**Question 1: Git and Notepad++ Setup**

git --version

git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

git config --list

alias npp='"/c/Program Files/Notepad++/notepad++.exe"'

git config --global core.editor "'/c/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"

git config --global -e

mkdir GitDemo

cd GitDemo

git init

ls -a

echo "Welcome to Git" > welcome.txt

ls

cat welcome.txt

git status

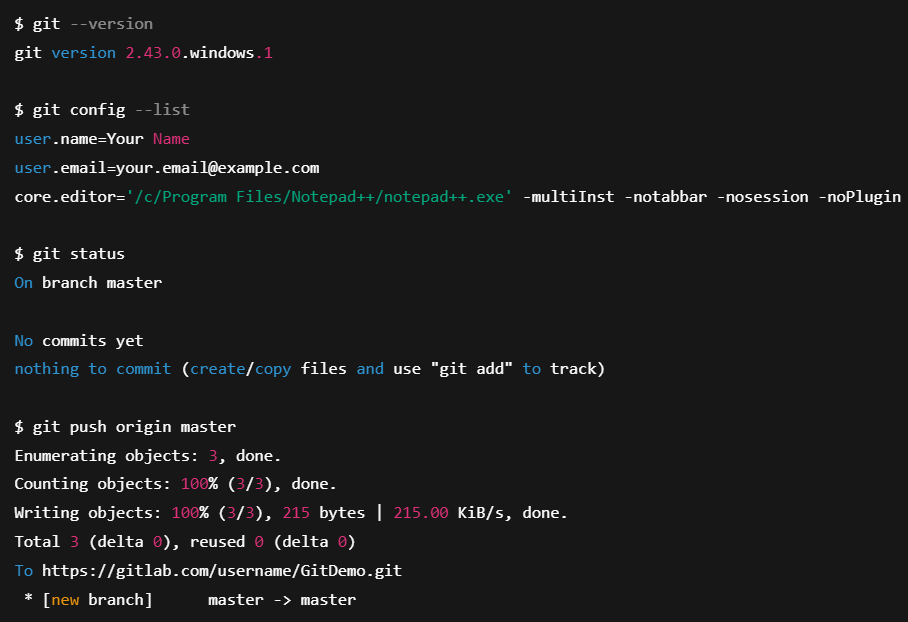
git add welcome.txt

git commit

git remote add origin https://gitlab.com/username/GitDemo.git

git pull origin master

git push origin master



**Question 2: Using .gitignore to Ignore Files and Folders**

touch debug.log

mkdir logs

touch logs/error.log

notepad++ .gitignore

\*.log

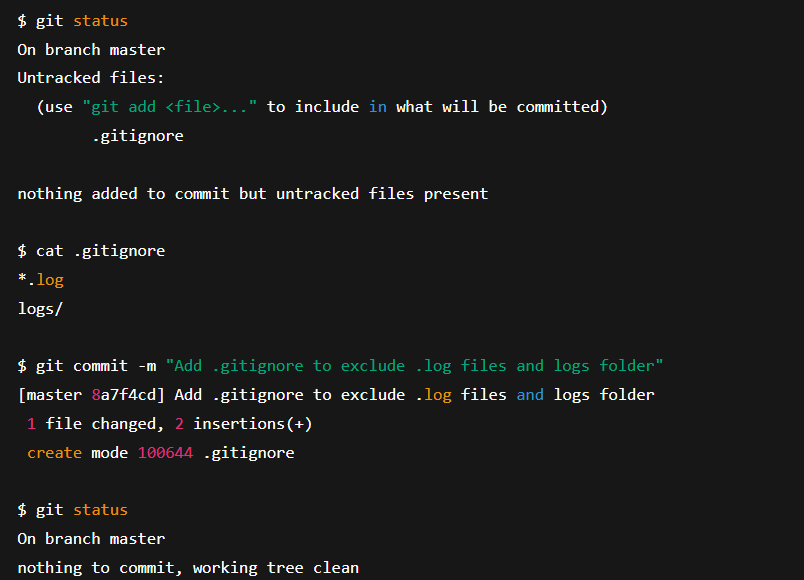
logs/

git status

git add .gitignore

git commit -m "Add .gitignore to exclude .log files and logs folder"

git status



**Question 3: Branching and Merging in Git**

git branch GitNewBranch

git branch -a

git checkout GitNewBranch

echo "Branch file content" > branchfile.txt

git add branchfile.txt

git commit -m "Add file in new branch"

git status

git checkout master

git diff master GitNewBranch

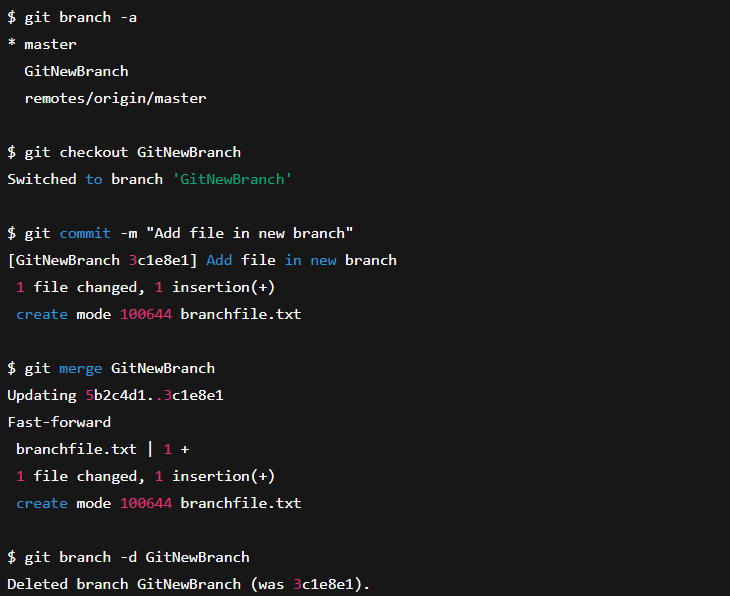
git difftool master GitNewBranch

git merge GitNewBranch

git log --oneline --graph --decorate

git branch -d GitNewBranch

git status



**Question 4: Resolving Merge Conflicts in Git**

git status

git branch GitWork

git checkout GitWork

echo "<msg>Hello from branch</msg>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml in branch"

git checkout master

echo "<msg>Hello from master</msg>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml in master"

git log --oneline --graph --decorate --all

git diff master GitWork

git difftool master GitWork

git merge GitWork

git mergetool

git add hello.xml

git commit -m "Resolve merge conflict for hello.xml"

echo "\*.bak" >> .gitignore

git add .gitignore

git commit -m "Add backup files to .gitignore"

git branch

git branch -d GitWork

git log --oneline --graph –decorate

