1.A

1. **Initialize a Git Repository:**
   * To start a new Git project, you can use the git init command. Here’s how you can do it:
   * cd /path/to/your/directory
   * git init
   * This creates a hidden .git directory within your project folder, which stores all the necessary information for version control.
2. **Add a Simple Text File:**
   * Now that you have a Git repository, let’s add a simple text file. You can create a new file using any text editor of your choice (e.g., nano, vim, or even a graphical editor).
   * For example, to create a file named hello.txt, run:
   * nano hello.txt
   * Add some content to the file and save it.
3. **Stage and Commit:**
   * Next, stage the changes using git add:
   * git add hello.txt
   * Then commit the changes with a meaningful message:
   * git commit -m "Initial commit: Added hello.txt"

And there you have it! You’ve initialized a Git repository, added a text file, and made your first commit. Feel free to replace hello.txt with any other filename you’d like to use.

2.A

Certainly! Let’s create a new branch named ‘feature’ and switch to it. Here are the steps:

1. **Create a New Branch:**
   * To create a new branch in Git, you can use the following command:
   * git branch feature
   * This creates a new branch named ‘feature’. You can replace ‘feature’ with any other name you prefer.
2. **Switch to the New Branch:**
   * To switch to the newly created branch, use the git checkout command:
   * git checkout feature
   * Now you’re on the ‘feature’ branch, and any changes you make will be isolated to this branch.
3. **Make Changes and Commit:**
   * Make your desired changes to the code (e.g., edit files, add new features, etc.).
   * Once you’ve made your changes, stage them using git add:
   * git add .
   * Then commit the changes with a meaningful message:
   * git commit -m "Added new feature: ..."

3.A

1. **Creating the Hotfix Branch:**
   * Suppose you encounter a severe bug in the current production release (let’s say version 1.2.1) that needs an immediate fix. You’ll create a hotfix branch from the master branch:
   * git checkout -b hotfix-1.2.1 master
   * The branch name (hotfix-1.2.1) should reflect the version you’re fixing.
2. **Fix the Issue:**
   * Now that you’re on the hotfix branch, address the bug by making necessary code changes.
   * Commit your changes:
   * git commit -s -a -m "Fixed severe production problem"
3. **Merging Back into Master:**
   * Once the hotfix is ready, merge it back into the master branch:
   * git checkout master
   * git merge --no-ff hotfix-1.2.1
   * The --no-ff flag ensures a merge commit is created, preserving the hotfix branch history.
4. **Merging into Develop (Optional):**
   * To ensure the bugfix is included in future releases, merge the hotfix into the develop branch as well:
   * git checkout develop
   * git merge --no-ff hotfix-1.2.1
5. **Cleanup:**
   * Finally, delete the temporary hotfix branch:
   * git branch -d hotfix-1.2.1