**1-3-8 动态分配的指针变量第一次使用前必须进行是否为NULL的判别。**

算法设计：

遍历整个代码，找到在代码中定义的指针变量，然后再继续遍历后续代码，找到代码中进行判断NULL的部分，然后找出判空的指针变量，将这个变量和前面代码中定义的变量进行比较，判断两者是否相等，如果相等，那么就判断判空了，否则报出提示信息。

**1-6-16 禁止使用已被释放了的内存空间。**

算法设计：

找到代码中free（）的地方，找出free的指针变量，然后在free之后的代码中进行查询，如果前面释放的指针变量在后续代码中进行了调用，那么我们判断被释放的内存空间被使用了，然后报出提示信息。

**1-6-17 被free的指针必须指向最初malloc、calloc分配的地址。**

遍历整个代码，找到free（）的部分，然后找出free的指针，判断在前面的代码中是否对指针进行了自加，自减，赋值等操作，如果进行了这些操作，那么就判断被free的指针并没有指向最初分配的地址，从而给出提示信息。

**变量在定义的时候建议先初始化**

**使用模式匹配的方法，遍历整个代码，找到代码中定义的变量，然后再变量后续代码，找出对变量进行赋值的代码，从中找出赋值的变量，然后判断定义的变量和后续赋值的变量是否统一，如果不统一，那么提示其需要对其进行初始化。**

**强制类型转换成整型的时候出现数据溢出的问题**

**遍历代码，找到定义整型的代码，然后取出对 Int型变量的赋值，将这个值存为字符串类型，然后进行字符串操作，得到这个值的整数部分，将这个整数和整型变量的取值范围进行比较，如果超出范围，那么就给出溢出的提示信息。**