Purpose:  Overload dynamic memory operators and use STL maps

 Part 1:  Overload operator new (scalar) and new[] (vector).

 Part 2:  Overload operator delete (scalar) and delete[] (vector).

 Part 3:  Write a class to monitor NEW and DELETE usage.  For NEW store the address and the number of bytes being allocated in an STL map. For DELETE, find the address in the STL map, get the bytes allocated for that address, and decrement the byte counter.

 Part 4:  Incorporate your overloaded NEW and DELETE into a standard C++ program to monitor NEW and DELETE usage.  The purpose is to determine if there are any memory leaks in a C++ program that uses NEW and DELETE.

Part 5:  To test your memory checking utility program, insert the C++ Lab3Test.cpp in Documents folder.  Introduce a leak, and see if your memory checker catches it.

Part 6:  Modify the pointers in Part5 to be smart pointers.  Remove all destructors, and Insert your memory tester and check for leaks.

Note:  If you find your overloaded New is recursing, please refer to the Map.cpp example 5.