Introduction:

The goal of this project is to develop a lexical analyzer that identifies keywords, separators, operators, identifiers, and numbers in a file. The program will read the input file, identify each character, and write the results to an output file. This report will outline the design and implementation of the program, as well as its limitations and potential improvements.

Design and Implementation:

To identify keywords, we used a hash table to store the keywords and checked each identifier against the keywords. Separators and operators were stored in a set, which allowed us to easily add or remove separators and operators in the future. We used a finite state machine (FSM) to detect different states of the program, such as reading a keyword, identifier, number, operator, or separator. Regular expressions were used to detect comments, including single-line comments (//) and block comments (/\* \*/). We modified the isdigit() and isalnum() functions to check for the period (.) character to identify floating-point numbers. We also improved user input handling by accepting file names as command-line arguments instead of hard-coding them in the program. Lastly, we wrote out the results to a file instead of printing them to the console.

Limitations and Potential Improvements:

One limitation of the program is that it is only capable of identifying pre-defined characters. To improve this, we can implement a machine learning algorithm to identify new characters that the program may not recognize. Another limitation is that the program does not identify floating-point numbers. To improve this, we can modify the program to recognize floating-point numbers using regular expressions. We can also improve the program's scalability by implementing multi-threading to improve its processing speed. Additionally, we can improve the user interface by providing a graphical user interface (GUI) for easier input and output handling.

Conclusion:

In conclusion, we have designed and implemented a lexical analyzer program that identifies keywords, separators, operators, identifiers, and numbers in a file. The program uses a hash table, a set, and a finite state machine to detect different states and identify different characters. The program has some limitations, but we have proposed potential improvements that can be implemented to make the program more accurate, scalable, and user-friendly.