



知识点总结



- 在C语言中，函数的**嵌套**调用指的是在一个函数内部调用另一个函数。这种机制允许将复杂的任务分解为多个较小的、更易于管理的函数，从而提高代码的可读性。

- 嵌套调用的形式：

假设有三个函数func_a、func_b和func_c，函数func_a调用func_b，而func_b又调用func_c，这就是一种典型的函数嵌套调用结构。

```
#include <stdio.h>
// 定义函数func_c
void func_c() {
    printf("这是func_c。\\n");
}
// 定义函数func_b，在func_b中调用func_c
void func_b() {
    printf("这是在调用函数func_c之前的func_b。\\n");
    func_c();
    printf("这是在调用函数func_c之后的func_b。\\n");
}
// 定义函数func_a，在func_a中调用func_b
void func_a() {
    printf("这是在调用函数func_b之前的func_a。\\n");
    func_b();
    printf("这是在调用函数func_b之后的func_a。\\n");
}
int main() {
    func_a();
    return 0;
}
```



在上述代码中，main函数调用func_a，func_a调用func_b，func_b调用func_c。

程序的执行流程如下：

1. main函数开始执行，调用func_a。
2. func_a开始执行，打印出相关信息，然后调用func_b。
3. func_b开始执行，打印出相关信息，接着调用func_c。
4. func_c开始执行，打印出信息后返回func_b。
5. func_b继续执行，打印出信息后返回func_a。
6. func_a继续执行，打印出信息后返回main函数。
7. main函数执行完毕，程序结束。

```
#include <stdio.h>
// 定义函数func_c
void func_c() {
    printf("这是func_c。\\n");
}
// 定义函数func_b，在func_b中调用func_c
void func_b() {
    printf("这是在调用函数func_c之前的func_b。\\n");
    func_c();
    printf("这是在调用函数func_c之后的func_b。\\n");
}
// 定义函数func_a，在func_a中调用func_b
void func_a() {
    printf("这是在调用函数func_b之前的func_a。\\n");
    func_b();
    printf("这是在调用函数func_b之后的func_a。\\n");
}
int main() {
    func_a();
    return 0;
}
```



运行结果如下：

这是在调用函数func_b之前的func_a。

这是在调用函数func_c之前的func_b。

这是func_c。

这是在调用函数func_c之后的func_b。

这是在调用函数func_b之后的func_a。

注意：

1、嵌套调用时，被调用的函数一定要提前定义好或者声明好。

2、嵌套的层次不宜过深。

```
#include <stdio.h>
// 定义函数func_c
void func_c() {
    printf("这是func_c。 \n");
}
// 定义函数func_b, 在func_b中调用func_c
void func_b() {
    printf("这是在调用函数func_c之前的func_b。 \n");
    func_c();
    printf("这是在调用函数func_c之后的func_b。 \n");
}
// 定义函数func_a, 在func_a中调用func_b
void func_a() {
    printf("这是在调用函数func_b之前的func_a。 \n");
    func_b();
    printf("这是在调用函数func_b之后的func_a。 \n");
}
int main() {
    func_a();
    return 0;
}
```



函数的定义不可以嵌套

函数的调用可以嵌套



题目1、在C语言程序中，下列说法中正确的是（ B ）

- A. 函数的定义可以嵌套，但函数的调用不可以嵌套
- B. 函数的定义不可以嵌套，但函数的调用可以嵌套
- C. 函数的定义和调用均不可以嵌套
- D. 函数的定义和调用均可以嵌套



题目2、有以下程序

```
#include<stdio.h>
int fun1(double a)
{
    return (int)(a * a);
}
int fun2(double x, double y)
{
    double a = 0, b = 0;
    a = fun1(x);
    b = fun1(y);
    return (int)(a + b);
}
main()
{
    double w;
    w = fun2(1.1, 2.0);
    printf("%4.2f", w);
}
```

程序执行后输出结果是 (B)

A. 5

B. 5.00

C. 5.21

D. 0.0



题目3、有以下程序

```
#include <stdio.h>
```

```
int func1(int x)
{
```

```
    return x * 2;
```

```
}
```

```
int func2(int y)
```

```
{
```

```
    return func1(y) + 3;
```

```
}
```

```
main()
```

```
{
```

```
    int result = func2(5);
```

```
    printf("%d\n", result);
```

```
}
```

程序执行后输出结果是 (C)

A. 5

B. 10

C. 13

D. 15