**实例1 三个数由小到大排序**

**【实例说明】**

**输人任意3个整数，编程实现对这3个整数进行由小到大排序井将排序后的结果显示在屏幕上。**

**【技术要点】**

**(1)本实例中用到了if语句。if语句的3种形式如下**

* **if（表达式）语句**

**其语义是：如果表达式的值为真，则执行其后的语句，否则不执行该语句。**

**• if（表达式）**

**语句1**

**else**

**语句2**

* **if（表达式1）**

**语句1**

**else if（表达式2）**

**语句2**

**else if（表达式3）**

**语句3**

**……**

**else if（表达式m）**

**语句m**

**else 语句n**

**其语义是：依次判断表达式的值，当出现某个值为真时，则执行其对应的语句。然后跳到整个if语句之外继续执行程序。如果所有的表达式均为假，则执行语句n,然后继续执行后续 程序。**

**(2) 3种形式的if语句中在if后面都有“表达式”，一般为逻辑表达式或关系表达式。在执行if语句时先对表达式求解，若表达式的值为0,按“假”处理，若表达式的值为非"0"'按 “真”处理，执行指定的语句。**

**(3)else子句不能作为语句单独使用，它必须是if语句的一部分，与if配对使用。**

**(4)if与else后面可以包含一个或多个内嵌的操作语句，当为多个操作语句时要用"{ }''**

**将几个语句括起来成为一个复合语句。**

**(5)if语句可以嵌套使用，即在if语句中又包含一个或多个if语句，在使用时应注意else**

**总是与它上面最近的未配对的if配对。**

**注意：本实例中使用scanf("%d%d%d",&a,&b,&c)从键盘中荻得3个任意数,在输入数据时在两个数据之间以一个或多个空格间隔，也可以用按回车键、Tab键，一定不能用逗号作为 两个数据间的分隔符，当以scanf("%d,%d,%d" ,&a,&b,&c); 格式输入数据时，两个数据之间要用","做间隔。**

**实例2 a²+b²**

**【实例说明】**

**编程要求输入整数a和 b, 若a²+b²的结果大与100, 则输出a²+b²的值，否则输出 a+b 的结果。**

**【技术要点】**

**本实例的基本思路是首先求出输入的两个数的平方和，再对该平方和进行判断， 如果平方和大于100, 则将平方和输出，否则输出 a+b 的值。**

**实例3 整数倍**

**【实例说明】**

**编程判断输入的数是否既是5又是7的整倍数，如果是输出yes, 否则输出no。**

**【技术要点】**

**实例的算法思想是对输入的数x用5和7分别整除，看是否能同时被5和7 整除，如果能，则输出yes, 否则输出no。**

**实例4 判断闰年**

**【实例说明】**

**从键盘上输入一个表示年份的整数，判断该年份是否是闰年，判断后的结果显示在屏幕上。**

**【技术要点】**

1. **计算闰年的方法用自然语言描述如下:如果某年能被4整除但不能被 100整除，或者该年能被400整除则该年为闰年。在本实例中我们用如下表达式来表示上面这句话：**

**year%4= =0&&year%100!=0)llyear%400==0。**

**除本实例外判断闰年还有许多方法，下面给出的算法（伪代码描述）也为其中一种：**

**{**

**if (某年能被 400 整除）**

**输出是闰年；**

**else if (该年能被 100 整除）**

**输出不是闰年；**

**else if (该年能被 4 整除）**

**输出是闰年；**

**else**

**输出不是闰年；**

**}**

**这种算法略显繁琐，读者可以根据实际情况选择适当的方法。**

1. **将判断闰年的自然语言转换成C语言要求的语法形式时锯要用到逻辑运算符&&、|| 、！。具体使用规则如下：**

**• &&逻辑与（相当于其他语言中的AND), a&&b 若a, b为真，则 a&&b 为真；**

**• II逻辑或（相当于其他语言中的 OR), allb 若a, b之一为真, 则 allb为真;**

**• ! 逻辑非（相当于其他语言中的NOT), a若为真, 则! a为假；**

**• 三者的优先次序是:！—>＆＆—>||, 即"!"为三者中最高的。**

**注意：程序编写过程中要注意"=="和"="使用的区别，“＝＝” 为关系运算符，方向＂自左至右＂, 而"="是赋值运算符，方向＂自右至左＂。**

**实例5 评定成绩**

**【实例说明】**

**编程对输入的百分制分数给出相应的等级，分数大于等于90为"A",80-89为"B",70-79 为"C",60-69为"D",60分以下为"E"。**

**【技术要点】**

**本实例的思路如下：输入成绩，当输入的成绩是100时，我们可以把它看成90分，因为100分和90分都是A等的，对输入的成绩使用除法便求出其高位数字，因为通过高位数字就可以判断出输出的成绩在哪个等级；使用 switch 语句对取出的高位数字进行选择判断，如果高位数为9则是A等，如果是8则是B等，依此类推，分别出现C、D、E等。**

**实例6 整数加减法练习**

**【实例说明】**

**练习者自己选择是进行加法还是减法运算，之后输入进行多少以内的加法或减法运算，具体数值会由计算机随机产生，输入答案，计算机会根据输人的数据判断结果是否正确。**

**【技术要点】**

**(1) 程序中用到rand()作用是产生一个随机数并返回这个数，a=rand()%max; 的具体含义就是产生max以内的任意随机数（不含max本身）。**

**(2)实例中用到以下语句signl=(sign==1?'-':'+');**

**其中(sign==1?'-':'+') 是一个条件表达式。它是这样执行的：如果（sign==1）条件为真，则条件表达式取值"-"'否则取值"＋"。**

**条件表达式的一般形式为:表达式1?表达式2:表达式3**

**• 条件运算符优先于赋值运算符，但比关系运算符和算术运算符低。**

**• 条件运算符的结合方向为“自右至左”。**

**• 条件表达式中，表达式1的类型可以与表达式2和表达式3的类型不同。**

**实例7 模拟ATM机界面程序**

**【实例说明】**

**模拟银行ATM机操作界面，主要实现取款功能，在取款操作前用户要先输入密码，密码正确才可进行取款操作，取款时将显示取款金额及剩余金额。操作完毕退出程序。**

**【技术要点】**

**本实例中主要用到switch语句，以下是对其使用方法的讲解： switch语句是多分支选择语句。它的一般形式如下：**

**switch(表达式）**

**{**

**case常量表达式1: 语句1；**

**case常量表达式2: 语句2；**

**……**

**case常量表达式n:语句n；**

**default: 语句n+1;**

**}**

**其语义是：计算表达式的值，并逐个与其后的常量表达式值相比较，当表达式的值与某个常量表达式的值相等时，即执行其后的语句，然后不再进行判断，继续执行case后的语句。如表达式的值与所有case后的常量表达式均不相同时，则执行 default 后的语句。**

**说明：**

**•每一个case 的常量表达式的值必须互不相同，否则就会出现相互矛盾的现象；**

**•各个case和default的出现次序不影响执行结果；**

**•在执行一个case 分支后，如果想使流程跳出switch 结构，即终止 switch语句的执行，可以在相应的语句后加 break来实现。最后一个default可以不加 break 语句。**

**实例8 用#打印三角形**

**【实例说明】**

**用＃打印如下所示的三角形，**

**#**

**###**

**#####**

**#######**

**#########**

**【技术要点】**

**本实例中多次用到 for 循环， 以下是对 for循环的详细讲解。**

**for 语句的一般形式为：**

**for (表达式1 ;表达式2;表达式 3) 语句**

**它的执行过程如下。**

**(1) 先求解表达式1。**

**(2) 求解表达式2, 若其值为非0, 则执行 for语句中指定的内嵌语句，然后执行下面第3步。若表达式2值为0, 则结束循环，转到下面第5步。**

**(3) 求解表达式 3。**

**(4) 返回第2步继续执行。**

**(5)循环结束，执行 for 语句下面的一个语句。**

**说明：**

**•表达式1通常用来给循环变量赋初值，一般是赋值表达式。也允许在 for语句外给循环变量赋初值，此时可以省略该表达式。**

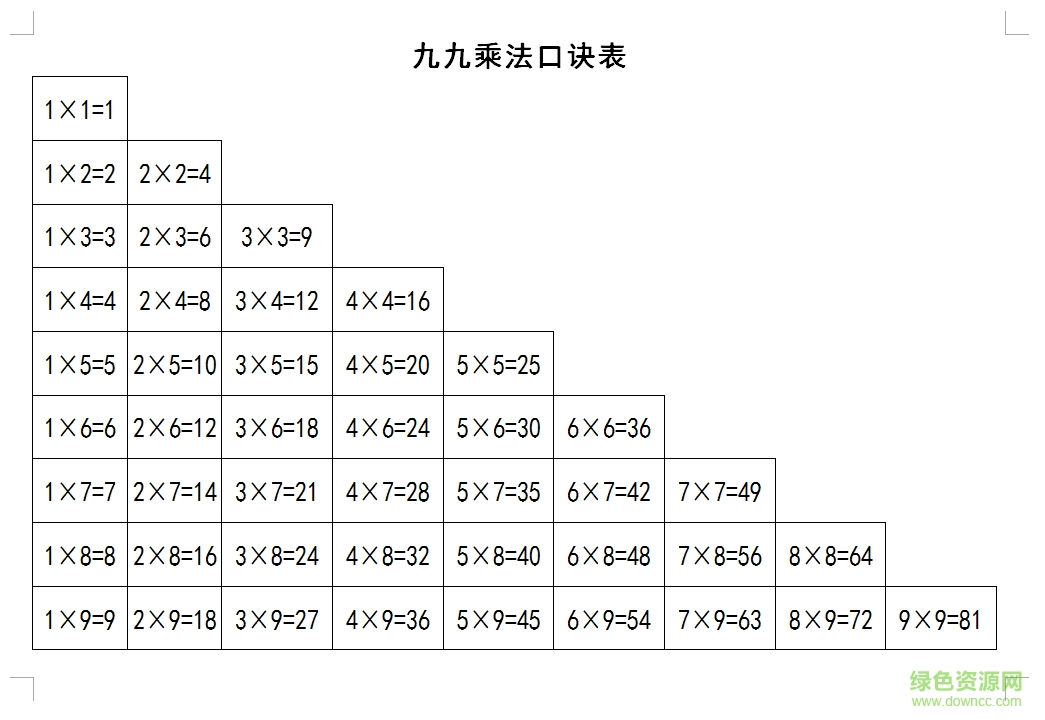
**•表达式2通常是循环条件，一般为关系表达式或逻辑表达式。如果表达式 2 省略，既不判断循环条件，也就是认为表达式2始终为真，则循环将无终止地进行下去。**

**•表达式3通常可用来修改循环变量的值，一般是赋值语句。表达式3也可以省略，但此时程序设计者应另外设法保证循环能正常结束。**

**实例9 打印乘法口诀表**

**【实例说明】**

**打印乘法口诀表。**

****

**【技术要点】**

**如何打印乘法口诀表关键是要分析程序的算法思想，本实例中两次用for循环， 第一次for循环即将它看成乘法口诀表的行数，同时也是每行进行乘法运算的第一个因子第二个 for循环范围的确定建立在第一个for循环的基础上，即第二个 for 循环的最大取值是第一个for循环中变量的值。**

**实例10 序列求和**

**【实例说明】**

**用while循环做一个简单的计算s=1 + 1/2+ 1/3+ ... + 1/n。**

**【技术要点】**

**本实例中用到while循环，这也是实现累加求和的关键，以下是对其使用方法的讲解。**

**while语句用来实现“当型”循环结构。其一般形式如下：**

**while (表达式）语句**

**其语义是：当表达式为非0值时，执行while语句中的内嵌语句。**

**其特点是：先判断表达式，后执行语句。**

**说明：**

**•while语句中的表达式一般是关系表达或逻辑表达式，只要表达式的值为真（非0）即可继续循环。**

**•循环体如果包含一个以上的语句，应该用花括弧括起来，以复合语句形式出现。如果不加花括弧则while语句的范围知道while后面第一个分号处。**

**•在循环体中应有使循环趋向于结束的语句以避免死循环。**

**提到while语句就不能不提到do-while,首先看下do-while语句的一般形式：**

**do**

**循环体语句**

**while (表达式）；**

**其语义是：先执行一次指定的循环体语句，然后判别表达式，当表达式的值为非零（真）时，返回重新执行循环体语句，如此反复，直到表达式的值等于0为止，此时的循环结束。**

**其特点是：先执行循环体， 然后判断循环条件是否成立。**

**示例11 用while语句求n!**

**【实例说明】**

**3! =3\*2\*1, 5! =5\*4\*3\*2\*1, 依此类推n!=n\*(n-1)\*…\*2\*1, 使用while语句求n!。**

**【技术要点】**

**(1)在写程序之前首先要理清求n!的思路。求一个数n的阶乘也就是用n\*(n-l)\*(n-2)\*···\*2\*1, 那么反过来从1一直乘到n依然成立。当n为0和1 的时候我们单独考虑，此时它们的阶乘均为1。**

**(2)求得的阶乘的最终结果这里我们要给它定义为单精度或双精度型，如果定义为整型就很容易出现溢出现象。**

**示例12 逆序存放数据**

**【实例说明】**

**任意输入5个数据，编程实现将这5个数据逆序存放，并将最终结果显示在屏幕上。**

**【技术要点】**

**本实例没有太多难点，但是通过本实例的练习能让读者明白在编程过程中如何运用一维数组。这里介绍一下一维数组的一些相关知识。**

**•一维数组的定义方式**

**类型说明符 数组名[常量表达式］**

**说明：常量表达式表示元素的个数，即数组长度。例如a[5],5表示数组中有5个元素，下标从0开始，到4结束。a[5]不能使用，数组下标越界。**

**•一维数组的初始化**

**（1）在定义数组的时可直接对数组元素赋初值。**

**(2) 可以只给一部分元素赋值，未赋值的部分元素值为0。例如： int a[5]={1,2}; 表示给前两个元素赋了初值，后面a[2] ,a[3] ,a[4]值均为0。**

**(3) 在对全部数组元素赋初值时，可以不指定数组长度。**

**示例13 相邻元素之和**

**【实例说明】**

**从键盘中任意输入10个整型数据存到数组a中，编程求出a中相邻两元素之和，并将这些和存在数组b中，按每行3个元素的形式输出。**

**【技术要点】**

**本实例的算法思想如下：输出10个元素存到数组a中，利用for循环将数组a中相邻的元素求和存到数组b中，相邻元素的表示形式为a[i-1]及a[i]。**

**示例14 统计各种字符操作**

**【实例说明】**

**输入一组字符，要求分别统计出其中英文字母、数字、空格以及其他字符的个数。**

**【技术要点】**

**（1）实例中定义了一个字符变量c，以下是关于字符变量的一些使用要点。**

**•字符变量用来存放字符常量，一个字符变量只能放一个字符，不能用来存放一个字符串。**

**•字符数据以ASCII码存储，实际上是以二进制形式存放的，他的存储形式与整数的存储形式类似，所以一个字符型数据既可以以字符形式输出，也可以以整数形式输出。**

**（2）使用getchar()函数给字符变量c赋值从而方便对输入字符进行判断。getchar()函数使用要点如下。**

**••一般形式为：getchar()。**

**作用是从终端输入一个字符，函数的值就是从输入设备得到的字符。**

**• getchar()函数只能接受一个字符。**

**与其对应的是 putchar()函数，它的作用是向终端输出个字符。**

**(3) while((c=getchar())!='\n') 这个while的表达式里用到了 "\n"'。**

**实例 15 字符串替换**

**【实例说明】**

**编程实现将字符串"today is Monday"替换变成"today is Friday"。**

**【技术要点】**

**本实例的算法思想如下：首先输人字符串1,再输入要替换的内容和替换的位置（字符串1中的位置），这时只需从替换位置开始将要替换的内容逐个拷到字符串1中，直到遇到字符串1的结束符或遇到替换字符串的结束符便结束替换。**

**实例16 不用strcat 连接两个字符串**

**【实例说明】**

**连接两个字符串，要求不使用strcat函数。**

**【技术要点】**

**(1)本实例的关键技术要点是在将后一个字符串连接到前一个字符串的时候要先判断前一个字符串的结束标志在什么位置，只有找到了前一个字符串的结束标志才能连接后一个字符串。**

**(2)程序中用到了字符数组的输入输出，这里详细介绍一下：**

**•字符数组可以逐个字符输人输出，用格式字符"%c"输入或输出一个字符；**

**•字符数组也可将整个字符串一次输入或输出。用"%s"格式符。**

**实例17 求字符串中字符的个数**

**【实例说明】**

**输入一个字符串，要求使用函数调用的方法计算出改字共有多少个字符。**

**【技术要点】**

**（1）本实例中自定义了一个函数length来实现统计字符串中字符个数的功能，那么该如何来定义一个函数呢，以下是关于函数定义的一些知识要点。**

**•无参函数的定义形式类型标识符函数名（）**

**{**

**声明部分**

**语句**

**}**

**•有参函数定义的一般形式**

**类型标识符 函数名（形式参数表列）**

**{**

**声明部分**

**语句**

**}**

**本实例中定义的这个函数 length 就是有参函数。**

**•空函数**

**类型说明符 函数名（）**

**{ }**

**（2）讲到函数定义的一般形式时提到了有参函数， 那么参数到底怎么使用可能还有部分读者不是很了解，这里简单介绍一下。参数分为两种，一种叫做形式参数， 另一种叫做实际参数。**

**• 在定义函数时函数名后面括弧中的变晁名称为 ”形式参数”。本实例中 char \*p 就是在定 义形式参数。**

**• 在主调函数中调用一个函数时，函数名后面括弧中的参数（可以是一个表达式）称为 “实际参数”。本实例主函数中调用 length 时括号中的 str 就是实际参数。**

**(3) 函数的返回值问题：**

**• 函数的返回值是通过函数中的 return语句获得的。本实例使用 return 语句将被调用函数 length中n的值带回到主调函数中。**

**• 在定义函数时要指定函数值的类型。**

**• 函数类型决定了返回值的类型。**

**• 为了明确表示“不带回值”,可以用"void"定义“无类型”。**

**实例18 求学生的平均身高**

**【实例说明】**

**输入学生数并逐个输入学生的身高， 输出身高的平均值。**

**【技术要点】**

**本实例主要采用了数组名作为函数参数，有以下几点需要说明。**

**(1)用数组名作参数，应该在主调函数和被调用函数中分别定义数组，像本例中主调函数定义的数组为 height, 被调用函数定义的数组为array。**

**(2)实参数组与形参数组类型应一致，本实例中都为 float。**

**(3)形参数组也可以不指定大小，在定义数组时在数组名后面跟一个空的方括弧。 本实例就没有指定形参数组的大小。**

**(4) 用数组名作函数实参时，不是把数组元素的值传递给形参，而是把实参数组的起始地址传递给形参数组。**

**实例19 猴子吃桃**

**【实例说明】**

**猴子吃桃问题：猴子第一天摘下若干个桃子， 当即吃了一半， 还不过瘾，又多吃了一个，第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。编写程序求第一天共摘了多少。**

**【技术要点】**

**本实例和前两个实例解题的思路基本上是一样的。就是先找出变量间的关系，找出了它们之间的关系后程序就基本上没有什么问题了。本题中读者要明确第一天桃子数和第二天桃子数之间的关系即第二天桃子数加1的2倍等于第一天的桃子数。**

**实例20 普通的位运算**

**【实例说明】**

**当a=2、b=4、c=6、d=8 时编程求 a&c、b|d、a^d、~a的值。**

**【技术要点】**

**本实例中涉及几个位运算符， 下面具体介绍。**

**• 按位与运算符（＆）**

**当两个相应的二进位都为1, 则该位与运算的结果为1, 否则为0。**

**• 按位或运算符(|)**

**两个相应的二进位中只要有一个为 I, 该位或运算结果值为 1, 当都为0 时，该位或运算的结果值才为0。**

**• 异或运算符(^)**

**当参加运算的两个二进位同号，则结果为0, 否则为1。**

**• 取反运算符（～）**

**～是一个单目运算符，作用是对一个二进制数按位取反，即0取反是1,1取反是0。**

**实例21 使用指针实现数据交换**

**【实例说明】**

**本实例实现使用指针变量实现交换两个变量(a和b) 的值。运行后，输入两个整型数值，将变量a, b中的值交换，然后输出到窗体上。**

**【技术要点】**

**本实例利用指针变量实现数据的交换。变量的指针就是变量的地址，存放地址的变量就是指针变量，用来指向另一个变量。在程序中使用一个"\*"表示“指向”，定义指针变量的一般形式为：**

**基类型 \*指针变量名**

**例如： int \*p；**

**char \*s；**

**float \*lp；**

**因为指针变量是指向一个变量的地址，所以将一个变量的地址值赋给这个指针变量后，这个指针变量就“指向”了该变量。例如，将变量i的地址存放到指针变量p中，p就指向i。**

**注意： 定义指针变量时必须指定基类型，因为要根据指定的类型决定分配的空间。例如，定义指针类型为整型，当指针移动一个位置时，其地址值加2；如果指针指向一个实型变量，则增加值为4。**

**实例22 指向结构体变量的指针**

**【实例说明】**

**本实例通过结构体指针变量实现在窗体上显示学生信息。运行程序后，将学生信息输出在窗体上。**

**【技术要点】**

**一个结构体变量的指针就是该变量所占据的内存段的起始地址。用一个指针变量指向结构体变量，此时该指针变量的值是结构体变量的起始地址。**

**实例23 使用指针输出数组元素**

**【实例说明】**

**本实例通过指针变址输出数组的各元素值，运行程序后，输人10个数值，运行后，可以看到输出的数组元素值。**

**【技术要点】**

**本实例应用指向数组的指针实现输出数组元素。定义一个指向数组元素的指针变量的方法与定义指向变盐的指针变益相同。例如：**

**int a[10]; /\*定义一个包含十个元素的数组\*/**

**int \*p; /\*定义整型指针变量\*/**

**p=&a[0]; /\*对指针变量赋值\*/**

**上面第3句代码把数组a的a[0]元素的地址赋给指针变量p。也就是说，p指向a数组的首元素。**

**实例2466 使用指针查找数列中最大值、最小值**

**【实例说明】**

**本实例实现在窗体上输入10个整型数，自动查找这些数中的最大值和最小值，并显示在窗体上。**

**【技术要点】**

**本实例使用指向一维数组的指针，遍历一维数组中数据，从而实现查找数组中的最大值最小值。**

**在本实例中，自定义函数max\_min()用于将求得的最大值和最小值分别存放在这两个变量max和min中。变量max和min是在main()函数中定义的局部变量，将这两个变量的地址作为函数参数传递给被调用函数max\_min(),函数执行后将数组中最大值和最小值存储在max和 min中并返回。这是数值的传递过程。**

**下面介绍如何实现查找数组中最大值和最小值。在自定义函数max\_min()中，定义了指针变量p指向数组，其初值为a+1,也就是使p指向a(1)。循环执行p++,使p指向下一个元素。每次循环都将\*p和\*max与\*min比较，将大值存放在max所指地址中，将小值存放在min所指地址中。**

**实例25 实现输入月份号输出该月份英文名**

**【实例说明】**

**本实例使用指针数组创建一个含有月份英文名的字符串数组，并使用指向指针的指针指向这个字符串数组，实现输出数组中的制定字符串。运行程序后，输入要显示英文名的月份号，将输出该月份对应的英文名。**

**【技术要点】**

**与上个实例一样，本实例使用指针的指针实现对字符串数组中字符串的输出。这里首先定义了一个包含月份英文名的字符串数组，并定义了一个指向指针的指针变量指向该数组。使用该变量输出字符串数组的字符串。**