/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 第三章知识点总结。以后学习中模仿此例，归纳总结知识点。

\* 编写这样的程序是学习计算机编程的"秘籍"。

\* 编程素材来源于课件以及每次上机的作业。

\* 不断积累，总结出一套自己的"宝典"。

\* 期末复习时，信手拈来，事半功倍。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h> // 包含头文件，说明本程序用到的输入输出库函数scanf(),printf()

#include <math.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

【例3.2】有人用温度计测量出用华氏法表示的温度(如64°F），

今要求把它转换为以摄氏法表示的温度(如17.8℃)。

f: 代表华氏温度，c代表摄氏温度， c = (5/9)(f-32)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main()

{

float f,c; //定义f和c为单精度浮点型变量

f=64.0; //指定f的值

// 注意：在C语言中，5/9和5.0/9的计算结果不同。整数/整数，其结果是整数。

c=(5.0/9)\*(f-32); //利用公式计算c的值

printf("f=%f\nc=%f\n",f,c); //输出c的值

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

【例3.2】计算存款利息。有1000元，想存一年。有3种方法可选:

(1)活期，年利率为r1；

(2)一年期定期，年利率为r2；

(3)存两次半年定期，年利率为r3。

请分别计算出一年后按3种方法所得到的本息和。

解题思路: 关键是确定计算本息和的公式。从数学知识可知，若存款额为p0，则:

活期存款一年后本息和为：p1=p0(1+r1)

一年期定期存款，一年后本息和为：p2=p0(1+r2)

两次半年定期存款，一年后本息和为：p3=p0(1+r3/2)(1+r3/2)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main2()

{

//定义变量,必要时可以给变量赋值, p0存款额，r1,r2,r3代表三种存法的利率，p1,p2,p3对应其本息和

float p0=1000, r1=0.0036, r2=0.0225, r3=0.0198, p1, p2, p3;

p1=p0\*(1+r1); //计算活期本息和

p2=p0\*(1+r2); //计算一年定期本息和

p3=p0\*(1+r3/2)\*(1+r3/2); //计算存两次半年定期的本息和

printf("p1=%f\np2=%f\np3=%f\n",p1, p2, p3); //输出结果

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

常量：

整型常量 1000，0，-5

实型常量，小数形式123.456，指数形式-34.8E-23

字符型常量 'a','#' // 引文单引号

字符串常量 "123", "boy" // 引文双引号

符号常量 #define PI 3.1416 // 注意行末没有分号

const float pi=3.1416

转义字符 \特殊字符， 例如, "hello \n"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main3()

{

int a = 1;

float f = 123.456;

double d = -34.8E-23;

char c = 'a';

// \n代表换行+回车(windows)

printf("Hello\n");

// n作为普通字符处理

printf("Hellon");

/\*\*

(a) '\'是转义字符，输出单引号

(b) 字符唯一对应一个整数（该字符对应的ASCII码）

(c) printf()函数，引号内的字符串"原样输出"，

格式描述符对应替换为后面的变量列表进行输出

printf("\'%c\'的ASCII码是%d\n",c,c);

==> %c对应第一个c, 以字符显示方式输出；

==> %d对应第二个c, 以

\*\*/

printf("\'%c\'的ASCII码是%d\n",c,c); // 'a'的ASCII码是97

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

标识符就是一个对象的名字。用于标识变量、符号常量、函数、数组、类型等

标识符只能由字母、数字和下划线3种字符组成，且第1个字符必须为字母或下划线

变量代表一个有名字的、具有特定属性的一个存储单元。

变量用来存放数据，也就是存放变量的值。

在程序运行期间，变量的值是可以改变的。

变量必须先定义，后使用。

变量名中区分大小写字母

不能使用关键字作为变量名

变量的名字应该尽量反映变量在程序中的作用与含义

常变量，与#define，一般在程序代码文件<include>后定义，供整个文件使用。

#define PI 3.1416 //定义符号常量,

const float pi=3.1416; //定义常变量

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define PI 3.141592 // 注意行末没有分号

int main4()

{

int a = 3; // 变量名，变量值(初始值)，存储单元

const float pi = 3.1416;

float s1,s2;

s1 = pi\*a\*a; // pi占用一个float存储空间

s2 = PI\*a\*a; // 预编译为3.1416\*a\*a

printf("%f,%f",s1,s2);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

数据类型，

就是对数据分配存储单元的安排，包括存储单元的长度(占多少字节)以及数据的存储形式。

不同的类型分配不同的长度和存储形式。

说明: C标准没有具体规定各种类型数据所占用存储单元的长度，只要求

sizeof(short) ≤sizeof(int) ≤sizeof(long) ≤sizeof(long long)，具体由各编译系统自行决定的。

sizeof是测量类型或变量占用存储单元的长度的运算符。

char,int,float,double, short, long

sigend,unsigned // 有符号和无符号定义

指数形式：d表示有效位数字(6~7位)

d.ddddE+n 小数点向右移动n位,d.dddd\*10^n

d.ddddE-n 小数点向左移动n位,d.dddd\*10^(-n)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main5()

{

char c = 'a'; // 定义变量的同时可以赋初始值

char c1 = '1', c2 = 'b'; // 相同类型变量可以在一条语句中定义

char c3 = '\n', c4 = '\0', c5 = '\a'; // 特殊字符

signed int i1 = -10; // 等效 int i1 = -10;

short int i2;

long int i3 = 1000000L; // 严格的编译器要求带后缀L(或l)，表示长整型常量。否则编译器会有警告信息（warning）

unsigned int u = 0xFF; // 0XFF

float f = 3.14F; // 严格的编译器要求带后缀F(或f)，表示单精度常量。否则编译器会有警告信息（warning）

double d = 3.12E5; // E前必须有数字，E后必须是（+，-）整数（或0），+号可以省略。

printf("%c,%d,%d,%u,%f,%lf\n",c5,c5,i1,u,f,d); // '\a'的ASCII码是BEL(整数 7)

printf("%u,%X,%x\n",u,u,u);

printf("%.1f,%5.1f",f,f); // 表示保留一位小数

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

运算符和表达式（对照课件，写简单程序验证）

注意：整数相除的结果为整数。

关系表达式(如，a > b)的值："真"为1; "假"为0

逻辑表达式(如，if (a) {} ), a非0为"真"

逻辑运算符'=='不是'=' ， if (a == b) // 一定不能写成a = b, 如果是这样，条件成立否，取决于a的值(=b),a非零，条件成立

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main6()

{

int a = 1,b =2;

int c = 'a';

printf("%d,%d,%d\n",a/b,a<=b,a==b);

if (a == b) // 一定不能写成a = b, 如果是这样，条件成立否，取决于a的值(=b),a非零，条件成立

{

printf("a不等于b\n");

}

else

{

printf("a等于b\n");

}

printf("%d\n",a+b-c); // -94

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

优先级与结合性，

左结合性：自左至右的结合方向，一般二元运算为左结合，如 x+y-3 ==> ((x+y)-3)

右结合性：自右至左的结合方向，赋值运算为右结合，如，a=b=c; ==> (a=(b=c))

关系运算符优先级低于算术运算符优先级

复合赋值运算符(+=,\*=,/=,%=)优先级低于算术运算符，如 x += y+2 ==> x = x + (y+2), x %= y ==> x = x % y, x/y的余数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main7()

{

int a = 1;

char c = 'a';

// 表达式a + 2! = c - 100的值相当于(a+2) != (c-100)

// ==> 3 != (97-100)

// ==> 3 != -3

// ==> 1

printf("%d\n",a+2!=c-100); // 1

int x,y;

x = 10; y =20;

x += y; // ==> x = x+y

x += y+2; // ==> x = x+(y+2);

printf("%d,%d\n",x,y);

}

// 注意数学语义与C语言表达式的不同

int main8()

{

char c = 'a';

// 数学表达式'A'<=c<='Z' 编译器将从左到右结合 ==> ('A'<=c)<='Z',结果与数学意义不同。

// 与数学含义相同的关系表达式是'A'<=c && c<='Z' 或 c>='A' && c<='Z'

printf("%d\n",'A'<=c<='Z'); // ('A'<='a')<='Z' ==> 1<='Z' ==> 1

printf("%d\n",'A'<=c && c<='Z'); // 1 && 0 ==>0

// 典型用法

if ('A'<=c && c<='Z') printf("%c是大写字符\n",c);

else printf("%c是小写字符\n",c);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

整数变量的自增，自减运算

建议谨慎使用++和--运算符，只用最简单的形式，即i++，i--，且把它们作为单独的表达式。

复杂表达式中使用++(--)，不同的编译器可能有不同的理解。例如：a = num/2+5\*(1+num++)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main9()

{

int x=10,y;

// 先使用，再自增

y = (x++) + 1;

printf("%d,%d\n",x,y); // 11,11

// 先自增，再使用

y = ++x + 10;

printf("%d,%d\n",x,y); // 12,22

// 简单形式，同一条语句中不要出现变量自身以及它的自增、自减运算，如，a = num/2+5\*(1+num++)

int i = 3;

i++; // ++i

printf("%d",i); // 4

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

不同类型数据间的混合运算

不同类型的两个操作数，“向上”转换成同一类型,再计算。

转换的结果一定是三种数据类型：

int，long，double

字符在表达式中的值是其对应的ASCII码(整数)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main10()

{

int a = 10;

float f = 10;

double d = 100.3F;

char c = 'a';

// 表达式右端计算结果是double类型的数据，赋值语句自动转换为float,精度的损失是否接受，是程序员的责任。

float result = a - c + f + d; // a-c ==> int, a-c+f ==> double, a-c+f+d ==> double

printf("%lf\n",result);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

字符在表达式中的值是其对应的ASCII码(整数)

【例3.3】给定一个大写字母，要求用小写字母输出。

解题思路: 字符数据以ASCII码存储在内存中，形式与整数的存储形式相同。

所以字符型数据和其他算术型数据之间可以互相赋值和运算。

大小写字母之间的关系是：同一个字母，用小写表示的字符的ASCII代码比用大写表示的字符的ASCII代码大32。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main11()

{

char c1,c2;

c1 = 'A'; //将字符′A′的ASCII码(int)放到c1变量中

c2 = c1 + 32; //得到字符′a′的ASCII码(int)，放在c2变量中

printf("%c\n",c2); //输出c2的值，是一个字符

printf("%d\n",c2); //输出c2的值，是字符′a′的ASCII代码

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

强制类型转换运算符

(类型名)(表达式)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main12()

{

int a; float x,y; double d;

d = (double)a; // 将ａ转换成double型

a = (int)(x+y); // 将x+y的值转换成int型

x = (float)(5%3); // 将5%3的值转换成float型

x = (float)a/3; // 与x = a/3的计算结果不同

a = (int)x+y; // 只将x转换成整型，然后与y相加

x = 10.2;

a = (int)x;

// 进行强制类型运算(int)x后得到一个int类型的临时值，它的值等于ｘ的整数部分，把它赋给a，

// 注意x的值和类型都未变化，仍为float型。该临时值在赋值后就不再存在了。

printf("%d,%f\n",a,x); // 10,10.2

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

表达式语句由一个表达式加一个分号构成，最典型的是由赋值表达式构成一个赋值语句。例如:

a = 3

是一个赋值表达式，而

a = 3;

是一个赋值语句。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main13()

{

const int a = 10;

int b;

// 并不是任何形式的数据都可以作为左值的，左值应当为存储空间并可以被赋值。

// 变量可以作为左值，而算术表达式a+b就不能作为左值，常量也不能作为左值。

b = a;

// a = b; // [Error] assignment of read-only variable 'a'

// 可以用{ }把一些语句和声明括起来成为复合语句(又称语句块)。

// 不影响{}外的同名变量的值。

float area = 20;

if (b == 10)

{

float pi=3.14159, r=2.5, area; //定义变量

area=pi\*r\*r;

printf("area=%f\n",area);

}

else printf("area=%f\n",area);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

数据输入和输出

scanf("格式控制"，变量地址表列);

printf("格式控制"，变量表列);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main14()

{

int a; char c; float f; double d;

// 整数，单精度数，双精度数的输入

printf("请输入a f d\n"); // 输入前最好有提示

// "格式描述符" 与变量的类型一一对应

scanf("%d%f%lf",&a,&f,&d); // " "内原样输入,末尾不要有"\n",空格隔开,变量前缀'&', 回车结束输入

//printf("请输入a,f,d\n"); // 输入前最好有提示

//scanf("%d,%f,%lf",&a,&f,&d); // " "内原样输入,末尾不要有"\n",逗号隔开,变量前缀'&', 回车结束输入

// "格式描述符" 与变量的类型一一对应

printf("%d,%f,%lf\n",a,f,d); // " "内原样输输出,变量前缀不能有'&'

printf("%d\t%f\t%lf\n",a,f,d); // '\t'制表符隔开

// 由于输入数据间用空格或别的字符隔开，因此字符的输入最好用单独语句输入。

printf("请输入一个字符\n");

scanf("%c",&c); // 接收了上一句scanf()输入缓冲区中剩余的回车符'\n'

//scanf("%c",&c); // 上一句scanf()消费的'\n',这次才是真正接收输入的字符。 另外一种办法是在其他scanf()之前，写接收字符的scanf()

printf("%d,%f,%lf,%d,%c\n",a,f,d,c,c);

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

【例3.5】求ax2+bx+c=0方程的根。a,b,c由键盘输入，设b^2-4ac＞0。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main15()

{

double a,b,c,disc,x1,x2,p,q; //disc用来存放判别式(b\*b-4ac)的值

scanf("%lf%lf%lf",&a,&b,&c); //输入双精度型变量的值要用格式声明″%lf″

disc=b\*b-4\*a\*c;

p=-b/(2.0\*a);

q=sqrt(disc)/(2.0\*a);

x1=p+q;x2=p-q; //求出方程的两个根

printf("x1=%7.2f\nx2=%7.2f\n",x1,x2); //输出方程的两个根

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char c;

scanf("%c",&c); // 替代形式

c = getchar();

putchar(c); // 参数可以是整数，表示char对应的ASCII码

int

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main16()

{

int a=66,b=79,c=89;

putchar(a);

putchar(b);

putchar(c);

putchar ('\n');

a=getchar(); //从键盘输入一个字符，送给字符变量a

b=getchar(); //从键盘输入一个字符，送给字符变量b

c=getchar(); //从键盘输入一个字符，送给字符变量c

putchar(a); //将变量a的值输出

putchar(b); //将变量b的值输出

putchar(c); //将变量c的值输出

putchar('\n'); //换行

return 0;

}