

实验 1 算法问题求解基础

一、实验目的

1. 掌握算法的概念；
2. 了解问题的求解方法；
3. 理解递归思想，学会编写递归；

二、实验原理

1. 算法

一个算法是对特定问题求解步骤的一种描述，它是指令的有限序列。算法具有下列 5 个特征：

- 1、有穷性 (Finiteness)：算法的有穷性是指算法必须能在执行有限个步骤之后终止；
- 2、确切性 (Definiteness)：算法的每一步骤必须有确切的定义；
- 3、输入项 (Input)：一个算法有 0 个或多个输入，以刻画运算对象的初始情况，所谓 0 个输入是指算法本身定出了初始条件；
- 4、输出项 (Output)：一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的；
- 5、可行性 (Effectiveness)：算法中执行的任何计算步骤都是可以被分解为基本的可执行的操作步，即每个计算步都可以在有限时间内完成（也称之为有效性）。

2. 问题求解过程

理解问题、设计方案、实现方案、回顾复查。

3. 递归

在数学与计算机科学中，是指在函数的定义中使用函数自身的方法。也就是说，递归算法是一种直接或者间接调用自身函数或者方法的算法。通俗来说，递归算法的实质是把问题分解成规模缩小的同类问题的子问题，然后递归调用方法来表示问题的解。

递归的基本原理：

第一：每一级的函数调用都有自己的变量。

第二：每一次函数调用都会有一次返回。

第三：递归函数中，位于递归调用前的语句和各级被调用函数具有相同的执行顺序。

第四：递归函数中，位于递归调用后的语句的执行顺序和各个被调用函数的顺序相反。

第五：虽然每一级递归都有自己的变量，但是函数代码并不会得到复制。

递归的优缺点：

优点	缺点
1、实现简单； 2、可读性好；	1、递归调用，占用空间大； 2、递归太深，容易发生栈溢出； 3、可能存在重复计算；

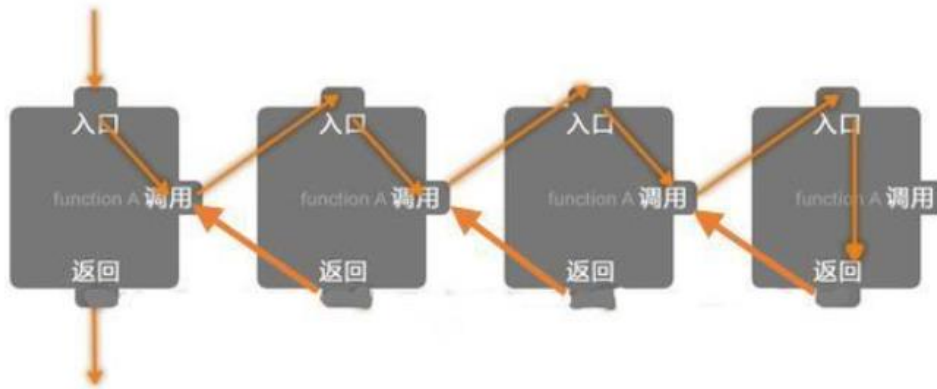
递归的三大要素：

第一要素：明确你这个函数想要干什么。先不管函数里面的代码什么，而是要先明白，你这个函数的功能是什么，要完成什么样的一件事；

第二要素：寻找递归结束条件。我们需要找出当参数为啥时，递归结束，之后直接把结果返回，请注意，这个时候我们必须能根据这个参数的值，能够直接知道函数的结果是什么；

第三要素：找出函数的等价关系式。我们要不断缩小参数的范围，缩小之后，我们可以通过一些辅助的变量或者操作，使原函数的结果不变；

递归的过程：



具体地说，如果递归函数调用自己，则被调用的函数也将调用自己，这将无限循环下去，除非代码中包含终止调用链的内容。通常的方法将递归调用放在 if 语句中。例如，void 类型的递归函数 recurs() 的代码如下：

```
1 void recurs(argumentlist){
2     statements1
3     if(test)           //test最终为false, 调用链将断开
4         recurs(arguments)
5     statements2
6 }
```

用文字描述这段代码块的内容：

只要 if 语句为 true，每个 recurs() 调用都将执行 statements1，然后再调用 recurs()，而不会执行 statements2。当前调用结束后，程序控制权将返回给调用它的 recurs()，而该 recurs() 将执行其 statements2 部分，然后结束，并将控制权返回给前一个调用，依次类推。

递归的使用

递归的强大之处在于它允许用户用有限的语句描述无限的对象。因此，在计算机科学中，递归可以被用来描述无限步的运算，尽管描述运算的程序是有限的。这一点是循环不太容易做到的。

编写正确的递归算法，一定要有“归”的步骤，也就是说递归算法，在分解问题到不能再分解的步骤时，要让递归有退出的条件，否则就会陷入死循环，最终导致内存不足引发栈溢出异常。

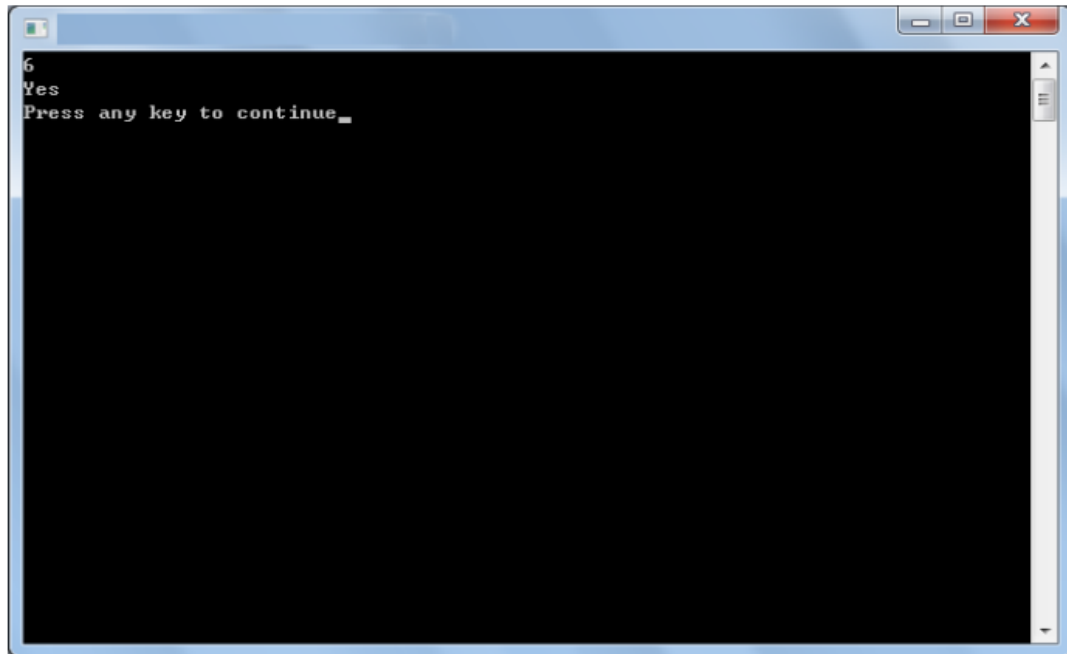
三、实验内容

第 1 题：编写一个递归程序求解下列问题

给定正整数 n ($n \leq 10000$)，确定 n 是否是它所有因子之和（完数）。

输入：正整数 n 输出：若 n 是它所有因子之和，输出 Yes，否则输出 No.

如下图所示：



什么是完数？

如果一个数恰好等于它的因子之和，则称该数为“完数”或“完全数”，例如第一个完全数是 6，它有约数 1、2、3、6，除去它本身 6 外，其余 3 个数相加， $1+2+3=6$ 。第二个完全数是 28，它有约数 1、2、4、7、14、28，除去它本身 28 外，其余 5 个数相加， $1+2+4+7+14=28$ 。第三个完全数是 496，有约数 1、2、4、8、16、31、62、124、248、496，除去其本身 496 外，其余 9 个数相加， $1+2+4+8+16+31+62+124+248=496$ 。后面的完全数还有 8128、33550336 等等。

四、本次实验要求

- 1、提交本次实验的代码并标上简单的注释内容；
- 2、附上本次实验运行的结果；
- 3、简单介绍本次实验的心得体会；
- 4、提交邮箱地址（周日前提交）：274704373@qq.com