PVS-studio

Practise1:

First Error:V557 Array overrun is possible. The 'i' index is pointing beyond array bound.
Type: Bug

It means the array have the possibilities to overflow, the last index in the for loop will be out of the range of the array.

Second Error:error: V614 Uninitialized variable 'value' used.
Type:Bug

It means the Uninitialized variable ‘value’ may have uncertain value, like old data, you will not know what value is inside it.

Third Error:error: V773 Visibility scope of the 'leak' pointer was exited without releasing the memory. A memory leak is possible.
Type: Performance Issues

It means the memory this pointer use will always be there, and can not release it. So the more you call this function, the less memory you can allocate. This is kind of Unnecessary Memory Allocation

Fourth Error: source>:16:1: warning: V799 The 'leak' variable is not used after memory has been allocated for it. Consider checking the use of this variable.
Type: Performance Issues

There is no use of the variable ‘leak’, so the memory allocated to it is useless. This is also a performance issue, kind of Unnecessary Memory Allocation.

Practise 2:

First Errors:V522 Dereferencing of the null pointer 'ptr' might take place.
Type:Bug

The memory allocated to a nullptr is not allowed to visit, if you want to visit it, the program will crash.

Fixing the null pointer dereferencing:Impact on reliability: Dereferencing a null pointer is a common cause of crashes in C++ programs. By checking whether the pointer is nullptr before accessing it, you avoid unexpected crashes and undefined behavior, increasing the reliability of the program.

Second Errors:warning: V557 Array overrun is possible. The value of 'i' index could reach 3.

Type:Bug

It means the program will access the element out of the range of array, it will be uncertain data, will affect the output, make it become a uncertain value.

Accessing memory out of the bounds of an array can lead to corruption of memory, or unpredictable behavior, increasing the reliability of the program.