Problem Statement 8:
Git Flow Workflow Explore the Git Flow workflow by initializing a repository following the Git Flow model. Simulate the process of feature development, release, and hotfix using Git Flow commands.

Solution :
The Git Flow workflow is a branching model that provides a set of guidelines for managing branches and releases in a software development project.

It defines specific branches for different purposes, such as feature development, releases, and hotfixes. Here's a step-by-step guide to simulating the Git Flow workflow:

Step 1: Install Git Flow
Ensure that Git Flow is installed on your system. You can install it using package managers like Homebrew, apt, or by downloading the scripts from the official repository.

Step 2: Initialize a Git Repository with Git Flow
```
# Initialize a new Git repository
git init

# Initialize Git Flow
git flow init
```

Follow the prompts during the initialization to set up the default branches for development, feature development, releases, and hotfixes.

Step 3: Feature Development
```
# Start a new feature
git flow feature start my-feature

# Make changes, commit, and finish the feature
git commit -m "Implement feature changes"
git flow feature finish my-feature
```

Step 4: Release

```
# Start a new release
git flow release start v1.0.0

# Bump version, make changes, commit
# This is where you would update version numbers and prepare for release
git commit -m "Prepare for release v1.0.0"

# Finish the release
git flow release finish v1.0.0
```

Step 5: Hotfix

```
# Start a new hotfix
git flow hotfix start hotfix-1.0.1

# Make changes, commit
git commit -m "Fix critical issue"

# Finish the hotfix
git flow hotfix finish hotfix-1.0.1
```

Step 6: View Branches and History

```
# View branches
git branch -a

# View commit history
git log --graph --oneline --all
```