CS 3723 Programming Languages

Assignment #2 – Memory Checker

Delivered Jan 24, 2019. Due Feb 9. 40 points, plus possible 10 point bonus.

1. (20 points) Given the attached **`mem1.c`** program, implement **`getMem()`** and **`freeMem()`** functions. The **`PRECHECK`** and **`POSTCHECK`** constants are to be placed before and after the User Data section to see if the user overwrote them.

2. (10 points) Compile and run the program, and submit both your code and the output from the program. Your program must output some information about errors, such as:

   > Memory Corruption at 0x17c90e8, bytes = 5000, sentinel = 6565656565656565 instead of aeaeaeaeaeaeaeae.

3. (10 points) The **`main()`** program contains four test cases. For each you must answer the following questions:
   a. Did a memory violation occur?
   b. If so, was it caught by the Memory Checker we implemented?

4. Bonus (10 points possible). If you completed parts 1-2-3 correctly, you can work on **`mem2.c`** which maintains a linked list of active memory blocks. If the program never calls **`freeMem()`** on a block, it won't be caught in **`mem1.c`**. It will, however, be caught by **`mem2.c`**, at program termination time. (In this case, don't worry about speed while traversing the linked list.) You should include information about un-freed blocks, such as:

   > Memory of size 5000 bytes, at 0x17c90e8 was never freed.

Note: You may have trouble with the invalid memory accesses. They may require some adjustments to lower the impact of smashing random memory.