Write a mergesort program in C++ based on the provided program skeleon (mergesort.cpp). Submit the appropriate program file (.cpp) and documentation file (.doc) via Canvas by the date/time due. (Remember: **do not** submit zip or exe files!) Although you will want to verify the functionality of your program by creating a personal input file, the house elf will use his own file for assessment and grading.

This lab gives you a chance to develop your own mergesort code. And in order to prove that you have cleaned up after yourself properly, use the Valgrind tool to check your code for memory leaks. (Please explain your Valgrind results in your documentation.)

Style and documentation will constitute 20% of the grading for this lab. Documentation within the program is required to follow and reflect your pseudocode, which in this case may follow the example of the provided code. In other words, your pseudocode may be embedded in the program but must be effectively more correct than that given.

<u>Crucial:</u> Make sure your program functions correctly on the UWB Linux servers. This means that it not only compiles, but also produces correct results in Linux using the Gnu C++ tool chain. This portion constitutes 80% of the grading.

The house elf looks forward to your submission!

==

Problem statement::

Write a program that builds on the provided code for mergesort. Implement the merge portion as efficiently as possible. (There is little room for modification of the mergesort function itself, so the real work will be done on merging the halves of the array at each nesting level.)

Consider how you might encapsulate your mergesort module, with the idea of providing it to a customer as a linkable object. In other words, how might you provide your code (without 'main', 'read', or 'write' of course) to someone who wants to sort an array of objects, generalizing the sorting capabilities to any array of objects of user defined type.

Please provide a description of how to do precisely that generalization in a .doc file along with your functional program. Your program does not need to implement this description, but only needs to sort the integer data contained in 'input.txt' which will be provided by the grader.