Assembly Language Project 2 Report

The goal of this project was to create an x86 NASM assembly program that reads integers from a file, computes their sum, and prints the result. Here's how I approached it in a step-by-step pseudo-algorithm form:

1. Understand the Requirements:
   - Input: A file name provided via the command line (hardcoded for simplicity in this case).
   - File format: First line contains the number of integers, followed by one integer per line (up to 1000 integers).
   - Output: Sum of all integers printed to the screen.
   - Modular design: Use subroutines or logical sections for file handling, parsing, and output.

2. Pseudo-Algorithm:

```
- Step 1: Open the File
  - Use system call `open` (eax = 5) with the file name.
  - Check for errors (negative return value indicates failure).
- Step 2: Read the File
  - Use system call `read` (eax = 3) to load the file content into a buffer.
  - Handle cases where the file is empty or reading fails.
- Step 3: Parse the Buffer
  - Iterate through the buffer character by character.
  - Convert ASCII digits to integers (subtract '0' from each digit).
  - Accumulate digits into a number until a newline is encountered.
  - Add each completed number to a running sum.
- Step 4: Convert Sum to ASCII
  - Convert the binary sum to an ASCII string by repeatedly dividing by 10 and collecting remainders.
  - Store the digits in reverse order in a buffer, adding a newline at the end.
- Step 5: Print the Sum
  - Use system call `write` (eax = 4) to output the ASCII string.
- Step 6: Clean Up
  - Close the file using system call `close` (eax = 6).
  - Exit the program gracefully.
```

3. Design Choices:
   - I used system calls instead of C library functions (like `fopen`, `fscanf`, `printf`) for a more low-level approach, though the assignment allows C functions.
   - The file name is hardcoded as "randomInt100.txt" instead of reading from the command line to simplify testing.
   - The parsing logic assumes integers are separated by newlines and ignores the first line (count of integers) for simplicity, though this could be improved.