

前七章

- 1、&代表的是取地址，在scanf输入的时候需要用到&。
- 2、C语言中的库函数并不参与编译，他们在创建的时候就已经被编译成为.obj文件，知识在链接的时候与C语句链接到一块。基本类型和枚举类型都是算数类型，算术类型和指针类型都是纯量类型。字符型类型也属于整型类型
- 3、%m.nf目标是输出的数据占多少列，n表示保留多少位小数
- 4、系统中判断真假的时候都是以0(假)和非0(真)进行判断的。
- 5、C语言是将字符串中的数组存储到一维数组中，字符串数据在输出时候没输出一个字符，就会检查后面是否是空操作符即\0。
- 6、一个C程序由若干个程序模块组成，每个程序模块作为一个源程序文件，每个源程序文件可以独立的进行编译，一个源程序文件可以供多个C程序使用，程序在编译时是以源程序为单位进行编译的。函数不可以嵌套定义，但是可以相互调用。从函数角度看，函数分为有参函数和无参函数，从用户的角度看，参数分为库函数和用户自己定义的函数。
- 7、函数中的形参在未出现函数调用的时候并不会为其分配存储单元，当出现函数调用的时候才会被临时分配存储单元。实参向形参进行的是值传递，单向传递，只能由实参传递给形参。自定义的函数的位置在主函数之后应该在主函数中事先对其进行声明，而后再进行使用。数组元素可以作为函数的实参，但是不可以作为形参，因为形参没有存储空间，实在函数调用的时候临时进行空间分配的。一维数组名既可以作为形参也可以作为实参和形参，作为形参时候传递的是数组首元素的地址。二维数组可以作为形参和实参，但是在函数定义声明参数时候，应注意不能忽略高维的大小，可以忽略一维的大小。
- 8、变量的存储方式有动态和静态两种，动态区主要存储没有使用static声明，形参，现场保护和返回地址。
- 9、局部变量存储类别：**自动变量**、**静态变量**（static修饰）和**寄存器变量**。**自动变量**存储在动态存储区。**该变量（静态）**在函数执行结束的时候存储空间并不会释放，而是会一直保留，以便下次调用函数的时候使用。静态变量存储在静态存储区，编译时赋初值且仅可以赋初值一次，再次调用时保留的是上一次运行的结果。对自动变量赋初值是在函数调用的时候进行的，而不是在函数编译的时候进行的。**寄存器变量**是指一些经常性使用的变量，为了提高变量的存取速度，将该变量存储在CPU寄存器之中。
- 10、全局变量的存储类别:全局变量都是存在静态存储区之中的。全局变量的作用域为**定义该变量的位置到文件结束**。如果在全局变量的定义点之前的函数引用该变量，需要使用extern将全局变量的作用域扩展到此位置。同时该关键字还可以将全局变量的作用域扩展到其他文件。
- 11、存储类型**static**关键字的使用：对于局部变量来说，是指定局部变量的存储区域，对于全局变量来说是指定其作用区域。
- 11、变量的声明和定义：需要建立存储空间的声明成为定义，如 `int a`。不需要建立存储空间的声明成为声明，如 `extern a`。外部变量的定义只能有一次，在函数外。但是外部变量的声明可以有 multiple 次，在函数内。
- 12、数据结构+算法= 程序。
- 13、算法的特性：有穷性、确定性、有零个或多个输入输出、有效性。
- 14、函数就是功能，每一个函数都用来实现一个特定的功能，函数的名字反应其代表的功能。

15、函数中的形参在未出现函数调用的时候并不会占用存储单元，只有在发生函数调用的时候才会被临时分配存储单元。

16、一个递归问题分为回溯和递推两个阶段。

指针

1、C语言中的地址被形象的成为指针，其包括位置信息和其所指向的数据的类型信息。指针变量就是专门用来存储另一个变量地址的变量，指针变量的值就是地址（指针）

2、使用指针法访问数组会使数组的访问速度更快。两个指针可以相减，但是只有在指向同一数组的时候才有意义。引用一个数组元素可以有下标法 `a[i]` 和指针法 `*(a+i)` 两种方法。

3、在使用指针法访问数组 `a` 的元素的时候，可以使用 `p=a;p++`，但是不可以直接使用 `a++`；这是因为 `a` 代表数组第一个元素的位置，是一个指针型常量。

4、使用数组作为函数的参数的时候，形参数组名只是一个指针变量，用来存储传输过来的实参数组的首地址，通过函数调用可以改变实参数组内元素的值。而对于普通的变量来说，实现的是值传递，通过函数调用并不可以直接改变实参的值。

5、形参数组名是一个**指针变量**，实参数组名是一个**指针常量**。

6、字符数组由若干个元素组成，每一个元素存放一个字符。字符指针变量中存放的是地址，可以对字符指针变量进行复制，但是不可以对数组名进行赋值。编译时，字符数组分配若干存储单元，但是字符指针变量只分配一个存储单元。字符指针的值是可以改变的，但是字符指针变量的值只是一个固定的值，不能改变。

7、指针也可以指向一个函数，函数名/指针代表的是函数的入口地址也就是起始地址。

8、指针数组：若一个数组中元素均为指针类型则称其为指针数组。

9 `void *` 表示无指向的指针，只有转换为有指向的指针的时候才能存取数据

10、字符变量和指针的比较

- 字符数组中存放的是元素，而指针变量存储的是地址
- 字符数组在编译的时候会被分配若干个存储单元，但是指针变量只有一个存储单元
- 可以对指针变量进行赋值，但是不可以对数组名进行赋值
- 指针变量的值可以改变，但是数组名代表一个固定的值。

11、在编译的时候，如果有函数存在，会将函数的源代码转换成为可执行代码并分配存贮空间，这段空间的起始地址就叫做 **函数的入口地址**，函数名就代表程序的入口地址。

结构体

1、结构体中的变量可以和程序中的变量具有相同的名称。

2、共用体类型,为了解决不同数据类型共享同一段内存的问题。共用体的内存长度等于最长的元素的内存长度。无法直接引用共用体名字，但是可以引用共用体内的元素。

- **共用体内只能存放一个值**，比如存97入共用体中，对于int类型的元素来说就是数字97，但是对于char类型的数字来说就是字符，对于float类型的元素来说就是0。
- 共用体中保存的是最后被赋值的元素的值。
- 共用体变量的地址和他的各个元素的地址是同一地址。
- 不能对共用体变量进行赋值，也不可以通过共用体变量获得任何值。
-

3、枚举

- C语言对于枚举元素按照常量进行处理。
- 每一枚举元素默认是一个常量，C语言在编译时默认他们的值为0、1、2、3、4、5、6、、、

对文件的输入输出

1、什么是文件

- 程序文件：源程序文件 (.c) 、目标文件(.obj)、可执行文件(.exe)
- 数据文件

2、文件的分类

数据文件可以分为阿斯克码文件（文本文件）和二进制文件（映像文件）。数据在**内存**中的存储是以**二进制**形式进行存储的。数据在磁盘可以以**阿斯克码**的性质进行存储也可以以**二进制**的形式进行存储。

3、文件指针

每一个被使用的文件都有文件信息区用以存储文件的有关信息。文件指针就是指向这个信息区开头的指针，可以通过文件指针找到与他相关联的文件。