**递归倒置字符数组**

问题描述

　　完成一个递归程序，倒置字符数组。并打印实现过程  
　　递归逻辑为：  
　　当字符长度等于1时，直接返回  
　　否则，调换首尾两个字符，在递归地倒置字符数组的剩下部分

输入格式

　　字符数组长度及该数组

输出格式

　　在求解过程中，打印字符数组的变化情况。  
　　最后空一行，在程序结尾处打印倒置后该数组的各个元素。

样例输入

Sample 1

5 abcde

Sample 2

1 a

样例输出

Sample 1

ebcda

edcba

edcba

Sample 2

a

C++代码

#include <iostream>

using namespace std;

#define N 100

int len;

char str[N];

void f(int n,char s[])

{

if(n<=1) return;

char c=s[0];

s[0]=s[n-1];

s[n-1]=c;

for(int i=0;i<len;i++)

cout<<str[i];

cout<<endl;

f(n-2,s+1);

}

int main()

{

cin>>len>>str;

f(len,str);

cout<<endl;

for(int i=0;i<len;i++)

cout<<str[i];

return 0;

}

C代码

#include<stdio.h>

#include<string.h>

void digui(char \*c,int top,int end)

{

char tmp;

if(top==end)

return;

if(top<end)

{

tmp=c[top];

c[top]=c[end];

c[end]=tmp;

puts(c);

digui(c,top+1,end-1);

}

}

int main(void)

{

char c[1000];

int n;

scanf("%d",&n);

getchar();

gets(c);

digui(c,0,n-1);

printf("\n");

puts(c);

return 0;

}

Java代码

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String[] string=scanner.nextLine().split(" ");

char[] arr = new char[Integer.valueOf(string[0])];

arr = string[1].toCharArray();

aa(arr, 0);

}

private static void aa(char[] arr, int ii) {

if (ii >= arr.length - 1 - ii) {

System.out.println();

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

System.out.print(arr[i]);

}

return;

} else {

char c = arr[ii];

arr[ii] = arr[arr.length - 1 - ii];

arr[arr.length - 1 - ii] = c;

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

System.out.print(arr[i]);

}

System.out.println();

aa(arr, ii + 1);

}

}

}