Polyspace Bug Finder

以下工程的编码标准报告: MISRA\_RULE\_JDSN

报告作者: lehuahu

Polyspace Bug Finder: 以下工程的编码标准报告: MISRA\_RULE\_JDSN

by 报告作者: lehuahu

Published 10-12月-2024 8:53:14

分析作者: lehuahu  
Polyspace 版本: Polyspace Bug Finder 23.2 (R2023b Update 9)  
工程版本: 1.0  
  
  
  
  
  
  
结果文件夹:  
C:\workspace\Polyspace\_Workspace\MISRA\_RULE\_JDSN\BF\_Result\_1

目錄

[第1章MISRA C:2004 编码标准 1](#_Toc184713249)

[**MISRA C:2004 编码标准摘要 - 按文件显示的违规** 1](#_Toc184713250)

[**MISRA C:2004 编码标准摘要 - 按规则显示的违规** 6](#_Toc184713251)

[**MISRA C:2004 编码标准 摘要(所有文件)** 9](#_Toc184713252)

[**MISRA C:2004 编码标准 摘要(已启用规则)** 14](#_Toc184713253)

[**MISRA C:2004 编码标准 违规** 21](#_Toc184713254)

[第2章附录 1 - 配置设置 152](#_Toc184713255)

[**Polyspace 设置** 152](#_Toc184713256)

[**编码标准配置** 153](#_Toc184713257)

[**具有编译错误的文件(未完整分析的文件)** 162](#_Toc184713258)

[第3章附录 2 - 定义 163](#_Toc184713259)

第1章MISRA C:2004 编码标准

MISRA C:2004 编码标准摘要 - 按文件显示的违规









MISRA C:2004 编码标准摘要 - 按规则显示的违规









MISRA C:2004 编码标准 摘要(所有文件)

| **文件** | **合计** |
| --- | --- |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_10\_1.c | 11 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_10\_2.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_1.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_10.c | 19 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_11.c | 12 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_12.c | 13 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_13.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_14.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_15.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_16.c | 13 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_2.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_3.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_4.c | 15 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_5.c | 12 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_6.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_7.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_8.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_9.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_1.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_2.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_3.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_4.c | 6 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_5.c | 15 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_6.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_7.c | 25 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_8.c | 14 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_9.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_1.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_2.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_3.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_4.c | 13 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_5.c | 4 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_1.c | 4 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_2.c | 19 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_3.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_4.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_5.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_6.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_7.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_8.c | 16 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_1.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_2.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_3.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_4.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_1.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_2.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_3.c | 17 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_4.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_1.c | 2 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_10.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_11.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_12.c | 2 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_13.c | 6 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_2.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_3.c | 14 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_4.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_5.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_6.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_7.c | 13 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_8.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_9.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_18\_1.c | 6 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_18\_2.c | 11 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_1\_1.c | 6 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_1\_2.c | 6 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_1\_3.c | 4 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_1.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_2.c | 4 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_3.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_4.c | 3 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_3\_1.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_3\_2.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_4\_1.c | 3 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_4\_2.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_4\_4.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_1.c | 4 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_2.c | 4 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_3.c | 3 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_4.c | 3 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_6\_1.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_1.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_10.c | 10 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_2.c | 11 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_3.c | 4 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_4.c | 6 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_5.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_5\_a.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_6.c | 16 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_7.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_8.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_9.c | 9 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_8\_1.c | 7 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_8\_2.c | 15 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_8\_3.c | 3 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_1.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_2.c | 20 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_3.c | 5 |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_4.c | 8 |
| C:\workspace\Polyspace\_Workspace\194 C Code\Rule194.h | 249 |
| C:\workspace\Polyspace\_Workspace\194 C Code\Rule194\_main.c | 98 |
| **合计** | **1198** |

MISRA C:2004 编码标准 摘要(已启用规则)

| **规则** | **描述** | **模式** | **合计** |
| --- | --- | --- | --- |
| 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. | required | 16 |
| 2.1 | Assembly language shall be encapsulated and isolated. | required | 0 |
| 2.2 | Source code shall only use /\* ... \*/ style comments. | required | 402 |
| 2.3 | The character sequence /\* shall not be used within a comment. | required | 0 |
| 2.4 | Sections of code should not be 'commented out'. | advisory | 0 |
| 3.4 | All uses of the #pragma directive shall be documented and explained. | required | 1 |
| 4.1 | Only those escape sequences which are defined in the ISO C standard shall be used. | required | 1 |
| 4.2 | Trigraphs shall not be used. | required | 1 |
| 5.1 | Identifiers (internal and external) shall not rely on the significance of more than 31 characters. | required | 0 |
| 5.2 | Identifiers in an inner scope shall not use the same name as an identifier in an outer scope, and therefore hide that identifier. | required | 1 |
| 5.3 | A typedef name shall be a unique identifier. | required | 0 |
| 5.4 | A tag name shall be a unique identifier. | required | 1 |
| 5.5 | No object or function identifier with static storage duration should be reused. | advisory | 1 |
| 5.6 | No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure and union member names. | advisory | 0 |
| 5.7 | No identifier name should be reused. | advisory | 6 |
| 6.1 | The plain char type shall be used only for the storage and use of character values. | required | 0 |
| 6.2 | Signed and unsigned char type shall be used only for the storage and use of numeric values. | required | 0 |
| 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. | advisory | 170 |
| 6.4 | Bit fields shall only be defined to be of type unsigned int or signed int. | required | 2 |
| 6.5 | Bit fields of type signed int shall be at least 2 bits long. | required | 1 |
| 7.1 | Octal constants (other than zero) and octal escape sequences shall not be used. | required | 2 |
| 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. | required | 90 |
| 8.2 | Whenever an object or function is declared or defined, its type shall be explicitly stated. | required | 0 |
| 8.3 | For each function parameter the type given in the declaration and definition shall be identical, and the return types shall also be identical. | required | 1 |
| 8.4 | If objects or functions are declared more than once their types shall be compatible. | required | 2 |
| 8.5 | There shall be no definitions of objects or functions in a header file. | required | 1 |
| 8.6 | Functions shall be declared at file scope. | required | 0 |
| 8.7 | Objects shall be defined at block scope if they are only accessed from within a single function. | required | 0 |
| 8.8 | An external object or function shall be declared in one and only one file. | required | 7 |
| 8.9 | An identifier with external linkage shall have exactly one external definition. | required | 3 |
| 8.10 | All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required. | required | 4 |
| 8.11 | The static storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage. | required | 0 |
| 8.12 | When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation. | required | 2 |
| 9.1 | All automatic variables shall have been assigned a value before being used. | required | 1 |
| 9.2 | Braces shall be used to indicate and match the structure in the non-zero initialisation of arrays and structures. | required | 0 |
| 9.3 | In an enumerator list, the '=' construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised. | required | 1 |
| 10.1 | The value of an expression of integer type shall not be implicitly converted to a different underlying type. | required | 3 |
| 10.2 | The value of an expression of floating type shall not be implicitly converted to a different type. | required | 5 |
| 10.3 | The value of a complex expression of integer type may only be cast to a type that is narrower and of the same signedness as the underlying type of the expression. | required | 0 |
| 10.4 | The value of a complex expression of floating type may only be cast to narrower floating type. | required | 0 |
| 10.5 | If the bitwise operator ~ and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediatly cast to the underlying type of the operand. | required | 0 |
| 10.6 | The 'U' suffix shall be applied to all constants of unsigned types. | required | 0 |
| 11.1 | Conversion shall not be performed between a pointer to a function and any type other than an integral type. | required | 1 |
| 11.2 | Conversion shall not be performed between a pointer to an object and any type other than an integral type, another pointer to object type or a pointer to void. | required | 1 |
| 11.3 | A cast should not be performed between a pointer type and an integral type. | advisory | 3 |
| 11.4 | A cast should not be performed between a pointer to object type and a different pointer to object type. | advisory | 2 |
| 11.5 | A cast shall not be performed that removes any const or volatile qualification from the type addressed by a pointer. | required | 1 |
| 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | advisory | 7 |
| 12.2 | The value of an expression shall be the same under any order of evaluation that the standard permits. | required | 2 |
| 12.3 | The sizeof operator should not be used on expressions that contain side effects. | required | 0 |
| 12.4 | The right hand operand of a logical && or || operator shall not contain side effects. | required | 1 |
| 12.5 | The operands of a logical && or || shall be primary-expressions. | required | 2 |
| 12.6 | The operands of a logical operators (&&, || and !) should be effectively Boolean. Expressions that are effectively Boolean should not be used as operands to operators other than (&&, || and !). | advisory | 2 |
| 12.7 | Bitwise operators shall not be applied to operands whose underlying type is signed. | required | 3 |
| 12.8 | The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand. | required | 1 |
| 12.9 | The unary minus operator shall not be applied to an expression whose underlying type is unsigned. | required | 1 |
| 12.10 | The comma operator shall not be used. | required | 2 |
| 12.11 | Evaluation of constant unsigned integer expressions should not lead to wrap-around. | advisory | 0 |
| 12.12 | The underlying bit representations of floating-point values shall not be used. | required | 3 |
| 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | advisory | 6 |
| 13.1 | Assignment operators shall not be used in expressions that yield a Boolean value. | required | 1 |
| 13.2 | Tests of a value against zero should be made explicit, unless the operand is effectively Boolean. | advisory | 2 |
| 13.3 | Floating-point expressions shall not be tested for equality or inequality. | required | 1 |
| 13.4 | The controlling expression of a for statement shall not contain any objects of floating type. | required | 1 |
| 13.5 | The three expressions of a for statement shall be concerned only with loop control. | required | 1 |
| 13.6 | Numeric variables being used within a for loop for iteration counting should not be modified in the body of the loop. | required | 1 |
| 13.7 | Boolean operations whose results are invariant shall not be permitted. | required | 9 |
| 14.1 | There shall be no unreachable code. | required | 8 |
| 14.2 | All non-null statements shall either have at least one side effect however executed, or cause control flow to change. | required | 1 |
| 14.3 | Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment provided that the first character following the null statement is white-space character. | required | 2 |
| 14.4 | The goto statement shall not be used. | required | 1 |
| 14.5 | The continue statement shall not be used. | required | 0 |
| 14.6 | For any iteration statement there shall be at most one break statement used for loop termination. | required | 1 |
| 14.7 | A function shall have a single point of exit at the end of the function. | required | 5 |
| 14.8 | The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement. | required | 3 |
| 14.9 | An if (expression) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement. | required | 2 |
| 14.10 | All if ... else if constructs should contain a final else clause. | required | 1 |
| 15.0 | A switch statement shall conform to MISRA-C syntax. | required | 0 |
| 15.1 | A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement. | required | 0 |
| 15.2 | An unconditional break statement shall terminate every non-empty switch clause. | required | 3 |
| 15.3 | The final clause of a switch statement shall be the default clause. | required | 2 |
| 15.4 | A switch expression should not represent a value that is effectively Boolean. | required | 1 |
| 15.5 | Every switch statement shall have at least one case clause. | required | 1 |
| 16.1 | Functions shall not be defined with variable numbers of arguments. | required | 1 |
| 16.2 | Functions shall not call themselves, either directly or indirectly. | required | 3 |
| 16.3 | Identifiers shall be given for all of the parameters in a function prototype declaration. | required | 2 |
| 16.4 | The identifiers used in the declaration and definition of a function shall be identical. | required | 1 |
| 16.5 | Functions with no parameters shall be declared with parameter type void. | required | 184 |
| 16.6 | The number of arguments passed to a function shall match the number of parameters. | required | 0 |
| 16.7 | A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object. | advisory | 0 |
| 16.8 | All exit paths from a function with non-void return type shall have an explicit return statement with an expression. | required | 0 |
| 16.9 | A function identifier shall only be used with either a preceding &, or with a parenthesised parameter list, which may be empty. | required | 1 |
| 16.10 | If a function returns error information, then that error information should be tested. | required | 81 |
| 17.1 | Pointer arithmetic shall only be applied to pointers that address an array or array element. | required | 0 |
| 17.2 | Pointer subtraction shall only be applied to pointers that address elements of the same array. | required | 1 |
| 17.3 | >, >=, <, <= shall not be applied to pointer types except where they point to the same array. | required | 0 |
| 17.4 | Array indexing shall be the only allowed form of pointer arithmetic. | required | 3 |
| 17.5 | The declaration of objects should contain no more than 2 levels of pointer indirection. | advisory | 0 |
| 17.6 | The address of an object with automatic storage shall not be assigned to an object that may persist after the object has ceased to exist. | required | 0 |
| 18.1 | All structure or union types shall be complete at the end of a translation unit. | required | 6 |
| 18.2 | An object shall not be assigned to an overlapping object. | required | 0 |
| 18.4 | Unions shall not be used. | required | 4 |
| 19.1 | #include statements in a file shall only be preceded by other pre-processor directives or comments. | advisory | 0 |
| 19.2 | Non-standard characters should not occur in header file names in #include directives. | advisory | 0 |
| 19.3 | The #include directive shall be followed by either a <filename> or "filename" sequence. | required | 0 |
| 19.4 | C macros shall only expand to a braced initialiser, a constant, a parenthesised expression,a type qualifier, a storage class specifier, or a do-while-zero construct. | required | 5 |
| 19.5 | Macros shall not be #define'd and #undef'd within a block. | required | 3 |
| 19.6 | #undef shall not be used. | required | 2 |
| 19.7 | A function should be used in preference to a macro. | advisory | 6 |
| 19.8 | A function-like macro shall not be invoked without all of its arguments. | required | 1 |
| 19.9 | Arguments to a function-like macro shall not contain tokens that look like pre-processing directives. | required | 1 |
| 19.10 | In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. | required | 2 |
| 19.11 | All macro identifiers in preprocessor directives shall be defined before use, except in #ifdef and #ifndef preprocessor directives and the defined() operator. | required | 1 |
| 19.12 | There shall be at most one occurrence of the # or ## pre-processor operators in a single macro definition. | required | 1 |
| 19.13 | The # and ## preprocessor operators should not be used. | advisory | 3 |
| 19.14 | The defined pre-processor operator shall only be used in one of the two standard forms. | required | 0 |
| 19.15 | Precautions shall be taken in order to prevent the contents of a header file being included twice. | required | 0 |
| 19.16 | Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor. | required | 0 |
| 19.17 | All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if or #ifdef directive to which they are related. | required | 0 |
| 20.1 | Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined. | required | 2 |
| 20.2 | The names of standard library macros, objects and functions shall not be reused. | required | 0 |
| 20.3 | The validity of values passed to library functions shall be checked. | required | 0 |
| 20.4 | Dynamic heap memory allocation shall not be used. | required | 2 |
| 20.5 | The error indicator errno shall not be used. | required | 0 |
| 20.6 | The macro offsetof, in library <stddef.h>, shall not be used. | required | 0 |
| 20.7 | The setjmp macro and the longjmp function shall not be used. | required | 0 |
| 20.8 | The signal handling facilities of <signal.h> shall not be used. | required | 0 |
| 20.9 | The input/output library <stdio.h> shall not be used in production code. | required | 67 |
| 20.10 | The library functions atof, atoi and atol from library <stdlib.h> shall not be used. | required | 0 |
| 20.11 | The library functions abort, exit, getenv and system from library <stdlib.h> shall not be used. | required | 0 |
| 20.12 | The time handling functions of library <time.h> shall not be used. | required | 0 |
| 21.1 | Minimisation of run-time failures shall be ensured by the use of at least one tool/technique. | required | 5 |
| **合计** |  |  | **1198** |

MISRA C:2004 编码标准 违规

Table 1.1. C:\workspace\Polyspace\_Workspace\194 C Code\R\_10\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 375 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1430 | 8.10 | All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required. Function 'R\_10\_1\_function' should have internal linkage | 文件范围 | Unset | Unreviewed |  |
| 356 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 688 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 321 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 519 | 16.5 | Functions with no parameters shall be declared with parameter type void. | R\_10\_1() | Unset | Unreviewed |  |
| 783 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_10\_1() | Unset | Unreviewed |  |
| 462 | 11.1 | Conversion shall not be performed between a pointer to a function and any type other than an integral type. | R\_10\_1() | Unset | Unreviewed |  |
| 499 | 11.2 | Conversion shall not be performed between a pointer to an object and any type other than an integral type, another pointer to object type or a pointer to void. | R\_10\_1() | Unset | Unreviewed |  |
| 596 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_10\_1() | Unset | Unreviewed |  |
| 776 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.2. C:\workspace\Polyspace\_Workspace\194 C Code\R\_10\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 463 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 558 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1018 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_10\_2() | Unset | Unreviewed |  |
| 690 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_10\_2() | Unset | Unreviewed |  |
| 546 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_10\_2() | Unset | Unreviewed |  |
| 889 | 11.5 | A cast shall not be performed that removes any const or volatile qualification from the type addressed by a pointer. Cast from type 'const int \*' to type 'int \*' removes qualifiers. | R\_10\_2() | Unset | Unreviewed |  |
| 448 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_10\_2() | Unset | Unreviewed |  |
| 1407 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 349 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.3. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 554 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 550 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 549 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_1() | Unset | Unreviewed |  |
| 876 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_1() | Unset | Unreviewed |  |
| 551 | 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | R\_11\_1() | Unset | Unreviewed |  |
| 607 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1358 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_1() | Unset | Unreviewed |  |
| 1144 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_1() | Unset | Unreviewed |  |

Table 1.4. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_10.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 589 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1081 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1190 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_10() | Unset | Unreviewed |  |
| 1140 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1067 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_10() | Unset | Unreviewed |  |
| 572 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_10() | Unset | Unreviewed |  |
| 487 | 12.12 | The underlying bit representations of floating-point values shall not be used. | R\_11\_10() | Unset | Unreviewed |  |
| 557 | 11.4 | A cast should not be performed between a pointer to object type and a different pointer to object type. | R\_11\_10() | Unset | Unreviewed |  |
| 569 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_10() | Unset | Unreviewed |  |
| 555 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_10() | Unset | Unreviewed |  |
| 343 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_10() | Unset | Unreviewed |  |
| 563 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_10() | Unset | Unreviewed |  |
| 567 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 468 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 961 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_10() | Unset | Unreviewed |  |
| 459 | 11.4 | A cast should not be performed between a pointer to object type and a different pointer to object type. | R\_11\_10() | Unset | Unreviewed |  |
| 565 | 12.12 | The underlying bit representations of floating-point values shall not be used. | R\_11\_10() | Unset | Unreviewed |  |
| 672 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_10() | Unset | Unreviewed |  |
| 579 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_10() | Unset | Unreviewed |  |

Table 1.5. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_11.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 514 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1007 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 627 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_11() | Unset | Unreviewed |  |
| 580 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_11() | Unset | Unreviewed |  |
| 1412 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1437 | 14.1 | There shall be no unreachable code. If-condition always evaluates to true. Dead branch from line 12 to line 14. | R\_11\_11() | Unset | Unreviewed |  |
| 1438 | 13.7 | Boolean operations whose results are invariant shall not be permitted. If-condition always evaluates to true. Dead branch from line 12 to line 14. | R\_11\_11() | Unset | Unreviewed |  |
| 678 | 13.1 | Assignment operators shall not be used in expressions that yield a Boolean value. | R\_11\_11() | Unset | Unreviewed |  |
| 384 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_11() | Unset | Unreviewed |  |
| 1300 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_11() | Unset | Unreviewed |  |
| 277 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_11() | Unset | Unreviewed |  |
| 874 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_11() | Unset | Unreviewed |  |

Table 1.6. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_12.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1058 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 587 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1015 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_12() | Unset | Unreviewed |  |
| 592 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_12() | Unset | Unreviewed |  |
| 1439 | 14.1 | There shall be no unreachable code. If-condition always evaluates to true. Dead branch from line 10 to line 12. | R\_11\_12() | Unset | Unreviewed |  |
| 1440 | 13.7 | Boolean operations whose results are invariant shall not be permitted. If-condition always evaluates to true. Dead branch from line 10 to line 12. | R\_11\_12() | Unset | Unreviewed |  |
| 585 | 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | R\_11\_12() | Unset | Unreviewed |  |
| 392 | 13.3 | Floating-point expressions shall not be tested for equality or inequality. | R\_11\_12() | Unset | Unreviewed |  |
| 855 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 680 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_12() | Unset | Unreviewed |  |
| 1388 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_12() | Unset | Unreviewed |  |
| 445 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_12() | Unset | Unreviewed |  |
| 1191 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_12() | Unset | Unreviewed |  |

Table 1.7. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_13.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 602 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 603 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1037 | 13.4 | The controlling expression of a for statement shall not contain any objects of floating type. The loop-counter 'i' shall not have floating type. | R\_11\_13() | Unset | Unreviewed |  |
| 597 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. ANSI C90 forbids mixed declarations and code. | R\_11\_13() | Unset | Unreviewed |  |
| 658 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_13() | Unset | Unreviewed |  |
| 422 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 599 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_13() | Unset | Unreviewed |  |
| 822 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_13() | Unset | Unreviewed |  |

Table 1.8. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_14.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 829 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 608 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 312 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_14() | Unset | Unreviewed |  |
| 458 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 513 | 13.5 | The three expressions of a for statement shall be concerned only with loop control. Third expression: There should be no persistent side effect other than modifying the loop counter (i) | R\_11\_14() | Unset | Unreviewed |  |
| 610 | 12.10 | The comma operator shall not be used. | R\_11\_14() | Unset | Unreviewed |  |
| 570 | 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | R\_11\_14() | Unset | Unreviewed |  |
| 600 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 699 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_14() | Unset | Unreviewed |  |
| 856 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_14() | Unset | Unreviewed |  |

Table 1.9. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_15.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1202 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1230 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 478 | 13.6 | Numeric variables being used within a for loop for iteration counting should not be modified in the body of the loop. Loop counter (i) should not be modified in the body of the loop | R\_11\_15() | Unset | Unreviewed |  |
| 615 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_15() | Unset | Unreviewed |  |
| 617 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. ANSI C90 forbids mixed declarations and code. | R\_11\_15() | Unset | Unreviewed |  |
| 624 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_15() | Unset | Unreviewed |  |
| 613 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_15() | Unset | Unreviewed |  |
| 1343 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1332 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.10. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_16.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 800 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1069 | 20.1 | Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined. The macro 'true' shall not be redefined. | 文件范围 | Unset | Unreviewed |  |
| 1181 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 621 | 14.7 | A function shall have a single point of exit at the end of the function. | R\_11\_16() | Unset | Unreviewed |  |
| 636 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 622 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_16() | Unset | Unreviewed |  |
| 630 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_16() | Unset | Unreviewed |  |
| 1050 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1441 | 14.1 | There shall be no unreachable code. If-condition always evaluates to true. Dead branch from line 11 to line 13. | R\_11\_16() | Unset | Unreviewed |  |
| 1442 | 13.7 | Boolean operations whose results are invariant shall not be permitted. If-condition always evaluates to true. Dead branch from line 11 to line 13. | R\_11\_16() | Unset | Unreviewed |  |
| 626 | 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | R\_11\_16() | Unset | Unreviewed |  |
| 591 | 12.6 | The operands of a logical operators (&&, || and !) should be effectively Boolean. Expressions that are effectively Boolean should not be used as operands to operators other than (&&, || and !). Right operand of logical operator '&&' should be effectively Boolean. | R\_11\_16() | Unset | Unreviewed |  |
| 1249 | 12.5 | The operands of a logical && or || shall be primary-expressions. | R\_11\_16() | Unset | Unreviewed |  |

Table 1.11. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 464 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 641 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 650 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_2() | Unset | Unreviewed |  |
| 1292 | 12.2 | The value of an expression shall be the same under any order of evaluation that the standard permits. Variable a is written and reused within this expression. | R\_11\_2() | Unset | Unreviewed |  |
| 488 | 12.2 | The value of an expression shall be the same under any order of evaluation that the standard permits. Variable a is written and reused within this expression. | R\_11\_2() | Unset | Unreviewed |  |
| 319 | 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | R\_11\_2() | Unset | Unreviewed |  |
| 536 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1159 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_2() | Unset | Unreviewed |  |
| 1216 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_2() | Unset | Unreviewed |  |

Table 1.12. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 872 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 653 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 723 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_3() | Unset | Unreviewed |  |
| 1350 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_3() | Unset | Unreviewed |  |
| 359 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_3() | Unset | Unreviewed |  |
| 998 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Expression must be a modifiable lvalue | R\_11\_3() | Unset | Unreviewed |  |
| 362 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.13. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 659 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 663 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 920 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_4() | Unset | Unreviewed |  |
| 661 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_4() | Unset | Unreviewed |  |
| 972 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1443 | 13.7 | Boolean operations whose results are invariant shall not be permitted. Unnecessary code, if-condition is always true. | R\_11\_4() | Unset | Unreviewed |  |
| 1213 | 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | R\_11\_4() | Unset | Unreviewed |  |
| 801 | 12.5 | The operands of a logical && or || shall be primary-expressions. | R\_11\_4() | Unset | Unreviewed |  |
| 981 | 12.4 | The right hand operand of a logical && or || operator shall not contain side effects. | R\_11\_4() | Unset | Unreviewed |  |
| 656 | 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | R\_11\_4() | Unset | Unreviewed |  |
| 340 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 474 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1011 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1115 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_4() | Unset | Unreviewed |  |
| 1091 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_4() | Unset | Unreviewed |  |

Table 1.14. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_5.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 363 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 664 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 940 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_5() | Unset | Unreviewed |  |
| 477 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1444 | 14.1 | There shall be no unreachable code. If-condition always evaluates to false. Dead branch from line 9 to line 11. | R\_11\_5() | Unset | Unreviewed |  |
| 1445 | 13.7 | Boolean operations whose results are invariant shall not be permitted. If-condition always evaluates to false. Dead branch from line 9 to line 11. | R\_11\_5() | Unset | Unreviewed |  |
| 1257 | 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | R\_11\_5() | Unset | Unreviewed |  |
| 1229 | 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | R\_11\_5() | Unset | Unreviewed |  |
| 1399 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_5() | Unset | Unreviewed |  |
| 1045 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_5() | Unset | Unreviewed |  |
| 668 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_5() | Unset | Unreviewed |  |
| 297 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_5() | Unset | Unreviewed |  |

Table 1.15. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_6.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 673 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 674 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 308 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_6() | Unset | Unreviewed |  |
| 913 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_6() | Unset | Unreviewed |  |
| 284 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 670 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_6() | Unset | Unreviewed |  |
| 1296 | 12.7 | Bitwise operators shall not be applied to operands whose underlying type is signed. Bitwise & on two operands of signed underlying types 'signed int' and 'signed int'. | R\_11\_6() | Unset | Unreviewed |  |
| 518 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 390 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_6() | Unset | Unreviewed |  |
| 1276 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_6() | Unset | Unreviewed |  |

Table 1.16. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_7.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 681 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 742 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 675 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_7() | Unset | Unreviewed |  |
| 679 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_7() | Unset | Unreviewed |  |
| 531 | 12.8 | The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand. | R\_11\_7() | Unset | Unreviewed |  |
| 677 | 12.7 | Bitwise operators shall not be applied to operands whose underlying type is signed. Bitwise << on the left hand operand of signed underlying type 'signed int'. | R\_11\_7() | Unset | Unreviewed |  |
| 1455 | 21.1 | Minimisation of run-time failures shall be ensured by the use of at least one tool/technique. Right operand of shift operation is negative or larger than the number of bits in the left operand data type. Additional Info: Expected values: Right operand must be in [0 .. 31]. Actual values: 32. Risk: If a number is shifted more than the size of the destination type, the behavior is undefined. Fix: Before a shift operation, handle negative and large values of right operand. Alternatively, check previous operations that lead to negative or large values. Or, simply use a larger data type for the left operand. | R\_11\_7() | Unset | Unreviewed |  |
| 634 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.17. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_8.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 926 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 492 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 682 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_8() | Unset | Unreviewed |  |
| 266 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_8() | Unset | Unreviewed |  |
| 753 | 12.9 | The unary minus operator shall not be applied to an expression whose underlying type is unsigned. | R\_11\_8() | Unset | Unreviewed |  |
| 1457 | 21.1 | Minimisation of run-time failures shall be ensured by the use of at least one tool/technique. Conversion from type int 32 bits to type unsigned int 32 bits might overflow. Additional Info: Expected values: [0 .. 2^32-1]. Actual values: -10. Risk: Truncation or wrap-around of value to fit destination type might lead to unexpected results. Fix: Ensure that the destination type is larger than or same as the source type. | R\_11\_8() | Unset | Unreviewed |  |
| 548 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1364 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_8() | Unset | Unreviewed |  |
| 1338 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_8() | Unset | Unreviewed |  |

Table 1.18. C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_9.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 504 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 689 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 730 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_9() | Unset | Unreviewed |  |
| 609 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_11\_9() | Unset | Unreviewed |  |
| 648 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 694 | 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | R\_11\_9() | Unset | Unreviewed |  |
| 841 | 12.10 | The comma operator shall not be used. | R\_11\_9() | Unset | Unreviewed |  |
| 902 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 286 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_11\_9() | Unset | Unreviewed |  |
| 612 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_11\_9() | Unset | Unreviewed |  |

Table 1.19. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 772 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1056 | 14.7 | A function shall have a single point of exit at the end of the function. | R\_12\_1() | Unset | Unreviewed |  |
| 620 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 697 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 279 | 14.1 | There shall be no unreachable code. Statement is unreachable | R\_12\_1() | Unset | Unreviewed |  |
| 1217 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_1() | Unset | Unreviewed |  |
| 1041 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_1() | Unset | Unreviewed |  |

Table 1.20. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 662 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 470 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 642 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_12\_2() | Unset | Unreviewed |  |
| 1114 | 14.2 | All non-null statements shall either have at least one side effect however executed, or cause control flow to change. Expression has no effect | R\_12\_2() | Unset | Unreviewed |  |
| 628 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.21. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 701 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1417 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 705 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 711 | 14.3 | Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment provided that the first character following the null statement is white-space character. A null statement shall appear on a line by itself. | 文件范围 | Unset | Unreviewed |  |
| 706 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1224 | 14.3 | Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment provided that the first character following the null statement is white-space character. A null statement shall appear on a line by itself. | 文件范围 | Unset | Unreviewed |  |
| 796 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.22. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 958 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 708 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 867 | 14.4 | The goto statement shall not be used. | R\_12\_4() | Unset | Unreviewed |  |
| 365 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 471 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_4() | Unset | Unreviewed |  |
| 566 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_4() | Unset | Unreviewed |  |

Table 1.23. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_5.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 817 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1218 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 625 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_12\_5() | Unset | Unreviewed |  |
| 713 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. ANSI C90 forbids mixed declarations and code. | R\_12\_5() | Unset | Unreviewed |  |
| 860 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_5() | Unset | Unreviewed |  |
| 486 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_5() | Unset | Unreviewed |  |
| 1227 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1446 | 14.1 | There shall be no unreachable code. If-condition always evaluates to false. Dead branch from line 11 to line 14. | R\_12\_5() | Unset | Unreviewed |  |
| 1447 | 13.7 | Boolean operations whose results are invariant shall not be permitted. If-condition always evaluates to false. Dead branch from line 11 to line 14. | R\_12\_5() | Unset | Unreviewed |  |
| 716 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_5() | Unset | Unreviewed |  |
| 915 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_5() | Unset | Unreviewed |  |
| 1317 | 14.6 | For any iteration statement there shall be at most one break statement used for loop termination. | R\_12\_5() | Unset | Unreviewed |  |
| 712 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1376 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_5() | Unset | Unreviewed |  |
| 720 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_5() | Unset | Unreviewed |  |

Table 1.24. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_6.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 793 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1329 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 417 | 14.7 | A function shall have a single point of exit at the end of the function. | R\_12\_6() | Unset | Unreviewed |  |
| 1175 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1047 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 760 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 560 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.25. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_7.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1260 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 341 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 749 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_7() | Unset | Unreviewed |  |
| 734 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_7() | Unset | Unreviewed |  |
| 517 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 316 | 15.2 | An unconditional break statement shall terminate every non-empty switch clause. | R\_12\_7() | Unset | Unreviewed |  |
| 1059 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_7() | Unset | Unreviewed |  |
| 446 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_7() | Unset | Unreviewed |  |
| 1411 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 733 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. ANSI C90 forbids mixed declarations and code. | R\_12\_7() | Unset | Unreviewed |  |
| 741 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_12\_7() | Unset | Unreviewed |  |
| 561 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 700 | 14.8 | The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement. The statement forming the body of a while statement shall be a compound statement. | R\_12\_7() | Unset | Unreviewed |  |
| 728 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_7() | Unset | Unreviewed |  |
| 292 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_7() | Unset | Unreviewed |  |
| 888 | 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | R\_12\_7() | Unset | Unreviewed |  |
| 738 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 655 | 14.8 | The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement. The statement forming the body of a for statement shall be a compound statement. | R\_12\_7() | Unset | Unreviewed |  |
| 1116 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_7() | Unset | Unreviewed |  |
| 437 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_7() | Unset | Unreviewed |  |
| 524 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 434 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_7() | Unset | Unreviewed |  |
| 714 | 14.8 | The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement. The statement forming the body of a do ... while statement shall be a compound statement. | R\_12\_7() | Unset | Unreviewed |  |
| 1138 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_7() | Unset | Unreviewed |  |
| 1342 | 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | R\_12\_7() | Unset | Unreviewed |  |

Table 1.26. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_8.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 747 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 288 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 427 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_12\_8() | Unset | Unreviewed |  |
| 746 | 13.2 | Tests of a value against zero should be made explicit, unless the operand is effectively Boolean. | R\_12\_8() | Unset | Unreviewed |  |
| 1082 | 14.9 | An if (expression) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement. An if (expression) construct shall be followed by a compound statement. | R\_12\_8() | Unset | Unreviewed |  |
| 1448 | 14.1 | There shall be no unreachable code. If-condition always evaluates to true. Dead branch at line 10. | R\_12\_8() | Unset | Unreviewed |  |
| 1449 | 13.7 | Boolean operations whose results are invariant shall not be permitted. If-condition always evaluates to true. Dead branch at line 10. | R\_12\_8() | Unset | Unreviewed |  |
| 1111 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_8() | Unset | Unreviewed |  |
| 989 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_8() | Unset | Unreviewed |  |
| 750 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 743 | 14.9 | An if (expression) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement. The else keyword shall be followed by either a compound statement, or another if statement. | R\_12\_8() | Unset | Unreviewed |  |
| 842 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_8() | Unset | Unreviewed |  |
| 878 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_8() | Unset | Unreviewed |  |
| 385 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.27. C:\workspace\Polyspace\_Workspace\194 C Code\R\_12\_9.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 370 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1310 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1139 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_9() | Unset | Unreviewed |  |
| 754 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_9() | Unset | Unreviewed |  |
| 775 | 14.10 | All if ... else if constructs should contain a final else clause. | R\_12\_9() | Unset | Unreviewed |  |
| 588 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_12\_9() | Unset | Unreviewed |  |
| 999 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_12\_9() | Unset | Unreviewed |  |
| 542 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.28. C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 960 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 755 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 758 | 15.3 | The final clause of a switch statement shall be the default clause. | R\_13\_1() | Unset | Unreviewed |  |
| 1057 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_1() | Unset | Unreviewed |  |
| 547 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_1() | Unset | Unreviewed |  |
| 756 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 826 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.29. C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 769 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 770 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 767 | 15.2 | An unconditional break statement shall terminate every non-empty switch clause. | R\_13\_2() | Unset | Unreviewed |  |
| 703 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_2() | Unset | Unreviewed |  |
| 762 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_2() | Unset | Unreviewed |  |
| 777 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1022 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_2() | Unset | Unreviewed |  |
| 921 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_2() | Unset | Unreviewed |  |
| 766 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_2() | Unset | Unreviewed |  |
| 1127 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_2() | Unset | Unreviewed |  |

Table 1.30. C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 731 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 433 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1070 | 15.3 | The final clause of a switch statement shall be the default clause. | R\_13\_3() | Unset | Unreviewed |  |
| 788 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_3() | Unset | Unreviewed |  |
| 1369 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_3() | Unset | Unreviewed |  |
| 779 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_3() | Unset | Unreviewed |  |
| 1125 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_3() | Unset | Unreviewed |  |
| 792 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 522 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_3() | Unset | Unreviewed |  |
| 785 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_3() | Unset | Unreviewed |  |

Table 1.31. C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1404 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 575 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 894 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_13\_4() | Unset | Unreviewed |  |
| 937 | 15.4 | A switch expression should not represent a value that is effectively Boolean. | R\_13\_4() | Unset | Unreviewed |  |
| 639 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 794 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_4() | Unset | Unreviewed |  |
| 643 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_4() | Unset | Unreviewed |  |
| 645 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_4() | Unset | Unreviewed |  |
| 852 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_4() | Unset | Unreviewed |  |
| 877 | 15.2 | An unconditional break statement shall terminate every non-empty switch clause. | R\_13\_4() | Unset | Unreviewed |  |
| 453 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 875 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_13\_4() | Unset | Unreviewed |  |
| 1192 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_13\_4() | Unset | Unreviewed |  |

Table 1.32. C:\workspace\Polyspace\_Workspace\194 C Code\R\_13\_5.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 532 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1196 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1284 | 15.5 | Every switch statement shall have at least one case clause. | R\_13\_5() | Unset | Unreviewed |  |
| 1204 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.33. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 632 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 804 | 16.1 | Functions shall not be defined with variable numbers of arguments. | R\_14\_1() | Unset | Unreviewed |  |
| 353 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 803 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.34. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 818 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 601 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1251 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_2' does not have complete prototype when called (argument list is missing). | R\_14\_2() | Unset | Unreviewed |  |
| 444 | 16.2 | Functions shall not call themselves, either directly or indirectly. Function R\_14\_2 shall not call itself, either directly or indirectly. | R\_14\_2() | Unset | Unreviewed |  |
| 824 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1072 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 809 | 8.8 | An external object or function shall be declared in one and only one file. Function 'R\_14\_2\_a' has external declarations in multiple files | 文件范围 | Unset | Unreviewed |  |
| 1379 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 837 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1198 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_2\_2' has no visible prototype at definition. | 文件范围 | Unset | Unreviewed |  |
| 1421 | 16.2 | Functions shall not call themselves, either directly or indirectly. Function R\_14\_2\_2 is called indirectly by R\_14\_2\_a. | 文件范围 | Unset | Unreviewed |  |
| 1428 | 8.10 | All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required. Function 'R\_14\_2\_2' should have internal linkage | 文件范围 | Unset | Unreviewed |  |
| 810 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1315 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_2\_a' does not have complete prototype when called (argument list is missing). | R\_14\_2\_2() | Unset | Unreviewed |  |
| 827 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1422 | 16.2 | Functions shall not call themselves, either directly or indirectly. Function R\_14\_2\_a is called indirectly by R\_14\_2\_2. | 文件范围 | Unset | Unreviewed |  |
| 815 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 811 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_2\_2' does not have complete prototype when called (argument list is missing). | R\_14\_2\_a() | Unset | Unreviewed |  |
| 978 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.35. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 389 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 836 | 8.8 | An external object or function shall be declared in one and only one file. Function 'R\_14\_3' has external declarations in multiple files | 文件范围 | Unset | Unreviewed |  |
| 512 | 16.3 | Identifiers shall be given for all of the parameters in a function prototype declaration. | 文件范围 | Unset | Unreviewed |  |
| 828 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 805 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 726 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 307 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 833 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.36. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1287 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 787 | 8.8 | An external object or function shall be declared in one and only one file. Function 'R\_14\_4' has external declarations in multiple files | 文件范围 | Unset | Unreviewed |  |
| 1299 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 280 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 795 | 16.4 | The identifiers used in the declaration and definition of a function shall be identical. Parameter number 1 has different names. | 文件范围 | Unset | Unreviewed |  |
| 845 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1066 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 839 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_14\_4() | Unset | Unreviewed |  |
| 485 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_14\_4() | Unset | Unreviewed |  |

Table 1.37. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_5.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 848 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 849 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1121 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1370 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1246 | 10.2 | The value of an expression of floating type shall not be implicitly converted to a different type. Implicit conversion of the expression from 'double' to 'signed int' that is not a wider floating type. | R\_14\_5\_a() | Unset | Unreviewed |  |
| 854 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1095 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 975 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 552 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_14\_5() | Unset | Unreviewed |  |
| 595 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_14\_5() | Unset | Unreviewed |  |

Table 1.38. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_6.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 863 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1002 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 831 | 14.7 | A function shall have a single point of exit at the end of the function. | R\_14\_6() | Unset | Unreviewed |  |
| 935 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_14\_6() | Unset | Unreviewed |  |
| 862 | 13.2 | Tests of a value against zero should be made explicit, unless the operand is effectively Boolean. | R\_14\_6() | Unset | Unreviewed |  |
| 1450 | 14.1 | There shall be no unreachable code. If-condition always evaluates to true. Dead branch from line 10 to line 12. | R\_14\_6() | Unset | Unreviewed |  |
| 1451 | 13.7 | Boolean operations whose results are invariant shall not be permitted. If-condition always evaluates to true. Dead branch from line 10 to line 12. | R\_14\_6() | Unset | Unreviewed |  |
| 866 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 865 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.39. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_7.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 979 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 781 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 545 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1019 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 988 | 16.5 | Functions with no parameters shall be declared with parameter type void. | R\_14\_7() | Unset | Unreviewed |  |
| 1060 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 657 | 16.9 | A function identifier shall only be used with either a preceding &, or with a parenthesised parameter list, which may be empty. Function identifier 'R\_14\_7\_exampleFunction' should be preceded by a '&' or followed by a parameter list. | R\_14\_7() | Unset | Unreviewed |  |
| 1098 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 348 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.40. C:\workspace\Polyspace\_Workspace\194 C Code\R\_14\_8.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 869 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1107 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1166 | 14.7 | A function shall have a single point of exit at the end of the function. | R\_14\_8() | Unset | Unreviewed |  |
| 310 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 893 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1345 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_14\_8() | Unset | Unreviewed |  |
| 583 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1156 | 12.6 | The operands of a logical operators (&&, || and !) should be effectively Boolean. Expressions that are effectively Boolean should not be used as operands to operators other than (&&, || and !). Operand of '!' logical operator should be effectively Boolean. | R\_14\_8() | Unset | Unreviewed |  |
| 1452 | 13.7 | Boolean operations whose results are invariant shall not be permitted. Unnecessary code, if-condition is always true. | R\_14\_8() | Unset | Unreviewed |  |
| 325 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 873 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1320 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1177 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 871 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_14\_8\_a() | Unset | Unreviewed |  |
| 373 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_8' does not have complete prototype when called (argument list is missing). | R\_14\_8\_a() | Unset | Unreviewed |  |
| 1024 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.41. C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 885 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 880 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1294 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_1() | Unset | Unreviewed |  |
| 884 | 11.3 | A cast should not be performed between a pointer type and an integral type. | R\_15\_1() | Unset | Unreviewed |  |
| 879 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_1() | Unset | Unreviewed |  |
| 466 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1219 | 17.4 | Array indexing shall be the only allowed form of pointer arithmetic. | R\_15\_1() | Unset | Unreviewed |  |
| 1122 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.42. C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 604 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 371 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 773 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_2() | Unset | Unreviewed |  |
| 895 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_2() | Unset | Unreviewed |  |
| 890 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_2() | Unset | Unreviewed |  |
| 1453 | 17.2 | Pointer subtraction shall only be applied to pointers that address elements of the same array. Subtracted pointers point to different array objects or have become null. The subtraction might cause undefined behavior. To fix this issue, make sure that both pointers point to elements of the same object, and that they are non-null before the subtraction. | R\_15\_2() | Unset | Unreviewed |  |
| 896 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.43. C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 899 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 744 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1188 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_3() | Unset | Unreviewed |  |
| 1406 | 11.3 | A cast should not be performed between a pointer type and an integral type. | R\_15\_3() | Unset | Unreviewed |  |
| 686 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_3() | Unset | Unreviewed |  |
| 568 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_3() | Unset | Unreviewed |  |
| 857 | 11.3 | A cast should not be performed between a pointer type and an integral type. | R\_15\_3() | Unset | Unreviewed |  |
| 898 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_3() | Unset | Unreviewed |  |
| 381 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 903 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.44. C:\workspace\Polyspace\_Workspace\194 C Code\R\_15\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 907 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1298 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 586 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_4() | Unset | Unreviewed |  |
| 904 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_4() | Unset | Unreviewed |  |
| 1308 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 934 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_15\_4() | Unset | Unreviewed |  |
| 1237 | 17.4 | Array indexing shall be the only allowed form of pointer arithmetic. | R\_15\_4() | Unset | Unreviewed |  |
| 909 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 938 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_15\_4() | Unset | Unreviewed |  |
| 296 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_15\_4() | Unset | Unreviewed |  |

Table 1.45. C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 922 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 635 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 944 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 927 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Incomplete type is not allowed | useR\_16\_1() | Unset | Unreviewed |  |
| 910 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.46. C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1223 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 745 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1142 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 919 | 17.4 | Array indexing shall be the only allowed form of pointer arithmetic. | R\_16\_2() | Unset | Unreviewed |  |
| 413 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 914 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_16\_2() | Unset | Unreviewed |  |
| 918 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_16\_2() | Unset | Unreviewed |  |
| 1289 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1030 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1349 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_16\_2' does not have complete prototype when called (argument list is missing). | R\_16\_2\_a() | Unset | Unreviewed |  |

Table 1.47. C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1023 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1327 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1031 | 18.4 | Unions shall not be used. | R\_16\_3() | Unset | Unreviewed |  |
| 590 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 932 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 924 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_16\_3() | Unset | Unreviewed |  |
| 1096 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_16\_3() | Unset | Unreviewed |  |
| 1225 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 421 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_16\_3() | Unset | Unreviewed |  |
| 970 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_16\_3() | Unset | Unreviewed |  |
| 936 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 618 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_16\_3() | Unset | Unreviewed |  |
| 838 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_16\_3() | Unset | Unreviewed |  |
| 925 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 963 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 740 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 844 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_16\_3' does not have complete prototype when called (argument list is missing). | R\_16\_3\_a() | Unset | Unreviewed |  |

Table 1.48. C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 605 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 710 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 939 | 18.4 | Unions shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 354 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1382 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'double' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 386 | 12.12 | The underlying bit representations of floating-point values shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1032 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 814 | 18.4 | Unions shall not be used. | R\_16\_4\_use\_union() | Unset | Unreviewed |  |
| 263 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.49. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1228 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 912 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.50. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_10.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 577 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1038 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 941 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1274 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 352 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.51. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_11.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1110 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1104 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1141 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 619 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 974 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 451 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 942 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_11' does not have complete prototype when called (argument list is missing). | R\_17\_11\_a() | Unset | Unreviewed |  |
| 748 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.52. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_12.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 784 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1182 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |

Table 1.53. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_13.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 684 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1263 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 951 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 905 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_13\_alternate' has no visible prototype at definition. | 文件范围 | Unset | Unreviewed |  |
| 1393 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1205 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.54. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1297 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 276 | 19.10 | In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. Parameter instance shall be enclosed in parentheses. | 文件范围 | Unset | Unreviewed |  |
| 652 | 19.4 | C macros shall only expand to a braced initialiser, a constant, a parenthesised expression,a type qualifier, a storage class specifier, or a do-while-zero construct. Macro 'INCREMENT' does not expand to compliant construct. | 文件范围 | Unset | Unreviewed |  |
| 1361 | 19.7 | A function should be used in preference to a macro. | 文件范围 | Unset | Unreviewed |  |
| 1079 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 752 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1099 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_17\_2() | Unset | Unreviewed |  |
| 454 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.55. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1250 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 267 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 501 | 19.5 | Macros shall not be #define'd and #undef'd within a block. Macros shall not be #define'd within a block. | 文件范围 | Unset | Unreviewed |  |
| 1172 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_17\_3() | Unset | Unreviewed |  |
| 1286 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_17\_3() | Unset | Unreviewed |  |
| 423 | 19.5 | Macros shall not be #define'd and #undef'd within a block. Macros shall not be #undef'd within a block. | 文件范围 | Unset | Unreviewed |  |
| 534 | 19.6 | #undef shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1269 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1103 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1278 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_3' does not have complete prototype when called (argument list is missing). | R\_17\_3\_a() | Unset | Unreviewed |  |
| 983 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1087 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 553 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_17\_3\_a() | Unset | Unreviewed |  |
| 887 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_17\_3\_a() | Unset | Unreviewed |  |

Table 1.56. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1086 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1063 | 19.6 | #undef shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 476 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1277 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 956 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1306 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_17\_4() | Unset | Unreviewed |  |
| 467 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_17\_4() | Unset | Unreviewed |  |

Table 1.57. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_5.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 968 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 967 | 19.7 | A function should be used in preference to a macro. | 文件范围 | Unset | Unreviewed |  |
| 1242 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1075 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 399 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 594 | 19.8 | A function-like macro shall not be invoked without all of its arguments. macro 'R\_17\_5' used with too few arguments | 文件范围 | Unset | Unreviewed |  |
| 1337 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_17\_5\_a() | Unset | Unreviewed |  |
| 959 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_17\_5\_a() | Unset | Unreviewed |  |
| 774 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Expected an expression | R\_17\_5\_a() | Unset | Unreviewed |  |
| 1194 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.58. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_6.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 718 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 977 | 19.4 | C macros shall only expand to a braced initialiser, a constant, a parenthesised expression,a type qualifier, a storage class specifier, or a do-while-zero construct. Macro 'R\_17\_6' does not expand to compliant construct. | 文件范围 | Unset | Unreviewed |  |
| 802 | 19.7 | A function should be used in preference to a macro. | 文件范围 | Unset | Unreviewed |  |
| 830 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | 文件范围 | Unset | Unreviewed |  |
| 1123 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | 文件范围 | Unset | Unreviewed |  |
| 521 | 19.13 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | 文件范围 | Unset | Unreviewed |  |
| 391 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1232 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1147 | 19.9 | Arguments to a function-like macro shall not contain tokens that look like pre-processing directives. Macro argument shall not look like a preprocessing directive. | 文件范围 | Unset | Unreviewed |  |

Table 1.59. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_7.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 780 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 825 | 19.4 | C macros shall only expand to a braced initialiser, a constant, a parenthesised expression,a type qualifier, a storage class specifier, or a do-while-zero construct. Macro 'R\_17\_7' does not expand to compliant construct. | 文件范围 | Unset | Unreviewed |  |
| 957 | 19.10 | In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. Parameter instance shall be enclosed in parentheses. | 文件范围 | Unset | Unreviewed |  |
| 638 | 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | 文件范围 | Unset | Unreviewed |  |
| 985 | 19.7 | A function should be used in preference to a macro. | 文件范围 | Unset | Unreviewed |  |
| 980 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1054 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | testR\_17\_7() | Unset | Unreviewed |  |
| 1264 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 990 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 782 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | testR\_17\_7() | Unset | Unreviewed |  |
| 702 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 982 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | testR\_17\_7() | Unset | Unreviewed |  |
| 1092 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | testR\_17\_7() | Unset | Unreviewed |  |

Table 1.60. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_8.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1253 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 685 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 751 | 19.11 | All macro identifiers in preprocessor directives shall be defined before use, except in #ifdef and #ifndef preprocessor directives and the defined() operator. 'UNDEFINED\_MACRO' is not defined. | 文件范围 | Unset | Unreviewed |  |
| 1396 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_17\_8() | Unset | Unreviewed |  |
| 631 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_17\_8() | Unset | Unreviewed |  |
| 771 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1215 | 19.5 | Macros shall not be #define'd and #undef'd within a block. Macros shall not be #define'd within a block. | 文件范围 | Unset | Unreviewed |  |

Table 1.61. C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_9.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1165 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 614 | 19.4 | C macros shall only expand to a braced initialiser, a constant, a parenthesised expression,a type qualifier, a storage class specifier, or a do-while-zero construct. Macro 'R\_17\_9' does not expand to compliant construct. | 文件范围 | Unset | Unreviewed |  |
| 807 | 19.7 | A function should be used in preference to a macro. | 文件范围 | Unset | Unreviewed |  |
| 1392 | 19.13 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | 文件范围 | Unset | Unreviewed |  |
| 997 | 19.13 | The # and ## preprocessor operators should not be used. The # and ## preprocessor operators should not be used. | 文件范围 | Unset | Unreviewed |  |
| 1309 | 19.12 | There shall be at most one occurrence of the # or ## pre-processor operators in a single macro definition. More than one occurrence of the # or ## preprocessor operators. | 文件范围 | Unset | Unreviewed |  |
| 995 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1340 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_17\_9\_example() | Unset | Unreviewed |  |
| 882 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.62. C:\workspace\Polyspace\_Workspace\194 C Code\R\_18\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1000 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 330 | 20.1 | Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined. The macro 'malloc' shall not be redefined. | 文件范围 | Unset | Unreviewed |  |
| 1387 | 19.4 | C macros shall only expand to a braced initialiser, a constant, a parenthesised expression,a type qualifier, a storage class specifier, or a do-while-zero construct. Macro 'malloc' does not expand to compliant construct. | 文件范围 | Unset | Unreviewed |  |
| 1005 | 19.7 | A function should be used in preference to a macro. | 文件范围 | Unset | Unreviewed |  |
| 692 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 992 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.63. C:\workspace\Polyspace\_Workspace\194 C Code\R\_18\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1363 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1009 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1314 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1267 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1071 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_18\_2() | Unset | Unreviewed |  |
| 533 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_18\_2() | Unset | Unreviewed |  |
| 1113 | 20.4 | Dynamic heap memory allocation shall not be used. | R\_18\_2() | Unset | Unreviewed |  |
| 757 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_18\_2() | Unset | Unreviewed |  |
| 962 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 420 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 847 | 20.4 | Dynamic heap memory allocation shall not be used. | R\_18\_2() | Unset | Unreviewed |  |

Table 1.64. C:\workspace\Polyspace\_Workspace\194 C Code\R\_1\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 928 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1014 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1234 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1013 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_1\_1() | Unset | Unreviewed |  |
| 1456 | 21.1 | Minimisation of run-time failures shall be ensured by the use of at least one tool/technique. Division by zero. | R\_1\_1() | Unset | Unreviewed |  |
| 424 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.65. C:\workspace\Polyspace\_Workspace\194 C Code\R\_1\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1016 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 908 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1153 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1266 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1021 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_1\_2() | Unset | Unreviewed |  |
| 496 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.66. C:\workspace\Polyspace\_Workspace\194 C Code\R\_1\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1403 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1152 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1252 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1025 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.67. C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1241 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 929 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 649 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1027 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1026 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_2\_1() | Unset | Unreviewed |  |
| 916 | 12.7 | Bitwise operators shall not be applied to operands whose underlying type is signed. Bitwise << on the left hand operand of signed underlying type 'signed int'. | R\_2\_1() | Unset | Unreviewed |  |
| 1458 | 21.1 | Minimisation of run-time failures shall be ensured by the use of at least one tool/technique. Operation << overflows. Additional Info: Expected values: Based on operand types, operation result must be in [-2^31 .. 2^31-1] to fit type int 32 bits. Actual values: x is 2^31-1. Risk: Overflow can result in unexpected values. Fix: Handle large values of operands before the operation or use larger types for operands. | R\_2\_1() | Unset | Unreviewed |  |
| 721 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.68. C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1132 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1341 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 911 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 883 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.69. C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1034 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1106 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1033 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1151 | 3.4 | All uses of the #pragma directive shall be documented and explained. | 文件范围 | Unset | Unreviewed |  |
| 1318 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.70. C:\workspace\Polyspace\_Workspace\194 C Code\R\_2\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 820 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 891 | 6.4 | Bit fields shall only be defined to be of type unsigned int or signed int. | 文件范围 | Unset | Unreviewed |  |
| 326 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.71. C:\workspace\Polyspace\_Workspace\194 C Code\R\_3\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1035 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1207 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1043 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1036 | 4.1 | Only those escape sequences which are defined in the ISO C standard shall be used. \x1b is not an ISO C escape sequence. | R\_3\_1() | Unset | Unreviewed |  |
| 1039 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.72. C:\workspace\Polyspace\_Workspace\194 C Code\R\_3\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1397 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 358 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1378 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 350 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1042 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_3\_2() | Unset | Unreviewed |  |
| 1048 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_3\_2() | Unset | Unreviewed |  |
| 1046 | 4.2 | Trigraphs shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1051 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.73. C:\workspace\Polyspace\_Workspace\194 C Code\R\_4\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1131 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 640 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 986 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.74. C:\workspace\Polyspace\_Workspace\194 C Code\R\_4\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1061 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 735 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1065 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1006 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_4\_2() | Unset | Unreviewed |  |
| 1233 | 5.2 | Identifiers in an inner scope shall not use the same name as an identifier in an outer scope, and therefore hide that identifier. | R\_4\_2() | Unset | Unreviewed |  |
| 1398 | 5.7 | No identifier name should be reused. Identifier 'externalVar' should not be reused. Already used as identifier (R\_4\_2.c line 5). | R\_4\_2() | Unset | Unreviewed |  |
| 671 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.75. C:\workspace\Polyspace\_Workspace\194 C Code\R\_4\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 538 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1384 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 342 | 8.4 | If objects or functions are declared more than once their types shall be compatible. Global declaration of 'myStruct' variable has incompatible type with its definition. | 文件范围 | Unset | Unreviewed |  |
| 414 | 5.7 | No identifier name should be reused. Tag name 'myStruct' should not be reused. Already used as tag name (R\_4\_4.c line 5). | 文件范围 | Unset | Unreviewed |  |
| 616 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. "myStruct" has already been declared in the current scope | 文件范围 | Unset | Unreviewed |  |
| 1077 | 5.4 | A tag name shall be a unique identifier. tag name 'myStruct' should not be reused. already used as tag name (R\_4\_4.c line 5). Tag name 'myStruct' should not be reused. Already used as tag name (R\_4\_4.c line 5). | 文件范围 | Unset | Unreviewed |  |
| 1068 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 442 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |

Table 1.76. C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1101 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1405 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 923 | 10.1 | The value of an expression of integer type shall not be implicitly converted to a different underlying type. Implicit conversion of the expression of underlying type 'signed short' to the type 'signed char' that is not a wider integer type of the same signedness. | R\_5\_1() | Unset | Unreviewed |  |
| 646 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.77. C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1073 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1335 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1333 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'signed char' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_5\_2() | Unset | Unreviewed |  |
| 328 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.78. C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1176 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1076 | 6.4 | Bit fields shall only be defined to be of type unsigned int or signed int. | 文件范围 | Unset | Unreviewed |  |
| 1173 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.79. C:\workspace\Polyspace\_Workspace\194 C Code\R\_5\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1090 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 261 | 6.5 | Bit fields of type signed int shall be at least 2 bits long. | 文件范围 | Unset | Unreviewed |  |
| 1183 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.80. C:\workspace\Polyspace\_Workspace\194 C Code\R\_6\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 840 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1362 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 527 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_6\_1() | Unset | Unreviewed |  |
| 408 | 7.1 | Octal constants (other than zero) and octal escape sequences shall not be used. | R\_6\_1() | Unset | Unreviewed |  |
| 846 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1275 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_6\_1() | Unset | Unreviewed |  |
| 717 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_6\_1() | Unset | Unreviewed |  |
| 1319 | 7.1 | Octal constants (other than zero) and octal escape sequences shall not be used. | R\_6\_1() | Unset | Unreviewed |  |
| 564 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.81. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1083 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1084 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 394 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1415 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1414 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.82. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_10.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 971 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 293 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1085 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1427 | 8.10 | All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required. Variable 'globalCounter' should have internal linkage | 文件范围 | Unset | Unreviewed |  |
| 764 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 304 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 323 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1093 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_7\_10\_PrintCounter() | Unset | Unreviewed |  |
| 1186 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_7\_10\_PrintCounter() | Unset | Unreviewed |  |
| 1360 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.83. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 901 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1029 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 259 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 964 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 503 | 8.3 | For each function parameter the type given in the declaration and definition shall be identical, and the return types shall also be identical. Function 'R\_7\_2' is not compatible with previous declaration. | 文件范围 | Unset | Unreviewed |  |
| 654 | 5.7 | No identifier name should be reused. Identifier 'R\_7\_2' should not be reused. Already used as identifier (Rule194.h line 66). | 文件范围 | Unset | Unreviewed |  |
| 693 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Declaration is incompatible with "void R\_7\_2(int)" (declared at line 66 of "C:\workspace\Polyspace\_Workspace\194 C Code\Rule194.h") | 文件范围 | Unset | Unreviewed |  |
| 1426 | 8.8 | An external object or function shall be declared in one and only one file. Function 'R\_7\_2' has an external declaration in another file that is not included in the current file. | 文件范围 | Unset | Unreviewed |  |
| 1429 | 8.4 | If objects or functions are declared more than once their types shall be compatible. Global declaration of 'R\_7\_2' function has a type incompatible with its definition. This defect occurs when linking the 2 translation units: Rule194\_main.c and R\_7\_2.c | 文件范围 | Unset | Unreviewed |  |
| 405 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 799 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.84. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1285 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1180 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 582 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 637 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.85. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1211 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1311 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1010 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 301 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 996 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_7\_4() | Unset | Unreviewed |  |
| 1325 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_7\_4() | Unset | Unreviewed |  |

Table 1.86. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_5.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 725 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 736 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 469 | 8.8 | An external object or function shall be declared in one and only one file. Variable 'myGlobalVar' that has external linkage should be declared in a header file. | 文件范围 | Unset | Unreviewed |  |
| 1295 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 897 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |

Table 1.87. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_5\_a.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1247 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 666 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 821 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 786 | 8.8 | An external object or function shall be declared in one and only one file. Variable 'myGlobalVar' that has external linkage should be declared in a header file. | 文件范围 | Unset | Unreviewed |  |
| 1105 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 502 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1102 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | useMyGlobalVar() | Unset | Unreviewed |  |
| 1100 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | useMyGlobalVar() | Unset | Unreviewed |  |

Table 1.88. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_6.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1117 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1062 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 715 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1355 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 540 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1189 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 493 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Variable "R\_7\_6" has already been initialized | 文件范围 | Unset | Unreviewed |  |
| 861 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 407 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 823 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 278 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1078 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 598 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Function "R\_7\_6\_function" has already been defined | 文件范围 | Unset | Unreviewed |  |
| 1133 | 5.7 | No identifier name should be reused. Identifier 'R\_7\_6\_function' should not be reused. Already used as identifier (R\_7\_6.c line 11). | 文件范围 | Unset | Unreviewed |  |
| 1112 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 329 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.89. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_7.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1238 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 578 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1425 | 8.9 | An identifier with external linkage shall have exactly one external definition. Global variable 'externalVar' multiply defined: This defect occurs when linking the 2 translation units: R\_7\_7.c and R\_4\_2.c | 文件范围 | Unset | Unreviewed |  |
| 1326 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 984 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1080 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1270 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_7\_7() | Unset | Unreviewed |  |
| 322 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_7\_7() | Unset | Unreviewed |  |
| 900 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.90. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_8.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 490 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1359 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1432 | 8.10 | All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required. Variable 'globalVarWithoutStatic' should have internal linkage | 文件范围 | Unset | Unreviewed |  |
| 931 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1094 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 709 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 401 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_7\_8() | Unset | Unreviewed |  |
| 1128 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_7\_8() | Unset | Unreviewed |  |

Table 1.91. C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_9.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1164 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1160 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1280 | 8.12 | When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation. Size of array 'R\_7\_9' should be explicitly stated. | 文件范围 | Unset | Unreviewed |  |
| 1380 | 8.8 | An external object or function shall be declared in one and only one file. | 文件范围 | Unset | Unreviewed |  |
| 1431 | 8.9 | An identifier with external linkage shall have exactly one external definition. undefined global variable 'R\_7\_9' | 文件范围 | Unset | Unreviewed |  |
| 1271 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1330 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1459 | 21.1 | Minimisation of run-time failures shall be ensured by the use of at least one tool/technique. Attempt to access R\_7\_9 outside its bounds. Additional Info: Actual values: Buffer offset is 0. Risk: Value read is invalid or program can hang or segfault. Fix: Check for possible programming error earlier, for instance, an incorrect loop bound. | useR\_7\_9() | Unset | Unreviewed |  |
| 629 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.92. C:\workspace\Polyspace\_Workspace\194 C Code\R\_8\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1226 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1372 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1120 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_8\_1() | Unset | Unreviewed |  |
| 1119 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_8\_1() | Unset | Unreviewed |  |
| 687 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_8\_1() | Unset | Unreviewed |  |
| 1454 | 9.1 | All automatic variables shall have been assigned a value before being used. Local variable unassigned is read before it is initialized. Additional Info: Risk: Reading non-initialized memory can result in unexpected values. Fix: Initialize the local variable before use. | R\_8\_1() | Unset | Unreviewed |  |
| 1124 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.93. C:\workspace\Polyspace\_Workspace\194 C Code\R\_8\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1239 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1134 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 683 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1145 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_8\_2() | Unset | Unreviewed |  |
| 406 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Expected an identifier | R\_8\_2() | Unset | Unreviewed |  |
| 1158 | 5.7 | No identifier name should be reused. Identifier '<error>' should not be reused. Already used as identifier (R\_8\_2.c line 7). | R\_8\_2() | Unset | Unreviewed |  |
| 1199 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Expected an identifier | R\_8\_2() | Unset | Unreviewed |  |
| 1148 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1221 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1136 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 850 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1150 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. A value of type "int" cannot be used to initialize an entity of type "struct 7" | R\_8\_2() | Unset | Unreviewed |  |
| 644 | 5.7 | No identifier name should be reused. Identifier '<error>' should not be reused. Already used as identifier (R\_8\_2.c line 7). | R\_8\_2() | Unset | Unreviewed |  |
| 1137 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Expected an identifier | R\_8\_2() | Unset | Unreviewed |  |
| 369 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.94. C:\workspace\Polyspace\_Workspace\194 C Code\R\_8\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1391 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1154 | 9.3 | In an enumerator list, the '=' construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised. | 文件范围 | Unset | Unreviewed |  |
| 1161 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.95. C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_1.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 819 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 945 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 489 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'long' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_9\_1() | Unset | Unreviewed |  |
| 1149 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_9\_1() | Unset | Unreviewed |  |
| 1163 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.96. C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_2.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 443 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1307 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1302 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1003 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 455 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 523 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1258 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1169 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1419 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1167 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'double' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_9\_2() | Unset | Unreviewed |  |
| 315 | 10.2 | The value of an expression of floating type shall not be implicitly converted to a different type. Implicit conversion of complex floating expression from 'float' to 'double'. | R\_9\_2() | Unset | Unreviewed |  |
| 1344 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1171 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 954 | 10.2 | The value of an expression of floating type shall not be implicitly converted to a different type. Implicit conversion of complex floating expression from 'float' to 'double'. | R\_9\_2() | Unset | Unreviewed |  |
| 1235 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1208 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_9\_2() | Unset | Unreviewed |  |
| 1354 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_9\_2() | Unset | Unreviewed |  |
| 1178 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1170 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1386 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.97. C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_3.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1168 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1187 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned char' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 581 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned char' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 768 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1305 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned char' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_9\_3() | Unset | Unreviewed |  |

Table 1.98. C:\workspace\Polyspace\_Workspace\194 C Code\R\_9\_4.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1193 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1324 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1262 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | R\_9\_4() | Unset | Unreviewed |  |
| 432 | 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. Integer constant does not fit within long int. | R\_9\_4() | Unset | Unreviewed |  |
| 949 | 10.1 | The value of an expression of integer type shall not be implicitly converted to a different underlying type. Implicit conversion of the expression of underlying type 'signed long long' to the type 'unsigned int' that is not a wider integer type of the same signedness. | R\_9\_4() | Unset | Unreviewed |  |
| 1195 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1053 | 20.9 | The input/output library <stdio.h> shall not be used in production code. | R\_9\_4() | Unset | Unreviewed |  |
| 955 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function printf is not used. | R\_9\_4() | Unset | Unreviewed |  |

Table 1.99. C:\workspace\Polyspace\_Workspace\194 C Code\Rule194.h

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1212 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 870 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 571 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 303 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 535 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 530 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 311 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 806 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 298 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1353 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 993 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1374 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 430 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 765 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1174 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 374 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 324 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 529 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 272 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 790 | 18.1 | All structure or union types shall be complete at the end of a translation unit. | 文件范围 | Unset | Unreviewed |  |
| 313 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 461 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 510 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1351 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 812 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 544 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 393 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1162 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 816 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 698 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1339 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 403 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 344 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 411 | 18.1 | All structure or union types shall be complete at the end of a translation unit. | 文件范围 | Unset | Unreviewed |  |
| 347 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 525 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 515 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 481 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 404 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 287 | 18.1 | All structure or union types shall be complete at the end of a translation unit. | 文件范围 | Unset | Unreviewed |  |
| 1184 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 987 | 18.1 | All structure or union types shall be complete at the end of a translation unit. | 文件范围 | Unset | Unreviewed |  |
| 383 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 511 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 380 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 508 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 355 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 647 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1367 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1423 | 5.5 | No object or function identifier with static storage duration should be reused. function 'R\_7\_3' conflicts with the function name 'R\_7\_3' (R\_7\_3.c line 5). | 文件范围 | Unset | Unreviewed |  |
| 576 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 295 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 633 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 290 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 361 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 520 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 378 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 498 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 948 | 8.5 | There shall be no definitions of objects or functions in a header file. Object 'R\_7\_6' should not be defined in a header file. | 文件范围 | Unset | Unreviewed |  |
| 1424 | 8.9 | An identifier with external linkage shall have exactly one external definition. The global variable 'R\_7\_6' has a tentative definition which is used in different translation units This defect occurs when linking the 2 translation units: R\_10\_2.c and R\_10\_1.c | 文件范围 | Unset | Unreviewed |  |
| 494 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 881 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1074 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 528 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 439 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1001 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1089 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1008 | 8.12 | When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation. Size of array 'R\_7\_9' should be explicitly stated. | 文件范围 | Unset | Unreviewed |  |
| 541 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 651 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1206 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 282 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 302 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1097 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1004 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 973 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 574 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 336 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 539 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 952 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 260 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 763 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 886 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 372 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 480 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 562 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 457 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'float' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 835 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 264 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned char' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 388 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'unsigned char' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1064 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 387 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 367 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1416 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 791 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 299 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 906 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 584 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 491 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 472 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 976 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 366 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1383 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 419 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 479 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 505 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 729 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 314 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1157 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 669 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 435 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 556 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 351 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1155 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 482 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 409 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1088 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 704 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 507 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 306 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1365 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 338 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 789 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 930 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1279 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 834 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1323 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 428 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1231 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 440 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1347 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 335 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 500 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1055 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 665 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 707 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 969 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 676 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 379 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 509 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1210 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 724 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 543 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 719 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 722 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 339 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 965 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 269 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 449 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 318 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1255 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 337 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 395 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 495 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1348 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1118 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 438 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 300 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 291 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 334 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1375 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 415 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 436 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 327 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 450 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 737 | 16.3 | Identifiers shall be given for all of the parameters in a function prototype declaration. | 文件范围 | Unset | Unreviewed |  |
| 660 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 953 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1352 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 431 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1146 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 696 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 452 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 917 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 410 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1143 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 946 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1409 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 537 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1268 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 798 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 382 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 429 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1401 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 332 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1214 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 289 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 305 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 271 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 559 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 273 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 526 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1413 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1265 | 18.1 | All structure or union types shall be complete at the end of a translation unit. | 文件范围 | Unset | Unreviewed |  |
| 484 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 868 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1049 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 294 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 397 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 396 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 425 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 813 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 732 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 843 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1368 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1244 | 18.4 | Unions shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1273 | 18.1 | All structure or union types shall be complete at the end of a translation unit. | 文件范围 | Unset | Unreviewed |  |
| 309 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 377 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 851 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 398 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 400 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 274 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 475 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 368 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 447 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 345 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1040 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 808 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 483 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1185 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1028 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 275 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1220 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 357 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 270 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 441 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 281 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1109 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 573 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 364 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 412 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 761 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 506 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 1200 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 346 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 285 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 376 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 333 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1044 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 268 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 456 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 864 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 465 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 943 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 262 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1390 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |
| 1012 | 16.5 | Functions with no parameters shall be declared with parameter type void. | 文件范围 | Unset | Unreviewed |  |
| 258 | 2.2 | Source code shall only use /\* ... \*/ style comments. C++ comments shall not be used. | 文件范围 | Unset | Unreviewed |  |

Table 1.100. C:\workspace\Polyspace\_Workspace\194 C Code\Rule194\_main.c

| **ID** | **规则** | **消息** | **函数** | **严重性** | **状态** | **注释** |
| --- | --- | --- | --- | --- | --- | --- |
| 1394 | 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. 'int' doesn't provide information about its size. Define and use typedefs clarifying type and size for numerical types or use one of the exact-width numerical types defined in <stdint.h>. | 文件范围 | Unset | Unreviewed |  |
| 1373 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'main' has no visible prototype at definition. | 文件范围 | Unset | Unreviewed |  |
| 966 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_1\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 360 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_1\_1 is not used. | main() | Unset | Unreviewed |  |
| 1304 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_1\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1130 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_1\_2 is not used. | main() | Unset | Unreviewed |  |
| 1126 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_1\_3\_function\_with\_a\_very\_long\_name\_exceeding\_the\_recommended\_length\_of\_identifiers' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 416 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_1\_3\_function\_with\_a\_very\_long\_name\_exceeding\_the\_recommended\_length\_of\_identifiers is not used. | main() | Unset | Unreviewed |  |
| 426 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_2\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 317 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_2\_1 is not used. | main() | Unset | Unreviewed |  |
| 1108 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_2\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 933 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_2\_2 is not used. | main() | Unset | Unreviewed |  |
| 739 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_2\_3' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1357 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_2\_3 is not used. | main() | Unset | Unreviewed |  |
| 1281 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_3\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1385 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_3\_1 is not used. | main() | Unset | Unreviewed |  |
| 1256 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_3\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1020 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_3\_2 is not used. | main() | Unset | Unreviewed |  |
| 1179 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_4\_1\_function\_with\_a\_very\_long\_identifier\_name\_exceeding\_thirty\_one\_characters' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 516 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_4\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1346 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_5\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1254 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_5\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 593 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_6\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1418 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 778 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_4' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1291 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_5' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1420 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'useMyGlobalVar' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1322 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_6\_function' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 727 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_7' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1395 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_8' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 858 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'useR\_7\_9' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 950 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_10\_UpdateCounter' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1261 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_7\_10\_PrintCounter' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1313 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_8\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 859 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_8\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 402 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_9\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1410 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_9\_2 is not used. | main() | Unset | Unreviewed |  |
| 947 | 10.2 | The value of an expression of floating type shall not be implicitly converted to a different type. Implicit conversion of the expression from 'double' to 'float' that is not a wider floating type. | main() | Unset | Unreviewed |  |
| 1259 | 10.2 | The value of an expression of floating type shall not be implicitly converted to a different type. Implicit conversion of the expression from 'double' to 'float' that is not a wider floating type. | main() | Unset | Unreviewed |  |
| 1301 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_9\_3 is not used. | main() | Unset | Unreviewed |  |
| 1197 | 10.1 | The value of an expression of integer type shall not be implicitly converted to a different underlying type. Implicit conversion of the expression of underlying type 'signed short' to the type 'unsigned char' that is not a wider integer type of the same signedness. | main() | Unset | Unreviewed |  |
| 1356 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_9\_4' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 994 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_10\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1283 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_10\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1400 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1245 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1052 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_3' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 667 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_4' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1334 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_5' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 460 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_6' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1328 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_7' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1248 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_8' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 283 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_9' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 497 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_10' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1135 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_11' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1288 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_12' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1243 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_13' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 265 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_14' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 623 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_11\_15' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1129 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_12\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1201 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_12\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 320 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_12\_3' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1389 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_12\_4' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1312 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_12\_5' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1331 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_12\_6 is not used. | main() | Unset | Unreviewed |  |
| 1336 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_12\_8' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1408 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_13\_4' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 797 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1290 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_2\_a' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 331 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_6' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1203 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_14\_6 is not used. | main() | Unset | Unreviewed |  |
| 1381 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_7' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1303 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_7\_exampleFunction' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 691 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_14\_8' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1272 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function R\_14\_8 is not used. | main() | Unset | Unreviewed |  |
| 892 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_15\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 473 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_15\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1017 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_15\_3' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 759 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_15\_4' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 832 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'useR\_16\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 853 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_16\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1282 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_16\_3' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1377 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_16\_4\_use\_union' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 611 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_1' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1209 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 606 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_3' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 991 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_4' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1293 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_5' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1371 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_6' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 418 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_7' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1321 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_8' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1402 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_9\_example' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1236 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_10' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 695 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_11' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1222 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_12' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1240 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_17\_13' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |
| 1316 | 16.10 | If a function returns error information, then that error information should be tested. Value returned by function myMalloc is not used. | main() | Unset | Unreviewed |  |
| 1366 | 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. Function 'R\_18\_2' does not have complete prototype when called (argument list is missing). | main() | Unset | Unreviewed |  |

第2章附录 1 - 配置设置

Polyspace 设置

| **选项** | **值** |
| --- | --- |
| -author | lehuahu |
| -bug-finder | true |
| -checkers |  |
| -compiler | gnu3.4 |
| -date | 09/12/2024 |
| -disable-checkers | all |
| -dos | true |
| -I | C:\workspace\Polyspace\_Workspace\194 C Code |
| -import-comments | C:\workspace\Polyspace\_Workspace\194 C Code\Module\_1\BF\_Result |
| -lang | C |
| -library | none |
| -misra2 | all-rules |
| -prog | MISRA\_RULE\_JDSN |
| -results-dir | C:\workspace\Polyspace\_Workspace\MISRA\_RULE\_JDSN\BF\_Result\_1 |
| -target | i386 |
| -verif-version | 1.0 |

编码标准配置

Table 2.1. MISRA C:2004 编码标准 配置

| **规则** | **描述** | **模式** | **注释** | **已启用** |
| --- | --- | --- | --- | --- |
| 1.1 | All code shall conform to ISO 9899:1990 'Programming languages - C', amended and corrected by ISO/IEC 9899/COR1:1995, ISO/IEC 9899/AMD1:1995, and ISO/IEC 9899/COR2:1996. | required | - | 是 |
| 1.2 | No reliance shall be placed on undefined or unspecified behaviour. | required | Not enforceable | 否 |
| 1.3 | Multiple compilers and/or languages shall only be used if there is a common defined interface standard for object code to which the language/compilers/assemblers conform. | required | Not enforceable | 否 |
| 1.4 | The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers. | required | Not enforceable | 否 |
| 1.5 | Floating point implementations should comply with a defined floating point standard. | advisory | Not implemented | 否 |
| 2.1 | Assembly language shall be encapsulated and isolated. | required | - | 是 |
| 2.2 | Source code shall only use /\* ... \*/ style comments. | required | - | 是 |
| 2.3 | The character sequence /\* shall not be used within a comment. | required | - | 是 |
| 2.4 | Sections of code should not be 'commented out'. | advisory | - | 是 |
| 3.1 | All usage of implementation-defined behaviour shall be documented. | required | Not enforceable | 否 |
| 3.2 | The character set and the corresponding encoding shall be documented. | required | Not enforceable | 否 |
| 3.3 | The implementation of integer division in the chosen compiler should be determined, documented and taken into account. | advisory | Not implemented | 否 |
| 3.4 | All uses of the #pragma directive shall be documented and explained. | required | - | 是 |
| 3.5 | If it is being relied upon, the implementation-defined behaviour and packing of bitfields shall be documented. | required | Not enforceable | 否 |
| 3.6 | All libraries used in production code shall be written to comply with the provisions of this document, and shall have been subject to appropriate validation. | required | Not enforceable | 否 |
| 4.1 | Only those escape sequences which are defined in the ISO C standard shall be used. | required | - | 是 |
| 4.2 | Trigraphs shall not be used. | required | - | 是 |
| 5.1 | Identifiers (internal and external) shall not rely on the significance of more than 31 characters. | required | - | 是 |
| 5.2 | Identifiers in an inner scope shall not use the same name as an identifier in an outer scope, and therefore hide that identifier. | required | - | 是 |
| 5.3 | A typedef name shall be a unique identifier. | required | - | 是 |
| 5.4 | A tag name shall be a unique identifier. | required | - | 是 |
| 5.5 | No object or function identifier with static storage duration should be reused. | advisory | - | 是 |
| 5.6 | No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure and union member names. | advisory | - | 是 |
| 5.7 | No identifier name should be reused. | advisory | - | 是 |
| 6.1 | The plain char type shall be used only for the storage and use of character values. | required | - | 是 |
| 6.2 | Signed and unsigned char type shall be used only for the storage and use of numeric values. | required | - | 是 |
| 6.3 | Typedefs that indicate size and signedness should be used in place of the basic types. | advisory | - | 是 |
| 6.4 | Bit fields shall only be defined to be of type unsigned int or signed int. | required | - | 是 |
| 6.5 | Bit fields of type signed int shall be at least 2 bits long. | required | - | 是 |
| 7.1 | Octal constants (other than zero) and octal escape sequences shall not be used. | required | - | 是 |
| 8.1 | Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call. | required | - | 是 |
| 8.2 | Whenever an object or function is declared or defined, its type shall be explicitly stated. | required | - | 是 |
| 8.3 | For each function parameter the type given in the declaration and definition shall be identical, and the return types shall also be identical. | required | - | 是 |
| 8.4 | If objects or functions are declared more than once their types shall be compatible. | required | - | 是 |
| 8.5 | There shall be no definitions of objects or functions in a header file. | required | - | 是 |
| 8.6 | Functions shall be declared at file scope. | required | - | 是 |
| 8.7 | Objects shall be defined at block scope if they are only accessed from within a single function. | required | - | 是 |
| 8.8 | An external object or function shall be declared in one and only one file. | required | - | 是 |
| 8.9 | An identifier with external linkage shall have exactly one external definition. | required | - | 是 |
| 8.10 | All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required. | required | - | 是 |
| 8.11 | The static storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage. | required | - | 是 |
| 8.12 | When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation. | required | - | 是 |
| 9.1 | All automatic variables shall have been assigned a value before being used. | required | - | 是 |
| 9.2 | Braces shall be used to indicate and match the structure in the non-zero initialisation of arrays and structures. | required | - | 是 |
| 9.3 | In an enumerator list, the '=' construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised. | required | - | 是 |
| 10.1 | The value of an expression of integer type shall not be implicitly converted to a different underlying type. | required | - | 是 |
| 10.2 | The value of an expression of floating type shall not be implicitly converted to a different type. | required | - | 是 |
| 10.3 | The value of a complex expression of integer type may only be cast to a type that is narrower and of the same signedness as the underlying type of the expression. | required | - | 是 |
| 10.4 | The value of a complex expression of floating type may only be cast to narrower floating type. | required | - | 是 |
| 10.5 | If the bitwise operator ~ and << are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediatly cast to the underlying type of the operand. | required | - | 是 |
| 10.6 | The 'U' suffix shall be applied to all constants of unsigned types. | required | - | 是 |
| 11.1 | Conversion shall not be performed between a pointer to a function and any type other than an integral type. | required | - | 是 |
| 11.2 | Conversion shall not be performed between a pointer to an object and any type other than an integral type, another pointer to object type or a pointer to void. | required | - | 是 |
| 11.3 | A cast should not be performed between a pointer type and an integral type. | advisory | - | 是 |
| 11.4 | A cast should not be performed between a pointer to object type and a different pointer to object type. | advisory | - | 是 |
| 11.5 | A cast shall not be performed that removes any const or volatile qualification from the type addressed by a pointer. | required | - | 是 |
| 12.1 | Limited dependence should be placed on C's operator precedence rules in expressions. | advisory | - | 是 |
| 12.2 | The value of an expression shall be the same under any order of evaluation that the standard permits. | required | - | 是 |
| 12.3 | The sizeof operator should not be used on expressions that contain side effects. | required | - | 是 |
| 12.4 | The right hand operand of a logical && or || operator shall not contain side effects. | required | - | 是 |
| 12.5 | The operands of a logical && or || shall be primary-expressions. | required | - | 是 |
| 12.6 | The operands of a logical operators (&&, || and !) should be effectively Boolean. Expressions that are effectively Boolean should not be used as operands to operators other than (&&, || and !). | advisory | - | 是 |
| 12.7 | Bitwise operators shall not be applied to operands whose underlying type is signed. | required | - | 是 |
| 12.8 | The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand. | required | - | 是 |
| 12.9 | The unary minus operator shall not be applied to an expression whose underlying type is unsigned. | required | - | 是 |
| 12.10 | The comma operator shall not be used. | required | - | 是 |
| 12.11 | Evaluation of constant unsigned integer expressions should not lead to wrap-around. | advisory | - | 是 |
| 12.12 | The underlying bit representations of floating-point values shall not be used. | required | - | 是 |
| 12.13 | The increment (++) and decrement (--) operators should not be mixed with other operators in an expression | advisory | - | 是 |
| 13.1 | Assignment operators shall not be used in expressions that yield a Boolean value. | required | - | 是 |
| 13.2 | Tests of a value against zero should be made explicit, unless the operand is effectively Boolean. | advisory | - | 是 |
| 13.3 | Floating-point expressions shall not be tested for equality or inequality. | required | - | 是 |
| 13.4 | The controlling expression of a for statement shall not contain any objects of floating type. | required | - | 是 |
| 13.5 | The three expressions of a for statement shall be concerned only with loop control. | required | - | 是 |
| 13.6 | Numeric variables being used within a for loop for iteration counting should not be modified in the body of the loop. | required | - | 是 |
| 13.7 | Boolean operations whose results are invariant shall not be permitted. | required | - | 是 |
| 14.1 | There shall be no unreachable code. | required | - | 是 |
| 14.2 | All non-null statements shall either have at least one side effect however executed, or cause control flow to change. | required | - | 是 |
| 14.3 | Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment provided that the first character following the null statement is white-space character. | required | - | 是 |
| 14.4 | The goto statement shall not be used. | required | - | 是 |
| 14.5 | The continue statement shall not be used. | required | - | 是 |
| 14.6 | For any iteration statement there shall be at most one break statement used for loop termination. | required | - | 是 |
| 14.7 | A function shall have a single point of exit at the end of the function. | required | - | 是 |
| 14.8 | The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement. | required | - | 是 |
| 14.9 | An if (expression) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement. | required | - | 是 |
| 14.10 | All if ... else if constructs should contain a final else clause. | required | - | 是 |
| 15.0 | A switch statement shall conform to MISRA-C syntax. | required | - | 是 |
| 15.1 | A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement. | required | - | 是 |
| 15.2 | An unconditional break statement shall terminate every non-empty switch clause. | required | - | 是 |
| 15.3 | The final clause of a switch statement shall be the default clause. | required | - | 是 |
| 15.4 | A switch expression should not represent a value that is effectively Boolean. | required | - | 是 |
| 15.5 | Every switch statement shall have at least one case clause. | required | - | 是 |
| 16.1 | Functions shall not be defined with variable numbers of arguments. | required | - | 是 |
| 16.2 | Functions shall not call themselves, either directly or indirectly. | required | - | 是 |
| 16.3 | Identifiers shall be given for all of the parameters in a function prototype declaration. | required | - | 是 |
| 16.4 | The identifiers used in the declaration and definition of a function shall be identical. | required | - | 是 |
| 16.5 | Functions with no parameters shall be declared with parameter type void. | required | - | 是 |
| 16.6 | The number of arguments passed to a function shall match the number of parameters. | required | - | 是 |
| 16.7 | A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object. | advisory | - | 是 |
| 16.8 | All exit paths from a function with non-void return type shall have an explicit return statement with an expression. | required | - | 是 |
| 16.9 | A function identifier shall only be used with either a preceding &, or with a parenthesised parameter list, which may be empty. | required | - | 是 |
| 16.10 | If a function returns error information, then that error information should be tested. | required | - | 是 |
| 17.1 | Pointer arithmetic shall only be applied to pointers that address an array or array element. | required | - | 是 |
| 17.2 | Pointer subtraction shall only be applied to pointers that address elements of the same array. | required | - | 是 |
| 17.3 | >, >=, <, <= shall not be applied to pointer types except where they point to the same array. | required | - | 是 |
| 17.4 | Array indexing shall be the only allowed form of pointer arithmetic. | required | - | 是 |
| 17.5 | The declaration of objects should contain no more than 2 levels of pointer indirection. | advisory | - | 是 |
| 17.6 | The address of an object with automatic storage shall not be assigned to an object that may persist after the object has ceased to exist. | required | - | 是 |
| 18.1 | All structure or union types shall be complete at the end of a translation unit. | required | - | 是 |
| 18.2 | An object shall not be assigned to an overlapping object. | required | - | 是 |
| 18.3 | An area of memory shall not be reused for unrelated purposes. | required | Not enforceable | 否 |
| 18.4 | Unions shall not be used. | required | - | 是 |
| 19.1 | #include statements in a file shall only be preceded by other pre-processor directives or comments. | advisory | - | 是 |
| 19.2 | Non-standard characters should not occur in header file names in #include directives. | advisory | - | 是 |
| 19.3 | The #include directive shall be followed by either a <filename> or "filename" sequence. | required | - | 是 |
| 19.4 | C macros shall only expand to a braced initialiser, a constant, a parenthesised expression,a type qualifier, a storage class specifier, or a do-while-zero construct. | required | - | 是 |
| 19.5 | Macros shall not be #define'd and #undef'd within a block. | required | - | 是 |
| 19.6 | #undef shall not be used. | required | - | 是 |
| 19.7 | A function should be used in preference to a macro. | advisory | - | 是 |
| 19.8 | A function-like macro shall not be invoked without all of its arguments. | required | - | 是 |
| 19.9 | Arguments to a function-like macro shall not contain tokens that look like pre-processing directives. | required | - | 是 |
| 19.10 | In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##. | required | - | 是 |
| 19.11 | All macro identifiers in preprocessor directives shall be defined before use, except in #ifdef and #ifndef preprocessor directives and the defined() operator. | required | - | 是 |
| 19.12 | There shall be at most one occurrence of the # or ## pre-processor operators in a single macro definition. | required | - | 是 |
| 19.13 | The # and ## preprocessor operators should not be used. | advisory | - | 是 |
| 19.14 | The defined pre-processor operator shall only be used in one of the two standard forms. | required | - | 是 |
| 19.15 | Precautions shall be taken in order to prevent the contents of a header file being included twice. | required | - | 是 |
| 19.16 | Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor. | required | - | 是 |
| 19.17 | All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if or #ifdef directive to which they are related. | required | - | 是 |
| 20.1 | Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined. | required | - | 是 |
| 20.2 | The names of standard library macros, objects and functions shall not be reused. | required | - | 是 |
| 20.3 | The validity of values passed to library functions shall be checked. | required | - | 是 |
| 20.4 | Dynamic heap memory allocation shall not be used. | required | - | 是 |
| 20.5 | The error indicator errno shall not be used. | required | - | 是 |
| 20.6 | The macro offsetof, in library <stddef.h>, shall not be used. | required | - | 是 |
| 20.7 | The setjmp macro and the longjmp function shall not be used. | required | - | 是 |
| 20.8 | The signal handling facilities of <signal.h> shall not be used. | required | - | 是 |
| 20.9 | The input/output library <stdio.h> shall not be used in production code. | required | - | 是 |
| 20.10 | The library functions atof, atoi and atol from library <stdlib.h> shall not be used. | required | - | 是 |
| 20.11 | The library functions abort, exit, getenv and system from library <stdlib.h> shall not be used. | required | - | 是 |
| 20.12 | The time handling functions of library <time.h> shall not be used. | required | - | 是 |
| 21.1 | Minimisation of run-time failures shall be ensured by the use of at least one tool/technique. | required | - | 是 |

具有编译错误的文件(未完整分析的文件)

Table 2.2. 具有编译错误的文件(未完整分析的文件)

| **文件** |
| --- |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_11\_3.c |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_16\_1.c |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_17\_5.c |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_4\_4.c |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_2.c |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_7\_6.c |
| C:\workspace\Polyspace\_Workspace\194 C Code\R\_8\_2.c |

第3章附录 2 - 定义

Table 3.1. 缩写

| **缩写** | **定义** |
| --- | --- |
| NA | 不可用 |