

TCPL学习问题和解答3

问题：只要函数内能调用函数，并且最后能够返回，就可以实现递归吗？main函数可以调用自己吗？

两个问题的回答都是Yes。函数直接或间接调用自己都可以形成递归，但是如果没有中止条件或递归深度太大，都会导致程序栈溢出而崩溃。术语Stack Overflow说的就是它，还有个著名的问答网站就叫这个名字。

main函数虽然是程序的入口，但也可以被自己调用，不过基本没人会这么用。如果它自己调用自己而不提供中止条件，也会崩溃。你可以在gdb中运行，等它崩溃后用bt看看调用栈。如果是普通函数foo自己递归，会列出一个非常深的调用栈；如果是main自己递归，看到的只有几层，估计是main的特殊性导致的。

这是我很早之前写的一篇文章：<https://zedware.github.io/STACKOVERFLOW/>，供参考。

问题：尾递归优化

```
1  /******
2
3  Welcome to GDB Online.
4  GDB online is an online compiler and debugger tool ;
5  C#, VB, Perl, Swift, Prolog, Javascript, Pascal, HTM
6  Code, Compile, Run and Debug online from anywhere in
7
8  *****/
9  #include <stdio.h>
10
11  int fact(int n, int product)
12  {
13      if (n == 1)
14          return product;
15      return fact(n - 1, n * product);
16  }
17
18  // int fact(int n)
19  // {
20  //     if (n == 1)
21  //         return n;
22  //     return n * fact(n - 1);
23  // }
24
25  int main()
26  {
27      printf("%d", fact(5, 1));
28
29      return 0;
30  }
31
```

今天和学生讨论阶乘函数的递归写法，在python中写成这个样子，可以不用担心栈溢出，会一直共用同一个栈帧空间：前提是return后面直接是递归调用，不能出现在表达式中（像注释的递归函数那样）利用参数进行表达式的运算，这样就类似循环，不用担心栈溢出了，请问怎么验证在

C语言中是不是也是如此? <https://stackoverflow.com/questions/34125/which-if-any-c-compilers-do-tail-recursion-optimization>

问题： 怎么避免 vscode 格式化某段代码？

程序中有一个数组表示棋盘，如下：

```
char board[SIZE][SIZE * CHARSIZE + 1] =
{
```

[illegible]

}.

如果在 vscode 中选择自动格式化文档，则会变成：

[illegible]

明显不如原始的代码清楚。

1. vscode 默认配置的是 clang-format 这个格式化工具。

2. 很多格式化功能都可以通过特定的注释来临时关闭。

3. clang-format 的这个特殊注释是:

```
// clang-format off
```

```
char board[SIZE][SIZE * CHARSIZE + 1] =
```

{

[illegible] $\}_{i}$

```
// clang-format on
```

问题：

