In this assignment we look further into the usage of malloc() and free() and how they aid in dynamic memory allocation. We can understand the syntax of malloc and know how it returns a pointer value. The free() function indicates that the memory allocation can be freed from the current pointer once its use is complete. Here we create our own malloc() and free() to do the same functions as above but additionally also catch and handle common errors that arise while using these functions. We catch, detect, and analyze these errors and print out valuable user outputs to point out the errnos. By using the ___FILE___ and ___LINE___ printf directives we print information on where the error is found, in the malloc() or free().

We next test the code to check its proper functioning b creating memgrind.c with three workloads. Each of these workloads test the error handler to ensure they work correctly. The two workloads that were self-made are explained in detail in testplan.txt. During the execution of each of these workloads 50 times, their compilation time is recorded and outputted as a sequence.