编译警告信息表

0	Macro ‘name’ redefined (宏'name'被重新定义)	用#define 定义的符号被用于不同的参数或形式表重新定义
1	Macro formal parameter ‘name’ is never refer—enced (宏形式参数'name'被重新定义	#defing 形式参数从未在参数字符串中出现
2	Macro ‘name’ is already#undef (宏'name'被重新定义)	#undef 被用于不是宏观的符号
3	Macro ‘name’ called with empty parameter(s) (用空参数调用宏'name')	用零长度（zero-length）的参数不清调用在#define 中定义的参数化宏观
4	Macro ‘name’ is called recursively;not expanded （宏'name'被递归调用；不扩展）	递归宏调用使预处理器停止该宏的进一步扩展
5	Undefined symbol ‘name’ in #if or #elif;assumed zero (在#if 或#elif 中未定义的符号'name';假设为零)	在#if 或#elif 表达式中把非宏符号作为零来处理被认为是错误的编程实践。使用以下两个中任一个：#ifdef symbol 或#if defined（symbol）
6	Unkown escape seguence (‘/c’);assume'c' （未知的转义序列(‘/c’)；假设为'c'）	在字符常数中发现反斜杠（\）或字符文字后随未知的转义字符
7	Nested comment found without using the ‘c’ option （发现嵌套的注释未使用‘c’选项）	在注释中发现字符序列/*且被忽略
8	Invalid type-specifier for field;assumed “int” (无效的域类型识别符；假设为（整型）“int”)	在此执行中，位域仅能被指定为 int 或 unsigned int（无符号整型）
9	Undeclared function parameter ‘name’; assumed “int” (未声明的函数参数'name';假设为“int”)	缺省情况下，K&R 函数定义头部中未声明的识别符被给予类型 int（整型）
10	Dimension of array ignored:array assumed pointer （忽略数组的元素数；数组假设为指针）	具有显示元素数的数组被规定为形式参数，编译器把它作为指向对象的指针来处理
11	Storage class “static” ignored;'name' declared “extern” (存储类"static"被忽略；'name'被声明为"extern")	对象或函数首先被声明为 extern（外部）（显示地或缺省），后来被声明为 static (静态)。静态声明被忽略
12	Incompletely bracketed initializer (括号不完整的初始化)	为了避免模糊，初始化应当只使用一层{}括号或者被{}括号完整地包围
13	Unreferenced label ‘name’ (未引用的标号'name')	括号被定义但从未被引用

14	Type specifier missing;assumed “int” (遗漏了类型指定符；假设为“int”)	在声明中未给出类型定义符-假设 int（整型）
15	Wrong usage of string operator(‘#” or ‘##′);igno- -red (字符串运算符（′#′′##′）错误使用；被忽略) 此外，#运算符必须先于形式参数： #define mac(p1) #p1 /*Becomes “p1″*/ #define mac(p1,p2) p1+p2##add this /*Merged p2*/	这限制了#和##运算符用于参数化宏的标记域（token-field）
16	Non-void function:″return″with <expression>; expected (非空函数：用<表达式>返回)	在所有地方非空（non-void）函数定义应当用已定义的返回值退出
17	Invalid storage class for function;assumed to be “extern” (无效的函数存储类别；假设为″extern″)	无效的函数存储类别一被忽略。有效的类别是 extern,static，或 typedef
18	Redeclared parameter′s storage class (重新声明的参数存储类别)	在后续的声明/定义中函数形式参数的存储类别从 register 变为 auto，或从 auto 变为 register。
19	Storage class “extern” ignored;′name′ was first declared as “static” (存储类″extern″被忽略；′na--me′首先被声明为″static″	被声明为 static（静态）的识别符后来被显示或隐含地声明为 extern（外部），extern（外部）声明被忽略
20	Unreachable statement(s) (不能到达的语句)例如：Break; I=2;/*never executed*/	无条件转移或返回在一条或多条语句之前，使得这条或多条语句从来不会被执行。
21	Unreachable statement(s) at unreachable label ‘na- -me′ (在未卜先知被引用的标号′name′处不能到达的语句) 例如： Break; Here; I=2;/*never executed*/	无条件转移或返回在有标号的一条或多条语句之前，但是标号从未被引用，所以这条或多余语句从来不会被执行
22	Non-void function:explicit “return” <expression>; Expected (非空函数；显示的“return”〈表达式〉；被预期)	非空（non-void）函数产生隐含返回。这可能是从循环或开关语句中非预期的退出。注意：不带 default 的开关语句总是被编译器当作‘可退出的’而不管 case 的结构如何
23	Undeclared function ‘name′;assumed “extern″″int″ （未声明的函数‘name’；假设为“extern″″int″）	对未声明函数的引用导致使用缺省的声明。函数被假设为具有 K&R 类型，具有外部存储类别，并返回 int（整型）
24	Static memeory option converts local “auto”	用于静态存储器分配的命令行选

	or ""register" to "static" (静态存储器选项把局部"auto" 或"register"转换为"static")	项使 auto(自动)和 register（寄存器）声明被当作 static（静态）来处理
25	Inconsistent use of K&R function-varying number of parameters (K&R 函数的不一致使用改变了参数的数目)	用改变了的参数类型调用 K&R 函数
26	Inconsistent use of K&R function-changing type of parameters (K&R 函数的不一致使用改变了参数的类型 例如： myfunc(34);/*int argument*/ myfunc(34,6);/*float argument*/	用改变了的参数类型调用 K&R 函数
27	Size of "extern" object 'name' is unkown (外部对象'name'的大小未知)	Extern（外部）数组应当用 size 声明
28	Constant [index] outside array bound (常数[索引]超出数组边界)	存在超出已声明数组边界的常数索引
29	Hexadecimal escape sequence larger than "char" (十六进制转义序列大于"char")	转义序列被截断以适合于放入 char（字符）中

30	Attribute ignored(属性被忽略)
	例子：
	const struct s
	{
	…
	};/*no object declared.const ignored - warning*/
	const int myfunc(void);
	/*function returning const int - warning*/
	const int (*fp)(void);
	/*pointer to function returning
	const int - warnig*/
	int (*const fp)(void);
	/*const pointer to function returning int - ok
	no warning*/
		因为 const（常量）或 volatile（易失的）是对象的属性，所以当它们与 structure（结构），union（联合），或 enumeration（枚举）标签定义一起给出时讲被忽略，上述定义没有和对象同时声明。而且，函数被认为不能返回 const 或 volatile

31	Incompatible parameters of K&R functions(K&R 函数的参数不兼容) 在下列范围之一使用指针: pointer - pointer, expression ? ptr : ptr, pointer equality_op pointer pointer = pointer formal parameter vs actual parameter	指向函数的指针（可能是间接的）或 K&R 函数声明具有不兼容的参数类型
32	Incompatible numbers parameters of K&R	指向函数的指针（可能是间接的）

	functions (K&R 函数的参数数目不兼容)在下列范围之一使用指针: pointer - pointer, expression ? ptr : ptr, pointer equality_op pointer pointer = pointer formal parameter vs actual parameter	或 K&R 函数声明具有不兼容的参数类型
33	Local or formal ‘name’ was never referenced (局部或形式参数'name'从未被引用)	在函数定义中未使用形式参数或局部变量对象。
34	Non-printable character ‘/xhh’ found in literal or character constant (在文字或字符常数中发现不可打印字符号‘/xhh’)	在字符串文字或字符常数中使用不可打印（non-printable）字符被认为是一种不好的编程习惯。为了得到同样的结果可使用'\0xhhh
35	Old-style(K&R)type of function declarator (老式(K&R)类型的函数声明)	发现老式(K&R)函数声明。只有正在使用-gA 选项时才发出这种警告
36	Floating point constant out of range (浮点常数超出范围)	浮点值太大或太小以致不能使用目标的浮点系统来表示
37	Illegal float operation :division by zero not allowed (非法浮点运算：不允许除以零)	在常数算术运算时发现除零
38	Tag identifier ‘name’ was never defined (从未定义标签识别符‘name’)	
39	Dummy statement.Optimized away!	发现多余的代码。这通常表示用户代码中打印错误或可能产生于使用有点不太通用的宏时（这不是错误）。例如：a+b;

40	Possible bug!"if"statement terminated
	(可能是缺陷!“if”语句被中止)
		这通常表示用户代码中的代码错误。例如：
		if(a==b);
		{
		<if body>
		}

41	Possible bug!Unintialized variable
	(可能是缺陷! 未初始化的变量)
		在初始化之前使用变量（变量具有随机值）。
		例如：
		void func(p1)
		short a;
		p1+=a;
		}

42	Possible bug!Integer promotion may cause probl--ems.Use cast to avoid it	整数提升规则指出所有整数运算必须产生这样的结果：当它们具

	(可能是缺陷！整数提升可能产生问题。使用 cast 以避免此问题)
	例如:
	short tst(unsigned char a)
	{
	if (-a)
	return (1);
	else
	return (-1);
	}
	此例将始终返回 1，即使对于数值 0xff 也是如此，其原因是整数提升首先使变量 a 变为 0x00ff，然后执行位非（bit not)。
	整数提升被许多其他 C 编译器所忽略，因此当用 IAR 系统编译器重新编译已有的程序时，可能产生此警告.
		有比 int（整型）低的精度时，就好像它们是 int（整型）一样。这有时可能导致未预期的结果

43	Possible bug!Single ‘=’ instead of ‘==’ used in “if” Statement
	(可能是缺陷！在 “if”语句中用 ‘==’代替单个 ‘=’)
		这通常表示用户代码中的打印错误例如:。
		if (a=1)
		{
		<if body>
		}

44	Redundant expression. Example:multiply with 1.Add with 0（ 多余的表达式。例如：乘以 1，加上 0）	这可能表示用户代码中的打印错误，但是它也是可能是，有 case 工具产生的错误代码的结果。
45	Possible bug!Strange or faulty expression.Example: division by zero (可能是缺陷！奇怪或错误的表达式。)例如：除以零)	这通常表示用户代码中的缺陷
46	Unreachable code deleted by the global optimizer (由全局优化删除不能到达的代码) 例如：除以零)	在用户代码中多余的代码块。它可能是 bug（缺陷）的结果，但通常仅是不完善代码的信号
47	Unreachable returns.The function will never return (不能到达的返回，函数将永远不返回)	函数将永远不能返回到调用的函数。这可能是程序缺陷的结果，但通常当在 RTOS 系统中具有永不结束循环时产生。

48	Unsigned compare always true/false
	(无符号的比较总是为真/假)
		这表示用户代码中的缺陷！通常的原因是遗漏了-c 编译器开关
		例如：
		for (uc=10;uc>=uc--);
		{
		<loop body>
		}

		因为无符号的值永远大于或等于零，所以这是永不结束的循环。
49	Signed compare always true/false (有符号的比较总是为真/假)	这表示用户代码中的缺陷!