Joseph Pennington
2912079

In valgrind_test.c, the first error resulted from a conditional jump that depended on uninitialized_variable. To fix this error, I simply initialized the variable at the beginning of the for loop in line 20 by setting it equal to 0.

To fix the still reachable memory leak, I called the free function on still_reachable. This leak was caused by allocating global space but never freeing it before the program ended.

To fix the indirectly lost memory leak, I called the free function on definitely_lost within the for loop. This leak was caused because definitely_lost was out of scope after the for loop ended, and therefore, indirectly lost.

To fix the possibly lost memory leak, I subtracted 4 from possibly_lost and then called the free function on possibly_lost. This leak was caused because the pointer was pointing to the middle of the allocated block. By subtracting 4, it put the pointer back at the beginning of the block.

Lastly, to fix the definitely lost memory leak, I called the free function on the definitely_lost pointer. This leak was caused because the pointer was created using global space but never pointed to anything.