

The chart below is based on `au-misra2.lnt`, our configuration file for MISRA C 2004. Although we have taken great care in creating this chart, we make no guarantees as to its completeness and accuracy. If you find any discrepancies or errors, please send an email to “sales” at “gimpel.com” with the details.

	Checked	Partially Checked	Not Statically Checkable	Not Yet Checked	Totals
Required Rules	115	1	5	1	122
Advisory Rules	17	1	2	0	20
Totals	132	2	7	1	142

Rule	Required	Advisory	Checked	Description
1.1	req		Yes	strict ANSI checking
1.2	req		Yes	no undefined or unspecified behavior
1.3	req		n/a	not statically checkable
1.4	req		Yes	identifier names must be unique in first 31 characters
1.5		adv	n/a	not statically checkable
2.1	req		Yes	no in-line assembly language
2.2	req		Yes	do not use // comments
2.3	req		Yes	no nested comments
2.4		adv	partial	no nested comments
3.1	req		n/a	not statically checkable
3.2	req		n/a	not statically checkable
3.3		adv	n/a	not statically checkable
3.4	req		Yes	report on pragma use
3.5	req		n/a	not statically checkable
3.6	req		Yes	libraries should conform to MISRA standard
4.1	req		Yes	do not use non-ANSI escape sequences
4.2	req		Yes	do not use trigraphs
5.1	req		Yes	identifier names must be unique in first 31 characters
5.2	req		Yes	complain about name hiding
5.3	req		Yes	do not reuse typedef names
5.4	req		Yes	do not reuse tag names
5.5		adv	Yes	complain about name hiding
5.6		adv	Yes	complain about name hiding
5.7		adv	Yes	do not reuse identifier names
6.1	req		Yes	disallowed use of plain char
6.2	req		Yes	disallowed use of non-numeric value
6.3		adv	Yes	do not use modifiers (int, char, etc) outside of a typedef
6.4	req		Yes	bitfield type should be unsigned int or signed int
6.5	req		Yes	small bit field should be unsigned int

Rule	Required	Advisory	Checked	Description
7.1	req		Yes	no octal constants
8.1	req		Yes	use function prototypes
8.2	req		Yes	function has no explicit type
8.3	req		Yes	declaration and definition should have identical parameter types
8.4	req		Yes	symbol redeclared
8.5	req		Yes	no definitions of objects or functions in a header file
8.6	req		Yes	function not declared at file scope
8.7	req		Yes	could define variable at block scope
8.8	req		Yes	object/function previously declared
8.9	req		Yes	symbol previously defined
8.10	req		Yes	symbol could be made static
8.11	req		Yes	symbol previously used as static
8.12	req		Yes	array has 0 dimension
9.1	req		Yes	symbol not initialized
9.2	req		Yes	omitted braces within an initializer
9.3	req		Yes	initialize either all enum members or only the first
10.1	req		Yes	loss of precision / possible loss of fraction
10.2	req		Yes	prototype coercion
10.3	req		Yes	prohibited cast of complex integer expression
10.4	req		Yes	prohibited cast of complex floating point
10.5	req		Yes	recasting required for << and - operators
10.6	req		Yes	unsigned integer literals require a 'U' suffix
11.1	req		Yes	cast pointer to non-pointer
11.2	req		Yes	cast pointer to non-pointer
11.3		adv	Yes	cast pointer to non-pointer
11.4		adv	Yes	cast pointer to pointer
11.5	req		Yes	attempt to cast away const/volatile from a pointer or reference
12.1		adv	Yes	dependence on C's operator precedence
12.2	req		Yes	order of evaluation
12.3	req		Yes	'sizeof' used on expressions with side effect
12.4	req		Yes	side effects on right hand side of logical operator
12.5	req		Yes	non-primary expression used with logical operator
12.6		adv	Yes	boolean expression required for logical operator
12.7	req		Yes	bitwise operator applied to signed underlying type
12.8	req		Yes	excessive shift value
12.9	req		Yes	unary minus applied to unsigned
12.10	req		Yes	comma operator used outside of 'for' expression
12.11		adv	Yes	overflow in computing constant
12.12	req		Yes	bit representation of a floating point type used
12.13		adv	Yes	increment or decrement combined with another operator
13.1	req		Yes	boolean test of assignment
13.2		adv	Yes	boolean test of assignment
13.3	req		partial	testing floats for equality

Rule	Required	Advisory	Checked	Description
13.4	req		Yes	floating point variable used as loop counter
13.5	req		Yes	'for' loop expressions do not match
13.6	req		Yes	detect loop variables modified within the loop
13.7	req		Yes	constant value Boolean
14.1	req		Yes	unreachable code
14.2	req		Yes	non-null statement has no effect
14.3	req		Yes	null statement not on line by itself
14.4	req		Yes	do not use goto
14.5	req		Yes	do not use continue
14.6	req		Yes	more than one 'break' terminates loop
14.7	req		Yes	return before function end
14.8	req		Yes	use braces around body of switch, else, for, do, while
14.9	req		Yes	use braces around body of if, else, else if
14.10	req		Yes	no 'else' at end of 'if ... else if' chain
15.0	req		Yes	use MISRA switch syntax
15.1	req		Yes	need a switch
15.2	req		Yes	use break to terminate case
15.3	req		Yes	case label follows default in switch statement
15.4	req		Yes	boolean value in switch statement
15.5	req		Yes	switch does not have a case
16.1	req		Yes	function has variable number of arguments
16.2	req		Yes	do not use recursive functions
16.3	req		Yes	use identifiers for all parameters in a prototype
16.4	req		Yes	function parameter list differs from prior declaration
16.5	req		Yes	function declaration is missing type info
16.6	req		Yes	too few or too many arguments for prototype
16.7		adv	Yes	use const on paramaters where appropriate
16.8	req		Yes	function should return a value
16.9	req		Yes	function identifier used without '&' or parenthesized parameter list
16.10	req		Yes	ignoring return value of function
17.1	req		Yes	pointer arithmetic used on non-array
17.2	req		Yes	relational or subtract operator applied to pointers
17.3	req		Yes	relational or subtract operator applied to pointers.
17.4	req		Yes	pointer arithmetic by increment or decrement
17.5		adv	Yes	more than two pointer indirection levels used
17.6	req		Yes	assigning address of auto to outer scope symbol or to static
18.1	req		Yes	vacuous type for variable
18.2	req		no	not currently supported
18.3	req		n/a	not statically checkable
18.4	req		Yes	unions shall not be used
19.1		adv	Yes	only preprocessor statements and comments before '#include'
19.2		adv	Yes	header file name with non-standard character
19.3	req		Yes	need < or " after #include

Rule	Required	Advisory	Checked	Description
19.4	req		Yes	expression-like macro not parenthesized
19.5	req		Yes	'#define / #undef ' used within a block
19.6	req		Yes	do not use '#undef '
19.7		adv	Yes	use function instead of function-like macro
19.8	req		Yes	syntax error in call of macro
19.9	req		Yes	preprocessor directives in macro invocation
19.10	req		Yes	expression-like macro not parenthesized
19.11	req		Yes	undefined preprocessor variable
19.12	req		Yes	multiple use of # or ## operators in macro definition
19.13		adv	Yes	do not use of # or ## operators in macro definition
19.14	req		Yes	non-standard use of 'defined' preprocessor operator
19.15	req		Yes	repeated include file
19.16	req		Yes	#endif or #else is not followed by EOL
19.17	req		Yes	#if not closed within file
20.1	req		Yes	do not redefine standard functions
20.2	req		Yes	re-use of reserved identifier
20.3	req		Yes	calls to standard library functions are checked
20.4	req		Yes	do not use dynamic heap allocation
20.5	req		Yes	do not use errno
20.6	req		Yes	do not use offsetof
20.7	req		Yes	do not use longjmp, setjmp
20.8	req		Yes	do not use signal, raise
20.9	req		Yes	do not use stdio.h
20.10	req		Yes	do not use atof, atio, atoll
20.11	req		Yes	do not use abort, exit, getenv, system
20.12	req		Yes	do not use time, strftime, clock, difftime, mktime
21.1	req		Yes	static analysis by using PC-lint/FlexeLint