Lab 1: Large Integer Arithmetic Expression

Objective: In this lab, you will implement a program that evaluates large integer arithmetic expressions with multiple operands and parentheses. You will work with large integers (up to 100 digits) and ensure the proper handling of basic arithmetic operations, respecting the precedence of operations, including the use of parentheses.

Problem Statement: Given a mathematical infix expression containing large integers (up to 100 digits) and multiple arithmetic operators (`+`, `-`, `*`, `/`), your task is to evaluate the expression and return the result. The expression may also include parentheses to group operations. Your program should handle expressions with multiple operands, correctly respecting the order of operations (PMDAS: Parentheses, Multiplication/Division, Addition/Subtraction).

Requirements:

1. **Input:**
   - A string representing the arithmetic expression. The expression may contain:
   - Large integers (up to 100 digits).
   - The four basic arithmetic operators (`+`, `-`, `*`, `/`).
   - Parentheses to denote precedence in operations.
   - Example Input: (`12345678901234567890 + (98765432109876543210 - 12345678901234567890)) * 100`
2. **Output:**
   - The result of evaluating the arithmetic expression, taking into account the proper order of operations, including the handling of parentheses.
   - Example Output: `11111111111111111110000000000000000000000`.
3. **Tasks:**
   - Replace StudentID with your student ID.
   - Take the input from the file name "tests.txt". Output the results to the file named "output_StudentID.txt".
   - Each line in the file "tests.txt" is an arithmetic expression.
   - Your program must be a C++ program. The cpp file containing the main function must be named with your student ID, i.e., StudentID.cpp, which takes **two command-line arguments** x and y where x=tests.txt and y=output_StudentID.txt, then prints the results of each expression in "tests.txt":

     For example, consider the case when tests.txt contains two expressions as follows:
     ```
     (12345678901234567890 + (98765432109876543210 -
     12345678901234567890)) * 100
     1 + 2 * 4
     ```
     When using the following command line in terminal or cmd (in Windows):
     g++ StudentID.cpp -o StudentID
     StudentID tests.txt output_StudentID.txt (or ./StudentID tests.txt output_StudentID.txt)
     the following results must be shown in the terminal or cmd and written in the file

output_StudentID.txt:
`11111111111111111110000000000000000000000`
9
(must have an endline (/n or endl) here)

- Implement a function that evaluates the arithmetic expression, hopefully ensuring that:
  - Parentheses are evaluated first.
  - Arithmetic operations respect operator precedence: multiplication and division before addition and subtraction.
  - Large integers (up to 100 digits) are handled correctly.
  - Negative integers are correctly processed.
  - Division by zero is properly handled with an appropriate error message.

4. **Example Expressions:**
   - `((12345678901234567890 + 98765432109876543210) * 100) / 100`
   - `10000000000000000000000000000000000000000 + (20000000000000000000000000000000000000000 / 50000000000000000000000000000000000000000)`
   - `((98765432101234567890 * 99999999999999999999) + 12345678901234567890) - (98765432109876543210 / 11111111111111111111)`

Evaluation Criteria:

- You are NOT ALLOWED to use any built-in data structure or library capable of handling large integers or expressions. For instance, `BigInteger` in C++ is prohibited to use.
- Your solution should handle errors gracefully, such as division by zero or malformed expressions, and output appropriate error messages.
- Correctness of results for various test cases, including those with parentheses and multiple operations.
- Proper handling of large integers and negative numbers.
- Correct implementation of operator precedence and parentheses handling.
- Efficient parsing and evaluation of expressions with multiple nested operations.

Report guidelines:

1. Submit your source code and report. All source codes must be placed in a folder named StudentID.zip. The report must be in the pdf format and named as StudentID.
2. Write a report to describe what you have done to finish this assignment.
3. The first page of your report should contain your full name, student ID, and a table that describes which part of your source code is written by yourself and which part is referenced. For the reference part, you must explicitly write the reference.

No.	Percentage understood	Content understood	Percentage Referenced	Content Referenced	Reference Source
1	10%		90%		
2					

4. Any violation of this guideline or the tasks will result in zero.