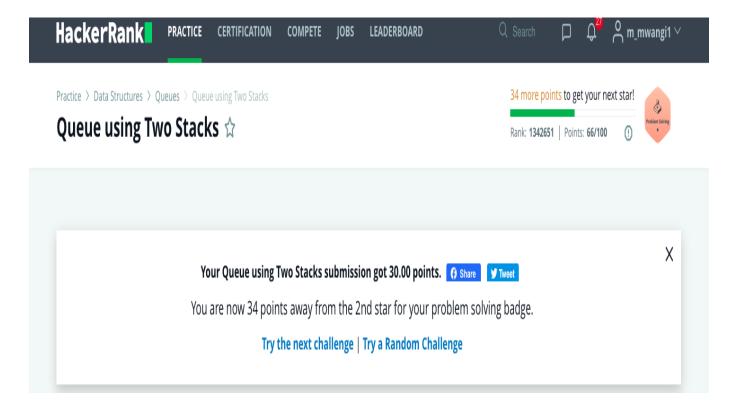
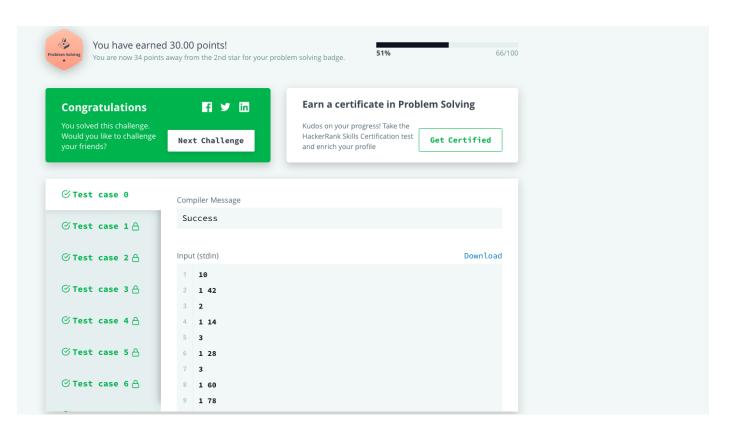
HackerrankName: Modester Mwangi





```
Congratulations
                                                                      Next Challenge
You solved this challenge. Would you like to challenge your friends? f in

    ∀ Test case 7 
    □

                      3 2
                      4 1 14
⊘Test case 8 🖰
                      5 3
                      6 1 28
⊘Test case 9 🖰
                      7 3
⊘Test case 10 △
                      8 1 60
                      9 1 78
                      10 2
⊘ Test case 11 △
                      11 2
⊗ Test case 12 🖰
                      Expected Output
                                                                             Download
⊘Test case 13 🖰
                      1 14
                      2 14
⊘Test case 14 🖰
```

```
▼ 10 €3 69

                                             Change Theme Python 3
# Enter your code here. Read input from STDIN. Print output to STDOUT
 2 ∨ class stack:
      def __init__(self):
        self.elements = []
      def push(self,item):
6 🗸
      return self.elements.append(item)
      def pop(self):
9 🗸
      return self.elements.pop()
12 🗸
      def __len__(self):
        return len(self.elements)
14
15 🗸
        def top(self):
        if self.elements:
16 🗸
        return self.elements[-1]
return None
17
18
20 ∨ class Queue:
21 ∨ def __init__(self):
        self.stack_add = stack()
self.stack_delete = stack()
def Enqueue(self,item):
         self.stack_delete.nush(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()
```