**Software Requirements Specification (SRS) for Formula Calculation Program**

**1. Purpose**

This SRS document specifies the requirements for a Formula Calculation Program implemented in C++. It defines the functionality, system interactions, user interactions, and constraints for a program that takes user-defined values, performs mathematical calculations based on these inputs, and outputs results in a command-line interface (CLI).

**a. Definitions**

* **Formula Calculation Program**: A command-line application to calculate results based on user inputs x and n using different formulas depending on the value of x.
* **Input Validation**: The program ensures all user inputs meet specified requirements before proceeding with calculations.
* **CLI**: Command-Line Interface for user input and output display.
* **Iterations**: Repeated calculations based on user-defined steps between upper and lower bounds.

**b. System Overview**

The Formula Calculation Program is a CLI application that:

1. Prompts users for input values.
2. Validates those inputs.
3. Performs a specific calculation depending on the values provided.
4. Outputs the results for each iteration across a defined range.

**c. References**

* **C++ Standard Library**: Used for input/output operations and mathematical calculations.
* **Documentation on Command-Line Interface (CLI)**: Provides user interaction guidelines.
* **Mathematical Functions in C++**: For using functions like pow and other mathematical operations.

**2. Overall Description**

**a. Product Perspective**

This program is a stand-alone, non-networked application built to operate independently in any compatible CLI environment. It uses modular functions for input handling, validation, calculation, and output.

**i. System Interfaces**

* **Operating System Compatibility**: Should be operable on any OS that supports C++ (e.g., Windows, Linux, macOS).
* **Compiler Requirements**: Compatible with C++11 or higher.

**ii. User Interfaces**

* **Command-Line Interface (CLI)**: The application operates through the CLI, prompting users to input values and displaying outputs directly in the terminal.

**iii. Hardware Interfaces**

* **Keyboard**: Users will need a standard keyboard to enter values and commands.
* **Display Monitor**: Output results and messages will be displayed on a terminal window.

**iv. Software Interfaces**

* **C++ Libraries**:
  + iostream: For handling input and output.
  + cmath: For performing mathematical operations, including pow.
  + limits: Used for error handling and clearing invalid inputs.

**v. Communication Interfaces**

The program does not require internet or other communication interfaces, as it operates locally within a single CLI environment.

**vi. Memory Constraints**

The program’s memory usage is minimal. It utilizes standard data types and iterates based on user-defined steps, with calculations stored temporarily during processing and outputted directly after each computation.

**b. Design Constraints**

**i. Operations**

* **Get Input**: Prompt user for inputs, including bounds, step size, and a value for nnn.
* **Validation**: Check that values meet specified criteria (e.g., n>7, a<b, non-negative step size).
* **Calculation**: Perform calculations based on whether x<3 or x≥3x.
* **Display Results**: Output results to the terminal, displaying either a single result or a series based on step intervals between bounds a and b.

**ii. Site Adaptation Requirements**

The program requires no specific environment setup beyond a CLI and C++ support, making it adaptable across most computing environments.

**c. Product Functions**

The program performs the following main functions:

1. **Input Collection and Validation**: Ensures user inputs adhere to required conditions (e.g., n>7, non-negative step sizes).
2. **Iterative Calculation**: Based on input values, calculates results for x across a specified range, using different formulas depending on the value of x.
3. **Output Results**: Outputs calculated results for each step iteration.

**d. User Characteristics**

Users are expected to:

* Be familiar with using CLI applications.
* Understand basic mathematical inputs and requirements (e.g., entering valid numbers).
* Be able to follow simple prompts to provide required inputs.

**e. Constraints, Assumptions, and Dependencies**

* **Constraints**: Input n must be greater than 7. Step size should be non-negative.
* **Assumptions**: Users will input values as prompted and follow constraints.
* **Dependencies**: Program depends on C++ standard libraries, requiring a C++ compiler for operation.

**3. Specific Requirements**

**a. External Interface Requirements**

* **Keyboard Input**: Used to enter values for variables x, n, step size and bounds.
* **Console Output**: Displays prompts, validation error messages, and calculation results.

**b. Functional Requirements**

1. **User Input Collection**:
   * Program prompts for bounds a and b, step size, and value n.
   * Repeatedly requests input until valid values are provided.
2. **Input Validation**:
   * Confirms n>7; if not, prompts user to re-enter.
   * Ensures step size is non-negative and that lower bound a is less than upper bound b.
3. **Calculation Logic**:
   * If x<3: Calculates result=5⋅x−2 and adjusts with a double summation based on user-defined bounds.
   * If x≥3: Computes a product and summation and calculates the final result based on these conditions.
4. **Display of Results**:
   * Outputs calculated results for each step within the range a to b or displays a single result if no iteration is needed.
5. **Run Again Prompt**:
   * Asks users if they want to run the program again with different inputs.

**c. Performance Requirements**

* Calculations and output should execute quickly, typically within a few seconds.
* Program responsiveness should allow immediate feedback on input validation errors.

**d. Logical Database Requirement**

* No data storage requirements; the program operates purely on user input and displays real-time calculations.

**e. Software System Attributes**

**i. Reliability**

* The program should handle input errors gracefully, displaying validation messages for incorrect inputs.

**ii. Availability**

* Program is designed to run on any compatible system and should be available as long as the operating environment supports C++.

**iii. Security**

* No specific security requirements; application operates locally with no sensitive data.

**iv. Maintainability**

* Modular functions (e.g., for input handling, validation, and calculation) make it easy to update and debug.

**v. Portability**

* Application should be portable to any operating system with C++ compiler support, with minimal adjustments required.

**f. Environment Characteristics**

**i. Hardware Requirements**

* Desktop or laptop computer with a standard keyboard and display.

**ii. Peripherals**

No additional peripherals required