**C语言总结20150114**

**--------------------------------------------------------------------**

**常量**

在程序运行过程中，其值不能被改变的量称为常量

分为字面常量也叫直接常量和符号常量两种

字面常量如：0 ，-3；(整形常量) 4.6(实型常量) 等

符号常量：#define P 3.14

**--------------------------------------------------------------------**

**整型常量的表示**

二进制: 0b1101;

八进制:0123 (等于10进制 1\*8\*8+2\*8+3) ； -011表示八进制-11，即使进制-9

十进制：123

十六进制：0x123

**--------------------------------------------------------------------**

**强制类型转换运算符**

一般形式：(类型名)(表达式)

**--------------------------------------------------------------------**

**逗号运算符和逗号表达式**

逗号表达式一般形式：表达式1，表达式2

求解过程：先求解表达式1，再求解表达式2，逗号表达式值为表达式2的值

扩展：表达式1，表达式2，表达式3，...

**--------------------------------------------------------------------**

**printf()输出格式**

%d:十进制带符号整数

%u:十进制无符号整数

%o:八进制无符号整数

%x:十六进制无符号整数

%p:输出地址

**--------------------------------------------------------------------**

**条件运算符**

一般形式：表达式1 ? 表达式2 ： 表达式3

求解过程：表达式1为真则整个表达式值为表达式2，反之为表达式3

**--------------------------------------------------------------------**

**switch语句**

一般形式：

switch(表达式)

{

case 常量表达式1：xxx;break;

default:xxxx;

}

**--------------------------------------------------------------------**

**goto语句**

一般形式：goto 语句标号;

注：语句标号命名同变量一样

**--------------------------------------------------------------------**

**for循环**

一般形式：for(表达式1，表达式2，表达式3) {语句；}

求解过程：先求表达式1，再求表达式2，若为真，执行语句，再求表达式3，再求表达式2...

**--------------------------------------------------------------------**

**break语句**

一般形式：break;

注：break只能用于循环语句(跳出本层循环体)和switch语句

**--------------------------------------------------------------------**

**continue语句**

一般性式：continue;

用法：结束本次循环，进行下一次循环的判断

**--------------------------------------------------------------------**

**二维数据**

一般形式：类型 数组名[常量表达式][常量表达式];

初始化：int a[3][4] = {1,2,3,4,5,7};或int a[3][4] = {{1,2}，{4,5}，{9}};

**--------------------------------------------------------------------**

**一些字符串处理函数**

strcat(字符数组1名，字符数组2名);

作用：将字符数组2接到字符数组1后面，返回字符数组1地址

strcpy(数组名1，字符串地址2)；

作用:将字符串2复制到数组1中去，包括最后的\0;

strcmp(字符串1，字符串2);

作用:比较字符串1和字符串2

相等返回0；串1大返回正整数，串2大返回负整数

**--------------------------------------------------------------------**

**strlen函数和sizeof**

strlen(字符数组名)；

作用：测试字符串长度，不包含\0；

sizeof(字符数组名、类新名、指针)；

字符数组名: 测试字符数组的大小

类型名: 测试类新所占字节大小

指针: 4字节

**--------------------------------------------------------------------**

**函数定义与声明**

函数的声明和函数的定义中的函数头部基本相同，只差一个分号

加上分号的函数头称为函数原型

函数原型一般形式：

函数返回值类型 函数名(参数类型1，参数类型2)；

函数返回值类型 函数名(参数类型1 参数1名，参数类型2 参数2名)；

**--------------------------------------------------------------------**

**编译过程**

c语言提供的预处理功能主要有以下3种

宏定义、文件包含、条件编译

注：编译过程分为：

**预编译**(宏替换，文件包含，条件编译，去注释生成.i文件)

**编译**(语法词法检查生成.S文件)

**汇编**(将汇编代码转换成二进制代码.o文件)

**链接**(生成可执行文件)

**--------------------------------------------------------------------**

**宏定义**

分为带参和不带参

不带参一般形式

#define 标识符 字符串

用标识符代替字符串，标识符也叫宏名

宏名的有效范围为定义命令之后到本源文件结束，或者用”#undef 宏名”终止宏名作用域

带参的一般形式

#define 宏名(参数表) 字符串

如：#define S(a,b) a\*b

各参数最好用括号括起来，#define S(a\*b) (a)\*(b)

**--------------------------------------------------------------------**

**文件包含**

文件包含即一个源文件将另外一个源文件全部内容包含进来

用#include命令来实现文件包含操作

一般形式：#include <文件名> 或 #include “文件名”

文件可为.h文件也可为.c文件

<>表示到到存放C库函数的头文件目录中寻找包含文件，””表示先按其中的给出路径查找，如找不到再到C库函数头文件目录中寻找，如#include “/etc/wang.h”

**--------------------------------------------------------------------**

**条件编译**

有以下几种形式

#ifdef 标识符

…1

#else

…2

#endif

作用：若标识符已被#define命令定义过，甚至类似这种 “#define 标识符”，则编译1

#ifndef 标识符

…3

#else

…4

#endif

作用:同上面的相反

#if 表达式

…5

#else

…6

#endif

作用：如果表达式为真，则编译5

**--------------------------------------------------------------------**

**位运算**

位运算符包含六个

&按位与 |按位或 ^按位异或(相异为真)

~取反 <<左移 >>右移

位运算只能用于整形或字符型数据

**--------------------------------------------------------------------**

**static extern**

C语言中每一个变量和函数都有两个属性

1：数据类型

2：存储类别

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\* 变量 \*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

对于存储类别，分为静态存储和动态存储

static 修饰的变量存于静态存储区

static 修饰局部变量，则局部变量整个运行期间都存在，且只初始化一次，存在于静态存储区

static 修饰全局变量，则全局变量作用范围只能在本源文件，别的源文件可用此全局变量名定义新的变量

extern 是对全局变量作用范围的扩展

extern 扩展本源文件中的全局变量到本源文件，在文件开头[extern 变量名]，则作用范围扩展到整个本源文件

extern 扩展本源文件中的全局变量到整个工程文件，哪个文件需要，只要在文件中[extern 变量名]就可以

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\* 函数 \*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

函数本质上是全局的，但是可以指定函数不能被其他源文件调用，根据函数是否能被其他源文件调用，将函数分为内部函数和外部函数

内部函数(静态函数)：在定义函数时，在函数名和函数类型前加static即可，此函数只在本源文件有作用，其他源文件可有重名函数

外部函数：在定义函数时，在函数名和函数类型前加extern,则表示此函数是外部函数，可供其他文件调用，在其他文件中要调用此函数，用extern对函数进行声明[extern 函数原型]即可

**--------------------------------------------------------------------**

**const**

1:const修饰的变量是只读不可写,所以最好初始化

2:在另一连接文件中引用const常量,extern const int i;

const 和指针

指向常量的指针 const char \*p;//指针指向单元不可变(可以保护指针所指向单元)，但指针可指向新单元

常指针 char \* const p = 地址;//指针指向单元，但单元内容可变

指向常量的常指针 const char \* const p =地址;//指针指向单元不可变，单元内容不可变

**--------------------------------------------------------------------**

**typedef**

typedef的语法规则其实很简单，一句话来说就是定义对象的语法前加关键字typedef，剩下的不变，原本定义的对象标识符换成类型标识符，对应语义从**定义一个对象改成定义一个类型别名**。

如

函数原型：void(\* signal(int signum,void(\*handler)(int)) )(int);

用typedef声明新的类型名

typedef void (\*RP)(int);

RP signal(int signum,RP handler);

**--------------------------------------------------------------------**

**枚举**

若一个变量只可能有几个值,可定义为枚举类型

声明枚举类型

enum weekday{sun,mon,tue,wed,thu,fri,sat}; //sun等称为枚举元素,枚举元素是常量

enum weekday{sun=7,mon=1,tue,wed,thu,fri,sat}; //sun等称为枚举元素,枚举元素是常量

定义枚举变量

weekday oneday;

赋值

oneday=sat;