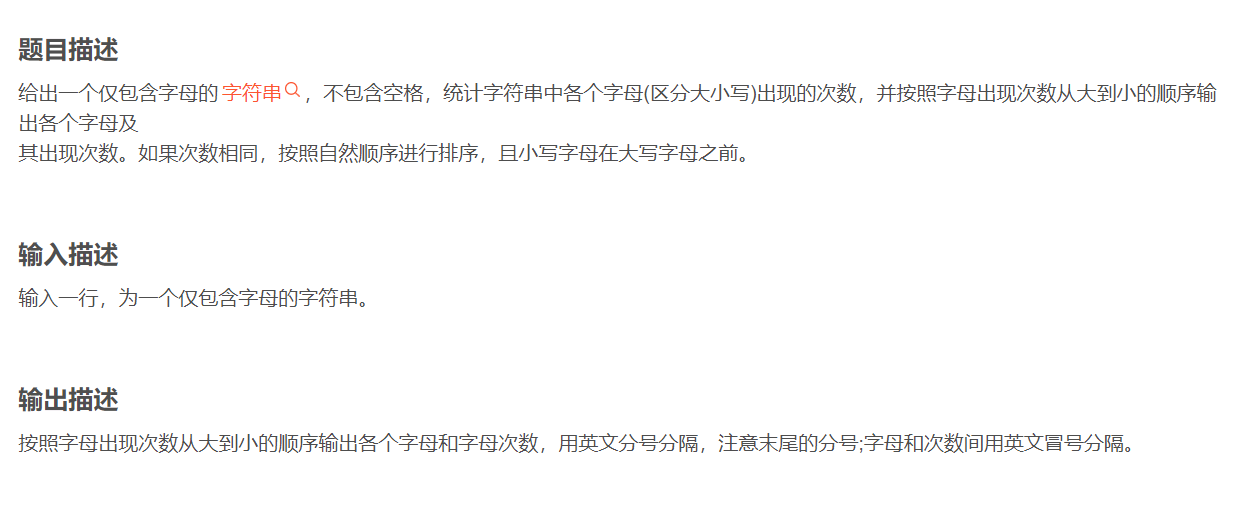
# **E卷-字符统计及重排[100分]（ Java | Python3 | C++ | C语言 | JsNode | Go ）**

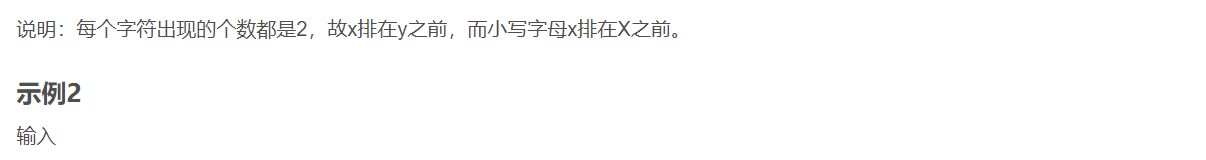




xyxyXX



x:2;y:2;X:2;



Abababb



b:4;a:3;





import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String s = scanner.nextLine(); // 读取输入字符串

int[] mp = new int[300]; // 初始化一个大小为300的数组，用于存储字符出现次数

// 统计字符出现次数

for (char ch : s.toCharArray()) {

mp[ch]++;

}

while (true) {

int mx = 0; // 记录当前出现次数最多的字符的次数

Integer ch = null; // 用于记录出现次数最多的字符

// 找到出现次数最多的字符

for (int key = 0; key < 300; key++) {

if (mp[key] > mx) {

mx = mp[key];

}

}

// 检查小写字母

for (int key = (int) 'a'; key <= (int) 'z'; key++) {

if (mp[key] == mx) {

ch = key; // 找到出现次数最多的字符

break;

}

}

// 如果没有找到小写字母中的最大出现次数字符，检查大写字母

if (ch == null) {

for (int key = (int) 'A'; key <= (int) 'Z'; key++) {

if (mp[key] == mx) {

ch = key;

break;

}

}

}

// 如果没有找到任何字符（所有字符次数都为0），退出循环

if (mx == 0) {

break;

}

// 打印出现次数最多的字符及其出现次数

System.out.print((char) ch.intValue() + ":" + mx + ";");

// 将该字符的出现次数清零

mp[ch] = 0;

}

scanner.close(); // 关闭输入流

}

}



s = input()

mp = [0]\*300

# 统计字符出现次数

for char in s:

mp[ord(char)] += 1

while True:

mx = 0

ch = None

# 找到出现次数最多的字符

for key in range(300):

if mp[key] > mx:

mx = mp[key]

for key in range(ord('a'), ord('z')+1):

if mp[key] == mx:

mx = mp[key]

ch = key

break

if ch == None:

for key in range(ord('A'), ord('Z')+1):

if mp[key] == mx:

mx = mp[key]

ch = key

break

if mx == 0:

break

# 打印出现次数最多的字符及其出现次数

print(f"{chr(ch)}:{mx};", end="")

# 将该字符的出现次数清零

mp[ch] = 0



#include <iostream>

#include <string>

#include <vector>

using namespace std;

int main() {

string s;

getline(cin, s); // 读取输入字符串

vector<int> mp(300, 0); // 初始化一个大小为300的数组，用于存储字符出现次数

// 统计字符出现次数

for (char ch : s) {

mp[ch]++;

}

while (true) {

int mx = 0; // 记录当前出现次数最多的字符的次数

int ch = -1; // 用于记录出现次数最多的字符

// 找到出现次数最多的字符

for (int key = 0; key < 300; key++) {

if (mp[key] > mx) {

mx = mp[key];

}

}

// 检查小写字母

for (int key = 'a'; key <= 'z'; key++) {

if (mp[key] == mx) {

ch = key; // 找到出现次数最多的字符

break;

}

}

// 如果没有找到小写字母中的最大出现次数字符，检查大写字母

if (ch == -1) {

for (int key = 'A'; key <= 'Z'; key++) {

if (mp[key] == mx) {

ch = key;

break;

}

}

}

// 如果没有找到任何字符（所有字符次数都为0），退出循环

if (mx == 0) {

break;

}

// 打印出现次数最多的字符及其出现次数

cout << static\_cast<char>(ch) << ":" << mx << ";";

// 将该字符的出现次数清零

mp[ch] = 0;

}

return 0;

}



#include <stdio.h>

#include <string.h>

#include <stdlib.h>

// 定义一个字符串数组来存储输入的字符串

char s[10005];

// 定义一个数组来记录每个字母的出现次数，数组下标对应ASCII码

int cnt[150];

int main() {

// 读取输入字符串

gets(s);

// 获取字符串的长度

int n = strlen(s);

// 统计每个字符的出现次数

for (int i = 0; i < n; ++i) {

cnt[s[i]] += 1;

}

// 不断找出未输出的出现次数最多的字符并输出

while (1) {

int mx = 0; // 记录当前最大出现次数

char ch = '0'; // 记录当前最大出现次数的字符

// 找出当前最大出现次数

for (int i = 0; i < 150; ++i) {

if (cnt[i] > mx)

mx = cnt[i];

}

// 如果最大出现次数为0，则说明所有字符都已输出完毕

if (mx == 0)

break;

// 按照字母顺序，从'a'到'z'找出该出现次数的字符

for (int i = 'a'; i <= 'z'; ++i) {

if (cnt[i] == mx) {

ch = i;

break;

}

}

// 如果没有找到小写字母，则找大写字母

if (ch == '0') {

for (int i = 'A'; i <= 'Z'; ++i) {

if (cnt[i] == mx) {

ch = i;

break;

}

}

}

// 将该字符的出现次数置为0，表示已处理

cnt[ch] = 0;

// 输出字符及其出现次数

printf("%c:%d;", ch, mx);

}

return 0;

}



const readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question('', (s) => {

const mp = Array(300).fill(0); // 初始化一个大小为300的数组，用于存储字符出现次数

// 统计字符出现次数

for (let i = 0; i < s.length; i++) {

mp[s.charCodeAt(i)]++;

}

while (true) {

let mx = 0; // 记录当前出现次数最多的字符的次数

let ch = -1; // 用于记录出现次数最多的字符

// 找到出现次数最多的字符

for (let key = 0; key < 300; key++) {

if (mp[key] > mx) {

mx = mp[key];

}

}

// 检查小写字母

for (let key = 'a'.charCodeAt(0); key <= 'z'.charCodeAt(0); key++) {

if (mp[key] === mx) {

ch = key; // 找到出现次数最多的字符

break;

}

}

// 如果没有找到小写字母中的最大出现次数字符，检查大写字母

if (ch === -1) {

for (let key = 'A'.charCodeAt(0); key <= 'Z'.charCodeAt(0); key++) {

if (mp[key] === mx) {

ch = key;

break;

}

}

}

// 如果没有找到任何字符（所有字符次数都为0），退出循环

if (mx === 0) {

break;

}

// 打印出现次数最多的字符及其出现次数

process.stdout.write(String.fromCharCode(ch) + ":" + mx + ";");

// 将该字符的出现次数清零

mp[ch] = 0;

}

rl.close();

});



package main

import (

"fmt"

)

func main() {

var s string

fmt.Scan(&s)

mp := make([]int, 300) // 初始化一个大小为300的数组，用于存储字符出现次数

// 统计字符出现次数

for \_, ch := range s {

mp[ch]++

}

for {

mx := 0 // 记录当前出现次数最多的字符的次数

var ch int // 用于记录出现次数最多的字符

// 找到出现次数最多的字符

for key := 0; key < 300; key++ {

if mp[key] > mx {

mx = mp[key]

}

}

// 检查小写字母

for key := int('a'); key <= int('z'); key++ {

if mp[key] == mx {

ch = key // 找到出现次数最多的字符

break

}

}

// 如果没有找到小写字母中的最大出现次数字符，检查大写字母

if ch == 0 {

for key := int('A'); key <= int('Z'); key++ {

if mp[key] == mx {

ch = key

break

}

}

}

// 如果没有找到任何字符（所有字符次数都为0），退出循环

if mx == 0 {

break

}

// 打印出现次数最多的字符及其出现次数

fmt.Printf("%c:%d;", ch, mx)

// 将该字符的出现次数清零

mp[ch] = 0

}

}