小马加编信息学教案(十六)

C++ 函数(三)

一. 课程内容

1. 函数的递归调用的概念
2. 函数的递归调用的理解
3. 函数的递归调用的应用

二. 知识讲解

1. 函数的递归调用的概念

- 什么是函数递归调用?

> **函数自己调用自己，这种调用称为“递归”调用，这样的函数称为“递归函数”。**

> 举个有趣的例子：
> 从前有座山，山里有座庙，庙里有个老和尚对小和尚说：“从前有座山，山里有座庙，庙里有个老和尚对小和尚说：“从前有座山，山里有座庙.....””

> 执行递归函数将反复调用其自身，每调用一次就进入新的一层，当最内层的函数执行完毕后，再一层一层地由里到外退出。

2. 函数的递归调用的理解

> 下面我们通过一个求阶乘的例子，看看递归函数到底是如何运作的。
> 阶乘 $n!$ 的计算公式如下：
>
> $$n! = \begin{cases} 1 & (n = 0, 1) \\ n * (n-1)! & n > 1 \end{cases}$$
>
> 根据公式编写以下代码

```
#include <bits/stdc++.h>
using namespace std;

//求n的阶乘
long factorial(int n) {
    if (n == 0 || n == 1) {
        return 1;
    }
    else {
        return factorial(n - 1) * n;  // 递归调用
    }
}

int main() {
    int a;
    cin >> a;
    printf("Factorial(%d) = %ld\n", a, factorial(a));
    return 0;
}
```

> 输入: 5↙
> 运行结果

```
Factorial(5) = 120
```

> $factorial()$ 就是一个典型的递归函数。调用 $factorial()$ 后即进入函数体，只有当 $n == 0$ 或 $n == 1$ 时函数才会执行结束，否则就一直调用它自身。

> 由于每次调用的实参为 $n-1$，即把 $n-1$ 的值赋给形参 n，所以每次递归实参的值都减 1，直到最后 $n-1$ 的值为 1 时再作递归调用，形参 n 的值也为 1，递归就终止了，会逐层退出。

- 要想理解递归函数，重点是理解它是如何逐层进入，又是如何逐层退出的，下面我们以 5! 为例进行讲解。

递归的进入

> (1) 求 5!，即调用 $factorial(5)$ 。当进入 $factorial()$ 函数体后，由于形参 n 的值为 5 ，不等于 0 或 1 ，所以执行 $factorial(n-1)*n$，也即执行$factorial(4)*5$ 。为了求得这个表达式的结果，必须先调用 $factorial(4)$，并暂停其他操作。换句话说，在得到 $factorial(4)$ 的结果之前，不能进行其他操作。这就是第一次递归。

> (2) 调用 $factorial(4)$ 时，实参为 4 ，形参 n 也为 4 ，不等于 0 或 1，会继续执行 $factorial(n-1)*n$ ，也即执行 $factorial(3)*4$ 。为了求得这个表达式的结果，又必须先调用 $factorial(3)$。这就是第二次递归。

> (3) 以此类推，进行四次递归调用后，实参的值为 1 ，会调用 $factorial(1)$。此时能够直接得到常量 1 的值，并把结果 `return` ，就不需要再次调用 $factorial()$ 函数了，递归就结束了。

> 下表列出了逐层进入的过程

层次/层数	实参/形参	调用形式	需要计算的表达式	需要等待的结果
1	n=5	factorial(5)	factorial(4) * 5	factorial(4) 的结果
2	n=4	factorial(4)	factorial(3) * 4	factorial(3) 的结果
3	n=3	factorial(3)	factorial(2) * 3	factorial(2) 的结果
4	n=2	factorial(2)	factorial(1) * 2	factorial(1) 的结果
5	n=1	factorial(1)	1	无

递归的退出

> **当递归进入到最内层的时候，递归就结束了，就开始逐层退出了，也就是逐层执行 `return` 语句。**

> (1) n 的值为 1 时达到最内层，此时 `return` 出去的结果为 1，也即 $factorial(1)$ 的调用结果为 1。

> (2) 有了 $factorial(1)$ 的结果，就可以返回上一层计算 $factorial(1)*2$ 的值了。此时得到的值为 2， `return` 出去的结果也为 2 ，也即 $factorial(2)$ 的调用结果为 2。

> (3) 以此类推，当得到 $factorial(4)$ 的调用结果后，就可以返回最顶层。经计算，$factorial(4)$ 的结果为 24，那么表达式 $factorial(4)*5$ 的结果为 120 ，此时 `return` 得到的结果也为 120 ，也即 $factorial(5)$ 的调用结果为 120，这样就得到了 5! 的值。

> 下表列出了逐层退出的过程

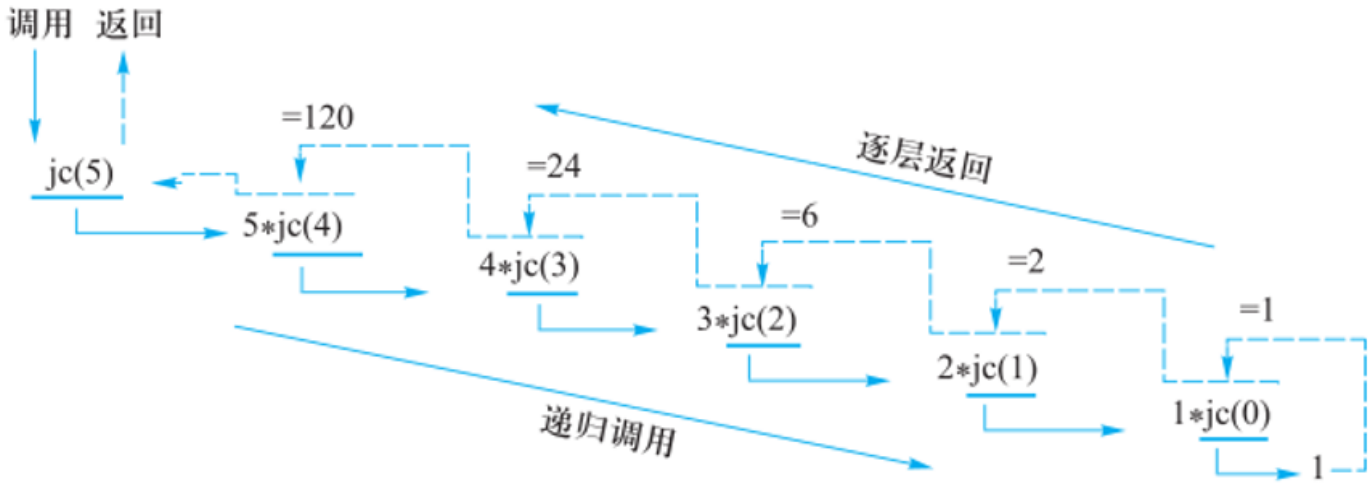
层次/层数	调用形式	需要计算的表达式	从内层递归得到的结果 (内层函数的返回值)	表达式的值 (当次调用的结果)
5	factorial(1)	1	无	1
4	factorial(2)	factorial(1) * 2	factorial(1) 的返回值，也就是 1	2
3	factorial(3)	factorial(2) * 3	factorial(2) 的返回值，也就是 3	6
2	factorial(4)	factorial(3) * 4	factorial(3) 的返回值，也就是 6	24
1	factorial(5)	factorial(4) * 5	factorial(4) 的返回值，也就是 24	120

递归的条件

> 每一个递归函数都应该**只进行有限次的递归调用**，否则它就会进入死胡同，永远也不能退出了，这样的程序是没有意义的。

> 要想让递归函数逐层进入再逐层退出，需要解决两个方面的问题：

- **存在限制条件，当符合这个条件时递归便不再继续**。对于 $factorial()$，当形参 n 等于 0 或 1 时，递归就结束了。
- **每次递归调用之后越来越接近这个限制条件**。对于 factorial()，每次递归调用的实参为 $n-1$，这会使得形参 n 的值逐渐减小，越来越趋近于 1 或 0。



3. 函数的递归调用的应用

> 一个问题要想用递归的方法（函数）来解决，必须要符合两个条件。
> (1) 可以把这个问题转化成一个新问题，而新问题的解法和原问题的解法完全相同，只是问题规模变小了；
> (2) 必须要有一个明确的递归结束条件（递归边界）。

> 这与递归的条件相对应，这也是我们要将问题抽象为递归模型的关键。
> 这种抽象的思维能力需要我们进行一定的练习才能够适应、强化。

三. 经典例题

1. 递归第一次

> 同学们在做题时常遇到这种函数
>
> $$f(x)=\begin{cases}5 & x\geq 0\\ f(x+1)+f(x+2) & x<0\end{cases}$$
>
> 下面就以这个函数为题做一个递归程序吧

> 输入格式
> 一个整数，表示$f(x)$中x值

> 输出格式
> 一个数表示$f(x)$值

输入样例	输出样例
0	5
-5	77

> 数据范围
> $x\geq -30$

2. 下楼问题

> 从楼上走到楼下共有 h 个台阶，每一步可以走 1 或者 2 个台阶。
> 现在要你求出从楼上走到楼下的总方案数

> 输入格式
> 一行一个正整数h

> 输出格式
> 一个正整数，表示方案数

输入样例	输出样例
2	2
5	8

> 数据范围
> $0<h\leq 20$

3. 求最大公约数

> 输入两个正整数 m 和 n ，求它们的最大公约数。

> 输入格式
> 一行两个正整数 m 和 n ，用一个空格隔开

> 输出格式
> 一行一个正整数，表示 m 和 n 的最大公约数。

输入样例	输出样例
24 36	12

> 数据范围
> $2 \leq m，n \leq 10000$

> 提示
> 由欧几里得"辗转相除法"可知：若设$gcd(a,b)$为a,b最大公约数,则

$$gcd(a,b) = \begin{cases} a & b == 0 \\ gcd(b, a\%b) & b \neq 0 \end{cases}$$

四. 提高巩固

1. 汉诺塔问题(经典递归，可查阅资料)

> 汉诺塔：汉诺塔（又称河内塔）问题是源于印度一个古老传说的益智玩具。
> 大梵天创造世界的时候做了三根金刚石柱子，在一根柱子上从下往上按照大小顺序摞着64片黄金圆盘。
> 大梵天命令婆罗门把圆盘从下面开始按大小顺序重新摆放在另一根柱子上。
> 并且规定，在小圆盘上不能放大圆盘，在三根柱子之间一次只能移动一个圆盘。
> 给出第一根柱子的圆盘个数，你要做的就是找出最快将圆盘全部移到第三根柱子的方法，并将方法总数输出。

> 输入格式
> 一个正整数n

> 输出格式
> 一个正整数，表示方案总数。

输入样例	输出样例

输入样例	输出样例
3	7

> 数据范围
> $n \leq 20$

2. 下楼升级版

> 从楼上走到楼下共有 h 个台阶，每一步可以走不大于 s 个台阶。
> 问可走出多少种方案。

> 输入格式
> 一行2个正整数，分别表示 h 和 s 的值

> 输出格式
> 一个正整数，表示走法数

输入样例	输出样例
1 3	1

> 数据范围
> $0 < h, s \leq 20$

3. 分解质因子

> 输入一个正整数 n，用递归方法从小到大输出它的所有质因子（因子是质数）。

> 输入格式
> 一行一个正整数 n

> 输出格式
> 一行若干个正整数，两数之间用一个空格隔开，从小到大输出。

输入样例	输出样例
18	2 3 3

> 数据范围
> $2 \leq n \leq 10000$

> 提示

$$zyz(n,p)=\begin{cases} n=1 & \text{返回} \\ n>1 & \begin{cases} n\,\%\,p=0 & \begin{cases} \text{输出 } p \\ zyz(n/p,p) \end{cases} \\ n\,\%\,p\neq 0 & zyz(n,p+1) \end{cases} \end{cases}$$