

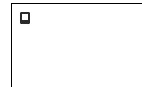


今日热门: 我市25岁女股民7年只做一只股 操作技巧惊呆众人

我市25岁女股民7年只做一只股 操作技巧惊呆众人

一夜2000当地散户得此消息 连夜关注炒盘内幕...[详情]

阅读(53320) | 评论(25885)



嘎啦顺的博客

<http://blog.sina.com.cn/zzblog2008> [订阅] [手机订阅]

[首页](#) [博文目录](#) [图片](#) [关于我](#)

个人资料



嘎啦顺

微博

加好友

发纸条

写留言

加关注

博客地图 world map

博客等级:

博客积分: **363**

博客访问: **12,252**

关注人气: **10**

获赠金笔: **1**

赠出金笔: **0**

荣誉徽章:



相关博文

实拍巴黎超恐怖地下墓穴
楚朋

探索神奇的“石林星球”
阿兹猫

比基尼美女盛行瑞士狂欢节
华拉那

全国97个县市最高温达40℃ 背后究
锦中一风水

中国古代历史上皇帝为什么必须杀
桥下街

茄子这样做好吃又省油,简单又快
用户352328266

龙头炼成记,操作十个字
概念爱好者

八月下旬财运红火的三大生肖
普渡斋

离婚,有时候也是一种负责任的
晓舟同志

新股加速开板有利于估值合理回
曹中铭

“剪刀差”带动市场投资机会
科德投资

正文

C语言基础——迭代与递归 (2009-04-26 07:28:20)

标签: it 分类: 蒸汽船

引言: 迭代与递归对于初学者可能会是一个难点, 现在就准备讲解这个。

一. 预备知识——函数

直接点来说C语言里的函数和数学里的函数完全是两回事, C语言里的函数就是子程序, 功能划分更细小的一段代码。为什么需要函数? 目的就是为了编写程序能够更方便。你要是问为什么, 那我举个例子, 你平时用得最多的两个函数scanf和printf, 这个其实就是别人编写好的函数, 在你包含了相应的头文件(也就是函数声明)以后, 你要用的时候只要按这两个函数的要求传递参数就能实现你所需要的功能了。有了函数这个东西, 你就不需要在你每次要进行输入输出的时候, 都要自己重复地写详细的实现过程, 而只需要简单地调用一下已经写好的代码。

以下就是一个简单的函数例子:

```
int isPrime(int n)//判断n是不是素数, 如果是就返回1
{
    int a;
    for(a=2; a<n; ++a) // a用于试除, 注意a<n
    {
        if(n % a == 0) // 余数为0就是说不是素数了
        {
            break;
        }
    }
    if(a < n)
    {
        return 0;
    }
    return 1;
}
```

你写好这个函数以后, 再需要判断一个数是不是素数的话, 很简单, 只要:

```
int a;
scanf("%d", &a)
if(isPrime(a))
{
    printf("yes");
}
else
{
    printf("no");
}
```

怎么样, 够简单吧, 和和printf输出结果一样那么轻松。

二. 迭代

迭代的一个经典例子是斐波那契数列, 这个数列是这样定义的:

$$f(n-1) + f(n-2) \quad (n \geq 2)$$

$$f(n) = \begin{cases} 1 & (1 \leq n \leq 2) \end{cases}$$

意思是说这个数列形如: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 前两个是1, 下一个数是它前面两个数的和。

字体大小: 大 中 小

转载 ▼

推荐为问题解答者赚取金币奖励？
-萧梦-

更多>>

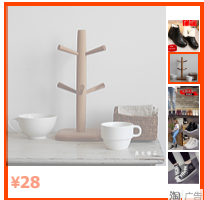
买房移民的国家

泰国买房

西班牙买房移民

C语言基础

澳大利亚买房



推荐博文

大爷抓贼6000多，弘扬正能量

中期向上趋势难以改变

秦汉时期吕氏各人物独领风骚

补贴大战能挽回Apple吗

51Talk自救乏力：豪赌“哈

台湾科技挣扎，人祸大于天灾？

收入份额=市场份额，虎嗅想干什

传奇的谢幕，谈岩田聪和他的任天

家常主食轻松做之——培根香葱花

意外及格

查看更多>>

谁看过这篇博文

gcx_49928...	7月15日
天啊为什...	7月5日
南條愛乃	6月22日
jerry	6月11日
不停的鱼	5月18日
星辰	5月4日
帅哥	4月17日
大暖小寒	2月2日
mmhhddddd...	1月27日
wjxmosquito	12月23日
云迪有正...	12月15日
柏舟	12月9日

现在问题来了，输入一个n，计算出这个数列的第n项是什么数。

例如输入：6

输出：8

你也许会想，既然是算第n个，那就要推算n次了吧，用个for循环就行了，然后你可能写出以下代码：

```
#include <stdio.h>
int main(void)
{
    int n, s;
    scanf("%d", &n);    // 输入要计算的n
    for(int a=2; a<=n; ++a) // a控制循环次数
    {
        //
    }
    printf("%d\n", s);    // s用来保存结果
    return 0;
}
```

好了，写到这里你多数都会卡住了，因为如果只用这些变量，根本不足以去计算。为此，从这个数列的定义观察，其中有：
 $f(n) = f(n-1) + f(n-2)$
如果用数组实现，那么就直观了。在假设输入的n最大是100的情况下，你可能写出如下代码：

```
#include <stdio.h>
int main(void)
{
    int n;
    int f[100] = {0,1};
    scanf("%d", &n);    // 输入要计算的n
    for(int a=2; a<=n; ++a) // a控制循环次数
    {
        f[a] = f[a-1] + f[a-2];
    }
    printf("%d\n", f[n]); // f[n]就是结果
    return 0;
}
```

运行结果是正确的，还不错吧！不过，你会不会想，计算这个可不可以不用数组呢？这个数组很浪费空间啊。为了确定能不能把这个数组去掉，你需要分析一下每一次计算的时候，使用了这个数组的哪些元素。很明显，只使用了f[a],f[a-1],f[a-2]再之前的就没有再使用了，要计算下一个数，只需要知道上一个数和上上个数。也就是说计算过程只需要用三个变量就可以了。我们声明为nLL,nL,s，分别是上上个数、上一个、和计算中的结果。代码可以改写为：

```
#include <stdio.h>
int main(void)
{
    int n;
    int nLL=0,nL=1,s=1;    // 初始化
    scanf("%d", &n);    // 输入要计算的n
    for(int a=2; a<=n; ++a) // a控制循环次数
    {
        s = nLL + nL;    // 计算当前的f(n)
        nLL = nL;    // 原来的上一个要变成上上一个
        nL = s;    // 当前的结果要变成上一个
    }
    printf("%d\n", s);    // s用来保存结果
    return 0;
}
```

怎么样，迭代也不难吧。

迭代习题：<http://yzfy.org/bbs/viewthread.php?tid=138>

本文章版权属于<http://yzfy.org>雨中飞燕之家论坛所有

三. 递归

本点是令新手头痛的一大问题，这里也会重点解释。

经典题目1:输入一个字符串，然后反序输出。

常规的解法是写入到一个char数组，然后从后向前输出。但是，如果你要用递归来解的话，思路就完全不同了。首先，由于是字符输入，我们需要一个一个字符来处理，所以可以选择用getchar()函数。然后，怎么去构造这个递归函数呢？首先，我们假设这个函数已经写好了，名字是fs()，它能完成这个功能。现在，我们写一个名字是f的函数，要调用它来完成整个功能（当然你不能直接把所有东西都交给它啊，只是假设）。

f函数运行的时候，需要读取字符和输出字符，你可以这样

[新浪首页](#) [登录](#) [注册](#)

再读一个字符，然后把剩下的交给fs函数，由fs函数把剩下的字符读取并且反向输出，最后你再把你自已读取的字符输出来，不就完成了么？
假如输入"abcd"，你读取了第一个字符'a'，然后剩下的由fs函数来处理，fs函数处理后输出反序的结果"dcb"后，你再输出你所读取的字符'a'，最后就是"dcba"，大功告成。

假想代码如下：

```
void f()
{
    int c = getchar(); //记录下输入的第一个字符
    fs(); //让这个假想的函数来完成剩下的字符反转
    putchar(c); //最后输出自己所记录下来的字符
}
```

现在你需要考虑一个问题：输入的那个字符如果是'\n'，就表示那一行已经输入结束了，就不需要再调用fs函数了，要不然你就不知道fs函数会干出些什么事情了，所以还得像下面这样补上一个判断语句：

```
void f()
{
    int c = getchar(); //记录下输入的第一个字符
    if(c!='\n')fs(); //如果输入没有结束就让这个假想的函数来完成剩下的
    putchar(c); //最后输出自己所记录下来的字符
}
```

好了，f函数也写好了。然后你就发现问题了，f和fs不就是在完成同一个功能么？你所写的f函数就是假想的fs函数所完成的功能了，所以你只需要把fs改成f，现在你就真正大功告成了：

```
#include <stdio.h>
void f()
{
    int c = getchar(); //记录下输入的第一个字符
    if(c!='\n')f(); //让这个函数来完成剩下的字符反转
    putchar(c); //最后输出自己所记录下来的字符
}
int main(void)
{
    f(); //调用这个函数
    return 0;
}
```

怎么样，运行一下试试吧，看看你理解了多少了。
假如输入abc回车，我们来看看详细是怎么运行的：
main调用f，第1层：

```
{
    先接收了一个'a'，再调用f:
    {
        接收了一个'b'，再调用f:
        {
            接收了一个'c'，再调用f:
            {
                接收了一个'\n'，结束
            }
            输出接收的'c'，结束
        }
        输出接收的'b'，结束
    }
    输出接收的'a'，结束
}
```

经典题目2:斐波那契数列

这个的题目和之前的那题一样，输入一个n，计算出这个数列的第n项是什么数。

如果用递归的方法，思考的方法也是差不多，先假设这个函数已经完成了，名字是fib(n)，你自己要编写的函数名字是f(n)，为了计算f(n)的结果，你需要知道它前两项的结果，这个不就很简单吗？只要调用fib函数就能得到结果了，所以就可以写出：

```
int f(n)
{
    return fib(n-1) + fib(n-2);
}
```

是不是很简单？先别急，现在你可要考虑特别的情况了。如果要计算f(2)或者f(1)，那么会进行fib(2-2)也就是fib(0)的计算，可是这个是没有定义的，不应该要让fib函数去计算它。由这个数列定义可以知道f(1)=f(2)=1，所以我们可以简单地加一个语句以避免出现计算fib(0)甚至fib(-1)的情况：

```
{
    if(n<=2)return 1;
    return fib(n-1) + fib(n-2);
}
```

现在这个f函数就完成了，结果同样，f函数和fib函数其实是一样的，所以你就可以把f改成fib，然后写上main函数：

```
#include <stdio.h>
int fib(n)
{
    if(n<=2)return 1;
    return fib(n-1) + fib(n-2);
}
int main(void)
{
    int n;
    scanf("%d", &n); //输入
    printf("%d", fib(n)); //计算并输出
    return 0;
}
```

怎么样？还算可以吧？慢慢理解一下吧:~)

15 1
喜欢 赠金笔

分享：

阅读 (2434) | 评论 (4) | 收藏 (0) | 转载 (3) | 喜欢 ▼ | 打印 | 举报

已投稿到： 排行榜

前一篇：C语言基础——初级程序结构的构思

后一篇：复活的历史

评论

重要提示：警惕虚假中奖信息

[发评论]

浮生若梦

这么好的东西怎么没人顶呢
我居然还有沙发坐
呵呵
顶一个~

2009-8-17 19:03

回复 (0)

蓝色忧郁



2010-11-3 15:40

回复 (0)

新浪网友

是好东西，但是我在学习还没用到数组，就已经开始将迭代和递归了。 所以使用数组来举例显然不是个好主义。

2013-2-10 17:33

回复 (0)

thinkerped

很nice

2016-3-3 18:55

回复 (0)

发评论

更多>>

登录名： 密码： 找回密码 注册 ☒ 记住登录状态

☒ 分享到微博 ☐ 评论并转载此博文

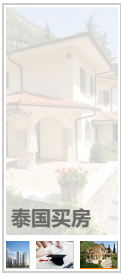
按住左边滑块，拖动完成上方拼图

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇
C语言基础——初级程序结构的构思

后一篇 >
复活的历史



新浪BLOG意见反馈留言板 不良信息反馈 电话：4006900000 提示音后按1键（按当地市话标准计费） 欢迎批评指正
新浪简介 | About Sina | 广告服务 | 联系我们 | 招聘信息 | 网站律师 | SINA English | 会员注册 | 产品答疑

Copyright © 1996 - 2017 SINA Corporation, All Rights Reserved
新浪公司 版权所有