Mathushan Mathyvannan
HW#7 Github

https://github.com/MathushanMathyvannan/CMSC313ClassKidd/tree/main

There has been confusion with classmates about if we are comparing the C code of each file with its assembly but I interpreted the assignment as comparing all three assembly files generated. If this is wrong, please let me know, and I will try to fix it.

For similarities of the assembly files generated from the for addmats.cpp, addmatsSimple.c, and addmatsSubr.c files, they all start the same. They start with .section, .build_version, .syntax unified, .glob, and .code 32. Then they start mentioning functions and the code for them. Also, they use @ for inline comments in all three assembly files. They also repeatedly use mov, ldr, str, add, and cmp in all three files to manipulate stuff. Also, they use bl ___stack_chk_fail as a safety check.

For differences between the assembly code of the two C files, there are many changes as a result of using functions. While the addmatsSimple.c only has the main function, we see in addmatsSubr.c that other functions can also be declared, like _addMatracies:. Also, since addmatsSimple.c writes everything in main, its assembly only has bl for bl ___stack_chk_fail, while addmatsSubr.c uses it to call its function. This makes it so that all of addmatsSimple.c's code is in the main function block, while addmatsSubr.c has the code split into main and _addMatracies.

For differences between the assembly code of the two C files and the C++ file, there are many changes as a result of using a class. Using a class resulted in more assembly code since you need to call more functions compared to not using a class, like the constructors and deconstructors. Also, the C++ file's assembly uses some very strange names, which I don't see in any of the C file's assembly, __ZN15MatrixFunctionsC1Ei. The stack also has more memory allocated and has more code surrounding it before and after.