Assignment Description: In this individual challenge, you will showcase your proficiency in Git version control while working on programming tasks. You'll tackle basic Git operations along with advanced techniques such as branching, merging, rebasing, resolving conflicts, cherry-picking, and reverting changes. Additionally, you will practice deleting files and leveraging other Git commands to enhance your understanding.

Instructions:

1. **Repository Setup**:
    - Create a new repository on GitHub.
2. **Initial Commit**:
    - Clone the repository to your local machine.
    - Create a new C/C++ file named "basic_program.c" or "basic_program.cpp".
    - Write a simple program (e.g., "Hello World") in the file.
    - Commit the changes to the repository.
3. **Branching and Merging**:
    - Create a new branch named "feature-branch".
    - Add a basic programming problem to the file (e.g., calculate the factorial of a number).
    - Commit the changes.
    - Switch back to the main branch.
    - Merge the changes from the "feature-branch" into the main branch.
    - Resolve any conflicts that may arise during the merge.
4. **Rebasing**:
    - Create a new branch named "experimental-branch".
    - Extend the previous programming problem (e.g., calculate the Fibonacci sequence) in the "experimental-branch".
    - Commit the changes.
    - Switch back to the main branch.
    - Rebase the changes from the "experimental-branch" onto the main branch.
    - Resolve any conflicts that may arise during the rebase.
5. **Advanced Git Tasks**:
    - **Cherry-pick**:
        - Identify a commit from another branch (e.g., the "feature-branch") containing a critical fix alongside a programming problem.
        - Cherry-pick that commit onto the main branch.
        - Resolve any conflicts that may arise during the cherry-pick, ensuring the integrated fix retains its functionality with the added problem.

        - Programming Problem Addition: Alongside the critical fix, add a new programming problem to the file (e.g., implement a function to find the nth prime number).
    - **Revert Changes**:
        - Select a commit that introduced an error affecting the solution to a programming problem.
        - Revert the selected commit to restore the previous state of the problem solution.
        - Resolve any conflicts that may arise during the revert, ensuring the restoration of the correct solution.
        - Programming Problem Change: Introduce a faulty commit that breaks the existing solution to the programming problem (e.g., modify the function that calculates the Fibonacci sequence to produce incorrect results)
    - **Reset**:
        - Reset the repository to a previous commit using the soft, mixed, and hard reset options.
        - Document the differences between each reset option.
    - **Squash Commits**:
        - Combine multiple commits into a single commit using interactive rebase.
        - Describe the rationale behind squashing commits and how it improves commit history readability.
6. **Deleting Files**:
    - Delete the "basic_program.c" or "basic_program.cpp" file from the repository.
    - Commit the deletion.
7. **Additional Git Tasks**:
    - **Stashing Changes**:
        - Stash changes made to the "basic_program.c" or "basic_program.cpp" file.
        - Apply the stashed changes to a different branch.