PROGRAM 12

Write the commands to undo the changes introduced by the commit with the ID "abc123".

STEP 1: git revert <COMMIT ID>:

This Git command is used to reverse the changes introduced by a specific commit. It creates a new commit that undoes the changes made by the original commit. How it works:

* Identify the Commit: You need to find the commit ID of the commit youwant to revert. You can use git log to find it.
* Execute the Command: Run the git revert <COMMIT ID> command, replacing <COMMIT ID> with the actual ID of the commit.
* Create a New Commit: Git creates a new commit that reverses the changes. The commit message will typically include information about the original commit and the revert.

Key Points:

* Preserves History: Revert creates a new commit, preserving the original commit in the history. This is useful for undoing mistakes or reverting unwanted changes.
* Resolves Conflicts: If there are conflicts between the revert and the current state of the branch, Git will prompt you to resolve them manually. * Multiple Reverts: You can revert multiple commits by providing their respective IDs.

Example:

git revert 1234567890abcdef

This command will revert the commit with the ID 1234567890abcdef.

Analyzing and Changing Git History with git revert

* Undoing Mistakes: If you realize that a previous commit introduced a bug or unwanted changes, you can use git revert to undo those changes.
* Reverting Specific Changes: You can revert specific commits without affecting other changes in the same branch.
* Creating a Clean History: By carefully using git revert, you can keep your Git history clean and understandable. Important Considerations:
* Careful Usage: Revert can sometimes introduce unintended side effects, especially if the reverted commit is part of a complex series of changes.