**Reorganize String**

Given a string S, check if the letters can be rearranged so that two characters that are adjacent to each other are not the same.

If possible, output any possible result.  If not possible, return the empty string.

class Solution **{**

public String reorganizeString**(**String S**)** **{**

char**[]** word **=** S**.**toCharArray**();** char**[]** count **=** **new** char**[**123**];**

int len **=** S**.**length**();**

**for(**int i **=** 0 **;** i **<** S**.**length**()** **;** i**++** **){**

count**[**word**[**i**]]++;**

**}**

int spChar **=** 0**;**

**for(**int i **=** 97**;** i **<** 123**;** i**++){**

**if(**count**[**i**]** **>** **(**len**+**1**)/**2**){**

**return** ""**;**

**}else** **if(**count**[**i**]==(**len**+**1**)/**2**){**

spChar **=** i**;**

**}**

**}** int j **=** 0**;** int k **=** 1**;**

char**[]** newWord **=** **new** char**[**len**];**

**if(**spChar **==** 0**){**

**for(**int i **=** 97**;** i**<**123**;** i**++){**

**while(**count**[**i**]!=**0 **&&** j**<**len**){**

newWord**[**j**]** **=** **(**char**)**i**;** j**+=**2**;**count**[**i**]--;**

**}**

**while(**count**[**i**]!=**0 **&&** k**<**len**){**

newWord**[**k**]** **=** **(**char**)**i**;**

k**+=**2**;**

count**[**i**]--;**

**}**

**}**

**}else{**

**for(**int i **=** 0**;**i**<**len**;**i**+=**2**){**

newWord**[**i**]** **=** **(**char**)**spChar**;**

**}**

**for(**int i **=** 97**;**i **<**123**;**i**++){**

**while(**count**[**i**]!=** 0 **&&** i **!=** spChar **&&**k**<**len**){**

newWord**[**k**]** **=** **(**char**)**i**;** k**+=**2**;**count**[**i**]--;**

**}**

**}**

**}**

String newWd **=** **new** String**(**newWord**);**

**return** newWd**;**

**}**

**}**