**Software Requirement Specification**

1. Load Program into Memory:
   1. The program must allow loading a BasicML program into memory, starting at memory location 00.
2. Execute Program:
   1. The program must execute the loaded BasicML program sequentially until a HALT instruction is encountered.
3. Read Input:
   1. The program must read a signed four-digit integer from the user and store it in a specified memory location.
4. Write Output:
   1. The program must write the value stored in a specified memory location to the screen.
5. Load Value into Accumulator:
   1. The program must load a value from a specified memory location into the accumulator.
6. Store Value in Memory:
   1. The program must store the value in the accumulator into a specified memory location.
7. Add Value to Accumulator:
   1. The program must add the value from a specified memory location to the accumulator.
8. Subtract Value from Accumulator:
   1. The program must subtract the value from a specified memory location from the accumulator.
9. Multiply Accumulator by Value:
   1. The program must multiply the accumulator by the value from a specified memory location.
10. Divide Accumulator by Value:
    1. The program must divide the accumulator by the value from a specified memory location, ensuring division by zero is prevented.
11. Branch to Memory Location:
    1. The program must branch (jump) to a specified memory location, updating the program counter.
12. Branch if Accumulator is Negative:
    1. The program must branch to a specified memory location if the accumulator is negative.
13. Branch if Accumulator is Zero:
    1. The program must branch to a specified memory location if the accumulator is zero.
14. Halt Execution:
    1. The program must halt execution when a HALT instruction is encountered.
15. Display Memory Contents:
    1. The program must display the contents of memory within a specified range.
16. Performance:
    * The program must execute BasicML instructions within a reasonable time frame (e.g., less than 1 second per instruction).
17. Usability:
    * The program must provide clear and concise prompts for user input and output, ensuring ease of use for students learning BasicML.
18. Error Handling:
    * The program must handle invalid inputs, invalid memory accesses, and division by zero gracefully, providing meaningful error messages to the user.