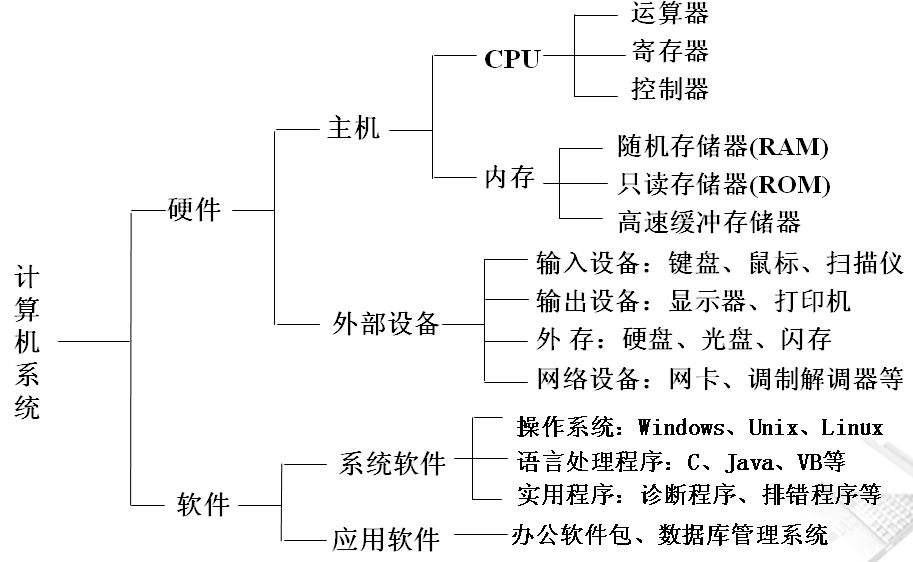
计算机导论与程序设计

一、计算机部件及原理

现代的计算机由多种部件构成。中央处理单元（CPU）承担绝大部分的运算工作。随机存取内存（RAM）是存储程序和文件的工作区；而永久内存存储设备（过去一般指机械硬盘，现在还包括固态硬盘）即使在关闭计算机后，也不会丢失之前存储的程序和文件。另外，还有各种外围设备（如，键盘、鼠标、触摸屏、监视器）提供人与计算机之间的交互。CPU负责处理程序。

CPU的工作非常简单。它从内存中获取并执行一条指令，然后再从内存中获取并执行下一条指令，诸如此类（一个吉赫兹的CPU一秒钟能重复这样的操作大约十亿次，因此，CPU能以惊人的速度从事枯燥的工作）。CPU有自己的小工作区——由若干个寄存器组成，每个寄存器都可以存储一个数字。一个寄存器存储下一条指令的内存地址，CPU使用该地址来获取和更新下一条指令。在获取指令后，CPU在另一个寄存器中存储该指令，并更新第1个寄存器存储下一条指令的地址。CPU能理解的指令有限（这些指令的集合叫作指令集）。而且，这些指令相当具体，其中的许多指令都是用于请求计算机把一个数字从一个位置移动到另一个位置。例如，从内存移动到寄存器。



二、计算机语言

人和人之间的交流需要通过语言，中国人之间用汉语，英国人用英语，俄罗斯用俄语，等等。人和计算机交流信息也要解决语言问题。需要创造一种计算机和人都能是别的语言，这就是计算机语言。计算机语言经历里以下几个发展阶段。

机器语言:计算机工作基于二进制，从根本上说，计算机只能识别和接受由0和1组成的指令。这种计算机能直接识别和接受的二进制代码称为机器指令（machine instruction）。机器指令的集合就是该计算机的机器语言（machine language）。在语言的规则中规定各种指令的表示形式以及它的作用。

显然，机器语言与人们习惯用的语言差别太大，难学，难写，难记，难检查，难修改，难以推广使用，因此初期只有极少数的计算机专业人员会编写计算机程序。

符号语言：为了克服机器语言的上述缺点，人们创造出符号语言（symbolic language)，它用一些英文字母和数字表示一个指令。

显然，计算机并不能直接识别和执行符号语言的指令，需要用一种称为汇编程序的软件把符号语言的指令转化成为机器指令。一般，一条符号语言的指令对应转化为一条机器指令。转化的过程称为“代真”或“汇编”。因此，符号语言又被称为符号汇编语言（symbolic assembler language）或汇编语言（assembler language）。

虽然汇编语言比机器语言简单好记一点，但仍然难以普及，只在专业人员中使用。

不同型号的计算机的机器语言和汇编语言是互不通用的。用甲机器的机器语言编写的程序在乙机器上不能使用。机器语言和汇编语言完全依赖于具体机器特性的，是面向机器的语言。由于它“贴近”计算机，或者说离计算机“很近”，故称为计算机低级语言（low level language）。

高级语言： 为了克服低级语言的缺点，20世纪50年代创造出了第一个计算机高级语言----FORTRAN语言。它很接近于人们习惯使用的自然语言和数学语言。程序中用到的语句和指令是用英文单词表示的，程序中所用的运算符和运算表达式和人们日常所用的数学式子差不多，很容易理解。程序运行的结果用英文和数字输出，十分方便。

这种语言功能很强，且不依赖于具体机器，用它写出的程序对任何型号的计算机都适用，它与具体机器的距离较“远”，故称为计算机高级语言（high level language）。

当然，计算机也是不能直接识别高级语言程序的，也要进行“翻译”。用一种称为编译程序的软件把高级语言写的程序（称为源程序（source program））转换为机器指令的程序（称为目标程序），然后让计算机执行机器指令程序，最后得到结果。高级语言的一个语句往往对应多条机器指令。

自从有了高级语言后，一般的科技人员，管理人员，大中学生乙己广大的计算机爱好者都能较容易的学会用高级语言编写程序，指挥计算机进行工作，而完全无须考虑什么机器指令，也可以不必深入的懂得计算机的内部结构和工作原理，就能得心应手的利用计算机进行各种工作，为计算机的推广普及创造了良好的条件，人们称高级语言的出现是计算机发展史上“惊人的成就”。

高级语言经历了不同的发展阶段：

1. 非结构话的语言
2. 结构话语言
3. 面向对象的语言





三、C语言的发展及其特点

（一）、C语言的发展

C语言诞生于美国的[贝尔](https://baike.baidu.com/item/%E8%B4%9D%E5%B0%94/1064241" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)实验室，由D.M.Ritchie以[B语言](https://baike.baidu.com/item/B%E8%AF%AD%E8%A8%80/1845842" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)为基础发展而来，在它的主体设计完成后，Thompson和Ritchie用它完全重写了UNIX，且随着UNIX的发展，c语言也得到了不断的完善。为了利于C语言的全面推广，许多专家学者和硬件厂商联合组成了C语言标准委员会，并在之后的1989年，诞生了第一个完备的C标准，简称“C89”，也就是“ANSI c”，截至2020年，最新的C语言标准为2017年发布的 “C17”。

（二）、C语言的特点

C语言的主要特点

1. 简洁的语言
2. 具有结构化的控制语句
3. （3）丰富的数据类型
4. 丰富的运算符
5. 可对[物理地址](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%9C%B0%E5%9D%80/2901583" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)进行直接操作
6. 代码具有较好的可移植性
7. 可生成高质量、目标代码执行效率高的程序

C语言是普适性最强的一种计算机程序编辑语言，它不仅可以发挥出高级编程语言的功用，还具有汇编语言的优点，因此相对于其它编程语言，它具有自己独特的特点。具体体现在以下三个方面：

其一，广泛性。C 语言的运算范围的大小直接决定了其优劣性。C 语言中包含了34种运算符，因此运算范围要超出许多其它语言，此外其运算结果的表达形式也十分丰富。此外，C 语言包含了字符型、[指针](https://baike.baidu.com/item/%E6%8C%87%E9%92%88/2878304" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)型等多种数据结构形式，因此，更为庞大的数据结构运算它也可以应付。

其二，简洁性。9 类控制语句和32个[KEYWORDS](https://baike.baidu.com/item/KEYWORDS/8284218" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)是C语言所具有的基础特性，使得其在计算机应用程序编写中具有广泛的适用性，不仅可以适用广大编程人员的操作，提高其工作效率，同 时还能够支持高级编程，避免了语言切换的繁琐。

其三，结构完善。C语言是一种结构化语言，它可以通过组建模块单位的形式实现[模块化](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97%E5%8C%96/3295536" \t "https://baike.baidu.com/item/c%E8%AF%AD%E8%A8%80/_blank)的应用程序，在系统描述方面具有显著优势，同时这一特性也使得它能够适应多种不同的编程要求，且执行效率高。

1. 简单的C程序设计

（一）、输出hello world！

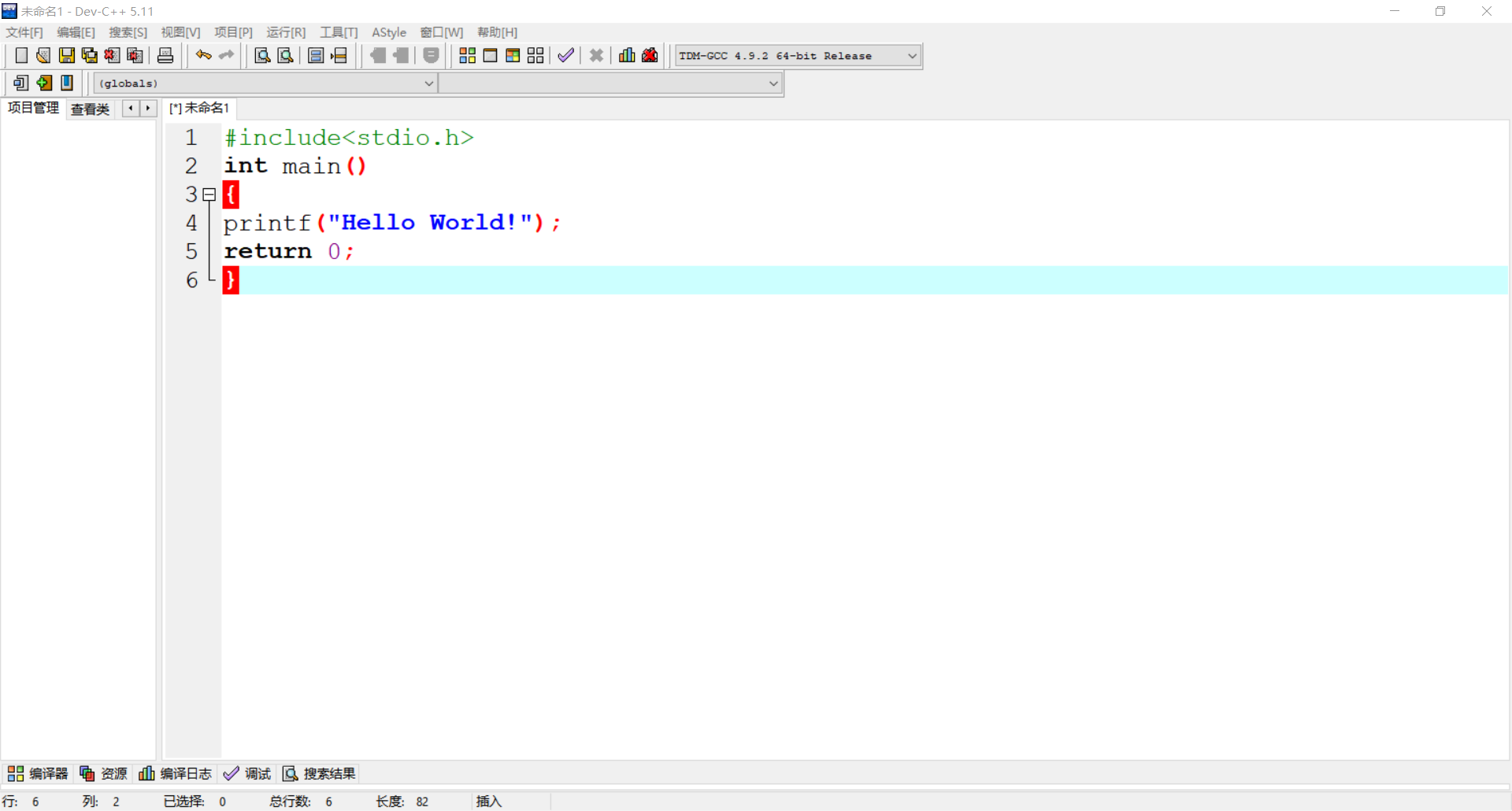
#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数

{ // 函数开始标志

printf(“Hello␣World!”); // 输出一行信息

return 0; // 函数执行完毕返回函数值0

} // 函数结束标志

（二）、求两个整数之和两个整数之和

#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数

{ // 函数开始标志

int a,b,sum; // 定义a,b,sum为整型变量

a=123;

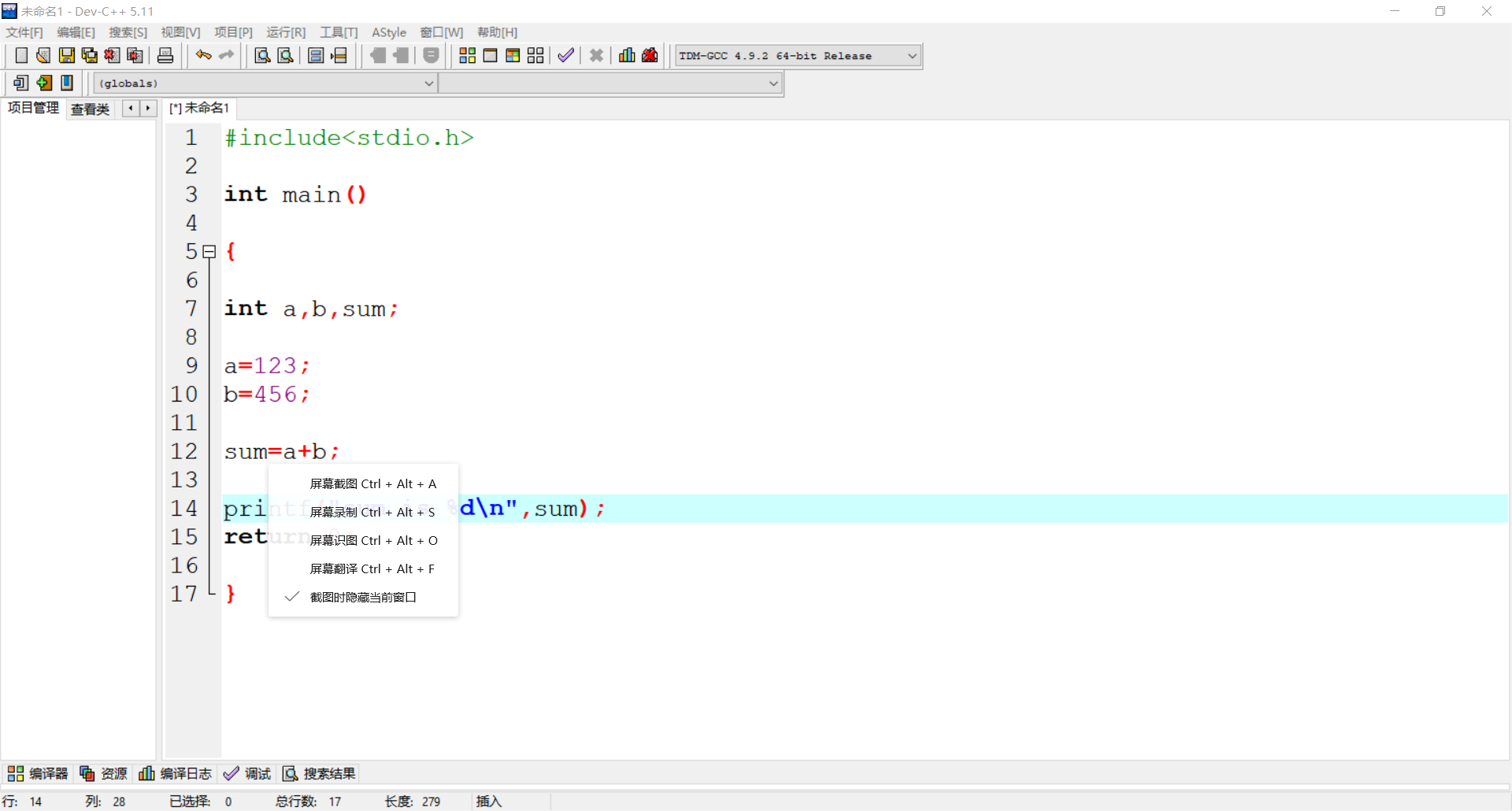
b=456; // 对a,b赋值

sum=a+b; // 计算a+b, 并把结果存放在变量sum中

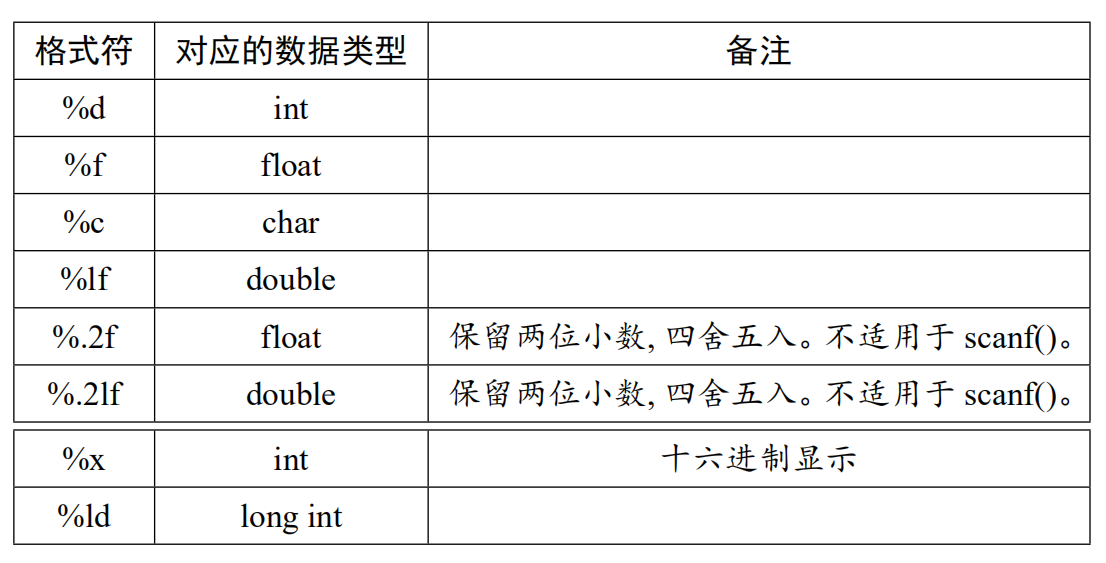
printf(“sum␣is␣%d\n”,sum); // 输出结果

return 0; // 函数执行完毕返回函数值0

} // 函数结束标志



注；



注： 输入语句 scanf(“% 变量格式符”, & 变量名);

输出语句 printf(“原样输出, % 格式符”, 对应变量值);

（三）、选择结构，if(条件表达式){表达式为真时执行语句; }

#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数

{ // 函数开始标志

int a=10; // 定义变量a为整型数值, 定义变量时，可以指定变量的初值

if(a>=10)

{

printf(“a>=10\n”); // \n为换行符

}

else

{

printf(“a<10\n”); // \n为换行符

}

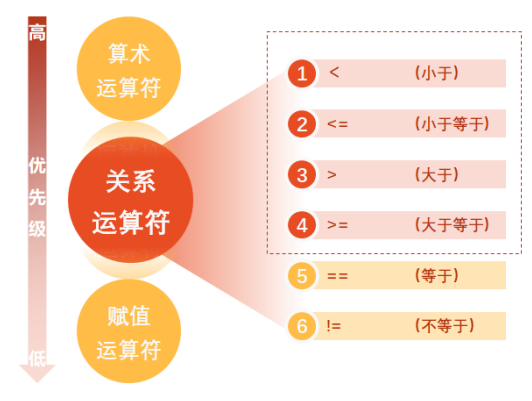
return 0; // 函数执行完毕返回函数值0

} // 函数结束标志

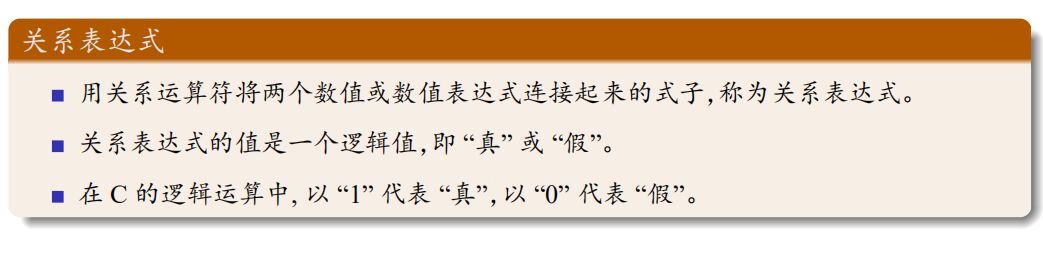
注：条件表达式: 关系表达式; 逻辑表达式; 数值表达式



注：关系运算符及其优先次序



注：关系表达式的值, 非 0 即真



注：逻辑运算符

int a=5,b=10,c=0; // 以int为例

if(!a) // 逻辑非(NOT), a是非0, 所以!a的值是0

{ ... }

if(a && b) // 逻辑与(AND), a,b均为非0, 所以(a && b)的值为1

{ ... }

if(a || c) // 逻辑或(OR), a,c之一是非0, 即为真

{ ... }

（四）、循环结构，while(条件表达式){表达式为真时执行的语句;}

#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数

{ // 函数开始标志

int a=10; // 定义变量a为整型数值, 定义变量时，可以指定变量的初值

while(a>=0)

{

printf(“a=%d\n”,a); // \n为换行符

a--; // a= a - 1

}

return 0; // 函数执行完毕返回函数值0

} // 函数结束标志

注： 算术运算符 +, — , \*, /, %, ++, −−

整数 = 整数/整数, 结果不会四舍五入。

余数 r=a%b, a,b 必须是整数。

++i, - -i: 先加 (减)1, 再使用。i++, i- -: 先使用, 再加 (减)1

注；数学库函数

