

## C 语言最重要的知识点

总体上必须清楚的:

- 1)程序结构是三种: 顺序结构, 循环结构(三个循环结构), 选择结构(if 和 switch)
- 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)。
- 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。
- 5)一定要记住 二进制 如何划成 十进制。

概念常考到的:

- 1、编译预处理不是 C 语言的一部分, 不再运行时间。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、每个 C 语言程序中 main 函数是有且只有一个。
- 3、在函数中不可以再定义函数。
- 4、算法的是一定要有输出的, 他可以没有输入。
- 5、break 可用于循环结构和 switch 语句。
- 6、逗号运算符的级别最低。

## 第一章

### 1) 合法的用户标识符考查:

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。

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。

### 2) 实型数据的合法形式:

2.333e-1 就是合法的, 且数据是  $2.333 \times 10^{-1}$ 。 考试口诀: e 前 e 后必有数, e 后必为整数。.

### 3) 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。

### 4) 整型一般是两个字节, 字符型是一个字节, 双精度一般是 4 个字节:

考试时候一般会说, 在 16 位编译系统, 或者是 32 位系统。碰到这种情况, 不要去管, 一样做题。掌握整型一般是两个字节, 字符型是一个字节, 双精度一般是 4 个字节就可以了。

### 5) 转义字符的考查:

在程序中 int a = 0x6d, 是把一个十六进制的数给变量 a 注意这里的 0x 必须存在。

在程序中 int a = 06d, 是一个八进制的形式。

在转义字符中, '\x6d' 才是合法的, 0 不能写, 并且 x 是小写。

'\141' 是合法的, 0 是不能写的。

'\108' 是非法的, 因为不可以出现 8。

### 6) 算术运算符的优先级别:

同级别的有的是从左到右, 有的是从右到左。

### 7) 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。

### 8) 表达式的考查:

是表达式就一定有数值。

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。

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。

考试口诀: ++在前先加后用, ++在后先用后加。

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。

## 9) 位运算的考查:

会有 1~2 题考试题目。

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。

一定要记住, 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。

10) 018 的数值是非法的, 八进制是没有 8 的, 逢 8 进 1。

11) % 符号两边要求是整数。不是整数就错了。

12) 三种取整丢小数的情况:

1、`int a = 1.6;`

2、`(int)a;`

3、

## 第二章

### 1) printf 函数的格式考查:

`%d` 对应整型; `%c` 对应字符; `%f` 对应单精度等等。宽度的, 左对齐等修饰。

`%ld` 对应 long int; `%lf` 对应 double。

### 2) 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。

### 3) 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。

### 4) 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.`

### 5) 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)

这个有推广的意义, 注意 `x = (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。

### 1) 关系表达式:

表达式的数值只能为 1 (表示为真), 或 0 (表示假)

当关系的表达是为真的时候得到 1。如 `9>8` 这个是真的, 所以表达式的数值就是 1;

### 2) 逻辑表达式:

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。

d) 要表示 `x` 是比 0 大, 比 10 小的方法。`0<x<10` 是不可以的 (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。

### 3) if 语句

`else` 是与最接近的 `if` 且没有 `else` 的相组合的。

### 4) 条件表达式:

表达式 1 ? 表达式 2 : 表达式 3

注意是当非 0 时候 是表达式 2 的数值, 当为 0 是就是表达式 2 的数值。 考试口诀: 真前假后。

### 5) switch 语句:

a) 一定要注意 有 `break` 和没有 `break` 的差别, 书上 (34 页) 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 `swic` 语句。

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

## 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();的分号一定不能丢。(当心上机改错)

## 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 break 就退出真个一层循环。

continue: 是继续的意思，(继续循环运算)，但是要结束本次循环，就是循环体内剩下的语句不再执行，跳到循环开始，然后判断循环条件，进行新一轮的循环。

## 3) 嵌套循环

就是有循环里面还有循环，这种比较复杂，要一层一层一步一步耐心的计算，一般记住两层是处理二维数组的。

## 4) while ((c=getchar()) != '\n') 和 while (c=getchar() != '\n') 的差别

先看  $a = 3 \neq 2$  和  $(a=3) \neq 2$  的区别：

( $\neq$ 号的级别高于 $=$ 号 所以第一个先计算  $3 \neq 2$ ) 第一个 a 的数值是得到的 1；第二个 a 的数值是 3。考试注意点：括号在这里的重要性。

## 第五章

函数：是具有一定功能的一个程序块；

### 1) 函数的参数，返回数值 (示意图)：

```
main()
{
int a = 5,b=6,c;
    c = add(a,b);
    printf("%d",c);
}
```

调用函数

a,b 是实参

整个函数得到一个数值就是

Add 函数的返回数值。

```
int add ( int x, int y)
```

```
{
    int z;
    z=x+y;
    return z;
}
```

被调用函数

x, y 是形式参数

函数返回数值是整型

z 就是这个 add 函数计算后得到的结果，就是函数返回给主程序的返回数值。

程序是在从上往下顺序执行，当碰到了函数 add 后，把 a, b 的数值穿给调用函数，程序暂时中断等待返回数值。

当得到了返回数值后，再顺序的往下执行

### 2) 一定要注意参数之间的传递

实参和形参之间 传数值，和传地址的差别。(考试的重点)

传数值的话，形参的变化不会改变实参的变化。

传地址的话，形参的变化就有可能改变实参的变化。

### 3) 函数声明的考查：

一定要有：函数名，函数的返回类型，函数的参数类型。

不一定要有：形参的名称。

## 第六章

指针变量的本质是用来放地址，而一般的变量是放数值的。

int \*p 中 \*p 和 p 的差别：

\*p 可以当做变量来用；\*的作用是取后面地址 p 里面的数值

p 是当作地址来使用。

\*p++ 和 (\*p)++ 之间的差别：改错题中很重要

\*p++ 是 地址会变化。

(\*p)++ 是数值会变化。

三名主义：（考试的重点）

数组名：表示第一个元素的地址。数组名不可以自加，他是地址常量名。（考了很多次）

函数名：表示该函数的入口地址。

字符串常量名：表示第一个字符的地址。

## 第七章

### 1 一维数组的重要概念：

对 a[10] 这个数组的讨论。

1、a 表示数组名，是第一个元素的地址，也就是元素 a[0] 的地址。

2、a 是地址常量，所以只要出现 a++，或者是 a=a+2 赋值的都是错误的。

3、a 是一维数组名，所以它是列指针，也就是说 a+1 是跳一列。

对 a[3][3] 的讨论。

1、a 表示数组名，是第一个元素的地址，也就是元素 a[0][0] 的地址。

2、a 是地址常量，所以只要出现 a++，或者是 a=a+2 赋值的都是错误的。

3、a 是二维数组名，所以它是行指针，也就是说 a+1 是跳一行。

4、a[0]、a[1]、a[2] 也都是地址常量，不可以对它进行赋值操作，同时它们都是列指针，a[0]+1，a[1]+1，a[2]+1 都是跳一列。

5、注意 a 和 a[0] a[1] a[2] 是不同的，它们的基类型是不同的。前者是一行元素，后三者是一列元素。

二维数组做题目的技巧：

如果有 a[3][3]={1,2,3,4,5,6,7,8,9} 这样的题目。

步骤一：把他们写成：

	第一列	第二列	第三列	
a[0]	1	2	3	—>第一行
a[1]	4	5	6	—>第二行
a[2]	7	8	9	—>第三行

步骤二：这样作题目间很简单：

\*(a[0]+1) 我们就知道是第一行的第一个元素往后面跳一列，那么这里就是 a[0][1] 元素，所以是 1。

\*(a[1]+2) 我们就知道是第二行的第一个元素往后面跳二列。那么这里就是 a[1][2] 元素，所以是 6。

一定记住：只要是二维数组的题目，一定是写成如上的格式，再去做题目，这样会比较简单。

数组的初始化，一维和二维的，一维可以不写，二维第二个一定要写

int a[]={1, 2} 合法。 int a[][4]={2, 3, 4} 合法。 但 int a[4][]={2, 3, 4} 非法。

二维数组中的行指针

int a[1][2];

其中 a 现在就是一个行指针，a+1 跳一行数组元素。 搭配 (\*) p[2] 指针

a[0], a[1] 现在就是一个列指针。a[0]+1 跳一个数组元素。搭配 \*p[2] 指针数组使用

还有记住脱衣服法则：

a[2] 变成 \*(a+2) a[2][3] 变成 \*(a+2)[3] 再可以变成 \*((a+2)+3) 这个思想很重要！

其考试重点

文件的复习方法：

把上课时候讲的文件这一章的题目要做一遍，一定要做，基本上考试的都会在练习当中。

1) 字符串的 strlen() 和 strcat() 和 strcmp() 和 strcpy() 的使用方法一定要记住。他们的参数都是地址。

其中 strcat() 和 strcmp() 有两个参数。

2) strlen 和 sizeof 的区别也是考试的重点；

3) define f(x)(x\*x) 和 define f(x) x\*x 之间的差别。一定要好好的注意这写容易错的地方，替换的时候有括号和没有括号是很大的区别。

4) int \*p;

p = (int \*) malloc (2);

p = (int \*) malloc (sizeof (int)); 以上两个等价

当心填空题，malloc 的返回类型是 void \*

5) 还有 main (int argc, char \*\*argv) {} 这种含有参数的题目，是很呆板的题目。第一个参数是表示输入的字符串的数目，第二个参数是指向存放的字符串。

6) 函数的递归调用一定要记得有结束的条件，并且要会算简单的递归题目。要会作递归的题目

7) 结构体和共用体以及链表要掌握最简单的。typedef 考的很多，而且一定要知道如何引用结构体中的各个变量，链表中如何添加和删除节点，以及如何构成一个简单的链表，一定记住链表中的节点是有两个域，一个放数值，一个放指针。

8) 函数指针的用法 (\*f)() 记住一个例子：

```
int add(int x, int y)
{....}
main()
{ int (*f)();
  f=add;
}
```

赋值之后：合法的调用形式为 1、add(2, 3);

2、f(2, 3);

3、(\*f)(2, 3)

9) 两种重要的数组长度：

char a[]={ 'a' , 'b' , 'c' }; 数组长度为 3，字符串长度不定。sizeof(a)为 3。

char a[5]={ 'a' , 'b' , 'c' } 数组长度为 5，字符串长度 3。sizeof(a)为 5。

10) scanf 和 putchar 的数据：

如果输入的是 good good study!

那么 scanf(“%s”,a); 只会接收 good. 考点：不可以接收空格。

getchar(a); 会接收 good good study! 考点：可以接收空格。

11) 共用体的考查：

union TT

{ int a;

char ch[2];}

考点一： sizeof (struct TT) = 2;

考点二： TT t1; t1=0x1234;

那么 ch[0]=0x34; ch[1]=0x12

以下对改错题的改错方式做一些总结，当然这些总结只能对大部分改错行有效。

1、若错误行是函数首部，可分为以下几种情况：

A、该行最后若有分号则删除，中间若有分号则改成逗号

B、形参类型不一致的问题，特别是指针类型，若后面用到某形参时有指针运算则该形参必为指针类型；若形参是二维数组或指向 m 个元素的指针变量，则第二维的长度必须与 main 中对应数组的第二维长度相同

C、函数类型不一致的问题，若函数中没有 return 语句则函数类型为 void，若有 return 语句则函数的类型必须与 return 后变量的类型一致。

2、若错误行是 if 或 while 语句，则首先看有没有用小括号将整个表达式括起，若没有则加上小括号。

3、若错误行中有 if、while、for 则要特别注意条件表达式的错误问题：

A、指针变量的应用，若表达式中有指针变量且没有指针运算符，则加上指针运算符

B、若条件表达式中只有一个等于号，则改成两个等于号，若为其它比较运算符则一般是进行逆转或加一个等于号

C、for 中要用分号分隔表达式，而不是用逗号

4、语法错误

A、语句缺少分号，若错误行中有语句没有用分号结束，则加上分号。

B、大小写不对，若错误行中有大写字母则一般都改成小写字母。

5、指针变量的运用，若错误行中有指针变量，并且该变量名前没有指针运算符则一般都是加上指针运算符

6、若错误行为 return 语句，则首先看是否是缺少分号若是则加上分号即可；否则就是 return 后的变量或表达式错误(此时可通过看题意，来分析该返回哪一变量或表达式)

7、若错误行中见到整型 1 除以某个表达式时，一概改成 1.0。但若是整型变量或表达式则只能用强制转换

8、复合运算符写错

9、字符串结束符写错，若有字符串结束符则要特别注意有没有写错，但第 11 题例外，因为该题是要将数字字符转换成对应的数字。

10、若错误行是定义语句，则首先看类型是否符合;再看所赋初值是否正确;若以上均不是，则看是否少定义了某个变量或少了花括号。

11、表达式错误(占的份量最多，并且没有统一的改法，我们只能通过题目要求来分析并修改)

12、若错误行中有一条横线，则必须将横线删除再填空。填空题中亦是如此。