10.20.2023

符号常量：补码存储（#define预编译命令——在编译时展开，第八讲）

换行：换行，但是光标还在那一列

回车：不换行，回到首位，可以后续覆盖

符号常量（诨名）和常变量可以共存

符号常量可以改，常变量不行（提高安全性）（常变量是C++对C的改进，更好用）

字符常量：加单引号——八位（‘1’和1不一样！‘1’是49）

记忆ASCII！

字符串常量：双引号！“zhang san”“x”

字符串常量不能表示在字符变量里，**字符串末尾还自动加了’\0’**（字符串结束标志）

‘\0’ASCII为0，不可见，但占字

字符串变成字符的时候会扔掉“\0”

空串：“”也包含‘\0’；空字符’’不可见。

字符型：ASCII→补码，可以与整型通用

字符串中”\\”打印\（\转义表示）

数据构成计算表达式，从而写出C语句（加个分号）

运算符（44个，13类）：

算数运算符、关系~、逻辑~、位~（汇编用）、赋值~（=）、条件~、逗号~（，）、指针~（难）、求字节长度~、强制类型转换~、分量~、下标~、其它（函数调用~（（））……）

不用刻意记

表达式：运算符将常量、变量、函数连接起来构成的计算式

参与的量必须有确切的值，经过运算得到确切的值和类型

注意：运算符的结合性（有左往右也有右往左，小心）；优先级（与数学一样）；

数据类型转换（参与的类型必须一致，可能自动转换或者手动强制转换）

表达式→C语句→程序

表达式+；=语句（语句运算符：；）

按照语句进行编译

单目运算符（不考虑类型）；双目运算符（必须类型一致）

自动转换规则：按最长的转换（给出warning，符合意图可以忽略）

char、short→int

int→unsigned→long→double

float→double

强制转换：(int)a

强转很灵活，必须知道自己在做什么！可以用来解决一些warning。

最好类型都一样；如果不一样，强转可以代表自己的意愿，比较好

float b=3.14（3.14是double，会给warning），可以float b=(float)3.14

int+float都转成double

多个运算符，先考虑优先级，在考虑结合性

寄存器完成自动类型转换，不是改变变量类型

数据类型（下）

继续做砖

算数表达式；赋值表达式；逻辑表达式；关系表达式

第四章P62~73

算数运算符：5种双目——+-\*/%（求余数）

优先级：先\*/%，后+-，**括号先内后外**

结合性：左往右

赋值的优先级最低

对int，5/3=1（大多数情况下向零取整，注意编译器）

小数的除法？

a1=3;a2=5;bi=(float)(a1/a2)==0;b4=(float)a1/a2==0.60;

printf里普通字符原样输出

%4.2f：输出五列，保留两位小数：1387.34→实际长度输出1387.34；5.2→左对齐补空格

一般保留小数直接砍掉，但是小心编译系统差别

自增运算符++；自减--——单目运算符，目必须用于变量

++k前置（先加一再使用）；k++后置（先使用再加一）

int i=3,k,j;k=++I;，i=4，k=4

int i=3,k,j;j=i++;，i=4，j=3

++ --和负号优先级一样，比正号高，结合性：自右至左

记不住加括号！

好多个自增时a++a++++a这种不同编译器结果不一样，编程是为了确定性，不要这么写

（感兴趣可以试试，结合性和自增顺序问题）

赋值运算符：=

赋值表达式：<变量>=<表达式>，先计算表达式的值，再存储在左边的变量

结合性：从右至左，类型自动向左转换

复合赋值运算符：+=：y+=a即y=y+x（把两条指令变成一条指令，优化执行，锦上添花）

x=a+=b\*=c-=5，先做c-=5，再做b\*=c，……

一般不这么写

a\*=2+3即a\*=(2+3)

不会考结合性复杂的题目

关系运算符：<,<=,>,>=,==,!=

<,<=,>,>=的优先级高于==,!=

半角：1byte（ASCII），全角：2byte（中文）

都用半角！小心全角半角！（查错时候看不清）

优先级：算数高于关系高于赋值

结合性：从左至右

关系表达式计算结果：1或0

3<x<6结果一定是1

逻辑运算符：&&与；||或；！非

！单目，优先级最高——非0→0；0→非0

&&——均非0→1，否则0

&&和||双目，&&优先级更高，从左至右结合

&&和||是短路运算符，从左往右计算时只要确定结果了就不算后一半了（对++x不友好）

不过加括号优先级可能又能让++进行

用代码写推理（应用问题用代码表达）（穷举遍历，计算机高效）