INT16-C. Do not make assumptions about representation of signed integers

Although many common implementations use a two's complement representation of signed integers, the C Standard declares such use as implementation-defined and allows all of the following representations:

- Sign and magnitude
- Two's complement
- One's complement

This is a specific example of MSC14-C. Do not introduce unnecessary platform dependencies.

Noncompliant Code Example

One way to check whether a number is even or odd is to examine the least significant bit, but the results will be inconsistent. Specifically, this example gives unexpected behavior on all one's complement implementations:

```
int value;

if (scanf("%d", &value) == 1) {
  if (value & 0x1 != 0) {
    /* Take action if value is odd */
  }
}
```

Compliant Solution

The same thing can be achieved compliantly using the modulo operator:

```
int value;

if (scanf("%d", &value) == 1) {
  if (value % 2 != 0) {
    /* Take action if value is odd */
  }
}
```

Risk Assessment

Incorrect assumptions about integer representation can lead to execution of unintended code branches and other unexpected behavior.

Recommendation	Severity	Likelihood	Remediation Cost	Priority	Level
INT16-C	Medium	Unlikely	High	P2	L3

Automated Detection

Tool	Version	Checker	Description
Astrée	19.04	**bitop-type**	Partially checked
LDRA tool suite	9.7.1	**50 S, 120 S**	Partially Implemented
Parasoft C/C++test	10.4.2	**CERT_C-INT16-a**	Bitwise operators shall only be applied to operands of unsigned underlying type
PRQA QA-C	9.5	**2940, 2941, 2942, 2943, 2945, 2946, 2947, 2948**	
RuleChecker	19.04	**bitop-type**	Partially checked