**Basic Git Questions**

1. **What is Git?**
   * Git is a distributed version control system used to track changes in source code during software development.
2. **What is a repository in Git?**
   * A repository is a storage location for your project, which contains all the files and the entire history of changes.
3. **How do you initialize a Git repository?**
   * git init
4. **How do you clone a Git repository?**
   * git clone <repository\_url>
5. **What is a commit in Git?**
   * A commit is a snapshot of changes made to the files in the repository. It records changes to the repository.

**Intermediate Git Questions**

1. **How do you stage files for a commit?**
   * git add <file\_name> or git add . (to add all files)
2. **How do you commit changes?**
   * git commit -m "commit message"
3. **What is the difference between git pull and git fetch?**
   * git fetch updates the local repository with changes from the remote repository but does not merge them. git pull fetches and merges changes from the remote repository into the current branch.
4. **How do you create a new branch in Git?**
   * git branch <branch\_name>
5. **How do you switch to a different branch?**
   * git checkout <branch\_name> or git switch <branch\_name>

**Advanced Git Questions**

1. **How do you merge branches in Git?**
   * git merge <branch\_name>
2. **How do you resolve merge conflicts?**
   * Identify the conflicted files, manually edit them to resolve conflicts, stage the resolved files using git add, and then commit the changes.
3. **What is git rebase and how is it different from git merge?**
   * git rebase moves or combines a sequence of commits to a new base commit. Unlike git merge, it rewrites the commit history.
4. **How do you delete a branch in Git?**
   * Locally: git branch -d <branch\_name>
   * Remotely: git push origin --delete <branch\_name>
5. **What is a remote repository?**
   * A remote repository is a version of your project hosted on the internet or another network. Examples include repositories hosted on GitHub, GitLab, or Bitbucket.

**Practical Git Scenarios**

1. **How do you revert a commit?**
   * git revert <commit\_hash>
2. **How do you reset your branch to a previous commit?**
   * git reset --hard <commit\_hash> (this discards all changes after the specified commit)
3. **How do you view the commit history?**
   * git log
4. **How do you stash changes in Git?**
   * git stash (to save changes) and git stash pop (to reapply stashed changes)
5. **How do you track a new file in Git?**
   * git add <file\_name>

**Git Collaboration Questions**

1. **How do you fork a repository?**
   * Forking is done through the GitHub (or similar service) interface, typically by clicking the "Fork" button on the repository's page.
2. **How do you handle multiple collaborators working on the same project?**
   * Use branches to work on separate features, regularly pull changes from the remote repository, and resolve conflicts during merges.
3. **How do you protect branches in Git?**
   * Configure branch protection rules in the repository settings on platforms like GitHub, GitLab, etc.
4. **What are Git hooks?**
   * Git hooks are scripts that run automatically in response to certain events, like committing or merging.
5. **How do you create a pull request?**
   * Push your branch to a remote repository, then navigate to the repository on a platform like GitHub and use the interface to create a pull request.

Git branch –check branch \*\* git branch br