算法设计与分析实验报告

班级		学号		姓名	
实验名称	2.递归算法	日期	2023.9.21	成绩	
实验目的和要求	1.调试 1 个程序; 2.设计 2 个程序;				
实验准备	熟悉编程环境; 理解递归算法思想,进行编程实践				

实验内容、实验结果与分析	**问题 1：调试如下程序，写出运行结果**
	1 //递归求两个数的最大公约数
	2 //The greatest common divisor of 24 and 30 is 6
	3 #include <stdio.h>
	4 #include <stdlib.h>
	5
	6 int GCD(int m, int n)
	7 {
	8 if(m%n=0)
	9 return n;
	10 else
	11 return GCD(n, m%n);
	12 }
	13 int main(void)
	14 {
	15 int m=24; n=30;
	16 printf(" (%d, %d) = %d\n\n", m,n, GCD(m,n));
	17
	18 return 1;
	19 }
	20
	
	要求: 指出程序语句中的错误并改正,写到实验报告里(第?行*** 改为 ***); 再测试两个整数的最大公约数；所有运行出结果填写到如下的实验与结果中。
	
	实验结果与分析：
	
	第 8 行：if(m%n=0) 改为 if(m % n == 0)
	第 15 行：int m=24; n=30; 改为 int m = 24; int n = 30;
	
	结果：**(24, 30) = 6**					

问题 2：实现整数划分问题中求解划分数的递归程序，并进行 2-5 次的测试。思考：是否能把所有的划分列出来？

程序：

```c
#include<stdio.h>

int splitInteger(int a, int b) {
    // 输出整数 n 有多少种划分方式

    if (a == 1 || b == 1) {
        return 1;
    }
    else if ( a < b) {
        return splitInteger(a, a);
    }
    else if ( a == b ) {
        return 1 + splitInteger(a, a - 1);
    }
    else {
        return splitInteger(a, b - 1) + splitInteger(a - b, b);
    }
}

int main(void) {

    int n;
    do {
        printf("请输入希望划分的那个整数:");
        scanf_s("%d", &n);

        if (n < 0) {
            printf("请输入大于等于 0 的数！\n");
        }
    } while (n < 0);

    printf("%d 的不同划分方式有%d 种\n", n, splitInteger(n, n));

    return 0;
}
```

测试结果：

输入 6，输出：6 的不同划分方式有 11 种

问题 3：实现 Ackerman 函数的递归程序，并进行 2-5 次的测试。

程序：

```c
#include<stdio.h>

int Ackerman(int a, int b) {
    // 输出 Ackerman 函数的值

    if (a == 1 && b == 0) {
        return 2;
    }else if (a == 0) {
        return 1;
    }else if (b == 0) {
        return a + 2;
    }else {
        return Ackerman(Ackerman(a - 1, b), b - 1);
    }
}

int main(void) {
    int m, n;
    do {
        printf("请输入 m:");        scanf_s("%d", &m);
        printf("请输入 n:");        scanf_s("%d", &n);

        if (m < 0 || n < 0 || ( m == 0 && n == 0 )) {
            printf("请输入大于等于 0 的 m 和 n 的值，而且 m 和 n 不能同时为
0! \n");
        }
    } while (m < 0 || n < 0 || (m == 0 && n == 0));
    printf("Ackerman(%d, %d)的值为%d", m, n, Ackerman(m, n));
    return 0;
}
```

测试结果：

输入：m = 2, n = 2, 输出：Ackerman(2, 2)的值为 4
输入：m = 3, n = 2, 输出：Ackerman(2, 2)的值为 8
输入：m = 1, n = 5, 输出：Ackerman(2, 2)的值为 2
输入：m = 3, n = 3, 输出：Ackerman(2, 2)的值为 16
输入：m = 4, n = 2, 输出：Ackerman(2, 2)的值为 16

总结与讨论	

注：完成作业后，把电子版提交至 课堂派

作业名称：学号姓名实验 02 递归