:
            return self.stack_front.pop()
        return self.replace_stack_front().pop()

    def peek(self):
```

```

        if self.stack_front:
            return self.stack_front.top()
        return self.replace_stack_front().top()

def replace_stack_front(self):
    while self.stack_rear:
        self.stack_front.push(self.stack_rear.pop())
    return self.stack_front

#read through inputs and split it and store in list
def read_inputs():
    t = input().strip()
    line= t.split()
    t_type = int(line[0])

    if len(line) == 1:
        return(t_type, None)
    num = int(line[1])
    return t_type,num

def main():
    q = Queue()
    element = int(input().strip())

#loop through the list and print the front element
    for i in range(element):
        t_type,num = read_inputs()
        if t_type == 1:
            q.Enqueue(num)

        elif t_type == 2:
            q.Dequeue()

```

```
        elif t_type == 3:
            print(q.peek())
if __name__ == "__main__":
    main()
```