

Compile

```
shuqiny2@circinus-14 05:26:08 ~/253P/HW7/src
$ javac RomanNumeralConverter.java
shuqiny2@circinus-14 05:26:14 ~/253P/HW7/src
$ java RomanNumeralConverter
```

Input:

9
IX
476
CDLXXVI

Output:

IX
9
CDLXXVI
476

```
shuginy2@circinus-14 05:26:14 ~/253P/HW7/src
$ java RomanNumeralConverter
9
IX
IX
9
476
CDLXXVI
CDLXXVI
476
```

Input:

0
-1
4000

Output:

UNDEFINED
Error
Error

```
shuginy2@circinus-14 05:26:14 ~/253P/HW7/src
$ java RomanNumeralConverter
9
IX
IX
9
476
CDLXXVI
CDLXXVI
476
0
UNDEFINED
-1
Invalid Roman numerals! Your input has illegal characters.
4000
Invalid number! Your input should NOT exceed 3999.
```

Input:

ABC
X123
MMMM

Output:

Error
Error
Error

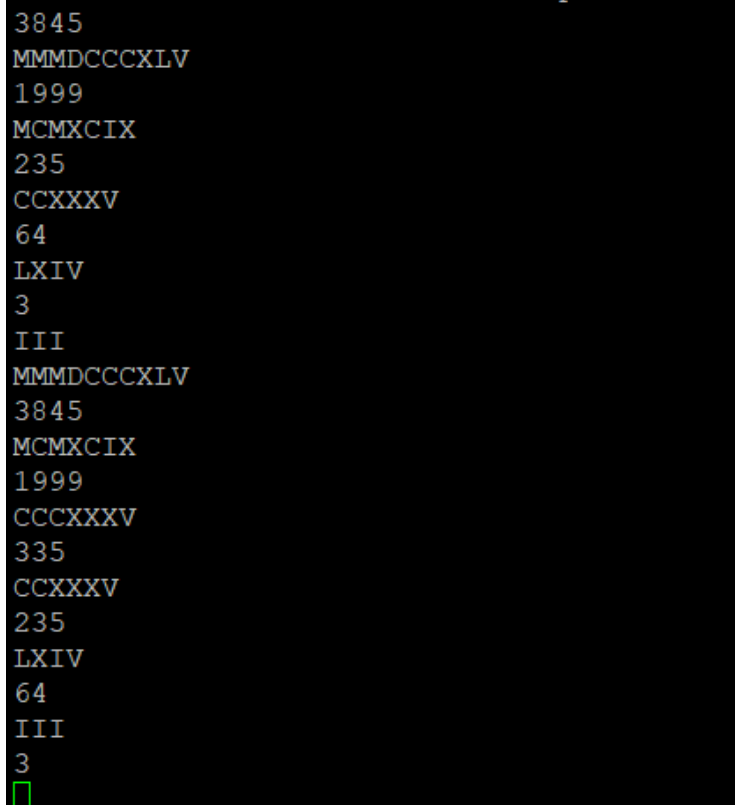
```
ABC
Invalid Roman numerals! Your input has illegal characters.
X123
Invalid Roman numerals! Your input has illegal characters.
MMM
3000
MMMM
Invalid Roman numerals! Your input should NOT exceed the limited value.
3045
```

Input:

3845
1999
235
64
3
MMMCCCCXLV
MCMXCIX
CCXXXV
LXIV
III

Output:

MMMCCCCXLV
MCMXCIX
CCXXXV
LXIV
III
3845
1999
235
64
3



```
3845
MMMCCCCXLV
1999
MCMXCIX
235
CCXXXV
64
LXIV
3
III
MMMCCCCXLV
3845
MCMXCIX
1999
CCXXXV
335
CCXXXV
235
LXIV
64
III
3
█
```

Leetcode 778. Swim in Rising Water

LeetCode

Explore 12 Problems new Mock new Contest Discuss Store

Description

Solution

Discuss (234)

Submissions

Success

Details >

Runtime: **4 ms**, faster than **91.57%** of Java online submissions for Swim in Rising Water.

Memory Usage: **39.5 MB**, less than **25.30%** of Java online submissions for Swim in Rising Water.




Next challenges:

The Maze

Minimize Malware Spread

Maximum Nesting Depth of Two Valid Parentheses Strings

Show off your acceptance:



Time Submitted	Status	Runtime	Memory	Language
11/18/2020 18:19	Accepted	4 ms	39.5 MB	java
11/18/2020 17:47	Accepted	166 ms	39.7 MB	java
11/18/2020 16:33	Wrong Answer	N/A	N/A	java
11/18/2020 16:31	Wrong Answer	N/A	N/A	java

Problems

Pick One

< Prev

778/1659

Next >

Java

Autocomplete

Testcase

Run Code Result

Debugger

Accepted

Runtime: 0 ms

Your input

[[0,2],[1,3]]
[[0,1,2,3,4],[24,23,22,21,5],[12,13,14,15,16],[11,17,18,19,20],[10,9,8,7,6]]

Output

3
16

Diff

Expected

3
16

Console

How to create a testcase

Run Code

Submit

Leetcode 42. Trapping Rain Water

LeetCode

Explore

Problems

Mock

Contest

Discuss

Store

☆ Premium

🔍

🔔

👤

Description

Solution

Discuss (999+)

Submissions

Success

Details >

Runtime: 1 ms, faster than 84.86% of Java online submissions for Trapping Rain Water.

Memory Usage: 38.5 MB, less than 75.76% of Java online submissions for Trapping Rain Water.

Next challenges:

Container With Most Water

Product of Array Except Self

Trapping Rain Water II

Pour Water

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language
11/18/2020 13:57	Accepted	1 ms	38.5 MB	java

Problems

✖ Pick One

< Prev

42/1659

Next >

i Java

Autocomplete

```
28     maxLeft = height[i];
29     maxLeftIndex = i;
30 }
31 }
32 for (int i = maxLeftIndex + 1; i < maxIndex; i++)
33     sum += maxLeft - height[i];
34 sumUpLeft(height, maxLeftIndex);
35 }
36
37 void sumUpRight(int[] height, int maxIndex) {
38     if (maxIndex >= height.length - 1)
39         return;
40     int maxRight = Integer.MIN_VALUE;
41     int maxRightIndex = 0;
42     for (int i = maxIndex + 1; i < height.length; i++) {
43         if (height[i] > maxRight) {
44             maxRight = height[i];
45             maxRightIndex = i;
46         }
47     }
48     for (int i = maxIndex + 1; i < maxRightIndex; i++)
49         sum += maxRight - height[i];
50     sumUpRight(height, maxRightIndex);
51 }
```

Testcase

Run Code Result

Debugger

Accepted

Runtime: 0 ms

Your input

[0,1,0,2,1,0,1,3,2,1,2,1]

Output

6

Diff

Expected

6

Console

How to create a testcase

Run Code

Submit

LeetCode 1224. Maximum Equal Frequency

LeetCode

Explore

Problems

Mock

Contest

Discuss

Store

Success Details >

Runtime: 8 ms, faster than 78.38% of Java online submissions for Maximum Equal Frequency.

Memory Usage: 49.7 MB, less than 51.35% of Java online submissions for Maximum Equal Frequency.

Next challenges:

Repeated DNA Sequences

Sentence Similarity

Subarrays with K Different Integers

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
11/18/2020 15:36	Accepted	8 ms	49.7 MB	java
11/18/2020 15:36	Runtime Error	N/A	N/A	java

Problems

Pick One

< Prev

1224/1659

Next >

i Java

Autocomplete

```
4 public int maxEqualFreq(int[] nums) {
5
6     int[] cnt = new int[N];
7     int[] freq = new int[N]; // count the freq of each # of appearance
8     int maxFreq = 0;
9     int res = 0;
10
11     for (int i = 0; i < nums.length; i++) {
12         cnt[nums[i]]++;
13         freq[cnt[nums[i]] - 1]--;
14         freq[cnt[nums[i]]]++;
15
16         maxFreq = Math.max(maxFreq, cnt[nums[i]]);
17
18         if (maxFreq * freq[maxFreq] + 1 == i + 1 // all maxF, current 1
19             || (maxFreq - 1) * freq[maxFreq - 1] + maxFreq == i + 1 // all maxF - 1,
20             || maxFreq == 1)
21             res = i + 1;
22     }
23     return res;
24 }
25 }
```

Testcase

Run Code Result

Debugger

Accepted Runtime: 0 ms

Your input [2,2,1,1,5,3,3,5]

Output 7

Diff

Expected 7

Run Code

Submit