C 语言的准备知识

1）bit 是位 是指为 0 或 1。 byte 是指字节，一个字节 = 八个位。

2）计算机的数据在电脑中保存是以二进制的形式，数据存放的位置就是它的内存地址。

3）进制的转换

十进制转换成二进制、八进制、十六进制。

二进制、八进制、十六进制转换成十进制。

4）整数的补码表示

整数在机器内大多用补码表示，最高位为 1 的表示负整数，其它数值部分按位取反后加 1。因此(-43)补=1111 1111 1101 0101

111 1111 1101 0101　取反后得到

000 0000 0010 1010　加 1 得到

000 0000 0010 1011　$2^5+2^3+2^1+2^0=32+8+2+1=43$

如果要写出-43 在机器内的二进制存放形式，步骤如下：

将 43 转换为二进制形式

000 0000 0010 1011　取反后得到

111 1111 1101 0100　加 1 后得到

111 1111 1101 0101　最后补上符号位

1111 1111 1101 0101

C 语言程序

1）编译预处理（如#include<stdio.h>）不是 C 语言的一部分，不占运行时间，不要加分号。C 语言编译的程序称为源程序，它以 ASCII 数值存放在文本文件中。

2）C 语言程序：可以没有输入，但是一定要有输出。

3）程序有三种结构：顺序结构 、分支结构 (选择结构)、循环结构。

4）程序都是从 main( )入口，然后从最上面顺序往下读(碰到循环做循环,碰到选择做选择)，有且只有一个 main 函数。

5）C 语言编写的程序称为源程序，又称为编译单位。

6）C 语言书写格式是自由的，每行可以写多个语句，可以写多行。注释不是 C 语言，不占运行时间，没有分号。不可以嵌套！

7）一个 C 语言程序有且只有一个 main 函数（main 函数必须有，普通函数可以没有），是程序运行的起点。

8）VC 是软件，用来运行写的 C 语言程序。

9）每个 C 语言程序写完后，都是先编译，后链接，最后运行。(.cpp——→.obj——→.exe)这个过程中注意.cpp 和.obj 文件时无法运行的，只有.exe 文件才可以直接运行。

C 语言的基础知识

- 标识符

1、标识符：

合法的要求是由字母，数字，下划线组成。有其它元素就错了。并且第一个必须为字母或则是下划线。第一个为数字就错了。

2、标识符分为关键词、准关键词、用户标识符。

关键词：不可以作为用户标识符号。main　define　scanf　printf 都不是关键词。迷惑你的地方 If 是可以做为用户标识符。因为 If 中的第一个字母大写了，所以不是关键词。

准关键词（也称预定义标识符）： scanf printf define include。记住准关键词可以做为用户标识符。

- 整数与实数

1）C 语言只有八、十、十六进制，没有二进制。但是运行时候，所有的进制都要转换成二进制来进行处理。

a、C 语言中的八进制规定要以 0 开头。018 的数值是非法的，八进制是没有 8 的，逢 8 进 1。

b、C 语言中的十六进制规定要以 0x 开头（a~f）。

2）小数的合法写法：C 语言小数点两边有一个是零的话，可以不用写。

1.0 在 C 语言中可写成 1.

0.1 在 C 语言中可以写成.1。

3）实型数据的合法形式：

a、2.333e-1 就是合法的，且数据是 2.333×10^{-1}。

e 前 e 后必有数，e 后必为整数。

4） 整型一般是 4 个字节， 字符型是 1 个字节，双精度一般是 8 个字节，单精度浮点型为 4 个字节：

long int x; 表示 x 是长整型。

unsigned int x; 表示 x 是无符号整型。

- 算术表达式和赋值表达式

核心：表达式一定有数值！

1、算术表达式：+, -, *, /, %

一定要注意：“/” 两边都是整型的话，结果就是一个整型。 3/2 的结果就是 1.

“/” 如果有一边是小数，那么结果就是小数。 3/2.0 的结果就是 0.5

“%”符号请一定要注意是余数，最容易算成了除号。）%符号两边要求是整数。不是整数就错了。[注意!!!]

2、自加表达式：

自加、自减表达式：假设 a=5，++a（是为 6）， a++（为 5）；

运行的机理：++a 是先把变量的数值加上 1，然后把得到的数值放到变量 a 中，然后再用这个++a 表达式的数值为 6，而 a++是先用该表达式的数值为 5，然后再把 a 的数值加上 1 为 6，

再放到变量 a 中。进行了++a 和 a++后 在下面的程序中再用到 a 的话都是变量 a 中的 6 了。

++在前先加后用，++在后先用后加。

3、关系表达式：

a、表达式的数值只能为 1（表示为真），或 0（表示假）。

如 9>8 这个关系表达式是真的，所以 9>8 这个表达式的数值就是 1。

如 7<6 这个关系表达式是假的，所以 7<6 这个表达式的数值就是 0

b、最容易错的：就是 int x=4,y=3,z=2;

x>y>z 是真还是假？带入为 4>3>2，从数学的角度出发肯定是真的，但是如果是 C 语言那么就是假的！因为 4>3 为真得到 1，表达式就变成了 1>2 那么运算结果就是假得到 0！

c、等号和赋值的区别！一定记住“=”就是赋值，“= =”才是等号。

关系运算符号：注意<=的写法，==和=的区别！

4、逻辑表达式：

核心：表达式的数值只能为 1（表示为真），或 0（表示假）。

a） 共有&&　||　！ 三种逻辑运算符号。

b）！>&&>||　优先的级别。

c） 注意短路现象。

d） 表示 x 小于 0 大于 10 的方法。

0<x<10 是不行的（一定记住）。是先计算 0<x 得到的结果为 1 或则 0；再用 0，或 1 与 10 比较得到的总是真（为 1）。所以一定要用 (0<x)&&(x<10)表示比 0 大比 10 小。

特别要注意：C 语言中是用非 0 表示逻辑真的，用 0 表示逻辑假的。

C 语言有构造类型（后面会学习），没有逻辑类型。

5、赋值表达式：

1、int x=y=10：错，定义时，不可以连续赋值。

2、int x,y;

x=y=10;　对，定义完成后，可以连续赋值。

3、赋值的左边只能是一个变量。

4、int x=7.7；对，x 就是 7

5、float y=7；对，x 就是 7.0

6、复合的赋值表达式：

int a=2;

a*=2+3；运行完成后，a 的值是 10。

一定要注意，首先要在 2+3 的上面打上括号。变成（2+3）再运算。

7、逗号表达式：

优先级别最低。表达式的数值逗号最右边的那个表达式的数值。

（2，3，4）的表达式的数值就是 4。

z=（2，3，4）（整个是赋值表达式）这个时候 z 的值为 4。

z= 2，3，4 （整个是逗号表达式）这个时候 z 的值为 2。

补充：

3、强制类型转换：

一定是 （int）a 不是 int（a），注意类型上一定有括号的。

注意（int）(a+b) 和（int）a+b 的区别。 前是把 a+b 转型，后是把 a 转型再加 b。

4、三种取整丢小数的情况：

1、int a =1.6;

2、(int)a;

3、1/2; 3/2;

- 字符

1）字符数据的合法形式::

'1' 是字符占一个字节，"1"是字符串占两个字节(含有一个结束符号’\0’)。

'0' 的 ASCII 数值表示为 48，'a' 的 ASCII 数值是 97，'A'的 ASCII 数值是 65。

一般表示单个字符错误的形式：'65'　"1"

字符是可以进行算术运算的，记住：　'0'-0=48

大写字母和小写字母转换的方法：　'A'+32='a'　相互之间是相差 32（'a'-'A'）

2）转义字符：

转义字符分为一般转义字符、八进制转义字符、十六进制转义字符。

一般转义字符： \0、 \n、 \’ 、 \” 、 \\。

八进制转义字符： ‘\141’ 是合法的， 前导的 0 是不能写的。

十六进制转义字符：’ \x6d’ 合法的，前导的 0 不能写，并且 x 是小写。

3、字符型和整数是近亲：两个具有很大的相似之处

char a = 65 ;

printf(“%c”, a); 得到的输出结果：a

printf(“%d”, a); 得到的输出结果：65

输入输出

- 数据输出

1、使用 printf 和 scanf 函数时，要在最前面加上#include “stdio.h”

2、printf 可以只有一个参数，也可以有两个参数。

3、printf (“ 第一部分 ”, 第二部分); 把第二部分的变量、表达式、常量以第一部分的形式展现出来！

4、printf (“a=%d, b=%d”, 12, 34)

一定要记住是将 12 和 34 以第一部分的形式现在在终端也就是黑色的屏幕上。核心为：

一模一样。在黑色屏幕上面显示为 a=12, b=34

printf (“a=%d, \n b=%d”, 12, 34) 那么输出的结果就是：a=12,
b=34

5、int x=017; 一定要弄清楚为什么是这个结果！过程很重要？？？？？？？？

printf (“%d”, x); 15？？？？？？

printf (“%o”, x); 17 ？？？？？

printf (“%#o”, x); 017？？？？？

printf (“%x”, x); 11 ？？？？

printf (“%#x”, x); 0x11 ？？？？？

6、int x=12, y=34;

char z= ‘a’;

printf (“%d ”, x, y); 一个格式说明，两个输出变量，后面的 y 不输出

printf (“%c”, z); 结果为：12a

7、一定要记住!!!!!!!!!!

格式说明	表示内容	格式说明	表示内容
%d	整型 int	%c	字符 char
%ld	长整型 long int	%s	字符串
%f	浮点型 float	%o	八进制
%lf	double	%#o	带前导的八进制
%%	输出一个百分号	%x	十六进制
%5d		%#x	带前导的十六进制

举例说明：

printf (“%2d”, 123); 第二部分有三位，大于指定的两位，原样输出 123

printf（“%5d”，123 ）；　第二部分有三位，小于指定的五位，左边补两个空格　123

printf（“%10f”，1.25 ）；　小数要求补足 6 位的，没有六位的补 0,。结果为　1.250000

printf（“%5.3f”，125 ）；　小数三位，整个五位，结果为 1.250（小数点算一位）

printf（“%3.1f”，1.25 ）；小数一位，整个三位，结果为 1.3（要进行四舍五入）

- 数据输入

1、scanf（“a=%d，b=%d”，&a，&b）超级重点！

一定要记住是以第一部分的格式在终端输入数据。核心为：一模一样。

在黑色屏幕上面输入的为　a=12，b=34 才可以把 12 和 34 正确给 a 和 b 。有一点不同也不行。

2、scanf（“%d，%d”，x，y）；这种写法绝对错误，scanf 的第二个部分一定要是地址！

scanf（“%d，%d”，&x，&y）；注意写成这样才可以！

3、特别注意指针在 scanf 的考察

例如：　int x=2；int *p=&x；

scanf（“%d”，x）；　错误　　scanf（“%d”，p）；正确

scanf（“%d”，&p）；　错误　　scanf（“%d”，*p）错误

4、指定输入的长度　（重点）

终端输入：1234567

scanf（“%2d%4d%d”，&x，&y，&z）；x 为 12，y 为 3456，z 为 7

终端输入：1 234567　　由于 1 和 2 中间有空格，所以只有 1 位给 x

scanf（“%2d%4d%d”，&x，&y，&z）；x 为 1，y 为 2345，z 为 67

5、字符和整型是近亲：

int x=97；

printf（“%d”，x）；　结果为 97

printf（“%c”，x）；　结果为 a

6、输入时候字符和整数的区别（注意注意）

scanf（“%d”，&x）；这个时候输入 1，特别注意表示的是整数 1

scanf（“%c”，&x）；这个时候输入 1，特别注意表示的是字符‘1’ASCII 为整数 48。

补充说明：

1）scanf 函数的格式

注意该函数的第二个部分是&a 这样的地址，不是 a；

scanf（“%d%d%*d%d”，&a，&b，&c）；跳过输入的第三个数据。

2）putchar ，getchar 函数

char　a = getchar() 是没有参数的，从键盘得到你输入的一个字符给变量 a。

putchar（‘y’）把字符 y 输出到屏幕中。

3）如何实现两个变量 x ，y 中数值的互换

不可以把 x=y ，y=x；要用中间变量 t=x；x=y；y=t。

4）如何实现保留三位小数，第四位四舍五入的程序，

y=（int）（x*100+0.5）/100.0　这个保留两位，对第三位四舍五入

y=（int）（x*1000+0.5）/1000.0 这个保留三位，对第四位四舍五入

y=（int）（x*10000+0.5）/10000.0 这个保留四位，对第五位四舍五入

注意 x = （int）x 这样是把小数部分去掉。