## Part II: Discussion (5 points each, 25 points).

1. Which version(s) are unsafe when multi-threaded? \_mmm2.c , mmm3.c, mm6.c \_

Why? \_\_\_If you were to run this program multiple times at the same time then same errors may occur since global variables with the same name will be changed in each occurrence of this program, similar with static variables, you may not get the correct results for your inputs\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Which version is the hardest to read and maintain? \_\_\_\_\_mmm1.c mmm5.c\_\_\_\_\_\_\_

Why? \_\_\_mm1.c has 2 functions that have mostly the same code, so if any significant changes need to be done to 1 function then the other is likely to require those other changes making more difficult to maintain, mmm5.c has a malloc which would require to be freed at the end of the program, which is easy to forget, also pointers are something that need to be taken accounted for when using them, making sure they are referenced or dereferenced correctly\_\_\_

1. Which version has the most redundant code? \_\_\_mmm1.c\_\_\_\_\_\_\_\_\_\_

Why? \_\_The code for small and large is the same except for 1 if statement\_\_\_

1. Which version(s) are closest to call-by-name? \_\_\_\_\_mmm7.c\_\_\_\_\_\_

Why? \_\_\_What is passed is Unevaluated and gets evaluated in the function\_\_\_\_\_\_\_\_\_

1. Which version(s) are closest to call-by-reference? \_\_\_mmm4.c, mmm6.c\_\_\_\_\_\_

Why? \_\_\_A pointer is passed containing the address in the function to be referenced, also there isn’t a return statement in these ones so the values are changed from the address being passed \_\_\_\_