

- 数据结构第一周总结
  - 一、关于提问
    - 1.1提问前你要干什么?
    - 1.2如何提问更高效?
    - 1.3在哪里提问?
  - 二、注重代码风格
  - 三、需要重视的知识点
    - 指针与内存
  - 四、问题回顾
    - 海岛问题总结

## 数据结构第一周总结

---

### 一、关于提问

(此部分引用并修改自2022猪脚说第1期)

#### 1.1提问前你要干什么?

1. 明确自己的问题到**代码行**;
2. 先到 **百度** 等搜索引擎搜寻自己的问题;
3. 若百度未能解决你的问题, 请到课程网站的论坛上搜寻相关问题;
4. 若课程网站的论坛未能解决你的问题, 请搜索课程微信群的历史消息, 看看有没有同学提到过相关问题;
5. 如果以上三步都没法解决你的问题, 那就提问吧!

#### 1.2如何提问更高效?

##### 1. 尽可能**一语中的**地提问

- "执行这句代码时, Dev-C++ 报 SIGSEGV 错误, 截图如下/代码链接如下:" - **精确到行(或尽量精确)+错误信息+代码截图/链接**
- "请问助教, 我这里哪错了?" - 请具体描述哪里不符合预期! 不精确到行的提问将相当耗费时间!

##### 2. 编写易读的代码:

- 请不要用 a, b, c, d, e, f 这样没有意义的名字作为变量/函数的名字, **建议用英文驼峰命名法**, 英文单词不太懂的, **用拼音也问题不大**;
- 没记清楚运算符优先级时, 要么马上查一下, 要么多用**小括号**;
- **就算 if/while 语句后面的执行语句只有一行, 也请不要省去大括号**;
- **勤加注释! 勤加注释! 勤加注释!**
- 整理好你的代码, Dev-C++ 中使用 Shift+Ctrl+A 即可**格式化代码**, 更美观的代码会让答疑者 更愿意帮你阅读代码;

3. 代码片段截图/代码片段链接, 告别微信群刷屏(**特别提出: 请不要抄袭盗用其他同学的代码, 我们有严格的代码查重!**) 代码截图, 完全可以使用微信自带的截图功能; **Markdown**(以下简称md)语法下的代码, 在论坛提问时使用: 课程论坛里支持使用md语法, 十分建议大家了解一些基本的md语法, 在以后会有非常大的作用; 如果要粘贴代码, md语法很简单, 用两行单独的"``"将你的代码框柱, 即英文半角下 键盘上

Esc键下方按键的字符，连续打3个。如： ``c #include <stdio.h> int main() { printf("Hello world!\n"); return 0; } ``

效果如下：

```
#include <stdio.h>
int main()
{
    printf("Hello world!\n");
    return 0;
}
```

其中，第一行 `` 字符后的 'c' 表示你粘贴的代码是C语言代码，md会将你的代码显示为 C语言风格。（其他语言同理）

3. 如果能够提供复现错误的方法, 就更好啦!

### 1.3在哪里提问?

1. 课程论坛;
2. 微信群;
3. 私聊助教. 问题解决后? 十分鼓励同学们在解决自己的问题后, 把解决思路写下来, 分享给大家. 可以是上传课程论坛, 可以是写 成文档在微信群里分享!

## 二、注重代码风格

代码风格真的很重要!!! 良好的代码风格是写出逻辑清晰的程序的前提, 易于debug, 也易于别人(包括一周后的自己)读懂你的代码。

好的代码风格体现在规范的排版、一目了然的变量函数的命名、合理的代码结构(例如减少代码复用、不要一main 到底)、简单易懂的注释等, 下面给出一些例子(大家如果想知道更规范详细的规则, 可以去查一些大厂的手册)

- 命名规则: 少用 a, b, c, d, e, f ..... 尽量用英文单词表示该变量或函数的含义或作用, 第一个单词首字母小写, 后面单词首字母大写。对于循环控制变量, 可以使用 i, j, k, l (尽量不要让这些变量出现在循环体外)
- 大括号位置: 函数定义、if、for 等语句, 左右括号分别独占一行, 并左侧对齐(当 for、if 只跟着一条语句时可以省略大括号)
- 合理利用缩进: 相同代码块(大括号)里的语句对齐, 并相对于大括号缩进
- 空格的使用:
  - 逗号、分号紧贴左边表达式, 右边加一空格(如果必要)
  - 单目运算符紧贴操作数
  - 双目或三目运算符左右两边均要加空格
  - if/for/while 与小括号中间加一空格
  - 函数定义和调用时函数名与小括号中间不加空格

## 三、需要重视的知识点

指针与内存

指针是c语言最重要的特点之一，与c语言的类型系统、数据存储、函数调用等都息息相关。大家如果对指针这方面还存在疑问，可以多看看老师上课的PPT，如果还不是很能理解，

对于基础的指针知识，可以在mooc上看翁恺老师的[网课](#)（第一单元-指针与字符串）

大家也可以看下面这篇文章：[C指针详解](#)

## 四、问题回顾

### 海岛问题总结

上节课PPT中给出了求凸海岛面积的算法，在这里我贴上求凹海岛问题的代码（如果同学们发现错误，可以随时来找qwp）

```
#include<stdio.h>
int a[50][50],n;
void dfs(int x,int y) {
    if(x>=0 && x<n && y>=0 && y<n) {
        if(a[x][y] == 0) {
            a[x][y] = -1;//岛外区域用-1表示
            dfs(x-1,y);
            dfs(x+1,y);
            dfs(x,y-1);
            dfs(x,y+1);
        }
    }
}
int main() {
    int i,j,area=0;
    scanf("%d",&n);
    for(i=0;i<n;i++) {
        for(j=0;j<n;j++){
            scanf("%d",&a[i][j]);
        }
    }
    //从边界开始递归
    for(i=0;i<n;i++) {
        dfs(i,0);
        dfs(0,i);
        dfs(i,n-1);
        dfs(n-1,i);
    }

    for(i=0;i<n;i++) {
        for(j=0;j<n;j++) {
            if(a[i][j] == 0) {
                area++;
            }
        }
    }
    printf("%d\n",area);
}
```

```
    return 0;  
}
```

如果同学们对有关海岛问题很感兴趣，可以通过下面的链接自行学习：[海岛问题总结](#) (PS : 里面很多知识我们之后才会学到，大家如果学有余力可以先了解一下)