Welcome

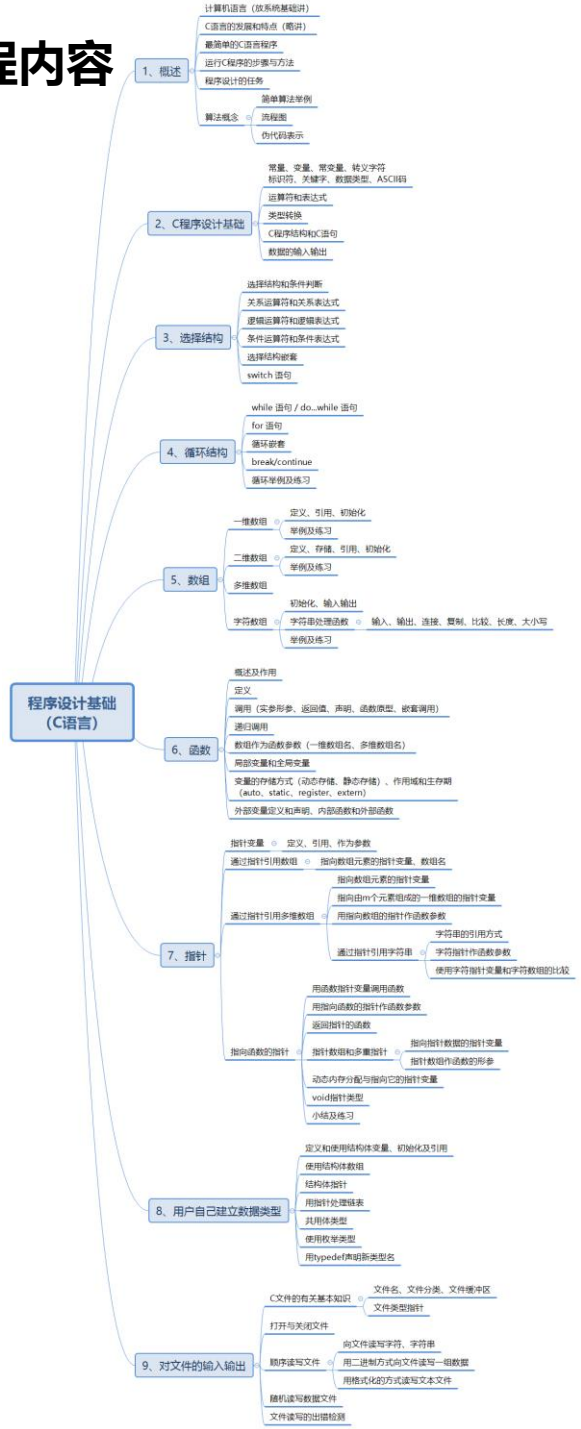
数据科学与大数据技术专业

程序设计基础(C语言)

上海体育学院经济管理学院

Wu Ying

课程内容



第5周 程序设计概述

计算机语言及C语言

- 最简单的C语言程序
  - 运行C程序的步骤和方法
  - 程序设计的任务

算法概念及简单算法

- 算法表示
  - 流程图
  - 伪代码

第6周 C程序设计基础

- 常量、变量及数据类型
  - 常量变量常变量转义字符
  - 标识符关键字数据类型

运算符和表达式及类型转换

C程序结构和C语句

数据的输入输出

第7周 选择结构

选择结构和条件判断

- 运算符和表达式
  - 关系运算符和关系表达式
  - 逻辑运算符和逻辑表达式
  - 条件运算符和条件表达式

选择结构嵌套

switch 语句

第8周 循环结构

while语句和do...while语句

for语句

循环嵌套，break，continue

循环练习

第9周 数组（一）

一维数组定义、引用、初始化

一维数组练习

二维数组定义引用初始化

多维数组及练习

第10周 数组（二）及函数（一）

字符数组初始化及输入输出

字符串处理函数

函数定义与调用

函数返回值、声明、原型、嵌套调用

第11周 函数（二）

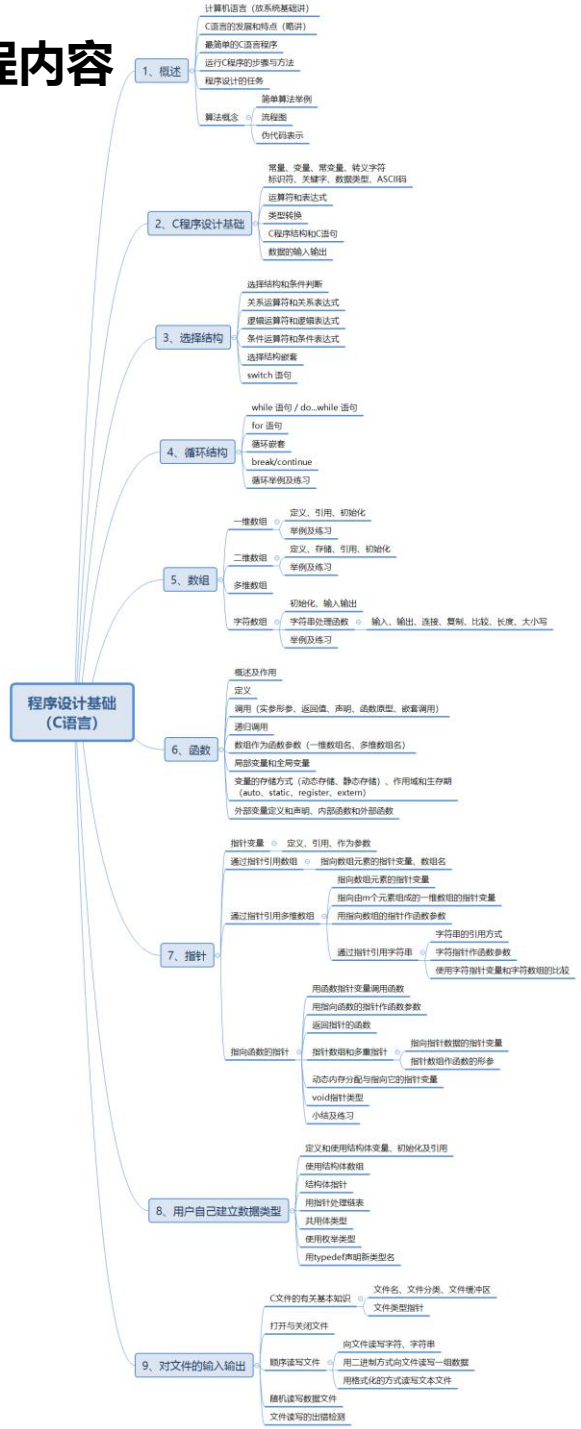
递归调用

数组作为函数参数

变量作用域和生存期

变量存储方式、内部函数与外部函数

课程内容



- 第12周 指针（一）
  - 指针变量
  - 通过指针引用数组
  - 通过指针引用多维数组（I）
    - 指向数组元素的指针变量
    - 指向一维数组的指针变量
- 第13周 指针（二）
  - 通过指针引用多维数组（II）
    - 用指向数组的指针作为函数参数
  - 通过指针引用字符串
    - 字符串的引用方式
    - 字符指针作函数参数
    - 使用字符指针变量和字符数组的比较
- 第14周 指针（三）
  - 指向函数的指针（I）
    - 用函数指针变量调用函数
    - 用指向函数的指针作函数参数
    - 返回指针的函数
  - 指向函数的指针（II）
    - 指针数组和多重指针
      - 指向指针数据的指针变量
      - 指针数组作函数的形参
    - 动态内存分配与指向它的指针变量
- 第15周 指针（四）及结构体（一）
  - 指向函数的指针（IV）
    - void指针类型
    - 小结及练习
  - 结构体
    - 定义和使用结构体变量
    - 使用结构体数组、结构体指针
- 第16周 结构体（二）及文件（一）
  - 用指针处理链表、共用体类型
  - 枚举类型、使用typedef声明新类型
  - C文件的基本知识
  - 打开与关闭文件
- 第17周 文件（二）
  - 顺序读写文件
  - 随机读写数据文件
  - 总复习练习
  - 总复习练习

第六章数组作业1：冒泡排序

发送对象：2020级数据1班 | 有效时段：2020-11-13 00:47 至 2020-11-25 00:47

请输入学号或姓名

选择	姓名
☐	孙奥
☐	何由
☐	曹思傲
☐	徐华镁
☐	何雨
☐	唐石金
☐	尤森煜
☐	陈佳骏

☐ 全选当前页

导出名单　批量加时

第六章数组作业1：冒泡排序

发送对象：2020级数据2班 | 有效时段：2020-11-13 00:47 至 2020-11-25 00:47

请输入学号或姓名

选择	姓名	学号/工号	状态	操作
☐	张雪玲	20760202	未查看	加时
☐	刘垚	20760208	未查看	加时
☐	宋金璇	20760214	未查看	加时
☐	陈帅	20760224	未查看	加时
☐	任一帆	20760125	已查看	加时
☐	褶家辉	20760203	已查看	加时
☐	蒋虎成	20760211	已保存　查看	加时

第五章循环作业2：8道编程题

发送对象：2020级数据1班 | 有效时段：2020-11-13 00:40 至 2020-11-30 00:41

请输入学号或姓名

姓名	学号/工号	状态
唐石金	20760105	未查看
曹思傲	20760108	未查看
徐华镁	20760121	未查看
何由	20760106	已保存 查看
朱振骐	20760117	已保存 查看
陈佳骏	20760118	已保存 查看
何雨	20760123	已保存 查看

返回

0:41

姓名	学号/工号	状态
任一帆	20760125	未查看
张雪玲	20760202	未查看
宋金璇	20760214	未查看
陈帅	20760224	未查看
褶家辉	20760203	已保存 查看

导出名单 督促学生

第五章循环作业1：6道编程题

发送对象：2020级数据1班 | 有效时段：2020-11-05 12:06 至 2020-11-14 12:06

请输入学号或姓名

姓名	学号/工号	状态
何由	20760106	未查看
徐华镁	20760121	未查看
宋金璇	20760214	未查看
唐石金	20760105	已查看

导出名单 督促学生

第五章循环作业1：6道编程题

发送对象：2020级数据2班 | 有效时段：2020-11-05 12:06 至 2020-11-14 12:06

请输入学号或姓名

姓名	学号/工号	状态
任一帆	20760125	未查看
刘垚	20760208	未查看
陈帅	20760224	未查看

习题 P215

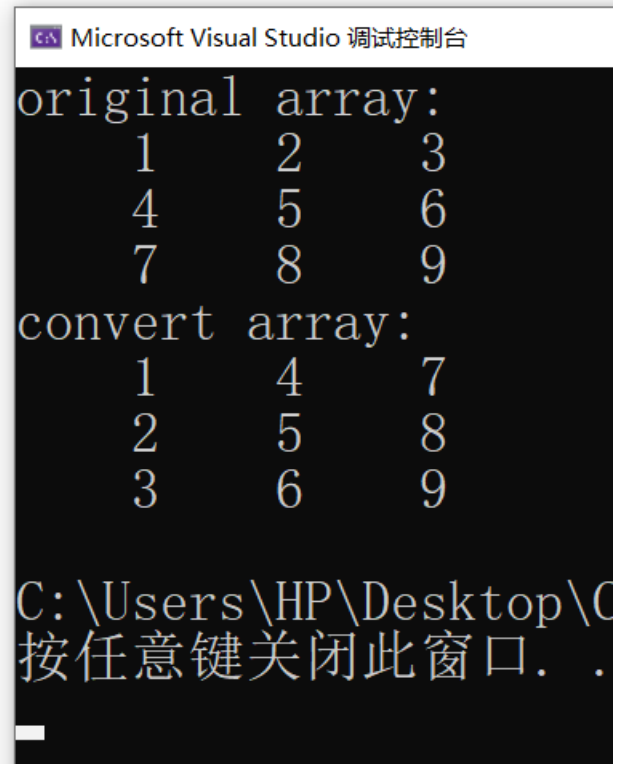
3.【作业】 写一个函数，使得给定的一个3*3的二位整型数组转置，行列互换

```c
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
void convert(int array[3][3])
{
    int i, j, t;
    for(i=0;i<3;i++)
        for (j = i + 1; j < 3; j++)
        {
            t = array[i][j];
            array[i][j] = array[j][i];
            array[j][i] = t;
        }
}

int main(void)
{
    int i, j;
    int a[3][3] = { 1,2,3,4,5,6,7,8,9 };
    printf("original array:\n");
    for(i=0;i<3;i++)
    {   for (j = 0; j < 3; j++)
            printf("%5d", a[i][j]);
        printf("\n");
    }

    convert(a);

    printf("convert array:\n");
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
            printf("%5d", a[i][j]);
        printf("\n");
    }
}
```



习题 P215

6 写一个函数，将两个字符串连接。

5 写一个函数，将一个字符串按逆序存放，在主函数中输入和输出字符串

数组作为函数参数

形式参数	实在参数
变量	常量、变量、表达式、数组元素
数组	数组

max(a,b);

myadd(a,100);

isprime(10+11); isprime(a[7]); ….isprime(a[i])…;isprime(11)

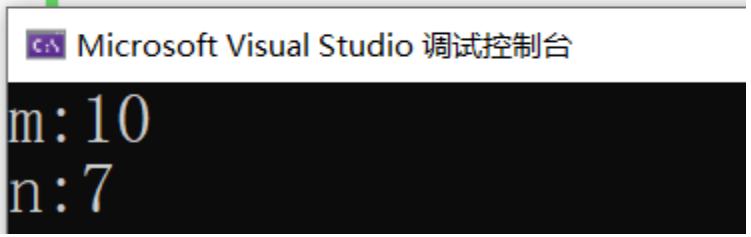
reverse(str);

mystrcat(s1,s2);

交换函数swift

```c
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
void swift(int* x,int* y)
{
    int t;
    t = *x;
    *x = *y;
    *y = t;
}

int main(void)
{
    int m, n;
    m = 7;
    n = 10;
    swift(&m, &n);
    printf("m:%d\nn:%d\n", m, n);
}
```



```c
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
void reverse(char s[])
{
    int i, j;
    char t;
    i = 0;
    j = strlen(s)-1;
    while (i <=j)
    {
        t = s[i];
        s[i] = s[j];
        s[j] = t;
        i++;j--;
    }
}

int main()
{
    char s1[100];
    printf("请输入字符串\n");
    gets_s(s1,99);
    reverse(s1);
    printf("%s\n", s1);
}
```

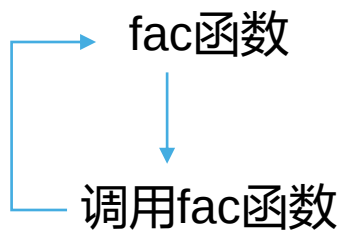
函数的递归调用

函数的递归调用

在调用一个函数的过程中又出现**直接或间接地调用该函数本身，称为函数的递归调用。**

$$n!=\begin{cases} n!=1 & (n=0,1) \\ n\cdot(n-1)! & (n>1) \end{cases}$$

```
int fac(int n)                    //定义fac函数
{
    int f;
    if(n<0)                       //n不能小于0
        printf("n<0,data error!");
    else
     {
        if(n==0||n==1)     //n=0或,1时n!=1
            f=1;                  //递归终止条件
        else
            f = n * fac(n-1);    //n>1时，n!=n*(n-1)!
     }
    return(f);
}
```



直接递归

函数的递归调用

函数的递归调用

```
f(6)
=> 6 * f(5)
=> 6 * (5 * f(4))
=> 6 * (5 * (4 * f(3)))
=> 6 * (5 * (4 * (3 * f(2))))
=> 6 * (5 * (4 * (3 * (2 * f(1)))))
=> 6 * (5 * (4 * (3 * (2 * 1))))
=> 6 * (5 * (4 * (3 * 2)))
=> 6 * (5 * (4 * 6))
=> 6 * (5 * 24)
=> 6 * 120
=> 720
```

递归是很多算法用的一种编程方法：定义即解决

三步走：

1）明确函数功能

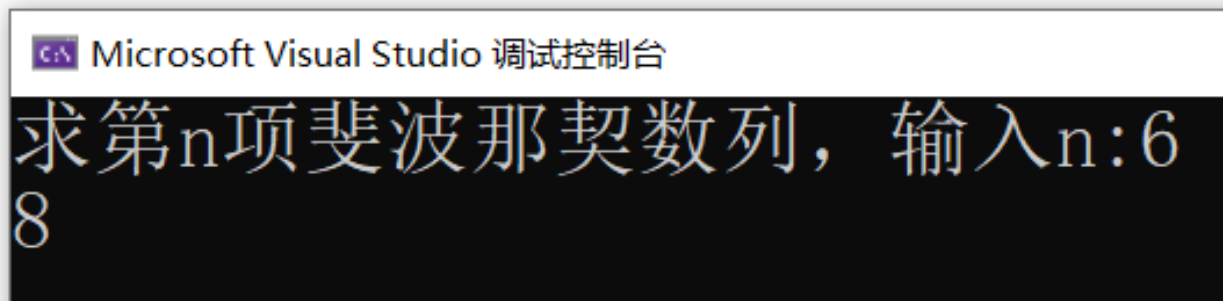
2）寻找递归结束条件

3）写出函数定义的关系表达式

斐波那契数列的是这样一个数列：1、1、2、3、5、8、13、21、34....，即第一项 f(1) = 1,第二项 f(2) = 1.....,第 n 项目为 f(n) = f(n-1) + f(n-2)。求第 n 项的值?

```c
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
int fibonacci(int m)
{
    if (m == 1 || m == 2)
        return 1;
    else
        return fibonacci(m - 1) + fibonacci(m - 2);
}


int main()
{
    int n;
    printf("求第n项斐波那契数列，输入n:");
    scanf("%d", &n);
    printf("%d\n",fibonacci(n));
}
```



小青蛙跳台阶

一只青蛙一次可以跳上1级台阶，也可以跳上2级。求该青蛙跳一个n级的台阶总共有多少种跳法。

三步走：

1）明确函数功能

2）寻找递归结束条件

3）写出函数定义的关系表达式

step1: 假设 f(n) 的功能是求青蛙跳上一个n级的台阶总共有多少种跳法

```
int f(int n){

}
```

step2: 递归结束的条件，n = 1时，f(1) = 1；n=2时， f(2) = 2

step3:写出函数定义的关系表达式

每次跳的时候，小青蛙可以跳一个台阶，也可以跳两个台阶，也就是说，每次跳的时候，小青蛙有两种跳法。

第一种跳法：第一次跳了一个台阶，那么还剩下n-1个台阶还没跳，剩下的n-1个台阶的跳法有f(n-1)种。

第二种跳法：第一次跳了两个台阶，那么还剩下n-2个台阶还没，剩下的n-2个台阶的跳法有f(n-2)种。

所以，小青蛙的全部跳法就是这两种跳法之和了，即 f(n) = f(n-1) + f(n-2)

函数的递归调用

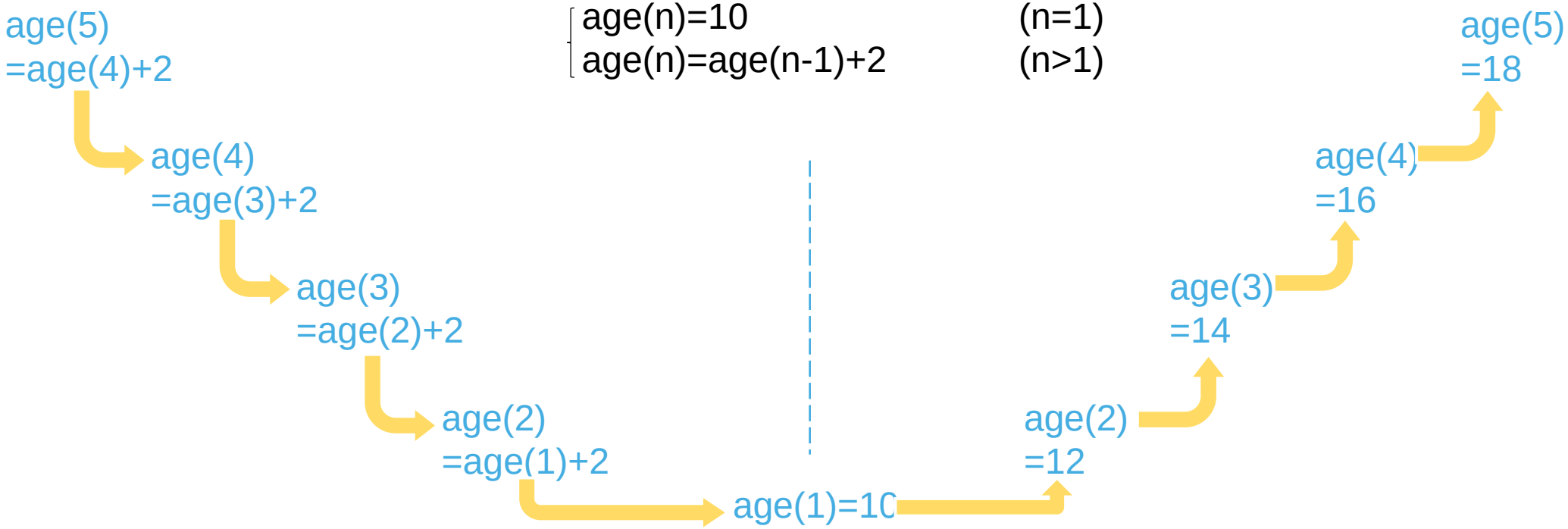
【例7.6】有5个学生坐在一起，问第5个学生多少岁，他说比第4个学生大2岁。问第4个学生岁数，他说比第3个学生大2岁。问第3个学生，又说比第2个学生大2岁。问第2个学生，说比第1个学生大2岁。最后问第1个学生，他说是10岁。请问第5个学生多大。

函数的递归调用

【例7.6】有5个学生坐在一起，问第5个学生多少岁，他说比第4个学生大2岁。问第4个学生岁数，他说比第3个学生大2岁。问第3个学生，又说比第2个学生大2岁。问第2个学生，说比第1个学生大2岁。最后问第1个学生，他说是10岁。请问第5个学生多大。

解题思路:

$$age(n)=\begin{cases}10 & (n=1)\\ age(n-1)+2 & (n>1)\end{cases}$$

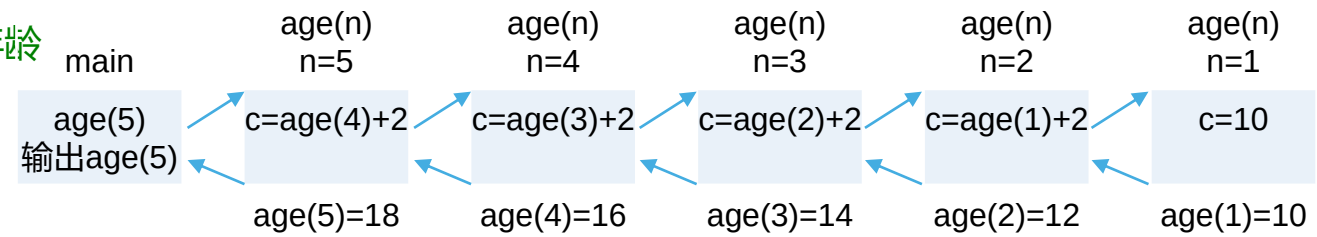


函数的递归调用

【例7.6】有5个学生坐在一起，问第5个学生多少岁，他说比第4个学生大2岁。问第4个学生岁数，他说比第3个学生大2岁。问第3个学生，又说比第2个学生大2岁。问第2个学生，说比第1个学生大2岁。最后问第1个学生，他说是10岁。请问第5个学生多大。

```
#include <stdio.h>
int main()
{     int age(int n);                    //对age函数的声明
      printf("NO.5,age:%d\n",age(5));    //输出第5个学生的年龄
      return 0;
}

int age(int n)                           //定义递归函数
{     int c;                             //c用作存放函数的返回值的变量
      if(n==1)                           //如果n等于1
            c=10;                        //年龄为10
      else                                     //如果n不等于1
            c=age(n-1)+2;                //年龄是前一个学生的年龄加2(如第4个学生年龄是第3个学生年龄加2)
      return(c);                         //返回年龄
}
```



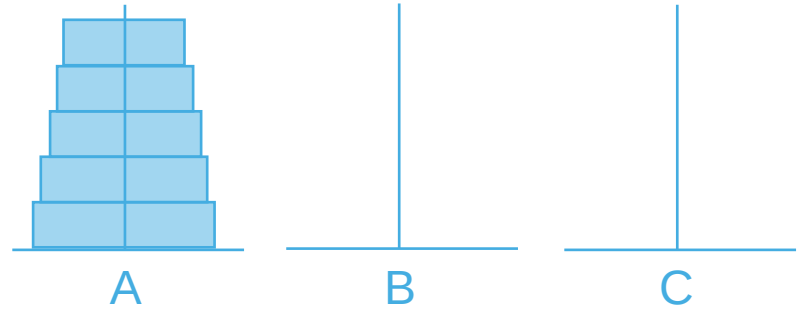
注意分析递归的终止条件。

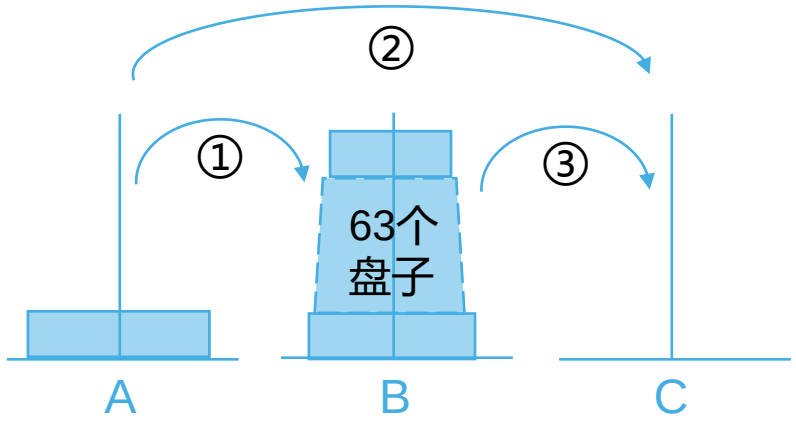
函数的递归调用

【例7.8】Hanoi（汉诺）塔问题。古代有一个梵塔，塔内有3个座A,B,C。开始时A座上有64个盘子，盘子大小不等，大的在下，小的在上。有一个老和尚想把这64个盘子从A座移到C座，但规定每次只允许移动一个盘，且在移动过程中在3个座上都始终保持大盘在下，小盘在上。在移动过程中可以利用B座。要求编程序输出移动盘子的步骤。

汉诺塔小游戏 (hannuota.cn)

http://www.hannuota.cn/





解题思路:

为便于理解，先分析将A座上3个盘子移到C座上的过程:

① 将A座上2个盘子移到B座上（借助C座）。

② 将A座上1个盘子移到C座上。

③ 将B座上2个盘子移到C座上（借助A座）。

其中第②步可以直接实现。第①步又可用递归方法分解为:

将A座上1个盘子从A座移到C座;

将A座上1个盘子从A座移到B座;

将C座上1个盘子从C座移到B座。

第③步可以分解为:

将B座上1个盘子从B座移到A座上;

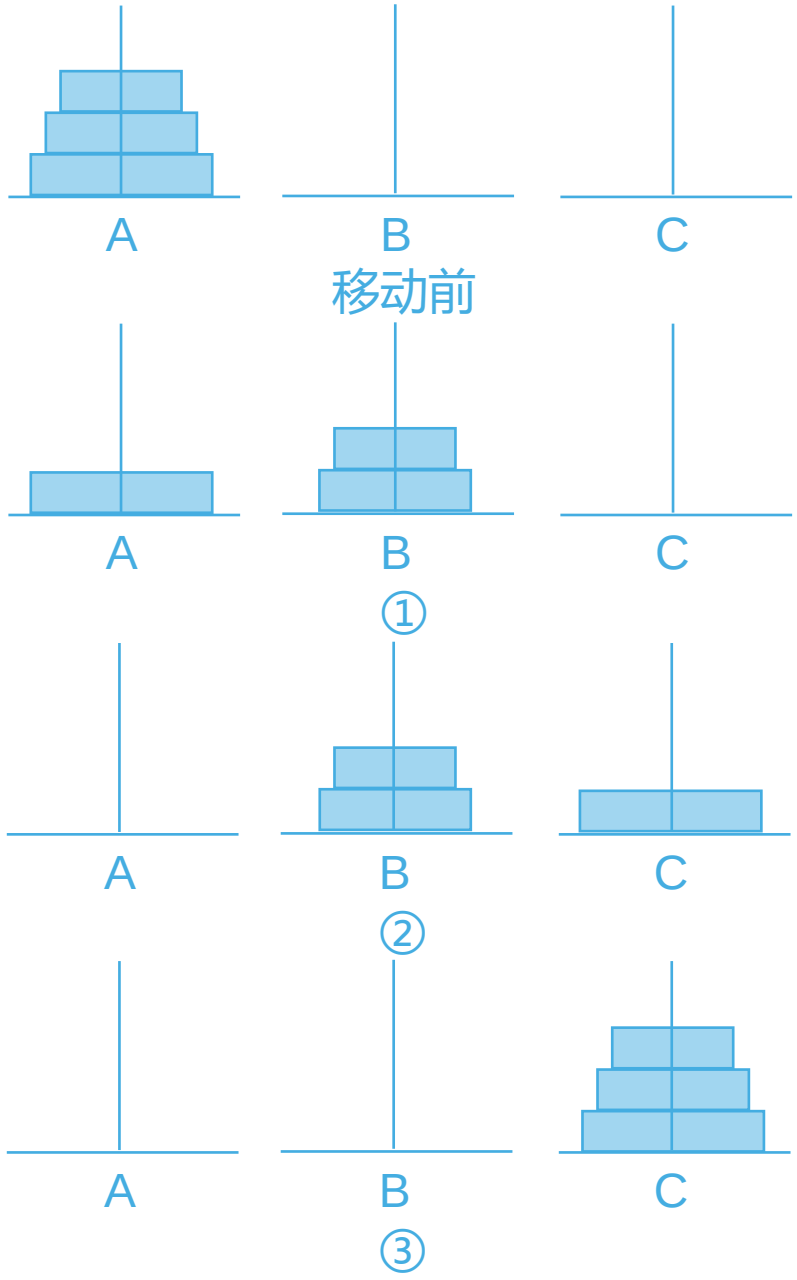
将B座上1个盘子从B座移到C座上;

将A座上1个盘子从A座移到C座上。

将以上综合起来，可得到移动3个盘子的步骤为:

A→C，A→B，C→B，A→C，B→A，B→C，A→C。

共经历7步。由此可推出: 移动n个盘子要经历(2^n-1)步。



```c
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
void move(char x, char y)
{
    printf("%c->%c\n",x, y);
}
void hanoi(int n, char one,char two,char three)
{
    if (n == 1)
        move(one, three);
    else
    {
        hanoi(n - 1, one, three, two);
        move(one, three);
        hanoi(n - 1, two, one, three);
    }
}

int main()
{
    int m;
    printf("input the number of disks:");
    scanf("%d", &m);
    printf("The step to move %d disks:\n", m);
    hanoi(m, 'A', 'B', 'C');
}
```

Microsoft Visual Studio 调试控制台

```
input the number of disks:3
The step to move 3 disks:
A->C
A->B
C->B
A->C
B->A
B->C
A->C

C:\Users\HP\Desktop\CPractice
按任意键关闭此窗口. . .
```

```c
#include <stdio.h>
int main()
{
    void hanoi(int n,char one,char two,char three);
    //对hanoi函数的声明
    int m;
    printf("input the number of diskes:");
    scanf("%d",&m);
    printf("The step to move %d diskes:\n",m);
    hanoi(m,'A','B','C');
}
```

```c
void hanoi(int n,char one,char two,char three) //定义hanoi函数
//将n个盘从one座借助two座,移到three座
{   void move(char x,char y);   //对move函数的声明
    if(n==1)
        move(one,three);
    else
    {   hanoi(n-1,one,three,two);
        move(one,three);
        hanoi(n-1,two,one,three);
    }
}
void move(char x,char y)            //定义move函数
{   printf("%c->%c\n",x,y); }
```

在本程序中，调用递归函数hanoi，其终止条件为hanoi函数的参数n的值等于1。显然，此时不必再调用hanoi函数了，直接执行move函数即可。
在本程序中move函数并未真正移动盘子，而只是输出移盘的方案（表示从哪一个座移到哪一个座）。