课外作业模板

001 最远距离

1. **问题分析**

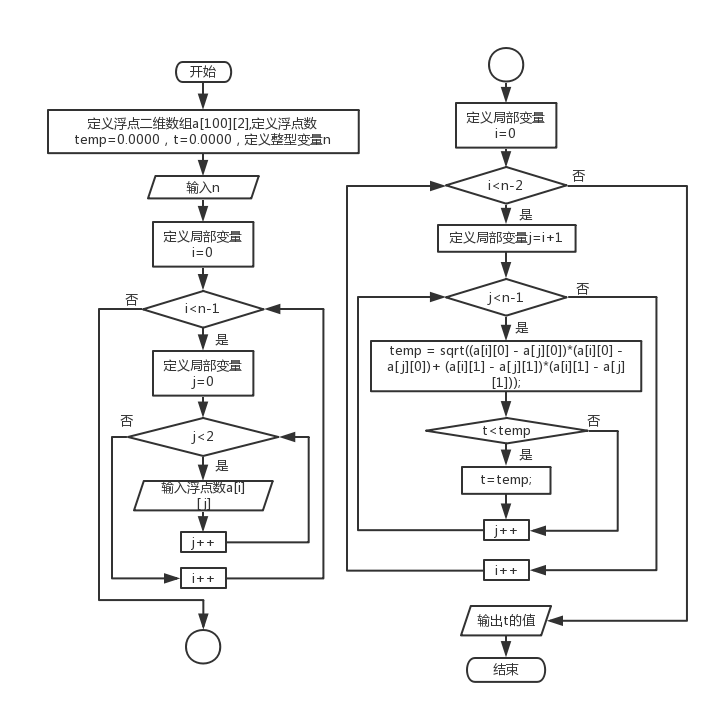
给定一组点 (x, y)（第一象限），求距离最远的两个点之间的距离，

只需要分别求出点之间的距离然后进行排序。

1. **解决方案**

定义一个二维数组，运用for语句进行赋值，再利用一次for语句进行循环计算和判断，最后输出所求值即可。

1. **算法设计**



1. **编程实现**

#include<stdio.h>

#include<math.h>

int main()

{

double a[100][2],temp = 0.0000, t = 0.0000;//定义二维数组a，浮点数temp，t

int n;

scanf("%d", &n);

for (int i = 0; i<n - 1; i++)//循环输入二维数组

{

for (int j = 0; j<2; j++)

scanf("%lf ", &a[i][j]);

}

for (int i = 0; i<n - 2; i++)//循环比较大小

{

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

{

temp = sqrt((a[i][0] - a[j][0])\*(a[i][0] - a[j][0])

+ (a[i][1] - a[j][1])\*(a[i][1] - a[j][1])); temp = sqrt((a[i][0] - a[j][0])\*(a[i][0] - a[j][0])

+ (a[i][1] - a[j][1])\*(a[i][1] - a[j][1]));

if (t<temp)

t = temp;

}

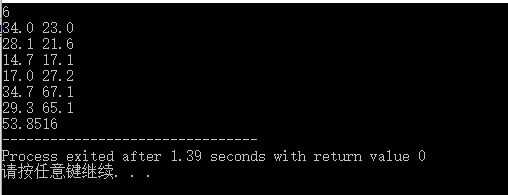
}

printf("%.4lf", t);//输出t

return 0;

}

1. **运行结果**



1. **结果分析**



1. **总结体会**

i：了解掌握sqrt函数的使用

ii：学会定义使用二维数组和巩固练习使用浮点数

003 角谷猜想

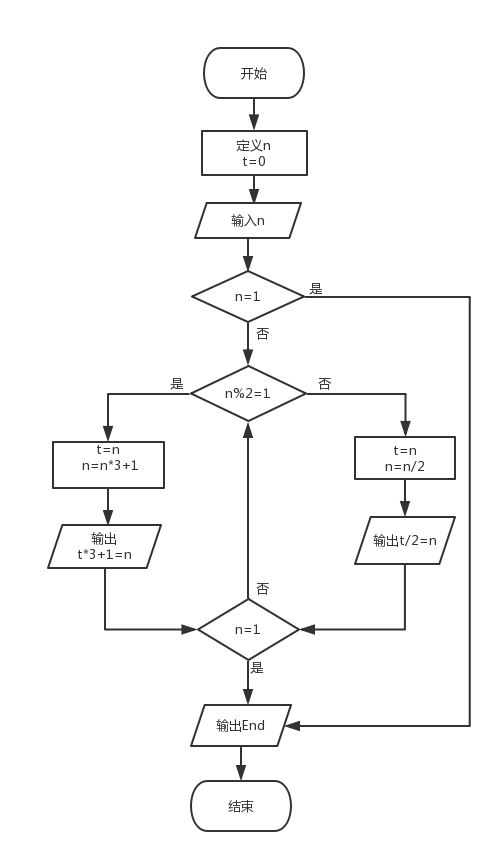
1. **问题分析**

所谓角谷猜想，是指对于任意一个正整数，如果是奇数，则乘 3 加 1，如果是偶数，则除以2，得到的结果再按照上述规则重复处理，最终总能够得到 1。如，假定初始整数为 5，计算过程分别为 16、8、4、2、1。程序要求输入一个整数，将经过处理得到 1 的过程输出来。

1. **解决方案**

输入一个数后先判断奇偶，然后根据判断结果执行不同的命令循环计算、输出即可，每计算一次输出一次，当n=1时输出End，结束程序。

1. **算法设计**



1. **编程实现**

#include<stdio.h>

int main()

{

int n, t = 0;

scanf("%d", &n);

if (n == 1)

{

printf("End");

return 0;

}

while (n != 1)//当n不等于1时执行该语句

{

if (n % 2 == 1)//当n不能整除2时

{

t = n;

n = n \* 3 + 1;

printf("%d\*3+1=%d\n", t, n);

}

else

{

t = n;

n = n / 2;

printf("%d/2=%d\n", t, n);

}

if (n == 1)

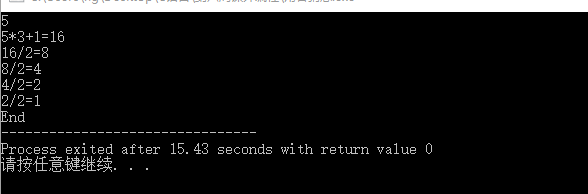
printf("End");

}

return 0;

}

1. **运行结果**



1. **结果分析**



（7**）总结体会**

i：熟练使用while、if等基本循环语句

ii：熟悉算法，懂得从多个角度解题

005 括号匹配问题

**1） 问题分析：**

在某个字符串（长度不超过 100）中有左括号、右括号和大小写字母；规定（与常见的算数式子一样）任何一个左括号都从内到外与在它右边且距离最近的右括号匹配。写一个程序，找到无法匹配的左括号和右括号，输出原来字符串，并在下一行标出不能匹配的括号。不能匹配的左括号用”$”标注, 不能匹配的右括号用”?” 标注.

可以通过字符数组分类讨论右括号能否匹配上左括号，再一一对应输出相应符号实现。

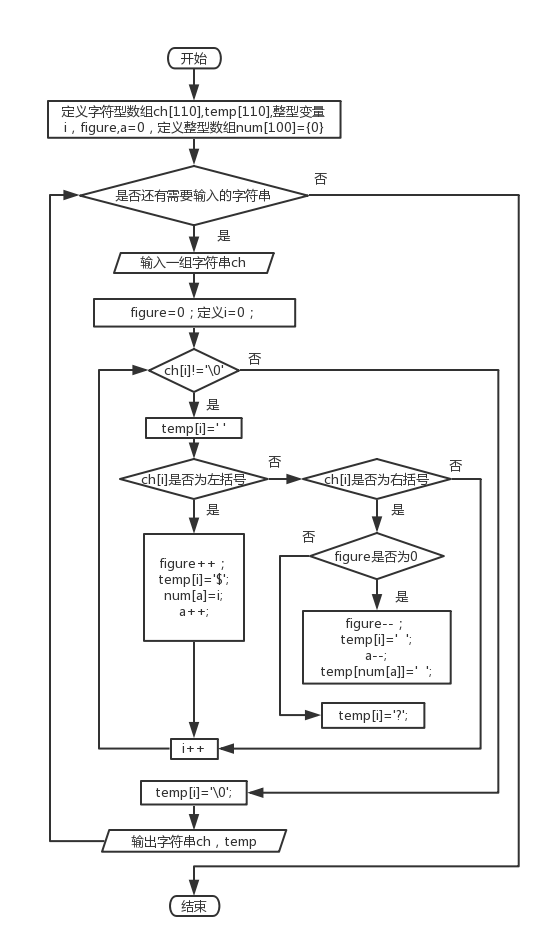
**2） 解决方案：**

1.先将要输出的字符初始化为空格，再分类讨论是否为左括号或右括号；

2.如果是左括号，左括号计数器加一

3,如果是右括号，先判断之前有没有未配对的左括号，有则都输出空格，没有就输出‘？’

1. **流程图：**

****

1. **编程实现：**

#include<stdio.h>

int main()

{

char ch[110], temp[110];

int i, figure;

int num[100] = { 0 }, a = 0;

while (gets\_s(ch)) //用gets语句输入字符串，当有字符串输入时执行循环

{

figure = 0; //左括号计数器清零

for (i = 0; ch[i] != '\0'; i++)

{

temp[i] = ' '; //默认为空格输出

switch (ch[i])

{

case '(':

figure++; //未配对左括号数加一

temp[i] = '$'; //将当前左括号设置为输出‘$’

num[a] = i; //用num[a]记录左括号地址

a++;

break;

case ')':

if (figure != 0) //如果之前已经出现过左括号

{

figure--; //未配对左括号数减一

temp[i] = ' '; //当前右括号设置为输出空格

a--;

temp[num[a]] = ' '; //将被配对的左括号设置为输出空格

}

else temp[i] = '?'; //如果没有可配对的左括号，输出‘？’

break;

}

}

temp[i] = '\0';//使输出结束而不跟一串乱码

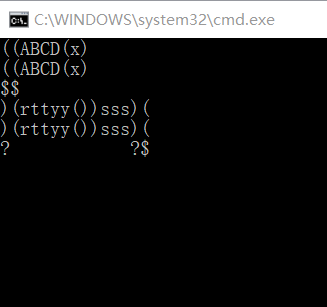
printf("%s\n%s\n", ch, temp);

}

return 0;

1. }**结果分析：**

**运算结果：**

****

**Open Judge运行结果：**

****

1. **反思总结：**

1.可以通过gets语句输入字符串；

2.可以用switch语句进行分类讨论，执行不同的操作；

3.需要多了解不同的算法，才能做出更多题。