**第十五章 函数**

1.C语言中子程序称为**函数**。

2.C语言是面向函数的，C程序本质上是函数的集合，每条语句属于并仅属于一个函数。

3.C语言程序总是从main函数开始，在main函数中调用其他函数，这些函数也可以依次调用更多函数，控制最终会返回main函数，main函数结束时程序结束。

4.**函数声明**称为**函数原型**，包括函数名称、返回值类型、输入值列表，以分号;结束。

5.**没有返回值的函数返回值类型为void**。

6.函数名称按标识符命名规则。

7.**帕斯卡命名法：**单词首字母大写 ，其余部分小写。

8.**骆驼式命令法：**单词首字母大写，其余部分小写，但第一个单词首字母小写。

9.**下划线法：**函数名中的每一个逻辑断点都有一个下划线来标记。

10.函数声明括号中描述函数需要输入的参数的类型和顺序，**可以不指明参数名而只写参数类型**，参数名也可以与函数定义时的形式参数不同。不需要参数时括号里为空。

10.在调用者内部被传给被调用者的值被称为**变元**。

11.函数定义括号中的是**形式参数列表**，变元与形式参数类型和顺序匹配。

12.**任何调用者的局部变量对被调用函数不可见。**

13.用**return**指明返回的值，必须与声明返回的类型匹配。

14.**若有返回值的函数没有使用return语句，则最后一条语句的值作为返回值返回给调用者。**

15.函数定义在函数调用前，可以不使用函数声明。

16.变元运算顺序自右向左。例：function(i,i++);。

17.一个函数能被任何一个函数调用（包括其本身）。

18.C函数在DLX底层实现时，参数个数若多于4个将使用存储器，且使用运行时栈机制，这段存储空间称为函数的**栈框架**或**活动记录**。

19.编译器为每一次函数调用，在存储器中分配一个活动记录；当函数返回时，它的活动记录将被回收，以便分配给后面的函数；每一次函数调用都会在存储器中为其局部数值获得它自己的空间。优点：允许函数递归。

20.R29栈指针指向栈顶，R30框架指针指向活动记录底。

21.**动态链接**：为调用者的框架指针制作的副本。

22.**叶函数**：在执行过程中没有调用其他函数的函数，且没有改变R31。

23.头文件包含函数声明，预处理宏，但不包含库函数的源代码，即头文件已编译。

24.数学库函数使用头文件math.h。

25.**printf函数格式说明少于参数时，将使用期望在存储器指定位置的垃圾值。**

26.**scanf函数返回值为该函数在输入流中成功扫描的格式说明的个数。**

27.**将C语言函数调用翻译为DLX汇编语言**

①通过栈指针R29对局部变量R16-R23、临时寄存器R8-R15、R24、R25、参数传递寄存器R2、R3压栈（只对将要修改但还需使用的寄存器压栈）。

subi r29 r29 #4

sw 0(r29) SR1

subi r29 r29 #4

sw 0(r29) SR2

……

subi r29 r29 #4

sw 0(r29) SRn

②参数由R4-R7 传递。

③返回值由R2和R3传递。

④通过栈指针R29将压栈的寄存器出栈。

lw SRn 0(r29)

addi r29 r29 #4

……

lw SR2 0(r29)

addi r29 r29 #4

lw SR1 0(r29)

addi r29 r29 #4

ret