**注：**

1、根据需求选择所需规则，格式**如下：**（~ 后有一个空格，每条规则后面必须有一个英文分号）

一条规则：

~ RULE\_3\_1\_A\_do\_not\_start\_filename\_with\_underbar;

多条规则：

~ RULE\_3\_1\_A\_do\_not\_start\_filename\_with\_underbar;

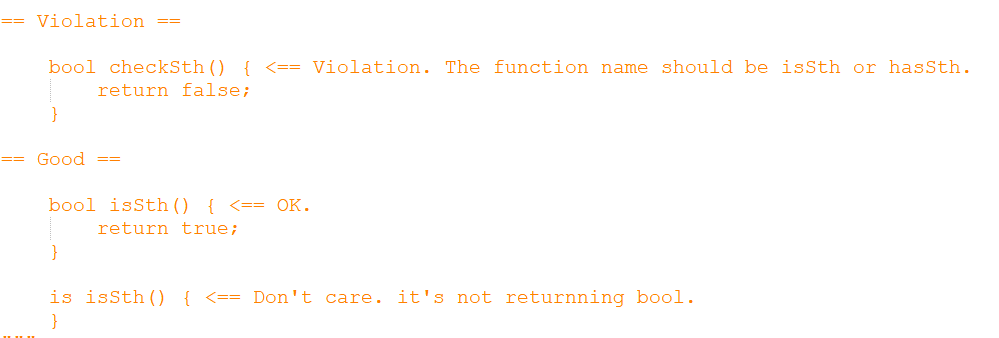
~ RULE\_3\_2\_B\_do\_not\_use\_same\_filename\_more\_than\_once;

———————————————————————————————————————

扩展规则如下：

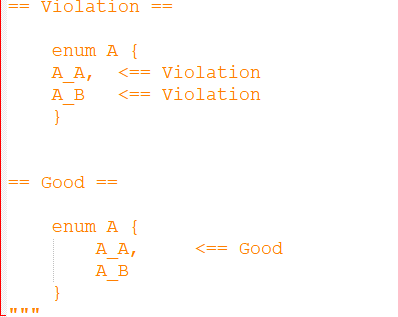
1、~ RULE\_3\_3\_A\_start\_function\_name\_with\_is\_or\_has\_when\_return\_bool

当返回值为布尔值的函数，其函数名要以is或者has开头



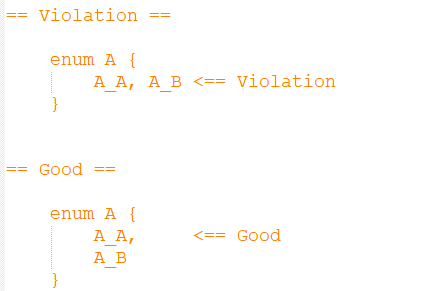
2、~ RULE\_4\_1\_B\_indent\_each\_enum\_item\_in\_enum\_block

定义枚举类型时，每一个枚举里面的枚举项都要进行缩进



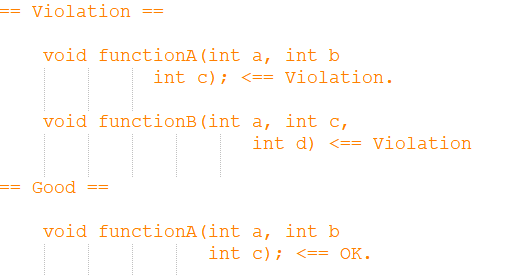
3、~ RULE\_4\_1\_B\_locate\_each\_enum\_item\_in\_seperate\_line

定义枚举类型时，每一个单独的枚举项占一行，若一行中出现了两个或多个枚举项则会报错



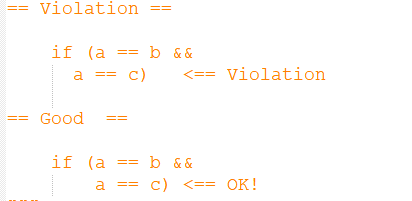
4、~ RULE\_4\_1\_C\_align\_long\_function\_parameter\_list

当第一个函数的参数在多行中定义时，从第二行开始，每一行的第一个参数对其第一行的第一个参数。



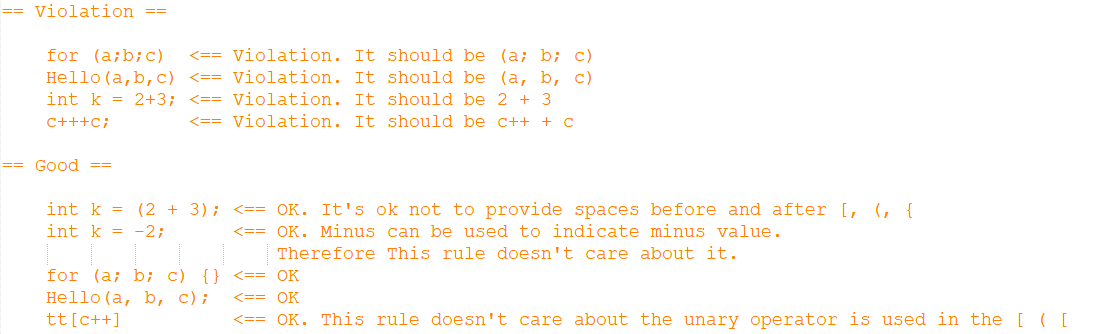
5、~ RULE\_4\_1\_E\_align\_conditions

在if和while字句中，如果字句中的条件列表为多行，那么从第二行开始，每一个行的条件要对其第一行。



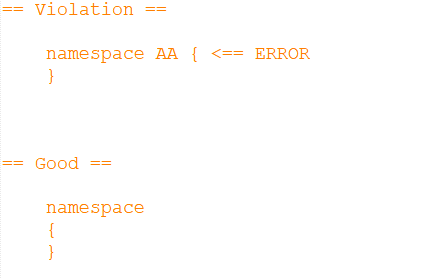
6、~ RULE\_4\_2\_A\_A\_space\_around\_operator

操作符周围应该使用空格



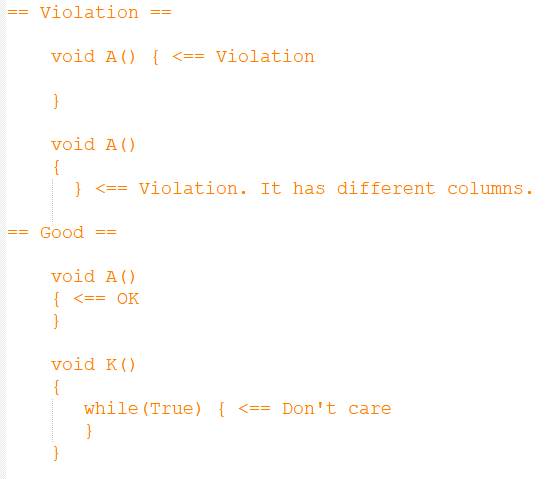
7、~ RULE\_4\_5\_A\_brace\_for\_namespace\_should\_be\_located\_in\_seperate\_line

名称空间的大括号应该位于单独的行中。



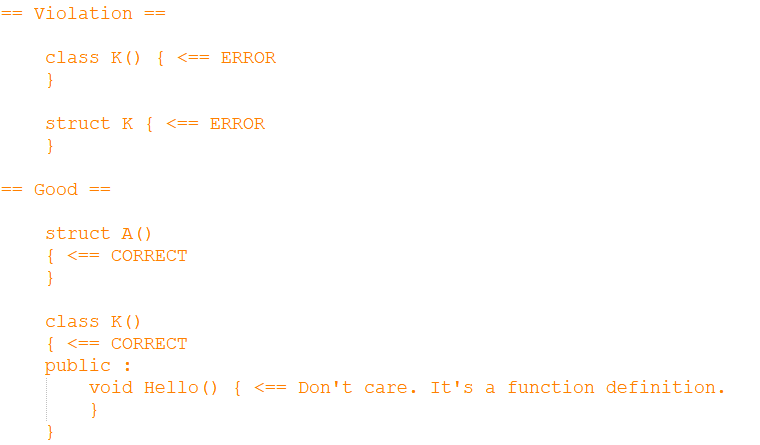
8、~ RULE\_4\_5\_A\_braces\_for\_function\_definition\_should\_be\_located\_in\_seperate\_line

函数定义的大括号应该位于单独的行中。



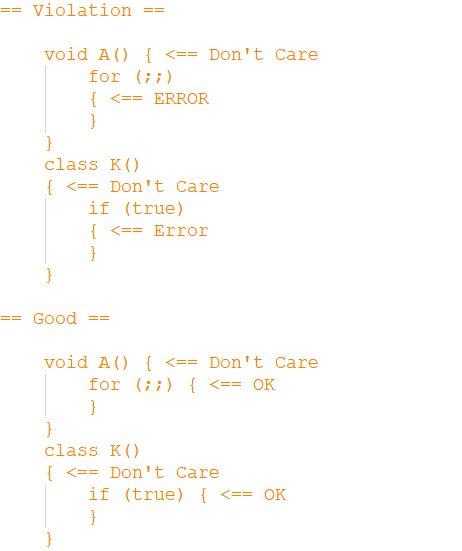
9、~ RULE\_4\_5\_A\_braces\_for\_type\_definition\_should\_be\_located\_in\_seperate\_line

类型定义(class / struct / union / enum)的大括号应该位于单独的行中。



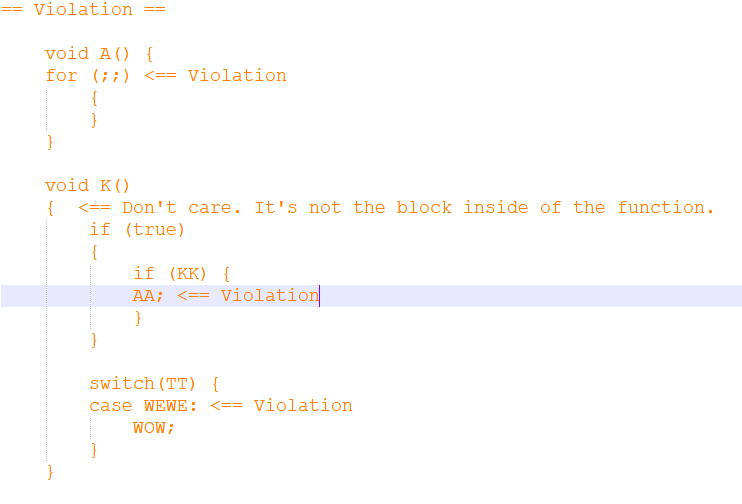
10、~ RULE\_4\_5\_A\_braces\_inside\_of\_function\_should\_be\_located\_in\_end\_of\_line

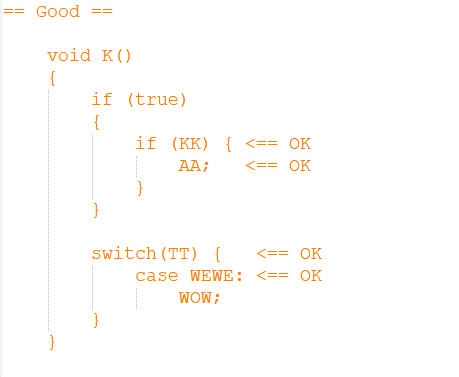
函数内部的大括号，应该位于每一行的结尾



11、~ RULE\_4\_5\_A\_indent\_blocks\_inside\_of\_function

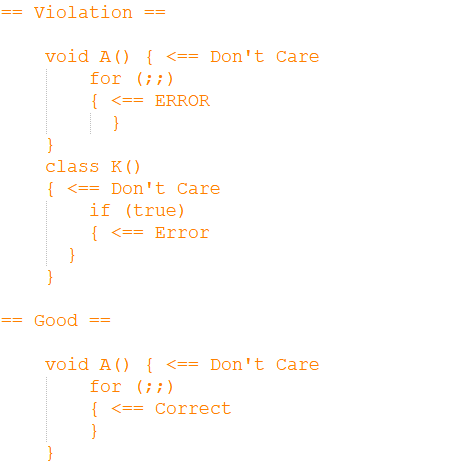
在函数内部，每一对大括号里面的代码要进行缩进，使之包含在大括号里





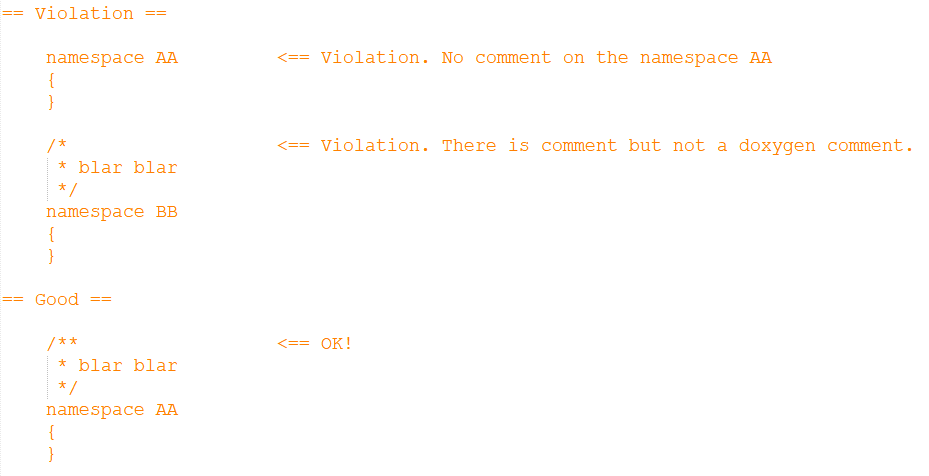
12、~ RULE\_4\_5\_A\_matching\_braces\_inside\_of\_function\_should\_be\_located\_same\_column

在函数内部，每一对大括号应该在同一列



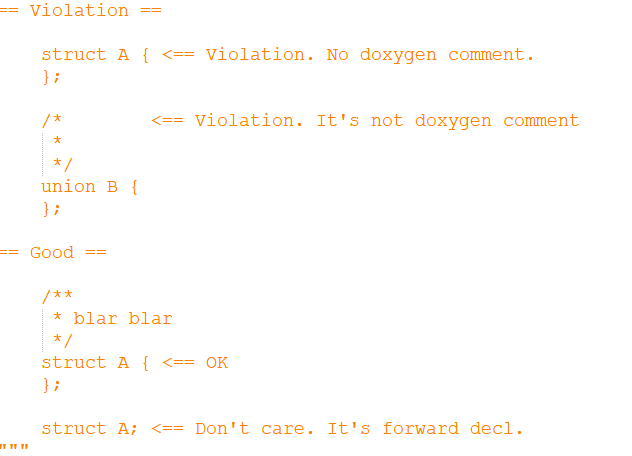
1. ~ RULE\_5\_2\_C\_provide\_doxygen\_namespace\_comment\_on\_namespace\_def

提供命名空间doxygen注释。它检查每个“名称空间”关键字前是否有doxygen sytle注释。



1. ~ RULE\_5\_2\_C\_provide\_doxygen\_struct\_comment\_on\_struct\_def

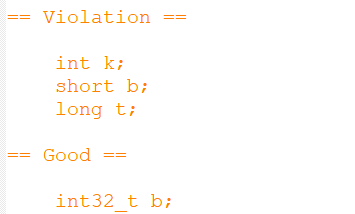
提供struct/union doxygen风格的注释。它检查每个结构体/联合体定义前面是否有doxygen sytle注释。



1. ~ RULE\_6\_2\_A\_do\_not\_use\_system\_dependent\_type

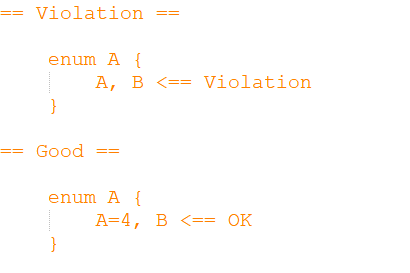
不要使用依赖于系统的类型(short、long、int)

而是分别使用与系统无关的类型(int16\_t、int64\_t、int32\_t)



1. ~ RULE\_6\_4\_B\_initialize\_first\_item\_of\_enum

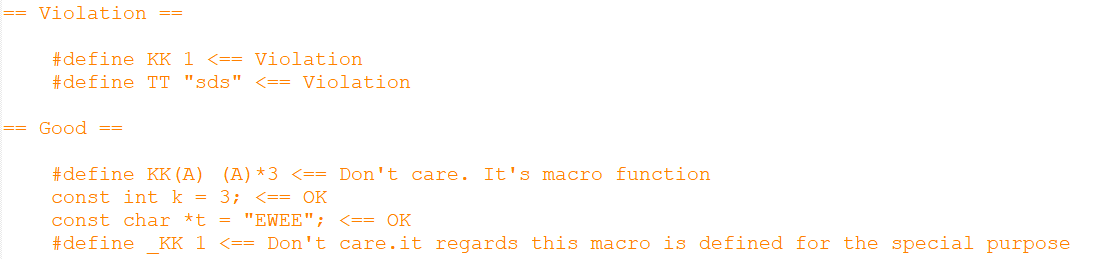
检查enum类型的第一项，并检查它是否已初始化（它检查第一个元素，并看到有= token）。



1. ~ RULE\_6\_5\_B\_do\_not\_use\_macro\_for\_constants

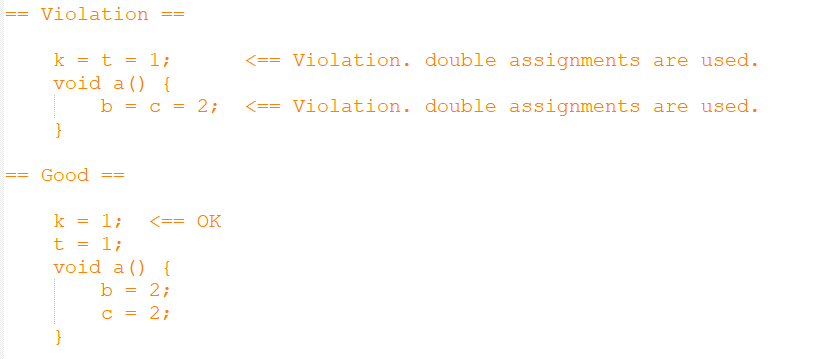
不要使用宏常量。相反，可以使用enum或const变量。不过，可以编写宏函数。

如果宏定义是以下划线开头，它认为这个宏是为特殊目的而定义的，不会报错。



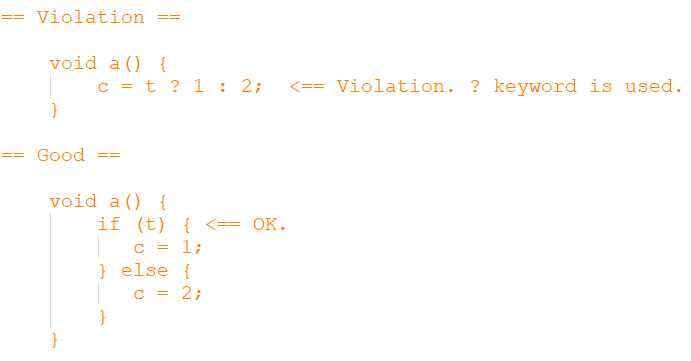
1. ~ RULE\_7\_1\_B\_A\_do\_not\_use\_double\_assignment

不用对多个变量使用双重赋值



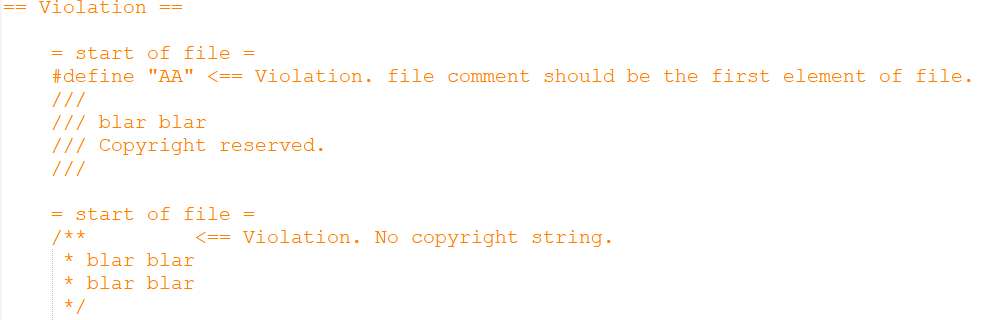
1. ~ RULE\_7\_1\_C\_do\_not\_use\_question\_keyword

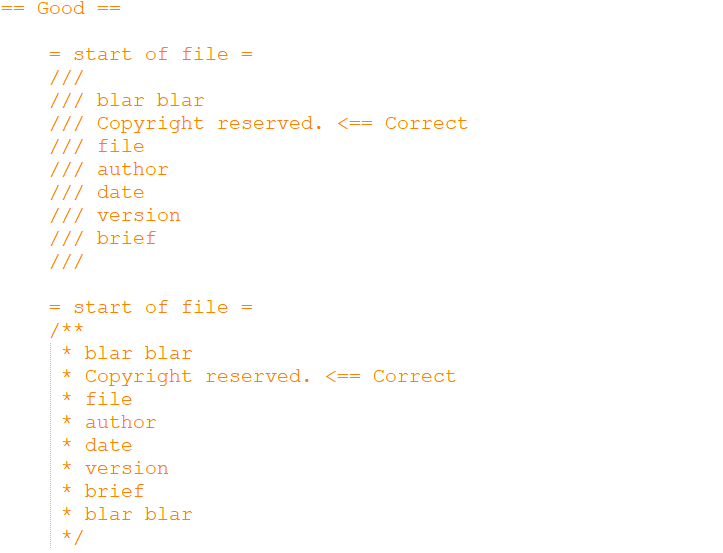
代码中不要使用三目运算符”?”



1. ~ RULE\_8\_1\_A\_provide\_file\_info\_comment

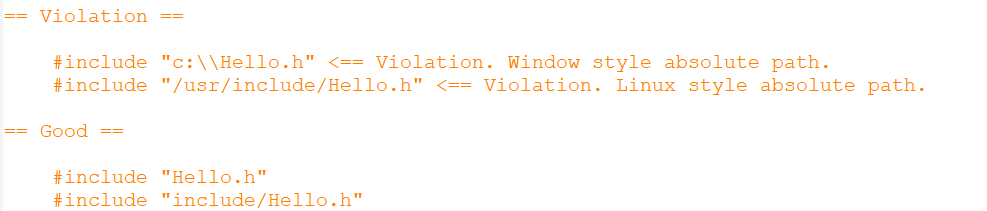
提供代码信息注释，包括版权信息，并且这些注释在代码文件的开头





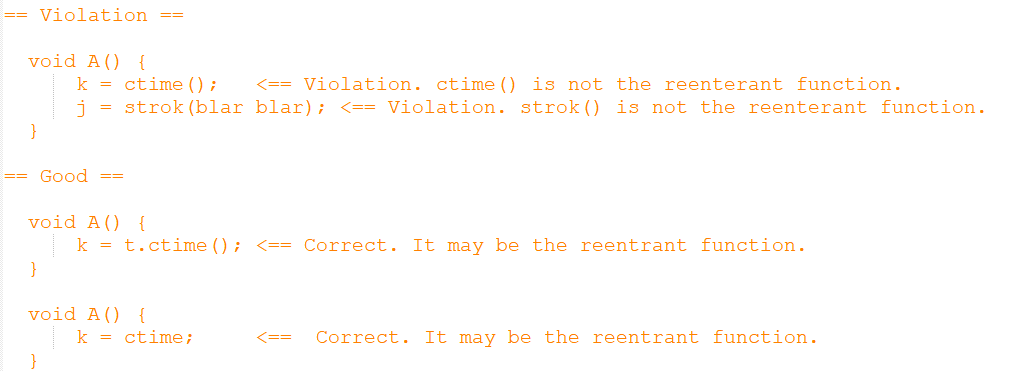
1. ~ RULE\_9\_1\_A\_do\_not\_use\_hardcorded\_include\_path

头文件路径不要使用绝对路径



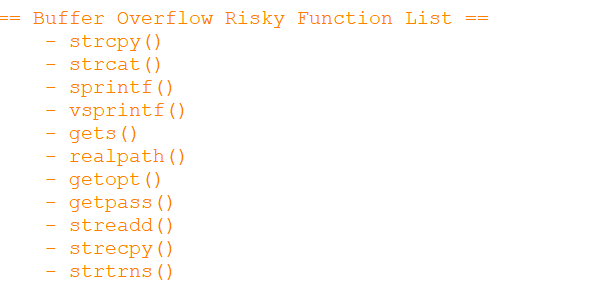
1. ~ RULE\_9\_2\_D\_use\_reentrant\_function

使用可重入函数，不要使用不可重入函数



1. ~ RULE\_10\_1\_A\_do\_not\_use\_bufferoverflow\_risky\_function\_for\_unix

不要使用unix环境下缓冲区溢出危险函数



1. ~ RULE\_10\_1\_B\_do\_not\_use\_bufferoverflow\_risky\_function\_for\_windows

不要使用windows环境下缓冲区溢出危险函数

