**C语言的基础知识**

**概念：**

⒈ 编译预处理 #include <stdio.h>。

⒉ 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。

⒊ 算法：可以没有输入，但是一定要有输出。

⒋ break可用于循环结构和switch语句。

⒌ 程序结构是三种: 顺序结构 、选择结构(分支结构)、循环结构。

**一、常量**

**⒈ 数字常量**

Ⅰ. 普通数字：1，35，2.7。

Ⅱ. 指数形式：2.45e-2等价于2.45\*10-2。

注意e大小写皆可，e前面的数字不能省，就算是1也不能省，后面的数字一定要是整数。

Ⅲ. 长整型，单精度浮点型：3235L，32.5F 分别表示3235是长整型数据，32.5是单精度浮点型左，若不写上L，F则表示3235是整型，32.5是双精度浮点型，L，F大小写皆可。

**⒉ 字符常量**

Ⅰ. 普通字符常量：用单引号把一个字符括起来，如‘A’,‘@’。

Ⅱ. 转义字符常量：一对单引号括起来并以“\”开头的字符序列，如‘\n’(回车), ‘\x23’(16进制23对应的字符)。

**二、标识符**

**⒈ 命名规则**

以数字，字母，下划线这三类字符组成，但只能以字母或下划线开头，而不能以数字开头，另外不能将关键字作为标识符。

**⒉ 变量名，函数名，数组名全都是标识符。**

**三、变量**

**⒈ 变量的定义格式**

类型名 变量名;

如 int a;定义了一个整型常量a。变量名是由人类随便定义的，符合命名规则的前提下，爱写啥就写啥。所以什么flag，cc，y1或者函数名fun,find等全部是自定的用来作为名字而已，没有更特别的意义。

**⒉ 类型名**

int 整型，long 长整型： 用于存放整数，只是数值范围不同

float 单精度浮点型 double 双精度浮点型：用于存放实数，数值范围，精度不同。

char字符型：用于存放字符。

**⒊ 变量赋值，初始化**

int a=3;定义的同时初始化。

a=6\*9;定义后在程序中进行赋值。

**⒋ 变量的值**

只有在赋值操作时才会被改变，即将其放在等号左边时才会改变它的值，或自增自减操作：a=5,a++,a--，像a+3并未改变a的值，只是使用了a的值而已。

**⒌ 自增自减运算**

变量++,++变量，变量--，--变量 使变量的值自增1或自减1

等价于 变量=变量+1 变量=变量-1。

*++，--放于变量前后效果的区别：*

*当自增自减运算作为表达式的一部分时，++，--放在变量前面是先自增自减再使用变量的值，放在变量后面则是先使用变量的值，再自增自减。*

如x=3; printf(“%d”,++x);则相当于执行了++x; printf(“%d”,x);这样的操作所以打印出4。

再如x=3; printf(“%d”,x++);则相当于执行了printf(“%d”,x); x++;这样的操作，则打印出3，当然最后x的值还是4。

**四、表达式**

**⒈ 算术表达式**

Ⅰ. 算术运算符：+,-,\*,/,%(求余)。

Ⅱ. 由算术运算符加上运算对象构成算术表达式,如3+3\*6-9/2。

Ⅲ. 值：跟我们小学时学的一样，就是表达式的计算结果。

Ⅳ. 整数除以整数结果取整数部分，故1/3得到的值是0。

Ⅴ. 5%3 结果为2，想想小学除法，求余得到的是余数不是商。

**⒉ 关系表达式**

Ⅰ. 关系运算符：< , > , <=, >=, = =(等于), !=(不等于)。

Ⅱ. 由关系运算符加上运算对象构成关系表达式,如3>=4, 2==a。

Ⅲ. 值：满足相应运算符所指定的关系的值为1，否则为0。

**⒊ 逻辑表达式**

Ⅰ. 逻辑运算符：&&(且) , ||(或) , !(非)。

Ⅱ. 由逻辑运算符加上运算对象构成逻辑表达式,如3&&4, x||!y。

Ⅲ. 值：满足相应运算符所指定的关系的值为1，否则为0。

Ⅳ. 进行 ||(或)运算时，若||左边的表达式值为1，则不再对右边的表达式进行运算。

Ⅴ. 进行 &&(且)运算时，若&&左边的表达式值为0，则不再对右边的表达式进行运算。

**五、输入输出函数**

**⒈ scanf(“格式控制串”，变量地址表列)**

如scanf(“%d%c%d”,&a,&ch,&b) scanf(“%4f”,&x)

**⒉ printf(“格式控制串”，输出项表列)**

如float x=7.5; printf(“%8.2f”,x);此处的意思是将x打印出来，且占8列，保留两位小数。

**常用格式符汇总：**

1. %d：输入输出整型数据， %c：输入输出字符型数据。
2. %f：输出单（双）精度浮点型数据，输入单精度型数据。

③ %s：输入输出一个字符串，用printf输出字符串时，输出项书写时可为字符串常量，或字符数组名。

如printf(“%s”, hello) char str[10]=“hi”

**六、C语言的语句**

**⒈ 表达式语句：**由表达式末尾加上分号构成。

**⒉ 选择结构语句：**if语句 switch语句。

**⒊ 循环语句：**for语句 while语句 do while语句。

**⒋ 复合语句：**用花括号｛｝将以上任意语句括起来构成一条复合语句。

**七、C程序的基本结构**

int main()

{

return 0;

}

**八、选择结构语句**

**⒈ if(表达式) 语句1; else 语句2;**

如果if语句的圆括号内的表达式值为非0，则执行语句1，值为0则执行语句2。

Ⅰ. 表达式可为任意表达式，if语句执行的实质是判断表达式的值是否为0来决定执行语句1还是语句2。

Ⅱ. 语句1和语句2都只能是一个语句，若要跟多条语句，切记用一对{}括起来，构成复合语句。

Ⅲ. if语句的三种结构

1. 单边： if(表达式) 语句。

⑵ 双边：if(表达式) 语句1; else 语句2;

1. 多层(重点掌握)：

if(表达式1) 语句1;

else if(表达式2) 语句2;

else if(表达式3) 语句3;

…

else 语句n;

**⒉ 条件运算符 表达式1?表达式2:表达式3**

若表达式1的值非0，则取表达式2的值作为整个表达式的值，否则取表达式3的值为整个表达式的值。如 3>4? 1:2 该表达式的值为2。

**⒊ switch语句**

switch(表达式)

{

case 表达式1：语句

case 表达式2：语句

…

case 表达式n：语句

default: 语句

}

语句执行过程：先计算表达式的值，然后判断该值与表达式1到表达式n中的哪个相等，若与表达式i的值相等，则执行表 达式i后的所有语句，当遇到break；语句时结束整个switch语句的执行。

**循环结构**

1. 三种循环结构：

for()； while()； do- while()三种。

① for循环当中必须是两个分号，千万不要忘记。

② 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。

③ do-while()循环的最后一个while();的分号一定不能够丢。，do－while循环是至少执行一次循环。

㈡ 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就退出整个一层循环。

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。

㈢ 嵌套循环

**九、数组**

**⒈ 定义格式：数据类型 数组名[整型常量]**；如 int a[10]; 定义了一个整型数组，数组名为a，这个数组含有10个元素。

**⒉ 引用数组元素：数组名[下标]** 切记下标值从0开始。下标可为常量，表达式，变量等，如int i=3； a[0]=5；a[3\*2]=9； a[i]=7。

**⒊ 初始化：数据类型 数组名[整型常量]={数据表列}**；将数据表列的各个值依次赋值给数组的各个元素。如int a[5]={0,1,2,3,4}；则数组a各元素a[0]到a[4]的值分别为0，1，2，3，4。

**⒋ 整型数组**

int a[10]; 整型数组里的各个元素存放的是整数。a [3] =3;

**⒌ 字符型数组**

char str[20]；字符型数组里的各个元素存放的是字符。 str[3]=‘A’。

**十、字符串函数**

**⒈ gets(字符数组名或字符指针变量)；**

如char str[10],\* str2; str2=str;则gets(str)；或gets(str2)；都是接收用户输入的字符串如“ABC”存入到字符数组str中

**⒉ puts(字符数组名或字符指针变量或字符串常量)；**

如char str[10]=“china”;char \*str2;str=str2; 则puts(str); 或puts(str2); 或 puts(“china”);都会在屏幕上打印出 china。

**⒊ strlen(字符数组名或字符指针变量);字符串测长函数**

char str[20]=“hello world!”;

int len; len=strlen(str);得出的结果是len的值为12

**⒋ strcat(字符串1的地址，字符串2的地址)；**

将字符串2的内容连接到字符串1的尾部。

char str1[20]= “ABC”,str2[20]= “xyz”;

strcat (str1, str2);

则程序运行的结果是str1内存放的字符串变为ABCxyz，当然str2存放的字符串还是xyz。

**⒌ strcmp(字符串1的地址，字符串2的地址)；**

比较串1和串2哪个比较大。

**⒍ strcpy(字符串1的地址，字符串2的地址)；**

将字符串2的内容复制到字符串1内。

char str1 [20] =“ABC”,str2[20]= “xyz”;

strcpy(str1,str2);此时str1的内容为“xyz”，当然str2的内容没变。

strcpy(str1,“uvw”);此时str1的内容又变成了“uvw”。

注意：要用上述str函数时必须包含头文件“string.h”