**1.** **What is prune in fetch ?**

**Ans :-** **Prune**

The Git prune command is used to delete all the files that are not reachable from the current branch. Unreachable objects are objects that are inaccessible from any of the available references.

The primary use of the Git prune command is to clean the working directory after we finish the work on the project.

If an object or commit is no more reachable from the current branch, the **git prune** command will delete that object or commit from the working directory.

**The git fetch -prune** <remote> command deletes all the objects that are not reachable from the git remote repository.

1. **what is squash and interactive interface of squash ?**

**Ans :- Squash**

To "squash" in Git means to combine multiple commits into one. This can help keep your repository's history more readable and understandable.

.We can do this at any point in time (by using Git's "Interactive Rebase" feature), though it is most often done when merging branches.

**Interactive Interface Squash**

Git interactive interface allows you to change individual commits, squash commits together, drop commits or change the order of the commits.

1. **What is merge conflicts and how to resolve it ?**

**Merge Conflicts :-**

Merge conflicts occur in version control systems, such as Git, when two or more branches of code have diverged, and the system cannot automatically determine how to merge them together. This happens when multiple contributors make changes to the same part of a file or to overlapping lines of code in separate branches. When you attempt to merge these branches, the version control system realizes there is a conflict because it doesn't know which changes to keep and which to discard. It then requires manual intervention to resolve the conflict.

**And how to resolve it :-**

1. Identify the Conflict:-

use the git status

1. Open the Conflicted File:-

<<<<<<< HEAD

// Your changes

=======

// Changes from the other branch

>>>>>>> branch name

1. Resolve the Conflict:-

Manually edit the conflicted file to decide which changes to keep

1. Add the Resolved File:-

git add path/to/conflicted file

1. Commit the Changes:-

git merge --continue

1. Complete the Merge:-

git rebase --continue

1. Push the Change:-

git push origin your\_branch\_name