


[Practice](#) [Compete](#) [Jobs](#) [Rank](#) [Leaderboard](#)

 bor0s

Dashboard > Algorithms > Strings > Super Reduced String

Badge Progress
☆☆☆

Points: 1046.20 Rank: 24600

Super Reduced String

 by [harshil7924](#)

[Problem](#)[Submissions](#)[Leaderboard](#)[Discussions](#)[Editorial](#)

Steve has a string, s , consisting of n lowercase English alphabetic letters. In one operation, he can delete any *pair of adjacent letters* with same value. For example, string "aabcc" would become either "aab" or "bcc" after 1 operation.

Steve wants to reduce s as much as possible. To do this, he will repeat the above operation as many times as it can be performed. Help Steve out by finding and printing s 's non-reducible form!

Note: If the final string is empty, print `Empty String`.

Input Format

A single string, s .

Constraints

- $1 \leq n \leq 100$

Output Format

If the final string is empty, print `Empty String`; otherwise, print the final non-reducible string.

Sample Input 0

```
aaabccddd
```

Sample Output 0

```
abd
```

Sample Case 0

Steve can perform the following sequence of operations to get the final string:

- aaabccddd \rightarrow abccddd
- abccddd \rightarrow abddd
- abddd \rightarrow abd

Thus, we print `abd`.

Sample Input 1

```
baab
```

Sample Output 1

```
Empty String
```

Explanation 1

Steve can perform the following sequence of operations to get the final string:

- baab \rightarrow bb
- bb \rightarrow Empty String

Thus, we print `Empty String`.

Sample Input 2

```
aa
```

Sample Output 2

Empty String

Explanation 2

Steve can perform the following sequence of operations to get the final string:

1. aa → Empty String

Thus, we print Empty String.

[f](#) [t](#) [in](#)Submissions: [34213](#)

Max Score: 10


Difficulty: Easy

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)  

C++  

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
8
9 int main() {
10     string input;
11     cin >> input;
12     for (int i = 0; i < input.size() - 1; i++) {
13         if (input.size() == 0) {
14             input = "Empty String";
15             break;
16         }
17         if (input[i] == input[i+1]) {
18             input.erase(i, 2);
19             i = -1;
20         }
21     }
22     cout << input << "\n";
23     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
24     return 0;
25 }
26
```

Line: 1 Col: 1

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

Run Code

Submit Code

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)