

C 语言基础知识总结

C 语言最重要的知识点复习资料

这个 a 版本，还会进行改正和补充，请大家认真熟悉这些基础知识！如有错误，请拨打！

总体上必须清楚的：

1) 程序结构是三种：顺序结构、选择结构(分支结构)、循环结构。

2) 读程序都要从 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。

3) 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他的地址。

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

概念常考到的：

1、编译预处理不是 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、`define PI 3.1415926;` 这个写法是错误的，一定不能出现分号。

3、每个 C 语言程序中 main 函数是有且只有一个。

4、在函数中不可以再定义函数。

5、算法：可以没有输入，但是一定要有输出。

6、break 可用于循环结构和 switch 语句。

7、逗号运算符的级别最低，赋值的级别倒数第二。

第一章 C 语言的基础知识

第一节、对 C 语言的基础认识

1、C 语言编写的程序称为源程序，又称为编译单位。

2、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。

3、一个C语言程序有且只有一个main函数，是程序运行的起点。

第二节、熟悉vc++

1、VC是软件，用来运行写的C语言程序。

2、每个C语言程序写完后，都是先编译，后链接，最后运行。（.c——.obj——.exe）这个过程中注意。c和.obj文件时无法运行的，只有.exe文件才可以运行。（常考！）

...文档交流 仅供参考...

第三节、标识符

1、标识符（必考内容）：

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、标识符分为关键字、预定义标识符、用户标识符。

关键字：不可以作为用户标识符号。main define scanf printf 都不是关键字。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，所以不是关键字。

...文档交流 仅供参考...

预定义标识符：背诵define scanf printf include。记住预定义标识符可以做为用户标识符。

...文档交流 仅供参考...

用户标识符：基本上每年都考，详细请见书上习题。

第四节：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。

第五节：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。（考过两

次)

...文档交流 仅供参考...

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。

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} 。

b、考试口诀:e前e后必有数,e后必为整数。请结合书上的例子。

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:

`long int x;` 表示x是长整型。

`unsigned int x;` 表示x是无符号整型。

第六、七节: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: +, -, *, /, %

考试一定要注意:“/”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，考试最容易算成了除号。) %符号两边要求是整数。不是整数就错了。

2、赋值表达式: 表达式数值是最左边的数值, a=b=5; 该表达式为5, 常量不可以赋值。

1、`int x=y=10;` 错啦，定义时，不可以连续

赋值。

2、int x, y;

x = y = 10; 对滴，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; 对滴，x 就是 7

5、float y=7; 对滴，x 就是 7.0

3、复合的赋值表达式：

int a=2;

a *=2+3; 运行完成后，a 的值是 12。

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。

4、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，

... 文档交流 仅供参考 ...

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。

考试口诀：++ 在前先加后用，++ 在后先用后加。

5、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。(有点难度哦！)

z = 2, 3, 4 (整个是逗号表达式) 这个时候 z 的值为 2。

补充：

1、空语句不可以随意执行,会导致逻辑错误。

2、注释是最近几年考试的重点,注释不是 C 语言,不占运行时间,没有分号。不可以嵌套!

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、 $1/2; 3/2;$

第八节、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' 的 ASCII 数值表示为 48,'a' 的 ASCII 数值是 97,'A' 的 ASCII 数值是 65.

一般考试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'

相互之间一般是相差 32.

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: 背诵 $\backslash 0$ 、 $\backslash n$ 、 \backslash' 、 \backslash'' 、 $\backslash \backslash$ 。

八进制转义字符: ' $\backslash 141$ ' 是合法的,前导的 0 是不能写的。

十六进制转义字符: ' $\backslash x 6d$ ' 才是合法的,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

第九章、位运算

1) 位运算的考查: 会有一到二题考试题目。

总的处理方法: 几乎所有的位运算的题目都要按这个流程来处理 (先把十进制变成二进制再变成十进制)。

例 1: `char a = 6, b;`

`b = a << 2;` 这种题目的计算是先要把 a 的十进制 6 化成二进制, 再做位运算。

例 2: 一定要记住, 异或的位运算符号 "`^`"。0 异或 1 得到 1。

0 异或 0 得到 0。两个女的生不出来。

考试记忆方法: 一男 (1) 一女 (0) 才可以生个小孩 (1)。

例 3: 在没有舍去数据的时候, `<<` 左移一位表示乘以 2; `>>` 右移一位表示除以 2。

第二章

第一节: 数据输出 (一) (二)

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。(选择题考过一次)

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!

... 文档交流 仅

供参考...

4、`printf ("a=%d, b=%d", 12, 34)` 考试重点!

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。考试核心为: 一模一样。在黑色屏幕

上面显示为 `a=12, b=34`

...文档交流 仅供参考...

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

`=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;` 注意这种题型

```
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: `12`

a

7、一定要背诵的

格式说明	表示内容	格式说明	表示内容
<code>%d</code>	整型 <code>int</code>	<code>%c</code>	字符 <code>char</code>
<code>%ld</code>	长整型 <code>long int</code>	<code>%s</code>	字符串
<code>%f</code>	浮点型 <code>float</code>	<code>%o</code>	八进制
<code>%lf</code>	<code>double</code>	<code>%#o</code>	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)

第三节 数据输入

1、`scanf(“a=%d, b=%d”, &a, &b)` 考试超级重点! 一定要记住是以第一部分的格式在终端输入数据. 考试核心为: 一模一样。

在黑色屏幕上输入的为 `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（考试重点）

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(考试超级重点)

`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` 函数的格式考察：

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` 函数的考查：

`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(要求背下来)

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, (要

求背下来)

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

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

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

这个有推广的意义, 注意 $x = (\text{int})x$ 这样是把小数部分去掉.

第三章

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, 没有逻辑类型。

关系运算符: 注意 $<=$ 的写法, $==$ 和 $=$ 的区别! (考试重点)

if 只管后面一个语句, 要管多个, 请用大括号!

1) 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、考试最容易错的: 就是 $\text{int } x=1, y=0, z=2;$

$x < y < z$ 是真还是假? 带入为 $1 < 0 < 2$, 从数学的角度出发肯定是

错的, 但是如果是 C 语言那么就是正确的! 因为要 $1 < 0$

为假得到 0,
表达式就变成

了 $0 < 2$ 那么运算结果就是 1, 称为了真的了!

c、等号和赋值的区别!一定记住“=”就是赋值,“==”才是等号。虽然很多人可以背

诵,但我依然要大家一定好好记住,否则,做错了,我一定会强烈的鄙视你!

2) 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. 考试比较喜欢考到。详细请见书上例子,一定要会做例 1 和例 2

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。

...文档交流 仅供参考...

3) if 语句

a、else 是与最接近的 if 且没有 else 的语句匹配。

b、交换的程序写法: $t=x; x=y; y=t;$

c、 $if(a < b) t=a; a=b; b=t;$

$if(a < b) \{t=a; a=b; b=t;\}$ 两个的区别,考试多次考到了!

d、单独的 if 语句: $if(a < b) t=a;$

标准的 if 语句: $if(a < b) \min=a;$

$else \min=b;$

嵌套的 if 语句: $if(a < b)$

```
if(b>c)printf("ok!");
```

```
多选一的if语句 if(a==t)printf("a");  
else if(b==t)printf("b");
```

```
else if(c==t)printf("c");  
else printf("d");
```

通过习题,要熟悉以上几种if语句!

经典考题:结合上面四种if语句题型做题,答错了,请自行了断!预备,开始!

```
int a=1, b=0;
```

```
if(!a) b++;
```

```
else if(a==0)
```

```
if(a) b+=2;
```

```
else b+=3; 请问b的值是多少?
```

如果没有看懂题目,你千万不要自行了断,这样看得懂不会做的人才会有理由的活着。

正确的是b为3.

```
int a=1, b=0;
```

```
if(!a) b++; 是假的不执行
```

```
else if(a==0) 是假的执行
```

```
if(a) b+=2; 属于else if的嵌套if语句,  
不执行。
```

```
else b+=3; if-else-if语句没有一个  
正确的,就执行else的语句!
```

4) 条件表达式:

表达式1 ?表达式2 :表达式3

a、考试口诀:真前假后。

b、注意是当表达式1的数值是非0时,才采用表达式2的数值做为整个运算结果,当表达式1的数值为0时,就用表达式3的数值做为整个的结果。

... 文档交流 仅供参考...

c、int a=1, b=2, c=3, d=4, e=5;

$k = a > b ? c : d > e ? d : e$; 求 k 的数值时多少? 答案为 san

5) switch 语句:

a) 执行的流程一定要弄清楚! 上课时候详细的过程讲了, 请自己一定弄清楚!

b) 注意有 `break` 和没有 `break` 的差别, 书上的两个例子, 没有 `break` 时候, 只要有一个 `case` 匹配了, 剩下的都要执行, 有 `break` 则是直接跳出了 `switch` 语句。 `break` 在 C 语言中就是分手, 一刀两断的意思。

c) `switch` 只可以和 `break` 一起用, 不可以和 `continue` 用。

d) `switch(x)` x : 是整型常量, 字符型常量, 枚举型数据。

```
    { case 1: ... }
    case 2: ...
    }
```

不可以是变量。

e) `switch` 是必考题型, 请大家一定要完成书上的课后的 `switch` 的习题。

第四章

1) 三种循环结构:

a) `for()` ; `while()` ; `do-while()` 三种。

b) `for` 循环当中必须是两个分号, 千万不要忘记。

c) 写程序的时候一定要注意, 循环一定要有结束的条件, 否则成了死循环。

d) `do-while()` 循环的最后一个 `while()`; 的分号一定不能够丢。(当心上机改错), `do-while` 循环是至少执行一次循环。

2) `break` 和 `continue` 的差别

记忆方法:

`break`: 是打破的意思, (破了整个循环) 所以看见 `br e`

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/885033223123011132>