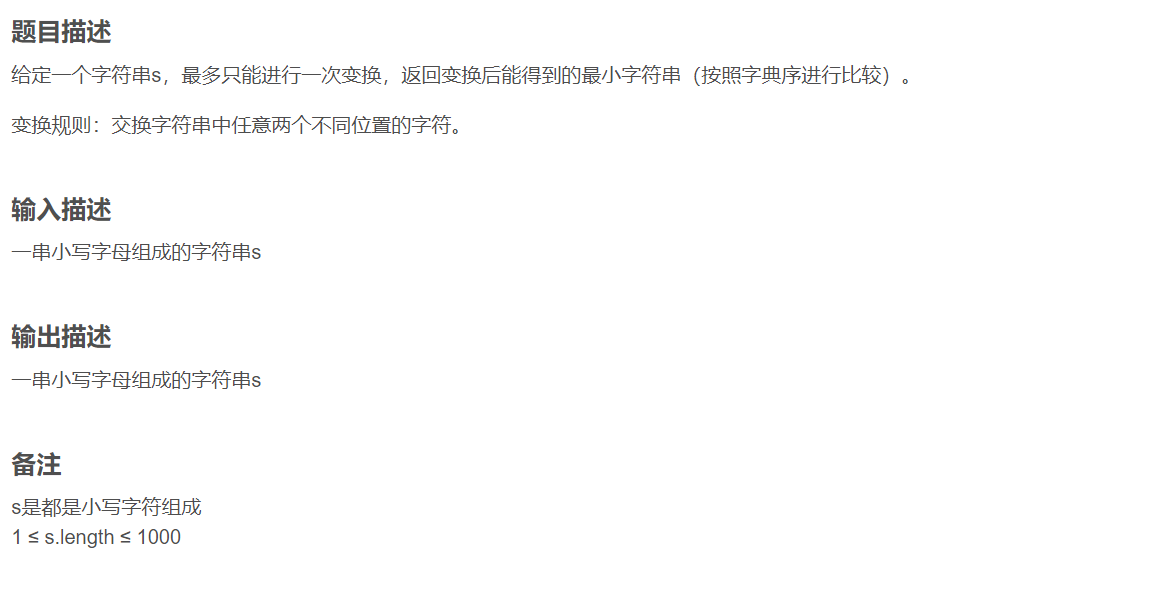
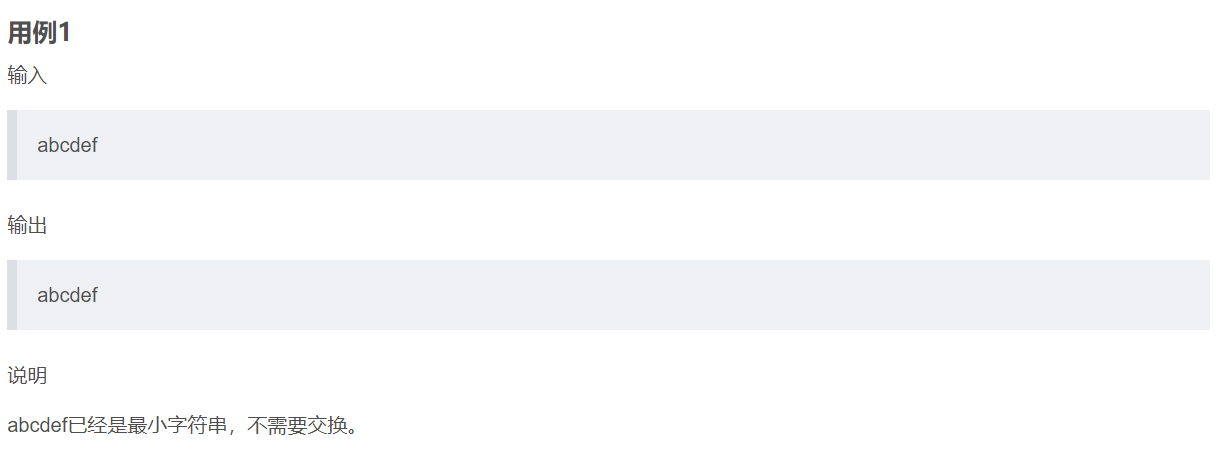
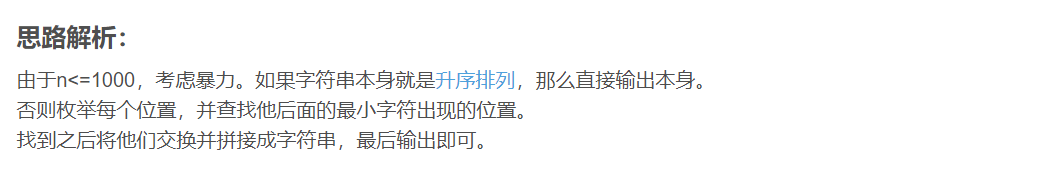
# **E卷-字符串变换最小字符串[100分]（ Java | Python3 | C++ | C语言 | JsNode | Go ）**











import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String line = sc.nextLine();// 读取一行输入作为字符串

char[] a = line.toCharArray();// 将字符串转换为字符数组

int n = a.length;// 获取字符数组的长度

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

int x = -1;// 初始化位置变量为-1

char k = 'z';// 初始化最小字符为'z'

for (int j = i + 1; j < n; j++) {

if (a[j] <= k) {// 找到比当前字符小的最小字符及其位置

x = j;

k = a[j];

}

}

if (k >= a[i] || x < 0) { // 如果没有找到需要交换的字符，则继续下一轮循环

continue;

}

char c = a[i];// 交换字符

a[i] = a[x];

a[x] = c;

System.out.println(String.valueOf(a));// 输出交换后的字符串

return;

}

System.out.println(line); // 如果没有找到需要交换的字符，则输出原始字符串

}

}



def main():

import sys

input = sys.stdin.read

line = input().strip() # 读取一行输入作为字符串

a = list(line) # 将字符串转换为字符数组

n = len(a) # 获取字符数组的长度

for i in range(n):

x = -1 # 初始化位置变量为-1

k = "z" # 初始化最小字符为'z'

for j in range(i + 1, n):

if a[j] <= k: # 找到比当前字符小的最小字符及其位置

x = j

k = a[j]

if k >= a[i] or x < 0: # 如果没有找到需要交换的字符，则继续下一轮循环

continue

a[i], a[x] = a[x], a[i] # 交换字符

print("".join(a)) # 输出交换后的字符串

return

print(line) # 如果没有找到需要交换的字符，则输出原始字符串

if \_\_name\_\_ == "\_\_main\_\_":

main()



#include <iostream>

#include <cstring>

using namespace std;

int main() {

string line;

getline(cin, line); // 读取一行输入作为字符串

int n = line.length(); // 获取字符串的长度

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

int x = -1; // 初始化位置变量为-1

char k = 'z'; // 初始化最小字符为'z'

for (int j = i + 1; j < n; j++) {

if (line[j] <= k) { // 找到比当前字符小的最小字符及其位置

x = j;

k = line[j];

}

}

if (k >= line[i] ||

x < 0) { // 如果没有找到需要交换的字符，则继续下一轮循环

continue;

}

swap(line[i], line[x]); // 交换字符

cout << line << endl; // 输出交换后的字符串

return 0;

}

cout << line <<

endl; // 如果没有找到需要交换的字符，则输出原始字符串

return 0;

}



#include <stdio.h>

#include <string.h>

int main() {

char line[1000];

fgets(line, 1000, stdin); // 读取一行输入作为字符串

int n = strlen(line) -

1; // 获取字符串的长度，减去末尾的换行符

line[n] = '\0'; // 去掉末尾的换行符

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

int x = -1; // 初始化位置变量为-1

char k = 'z'; // 初始化最小字符为'z'

for (int j = i + 1; j < n; j++) {

if (line[j] <= k) { // 找到比当前字符小的最小字符及其位置

x = j;

k = line[j];

}

}

if (k >= line[i] ||

x < 0) { // 如果没有找到需要交换的字符，则继续下一轮循环

continue;

}

char temp = line[i]; // 交换字符

line[i] = line[x];

line[x] = temp;

printf("%s\n", line); // 输出交换后的字符串

return 0;

}

printf("%s\n",

line); // 如果没有找到需要交换的字符，则输出原始字符串

return 0;

}



process.stdin.resume();

process.stdin.setEncoding('utf8');

let input = '';

process.stdin.on('data', (chunk) => {

input += chunk;

});

process.stdin.on('end', () => {

let line = input.trim(); // 读取一行输入作为字符串

let a = Array.from(line); // 将字符串转换为字符数组

let n = a.length; // 获取字符数组的长度

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

let x = -1; // 初始化位置变量为-1

let k = 'z'; // 初始化最小字符为'z'

for (let j = i + 1; j < n; j++) {

if (a[j] <= k) { // 找到比当前字符小的最小字符及其位置

x = j;

k = a[j];

}

}

if (k >= a[i] || x < 0) { // 如果没有找到需要交换的字符，则继续下一轮循环

continue;

}

[a[i], a[x]] = [a[x], a[i]]; // 交换字符

console.log(a.join('')); // 输出交换后的字符串

return;

}

console.log(line); // 如果没有找到需要交换的字符，则输出原始字符串

});



package main

import (

"bufio"

"fmt"

"os"

"strings"

)

func main() {

reader := bufio.NewReader(os.Stdin)

line, \_ := reader.ReadString('\n') // 读取一行输入作为字符串

line = strings.TrimSpace(line) // 去掉字符串前后的空格或换行符

a := []rune(line) // 将字符串转换为字符数组（rune数组以支持Unicode字符）

n := len(a) // 获取字符数组的长度

for i := 0; i < n; i++ {

x := -1 // 初始化位置变量为-1

k := 'z' // 初始化最小字符为'z'

for j := i + 1; j < n; j++ {

if a[j] <= k { // 找到比当前字符小的最小字符及其位置

x = j

k = a[j]

}

}

if k >= a[i] || x < 0 { // 如果没有找到需要交换的字符，则继续下一轮循环

continue

}

a[i], a[x] = a[x], a[i] // 交换字符

fmt.Println(string(a)) // 输出交换后的字符串

return

}

fmt.Println(line) // 如果没有找到需要交换的字符，则输出原始字符串

}