数据的表现形式及其运算：

在计算机高级语言中，数据有两种表现形式：常量和变量。

常量：其指不能被改变的量。（即数学中的常数）

常见的常量有以下几类：

1.整型变量。例：1000，0，-345等。

十进制小数，由数字和小数点组成。例：12.0，-0.6

2.实型变量

指数形式，例：12.34e3（代表12.34）

注意：由于在计算机输入或输出时，无法表示上角或下角，故规定以字母e或E代表以10为底的指数，E或e前必须有数字，且后面必须为整数。

3.字符常量包括普通字符和转义字符。

普通字符：用单括号括起来的一个字符，例：’a’，’Z’，’3’，’？’。

单撇号只是界限符，字符常量只能是一个字符，不包括单撇号。

字符常量存储在计算机存储单元中时，并不是存储字符本身，而是以其代码

（一般采用ASCLL代码）存储。对数字0~9字符的代码，后面一个数字的代码应该比前一个数字的代码大（如在ASCII字符集中，数字’2’的代码是50，数字’3’的代码是51，后者比前者的代码大1，符合要求）。

转义字符：意思将“\”后面的字符转换成另外的意义。

4.字符串常量：用双撇号把若干个字符括起来。如：”boy”，”123”。

单撇号只能包括一个字符。

双撇号可以包括一个字符串。

5.符号常量：用#define指令，指定用一个符号名称代表一个常量。

例：#define PI 3.1416 //注意行末没有分号

区别：不要把符号常量误认为是变量。符号常量不占内存，只是一个临时符号，在预编译后这个符号就不存在了，故不能对符号常量赋以新值。习惯上符号常量用大写表示，如：PI, PRICE等。

变量：代表一个有名字的、具有特定属性的一个存储单元。它用来存放数据，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。变量必须先定义变量的名字和类型，后使用。

常变量：常变量具有的基本属性：有类型，占存储单元，只是不改变其值，

例：const int a=3; 表示a被定义为一个整型变量，指定其值为3，而且在变量存在期间其值不能改变。

符号常量和常变量的区别：定义符号常量用#define指令，它是预编译指令，它只是用符号常量代表一个字符串，在预编译时仅仅是进行字符替换，在预编译后，符号常量就不存在了（全置换成别的内容了），对符号常量的名字是不分配存储单元的。而常变量要占用存储单元，有变量值，只是该值不改变而已，从使用的角度来看，常变量具有符号常量的优点，而且使用更方便。

标识符：在计算机高级语言中，用来对变量、符号常量名、函数、数组、类型等命名的有效字符序列统称为标识符（即一个对象的名字）。

C语言规定标识符只能由字母、数字、和下划线3种字符组成，且第一个字符必须为字母或下划线。

编译系统将大写字母和小写字母认为是两个不同的字符，变量名一般用小写字母表示，符号常量用大写表示。

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以及存储形式。

在计算机中，数据是存放在存储单元中的，它是具体存在的。而且存储单元是由有限的字节构成的，每一个存储单元中存放数据的范围是有限的，不能存放“无穷大”的数，也不能存放循环小数。用计算机进行的计算不是抽象的理论值的计算，而是用工程的方法实现的计算，在多数情况下只能得到近似的结果。

整型数据：基本整型（int），短整型（short int），长整型（long int），双长整型（long long int）。

整型变量的符号属性：变量值在存储单元中都是以补码形式存储的，存储单元中的第一个二进制代表符号，整型变量的值的范围包括负数到正数。

Unsigned表示“无符号整数”类型

Signed表示“有符号整数”类型

如果既未指定为signed也未指定为unsigned的，默认为有符号类型。

例：signedint int和int a等价。

只有整型（包括字符型）数据可以加signed或unsigned修饰符，实型数据不能加。

由于左面最高位不再用来表示符号，而是用来表示数值，因此无符号整型变量中可以存放的正数的范围比一般整型变量中正数的范围扩大一倍。

字符型数据

字符是以整数形式（字符的ASCII代码）存放在内存单元中的。

所有的127个字符都可以用7个二进位表示（ASCII代码为127时，二进制形式为1111111，7位全1）。

字符’1’和整数1是不同的概念，字符’1’只是一个形状为’1’的符号，在需要时按原样输出，在内存中以ASCII码形式存储，占1个字节，而整数1是以整数存储方式（二进制补码方式）存储的，占2个或4个字节。

整数运算1+1等于2，而字符’1’+’1’不等于整数2或字符’2’。

字符变量：用类型符char定义字符变量，实际上是一个字节的整型变量，可以把0~127之间的整数赋给一个字符变量，在输出字符变量时，可以选择以十进制整数形式输出，或以字符形式输出。

在使用有符号字符型变量时，允许存储的值为-128~127，但字符的代码不可能为负值，所以在存储字符时实际上只用到0~127这一部分，其第一位都是0。

如果将一个负整数赋给有符号字符型变量是合法的，但它不代表一个字符，而作为一字节整型变量存储负整数。

浮点型数据：用来表示具有小数点的实数。

在c语言中，实数是以指数形式存放在存储单元中的，一个实数表示为指数可以有不止一种形式，。

规范化的指数形式：指把小数部分中小数点前的数字为0、小数点后第一位数字不为0的表示形式。如 就是3.14159的规范化的指数形式。一个实数只有一个规范化的指数形式，在程序以指数形式输出一个实数时，必然以规范化的指数形式输出，如。

怎样确定常量的类型？：由单撇号括起来的单个字符或转义字符就是字符常量。对于数值常量按以下规律：

整型常量：不带小数点的数值是整型常量。

浮点型常量：凡以小数形式或指数形式出现的实数，是浮点型常量，在内存中都以指数形式存储。如：10是整型常量，10.0是浮点型常量。那么对浮点型常量是按单精度还是双精度处理呢？c编译系统把浮点型常量都按双精度处理，分配8个字节。

可以在常量的末尾加专用字符，强制指定常量的类型。如：3.14159后面加字母F或f，就表示是float型常量，分配4个字节。

每一个变量都属于一个确定的类型，类型是变量的一个重要的属性。变量是占用存储单元的，是具体存在的实体，在其占用的存储单元中可以存放数据。而类型是变量的共性，是抽象的，不占用存储单元，不能用来存放数据。

例：int a=3； //正确，对整型变量a赋值

Int=3； //错误，不能对类型赋值

向0取整就是求值后小数点后的数值舍弃。如：0.756=0。

%运算符要求参加运算的运算对象（即操作数）为整数，结果也是整数。如8%3=2，除%运算符以外的运算符的操作数都可以是任何算术类型。

自增、自减运算符：作用是使变量的值加1或减1。

例：+ +i，- -i (在使用i之前，先使i的值加（减)1）

i+ +，i- - （在使用i之后，使i的值加（减）1）

实例：i=3；

printf（”%d”,++i）；

输出4。

若改为

printf（”%d”,i++）；

则输出3。

注意：自增、自减运算符只能用于变量，而不能用于常量或表达式。

自增、自减运算符常用于循环语句中，使循环变量自动加1；也用于指针变量，使指针指向下一个地址。

赋值运算符：若有a=b=c，按右到左顺序，先把变量c的值赋给变量b，然后把变量b的值赋给变量a，即右结合性。

强制类型转换运算符

（double）a （将a转换成double类型）

（int）（x+y） （将x+y的值转换成int型）

（Float）（5%3） （将5%3的值转换成float型）

其一般形式为

（类型名）（表达式）

注意，表达式应该用括号括起来。

写成 （int）x+y

则只将x转换成整型，然后与y相加。

需要说明的是，在强制类型转换时，得到一个所需类型的中间数据，而原来变量的类型未发生变化。

例如： a=（int）x

如果已定义x为float型变量，a为整型变量，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。只是该临时值在赋值后就不再存在了。

复合的赋值运算符

在赋值符 = 之前加上其他运算符，可以构成复合的运算符。如果在“=”前加一个“+”运算符就成了复合运算符“+=”。例如，可以有以下的复合赋值运算：

a+=3 等价于 a=a+3

x\*=y+8 等价于 x=x\*（y+8）

x%=3 等价于 x=x%3

以“a+=3”为例来说明，它相当于使a进行一次自加3的操作。即：先使a加3，再赋给a。同样，“x\*=y+8”的作用是使x乘以（y+8），再赋给x。

凡是二元（二目）运算符，都可以与赋值符一起组成复合赋值符。

避免将占字节多的整型数据向占字节少的整型变量赋值，因为赋值后数值可能发生失真。

C语言本身并不提供输入输出语句，输入输出操作是由c标准函数库中的函数来实现的。printf和scanf不是c语言的关键字，而只是库函数的名字。实际上可以不用printf和scanf这两个名字，而另外编写一个输入输出函数，用来实现输入输出的功能，采用其他名字作为函数名。不把输入输出作为c语句的目的是使c语言编译系统简单精炼，因为将语句翻译成二进制的指令是在编译阶段完成的，没有输入输出语句就可以避免在编译阶段处理与硬件有关的问题，可以使编译系统简化，而且通用性好，可移植性好，在各种型号的计算机和不同的编译环境下都能适用，便于在各种计算机上实现。

格式字符

1. d格式符：用来输出一个有符号的十进制整数。

可以在格式声明中指定输出数据的域宽（所占的列数），如用“%5d”，指定输出数据占5行，输出的数据显示在此5列区域的右侧。

“%ld”代表long（长整型）数据

“%lld”代表long long（双长整型）数据

1. c格式符：用来输出一个字符。

一个整数，如果在0~127范围中，也可以用“%c”使之按字符形式输出，在输出前，系统会将该整数作为ASCII码转换成相应的字符

1. s格式符：用来输出一个字符串。
2. f格式符：用来输出实数（包括单、双精度、长双精度），以小数形式输出。
   1. 基本型，用%f。

不指定输出数据的长度，由系统根据数据的实际情况决定数据所占的列数。一般实数中的整数部分全部输出，小数部分输出6位。

* 1. 指定数据宽度和小数位数，用%m .nf。

不要轻易指定小数的位数为0，因为如果把小数部分指定为0，不仅不输出小数，而且小数点也不输出。

注意：一个双精度数只能保证15位有效数字的精确度，即使指定小数位数为50（如用%55.50f），也不能保证输出的50位都是有效的数字。

在使用%f输出时要注意数据本身能提供的有效数字，如float型数据的存储单元只能保证6位有效数字。double型数据能保证15位有效数字，不要以为计算机输出的所有数字都是绝对精确有效的。

* 1. 输出的数据向左对齐，用%-m .nf。

在m .n的前面加一个负号，其作用与%m .nf形式作用基本相同，但当数据长度不超过m时，数据向左靠，右端补空格。

* 1. e格式输入符：用格式声明%e指定以指数形式输出实数。

如果不指定输出数据所占的宽度和数字部分的小数位数，许多c编译系统（如visual c++）会自动给出数字部分的小数位数为6位，指数部分占5列（如e+002，其中“e”占1列，指数符号占1列，指数占3列）。

数值按标准化指数形式输出（即小数点前必须有而且只有1位非零数字）。

例： printf（“%e”，123.456）；

输出如下：

1.234560 e+002

6列 5列

格式符e也可以写成大写E形式。

以下几种初学时不太常用，只作介绍。

1. i格式符：按十进制整型数据的实际长度输出，作用与d格式符相同，一般习惯用%d而少用%i。
2. o格式符：以八进制整数形式输出。将内存单元中的各位的值（0或1）按八进制形式输出，因此输出的数值不带符号，即将符号位也一起作为八进制数的一部分输出。
3. x格式符：以十六进制数形式输出整数。
4. u格式符：用来输出无符号(unsigned)型数据，以十进制整数形式输出。
5. g格式符：用来输出浮点数，系统自动选f格式或e格式输出，选择其中长度较短的格式，不输出无意义的0。

字符数据的输入输出

1.用putchar（put character）函数输出一个字符。

putchar函数是输出字符的函数，它只能输出字符，即使输入整型数据也不能输出整数，只会将其对应的ASCII码输出。

2.用getchar（get character）函数输入一个字符。

如果是unsigned char 类型，可以存放0~255范围内的整数；

如果是signed char 类型，可以存放-128~127范围内的整数。