57 Static Arrays in C++ (std - -array)

C++标准库中用来处理的静态数组的类（不增长的数组）

1. 静态数组

```C++
std::array<int, 5> data;
data[0] = 2;
data[4] = 1;

//C风格的普通数组
int dataOld[5];
dataOld[0];
```

```C++
// 用模板传入std::array的size，避免显式调用
template<int T>
void PrintArray(std::array<int,T>& array)
{
    for(int i=0;i<T;i++)
    {
        std::cout << array[i] << std::endl;
    }
}

// 或是用iterator
for (std::array<int, data.size()>::iterator it = data.begin();
        it != data.end();it++)
    {
        std::cout << *it << std::endl;
    }
```

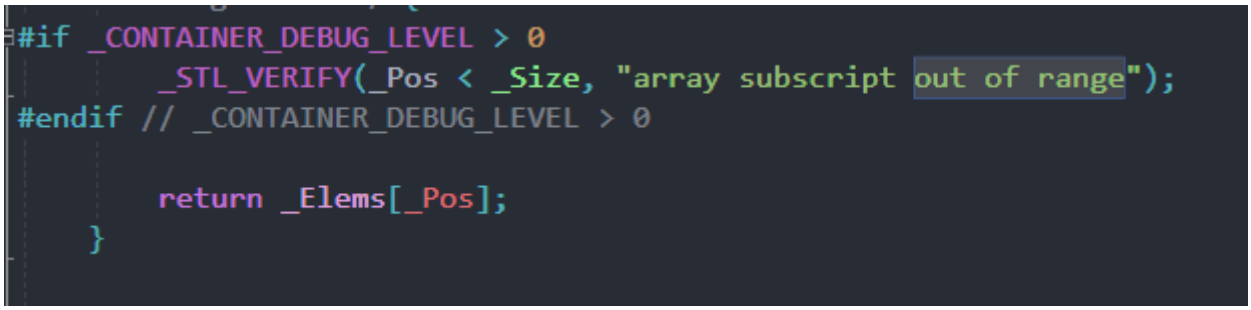
2. 静态数组和普通数组异同

`std::array`和普通数组在内存上形式是一样的，都在栈上分配，不像`std::vector`类是在堆上分配的。

但是`std::array`有边界检查（仅在Debug模式下），在最优化的情况下和普通数组性能是一样的。

`std::array`实际上不存储自己的size，size是你给它的一个模板参数，这意味着调用size function直接返回5而不是返回一个存储在内存中的size变量

可以看到边界检查是在一个宏中的，这意味着只有在那个调试级别才会发生，如果等级为0则返回跟C语言数组工作方式一样的。



你应该开始选择使用`std::array`而不是C语言风格数组，因为它增加了一层调试（在你期望对代码保护时），而且也没有性能成本，还可以让你记录数组的大小。