## 湖南师范大学信息科学与工程学院实验中心

### 《程序设计基础》 课程实验报告

|  |  |
| --- | --- |
| **记分** |  |

软件工程 **专业** 2018 **年级** 01 **班 学号** 201730184007 **姓名**  曾谞旺

**指导老师** 蔡美玲 **实验日期** 2018 **年** 11 **月** 15 **日**

**实验项目名称：** 实验九 递归程序设计

1. 实验目的

1.理解递归的思想，理解递归应用的场景。

2.理解递归程序执行的基本原理。

3.掌握设计递归程序的方法。。

4.应用递归解决实际问题。

1. 实验仪器

台式或笔记本电脑。

1. 实验内容
   1. 斐波那契数列
   2. 最大公约数
   3. 十进制转r进制
2. 实验步骤（源程序）

斐波那契数列：

1. #include <stdio.h>
2. #include <stdlib.h>
3. **long** Fabonaci(**int** n);
4. **int** main()
5. {
6. **int** n;
7. **long** output;
8. printf("Input n:\n");
9. scanf("%d",&n);
10. output = Fabonaci(n);
11. **if**(output == -1){
12. printf("Input error!\n");
13. }**else** {
14. printf("fabnaci[%d] = %d\n",n,output);
15. }
16. **return** 0;
17. }
19. //此函数求斐波那契数列的第n项的值。当返回是-1时，表明输入的n有误。
20. **long** Fabonaci(**int** n)
21. {
22. **long** fabonaci;
23. **if**(n < 1)**return** -1;
24. **if**(n == 1 || n == 2) {
25. fabonaci = 1;
26. }**else** {
27. fabonaci = Fabonaci(n - 1) + Fabonaci(n - 2);
28. }
29. **return** fabonaci;
31. }

最大公约数:

1. #include <stdio.h>
2. #include <stdlib.h>
3. **int** MaxCommonFactor(**int** a, **int** b);
5. **int** main()
6. {
7. **int** a,b,output;
8. printf("Input two positive integers,a,b:\n");
9. scanf("%d,%d",&a,&b);
10. **if**(a <= 0 || b <= 0){
11. printf("Input error!\n");
12. **return** 0;
13. }
14. output = MaxCommonFactor(a,b);
15. printf("MaxCommonFactor = %d\n",output);
16. **return** 0;
17. }
19. //函数功能： 计算两个正整数的最大公约数，-1表示没有最大公约数
20. //要求使用递归实现
21. **int** MaxCommonFactor(**int** a, **int** b)
22. {
23. **if**(a == b) **return** a;
24. **if**(a > b){
25. a = MaxCommonFactor(a - b,b);
26. }**else**{
27. b = MaxCommonFactor(a,b - a);
28. }
29. }

十进制转r进制:

1. #include <stdio.h>
2. #include <stdlib.h>
4. **void** TransformToR(**int** n,**int** r);
5. **void** PrintNumber(**int** n);
6. **int** main()
7. {
8. printf("Input n,r(r=2,8,16):\n");
9. **int** n,r;
10. scanf("%d,%d",&n,&r);
11. TransformToR(n,r);
12. printf("\n");
13. **return** 0;
14. }
16. //此函数实现将整数n转换为对应的r进制数进行输出
17. **void** TransformToR(**int** n,**int** r){
18. **if**(n>=r){
19. TransformToR(n/r,r);    //交换位置试试？利用栈的特性
20. PrintNumber(n%r);
21. }
22. **else** PrintNumber(n);
23. }
25. //此函数用于打印一个字符，用于各进制的字符"A"~"Z"形式的输出
26. **void** PrintNumber(**int** n){
27. **if**(n>=10) printf("%c",65+n-10);
28. **else** printf("%d",n);
29. }
30. 实验数据、分析与实验结果

**# 斐波那契数列:**

**这个数列本身的定义就是通过递归来定义的，所以，我们只需要写出边界条件(终止递归的条件)，以及给出的递归方程就能正确求解**

**# 最大公约数**

**这道求最大公约数的题，使用辗转相除法来求解，就能使用递归。**

**# 十进制转r进制**

**根据除k取余法的原理，我们知道递归方程，出口就是当这个数小于r的时候直接输出。**

1. 总结、心得体会

* 注意递归中某些语句的位置，对应出栈的顺序
* 要使用递归，首先得写出出口(边界条件)，其次才是递归方程
* 使用递归，原问题可以分解为若干个规模相当的子问题
* 递归适用的情景，递归不适用于内联函数和宏
* 求解一个问题，若能使用递归，则只需要展开到下一层，千万不要一直向下展开
* 递归还是很重要的，学好递归，对于我们后续学校数据结构中的二叉树以及算法中的动态规划，分支限界都是很重要的。