**C语言学习总结**

**目录**

**[1. VS2022初始代码模板创建 2](#_Toc23759)**

**[2. 数据类型（1个字节 = 8个比特位） 2](#_Toc5188)**

**[3. 局部变量与全局变量命名注意事项 2](#_Toc5979)**

**[4. printf()时的转意字符’\’ 2](#_Toc26403)**

**[6. ASCII码对照表 3](#_Toc11349)**

**[7. 数据在内存中以补码形式存储 3](#_Toc16396)**

**[8. static的用法 4](#_Toc6193)**

**[9. #define MAX(X,Y) (X>Y?X:Y) 的用法 4](#_Toc24155)**

**[10. 如何修改结构体中的数组成员 4](#_Toc18110)**

**[11. if 和 else的匹配机制（注意格式） 4](#_Toc10463)**

**[12. 文件的结束标志 4](#_Toc3280)**

**[13. for循环参数注意事项 4](#_Toc28016)**

**[14. 求1！+2！+……+10！ 5](#_Toc25574)**

**[15. 求数组中元素的个数 5](#_Toc15989)**

**[16. 以十六进制输出一个整数 5](#_Toc8227)**

**[17. scanf()函数注意事项 5](#_Toc25971)**

**[18. 循环中更新的变量不能使用浮点数 6](#_Toc23337)**

**[19. 静态变量不能跨函数使用 6](#_Toc23307)**

1. **VS2022初始代码模板创建**

①桌面右键VS，点击属性查看文件位置

②找到对应位置，进入“VC”文件夹，再进入“VCProjectItems”文件夹

③用notepad++打开newc++file.cpp文件

④将代码模板输入进去保存即可

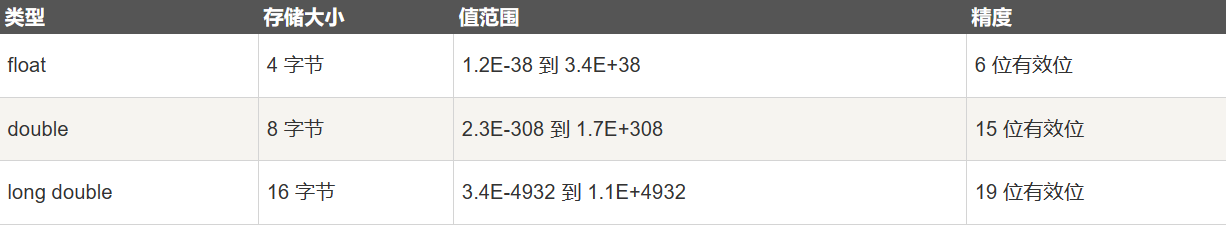
注：notepad++若保存不成功，点击使用管理员模式运行。

1. **数据类型（1个字节 = 8个比特位）**

①整数类型



②浮点类型



1. **局部变量与全局变量命名注意事项**

局部变量和全局变量的名字建议不要相同，容易产生误会或bug。二者名字相同时，局部变量优先。

1. **printf()时的转意字符’\’**

printf(“%d\n”,strlen(“c:\test\32\test.c”)); 结果为：13

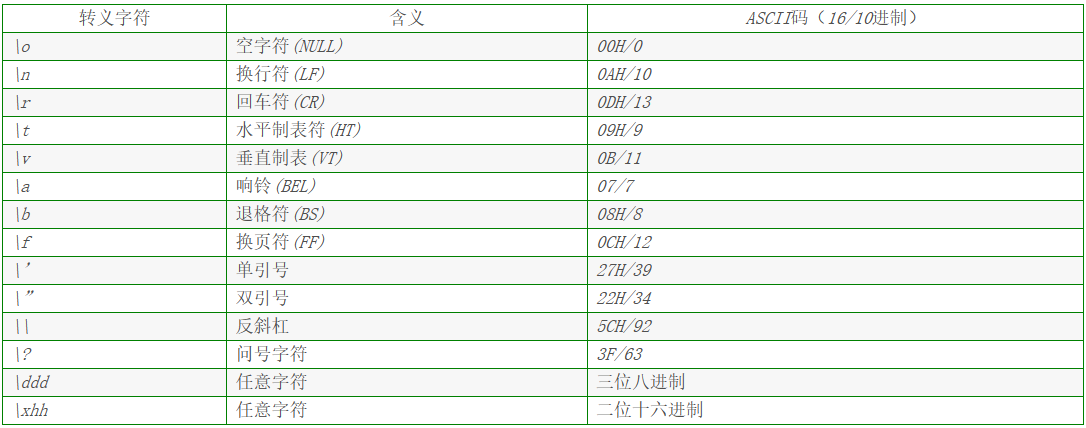
注：①\t：转义字符，计算字符串长度时算作一个字符。

②\32：此时32是8进制数。表示两个8进制数字对应的十进制数作为ASCII码值的对应字符，所以算作一个字符。即：8进制数：32 **=** 十进制数：26 **=** 字符：“->”。

③\ddd：ddd表示1~3个八进制数字。如：\130等。

④\xdd：dd表示2个十六进制数字。如：\x30。

1. **转意字符汇总**



1. **ASCII码对照表**



1. **数据在内存中以补码形式存储**

下面结果为：-1

int a = 0;

int b = ~a;

printf("%d",b);

注：①’~’:按位取反

②a:00000000/00000000/00000000/00000000

b:11111111/11111111/11111111/11111111

b为有符号的整数，在内存中以二进制的补码形式存储。最高位为符号位：1表示负数，0表示正数。printf()打印时打印的是原码，即10000000/00000000/00000000/00000001，十进制对应的就是-1。

③负数的十六进制表示：正数的原码、反码、补码相同；

负数的值：最高位表示符号位，其余位取反后加一即为该负数的绝对值。如下列输出结果为：-6

int main()

{

int i = 0xFFFFFFFa;

printf("%d\n", i);

return 0;

}

0xFFFFFFFa最高位为1，表示该十六进制数为负数，其余位为：111 1111 1111 1111 1111 1111 1111 1010，

取反加一：000 0000 0000 0000 0000 0000 0000 0110，即6（十进制值），所以0xFFFFFFFa的值为-6.

1. **static的用法**

①**修饰局部变量时**：局部变量的生命周期变长。

②**修饰全局变量时**：此全局变量只能在当前源文件内部使用，其他文件就无法调用了。

③**修饰函数时**：改变了函数的链接属性（外部链接->内部链接）。和全局变量类似，被修饰后的函数 只能在当前源文件内使用，其他文件无法调用。

1. **#define MAX(X,Y) (X>Y?X:Y) 的用法**

max = MAX(a,b) 等价于 max = (a>b?a:b)

1. **如何修改结构体中的数组成员**

修改结构体中数组成员，使用库函数<string.h>中的strcpy( , )函数

struct BOOK

{

char name[20];

};

struct BOOK b1 = {"C语言设计"};

strcpy(b1.name, "C++");

1. **if 和 else的匹配机制（注意格式）**

以下结果什么都没打印

int a = 0;

int b = 2;

if (a == 1)

if (b == 2)

printf("hehe\n");

else

printf("haha\n");

注：else和离得最近的if进行匹配。

1. **文件的结束标志**

文件结束标志：EOF，键盘输入Ctrl + Z 即可输入EOF。

1. **for循环参数注意事项**

for循环的初始化、调整和判断都可以省略，但判断部分被省略时，判断结果恒为真。

for (;;)

{

printf("hehe\n");

}

结果为死循环

1. **求1！+2！+……+10！**

int ret = 1;

int i = 0;

int n = 0;

int sum = 0;

for (i = 1; i <= 10; i++)

{

ret \*= i;

sum += ret;

}

1. **求数组中元素的个数**

int num1, num2 = 0;

int arr1[] = {1,2,3,4,5,6,7};

char arr2[] = "hello world";

num1 = sizeof(arr1) / sizeof(arr1[0]);

num2 = sizeof(arr2) / sizeof(arr2[0]) - 1; //等价于num4 = strlen(arr2);

printf("两数组的元素个数为：%d,%d",num1,num2); //结果是7和11

return 0;

**注**：字符串末尾还有一个元素是’\0’，strlen()函数计算字符串长度，遇到\0停止，但长度计算不包含\0， 当strlen()函数计算字符数组如：char arr[]={‘a’,’b’,’c’,’d’}时，结果为随机数，因为strlen()遇到\0 才停止计数。

1. **以十六进制输出一个整数**

int i = 47; //47是十进制数

printf("i = %x\n", i); //输出为2f

printf("i = %X\n", i); //输出为2F

printf("i = %#x\n", i); //输出为0x2f

printf("i = %#X\n", i); //输出为0x2F（推荐）

return 0;

1. **scanf()函数注意事项**

int i;

scanf("a%d", &i);

printf("i = %d\n", i);

return 0;

注：当键盘输入123时，输出错误，当键盘输入a123时，输出：i = 123（正确），当键盘输入a123mn时，输出：i = 123（正确），即scanf（）中非输入控制符（如a）必须原样输入。再举个例子：

int i, j, k;

scanf("%d, %d, %d", &i, &j, &k);

printf("i = %d, j = %d, k = %d\n", i, j, k);

return 0;

当键盘输入：1 2 3时，结果为i = 1, j = -858993460, k = -858993460，当键盘输入：1，2，3时，结果为i = 1, j = 2, k = 3。综上，使用scanf()编写代码规范如下：

①使用scanf()之前先使用printf()提示用户输入什么样的信息；

②scanf()中尽量不使用非输入控制符；

③应该编写代码对用户的非法输入做适当的处理。

如while ((ch = getchar()) != '\n')

continue;

1. **循环中更新的变量不能使用浮点数**

下列输出结果为：99.900002

float i = 99.9;

printf("%f", i);

注：float和double都不能保证可以精确的存储一个小数，只是近似值，所以在循环中更新的变量i（如 for(i = 0; i < 5; i ++)中的i）不能用浮点数格式。

1. **静态变量不能跨函数使用**

下列程序编译器没有报错，但逻辑上是有问题的：当f()函数调用完成后，变量i的内存被释放了，所以使用printf()函数打印\*p，即变量i的值时，存在内存访问越界。

void f(int\*\* q)

{

int i = 5;

\*q = &i;

}

int main()

{

int\* p;

f(&p);

printf("%d\n", \*p);

return 0;

}

可使用动态内存实现上述功能，将f()改写如下：

void f(int\*\* q)

{

\*q = (int\*)malloc(sizeof(int)); //malloc()声明包含在头文件<malloc.h>中

\*\*q = 5;

}

1. **分配内存和释放内存**

**分配内存**：操作系统将某一块内存空间的使用权分配给程序。

**释放内存**：操作系统将已经分配给程序的内存空间的使用权收回。

**注**：释放内存并没有把该内存的内容清空。

1. 鹏哥视频观看到P9