C语言有两种选择语句：（1）if语句，用来实现两个分支的选择结构；（2）switch语句，用来实现多分支的选择结构。

在系统对if语句编译时，若发现内嵌语句结束（出现分号），还要检查其后有无else，如果无else，就认为整个if语句结束，如果有else，则把else子句作为if语句的一部分。注意else子句不能作为一个完整语句单独使用，它必须是if语句的一部分，与if语句配对使用。

关系运算符和关系表达式

关系运算符及其优先次序

|  |  |  |
| --- | --- | --- |
| 关系运算符 | 关系 | 优先级 |
| < | 小于 | 优先级相同（高） |
| <= | 小于或等于 |
| > | 大于 |
| >= | 大于或等于 |
| == | 等于 | 优先级相同（低） |
| != | 不等于 |

!(非) （高）

算数运算符

关系运算符

&&和｜｜

赋值运算符 （低）

关系表达式：用关系运算符将两个数值或数值表达式连接起来的式子。

关系表达式的值是一个逻辑值，即“真”或“假”。例如，关系表达式“5==3”的值为“假”，“5>=0”的值为“真”。在c的逻辑运算中，以“非0”代表“真”，以“0”代表“假”，即将一个非0的数值认作为“真”，负数也是真。

逻辑运算符两侧的运算对象不但可以是0或1，或者是0和非0的整数，也可以是字符型、浮点型、枚举型或指针型的纯量型数据。

例：’c’&& ’d’的值为1（因为’c’和 ’d’的ASCII值都不为0，按“真”处理），所以1&&1的值为1。

在逻辑表达式的求解中，并不是所有的逻辑运算符都被执行，只是在必须执行下一个逻辑运算符才能求出表达式的解时，才执行该运算符。

例：

解析：当a=1，b=2，c=3，d=4，m和n的原值为1时，由于“a>b”的值为0，因此m=0，此时已能判断整个表达式不可能为真，不必再进行“n=c>d”的运算，因此n的值不是0而仍保持原值1。

逻辑型变量：这是c99所增加的一种数据类型，可以将关系运算和逻辑运算的结果存到一个逻辑型变量中，以便于分析和运算。

定义逻辑变量用类型符bool。

例：

#include <stdio.h>

int main()

{

float score;

scanf("%f",&score);

bool a,b; //a和b被定义为逻辑型变量

a=score>=60; //将关系表达式score>=60的值赋给逻辑变量a

b=score<=69; //将关系表达式score<=69的值赋给逻辑变量b

if(a&&b)

printf("the grade is c\n"); //如果a和b均为真，输出分数等级为c

else

printf("error\n");

}

条件运算符和条件表达式

条件运算符由两个符号（？和：）组成，必须一起使用。要求有三个操作对象，称为三目（元）运算符，它是c语言中唯一的一个三目运算符。

条件表达式的一般形式：表达式1？表达式2：表达式3

例：

相当于：

If(a>b)

Max=a;

Else

Max=b;

“表达式2“和“表达式3”不仅可以是数值表达式，还可以是赋值表达式或函数表达式。

例：

赋值表达式 a>b?(max=a):(max=b);

函数表达式 a>b?printf("%d",a):printf("%d",b);

If 与else的配对关系：else总是与它上面的最近的未配对的if配对。

Switch语句实现多分支选择结构，作用：根据表达式的值，使流程跳转到不同的语句。

（1）switch后面括号内的“表达式”，其值的类型应该是整数类型（包括字符型）。

（2）可以没有default标号，此时如果没有与switch表达式相匹配的case常量，则不执行任何语句，流程跳转到switch语句的下一个语句。

（3）各个case标号出现次序不影响执行结果。例：可先出现default标号，再出现case常量。

（4）case常量必须互不相同，否则会矛盾。

（5）多个case标号可以共用一组执行语句。

例： case‘a’:

case‘b’:

case‘c’:printf(“>60\n”);break;