# 实验五：函数(2 学时)

## 一、实验方式：上机

## 二、实验目的：

1、熟悉函数的定义、函数声明、函数调用和函数嵌套。
2、掌握通过参数在函数间传递数据的方法。
3、掌握全局变量和局部变量。
4、掌握常用算法—递归。

## 三、实验内容及其步骤：

1、编写程序实现 1~ n 的阶乘然后每项阶乘进行求和，用两个函数分别实现阶乘和求和的效果。

```
#include<stdio.h>
long Factorial(int n);                                  //函数声明
long Sum(int n);                                        //函数声明
int main(void)
{
    int num;
    printf("从键盘输入 n 的值：");
    scanf("%d", &num);
    printf("1!+2!+...%d! = %ld\n", num, Sum(num));
    //  调用求和函数 Sum()
    return 0;
}
long Sum(int n)                         //定义函数 Sum，实现累加求和
{
    int i;
    long result = 0;
    for (i = 1; i <= n; i++)
    {
        result += Factorial(i);   //嵌套调用求阶乘函数 Factorial()
    }
    return result;
}
long Factorial(int n)                   //定义函数 Factorial()，实现求阶乘
{
    int i;
    long result = 1;
    for (i = 1; i <= n; i++)
    {
```

```
            result *= i;
        }
        return result;
}
```

2、编写程序实现对 n 求阶乘，要求定义阶乘的函数，并且使用递归来实现。

```
#include<stdio.h>
long Factorial(int n)
{
        if (n == 0 || n == 1)                                   //递归的基线条件
        {
                return 1;
        }
        else
        {
                return Factorial(n - 1) * n;
        }
}
int main(void)
{
        int a;
        printf("Input a number: ");
        scanf("%d", &a);
        printf("Factorial (%d) = %ld\n", a, Factorial(a));    //调用自定义函数 Factorial()。
        return 0;
}
```

3、有 5 个人坐在一起，问第 5 个人多少岁，他说比第 4 个人大 2 岁。问第 4 个人多少岁，他说比第 3 个人大 2 岁。问第 3 人多少岁，他说比第 2 个人大 2 岁。问第 2 个人多少岁，他说比第 1 个人大 2 岁。最后问第 1 个人，他说他是 10 岁。

分析：该问题是一个递归问题。要求第 5 个人的年龄，必须先知道第 4 个人的年龄，显然第 4 个人的年龄也是未知的，但可以由第 3 个人的年龄推算出来。而想知道第 3 个人的年龄又必须先知道第 2 个人的年龄，第 2 个人的年龄则取决于第 1 个人的年龄。又已知每个人的年龄都比其前一个人的年龄大 2，因此根据题意，可得到如下几个表达式：

age(5)=age(4)+2

age(4)=age(3)+2

age(3)=age(2)+2

age(2)=age(1)+2

age(1)=10

自行编写程序，自定义一个年龄的函数，当输入第几个人时求出其对应的年龄，并在主函数中打印输出。

4、分析程序的运行结果，为什么是这个样子，了解全局变量和局部变量。

```
#include<stdio.h>

int n = 10;                          //全局变量

void Func1(void)
{
    int n = 20;                      //局部变量
    printf("Func1 n: %d\n", n);
}
void Func2(int n)
{
    printf("Func2 n: %d\n", n);
}

void Func3(void)
{
    printf("Func3 n: %d\n", n);
}

int main(void)
{
    int n = 30;                       //局部变量
    Func1();
    Func2(n);
    Func3();
    {                                 //代码块
        int n = 40;                   //局部变量
        printf("Block n: %d\n", n);
    }
    printf("main n: %d\n", n);
    return 0;
}
```