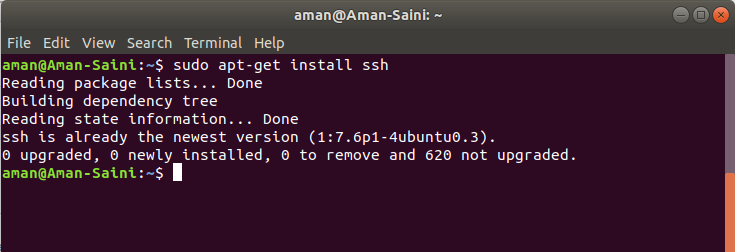
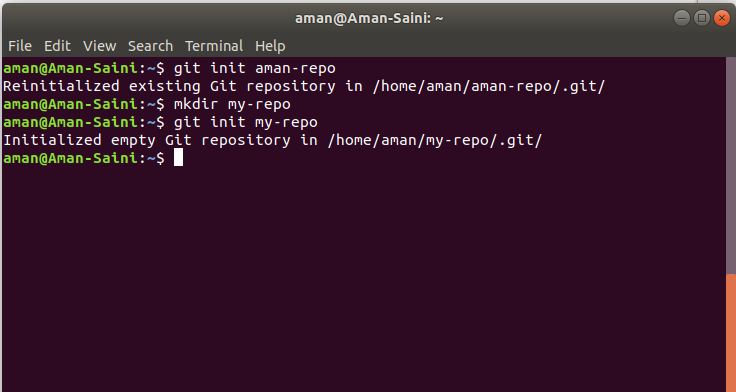
Q 1)-Git Setup

SS-

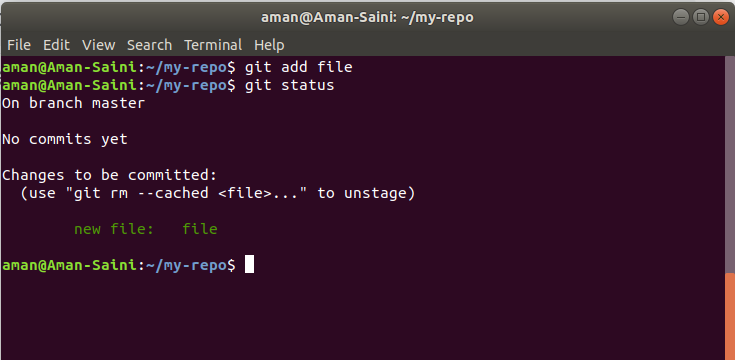
Q 2)-Initialize a Git Repository

SS-



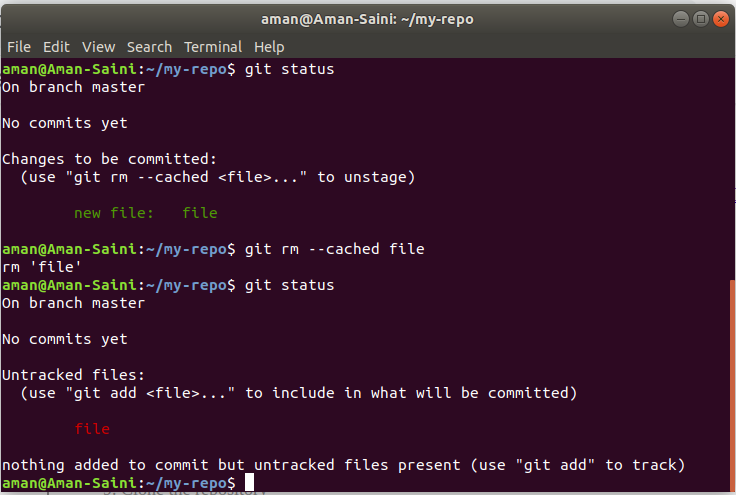
Q 3)-Add files to the repository

SS-



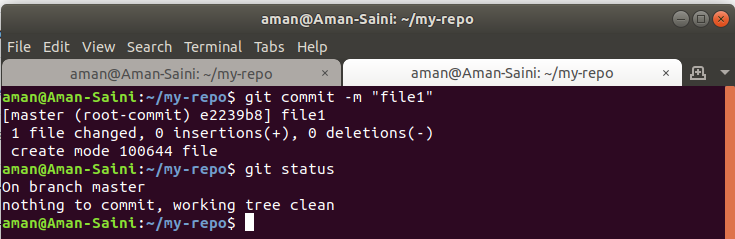
Q 4)-Unstage 1 file

SS-



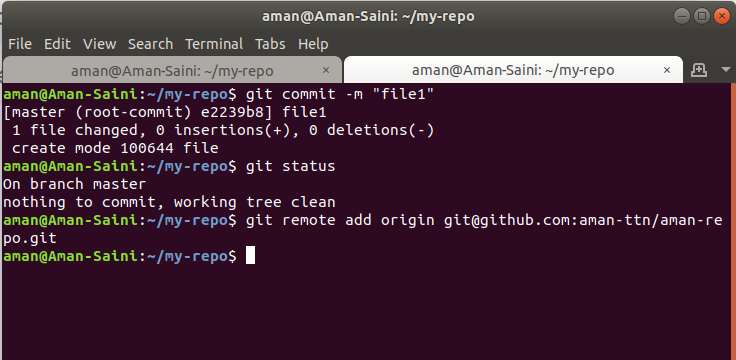
Q 5)-Commit the file

SS-



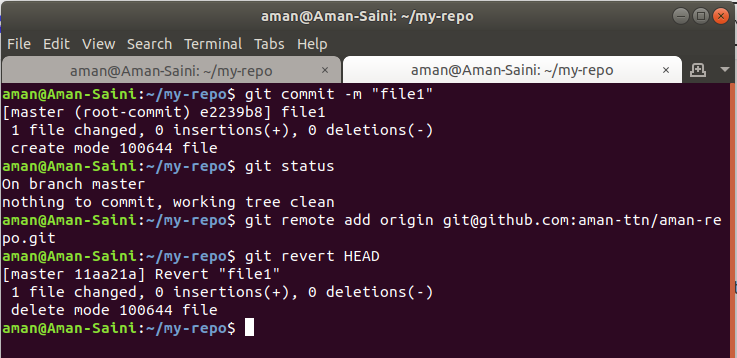
Q 6)-Add a remote

SS-



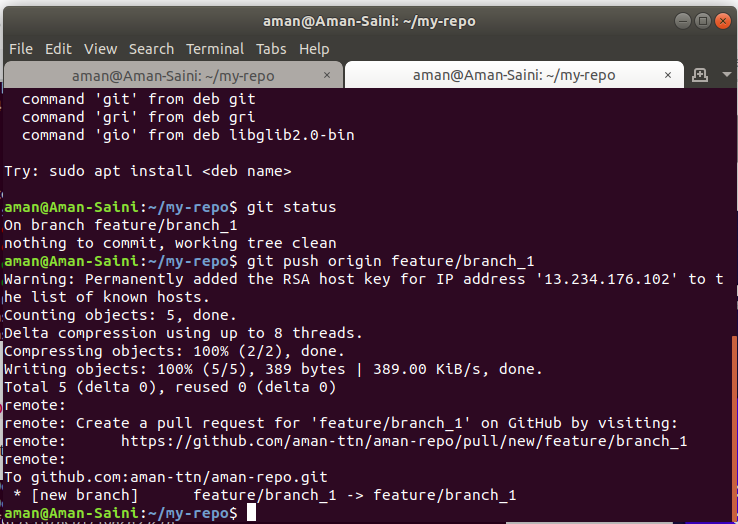
Q 7)-Undo changes to a particular file

SS-



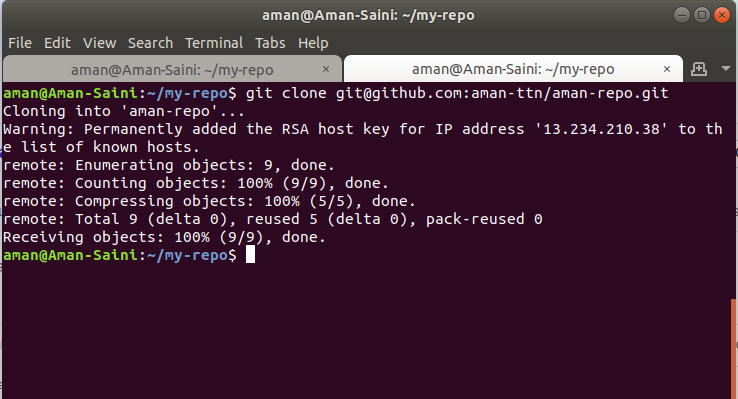
Q 8)-Push changes to Github

SS-



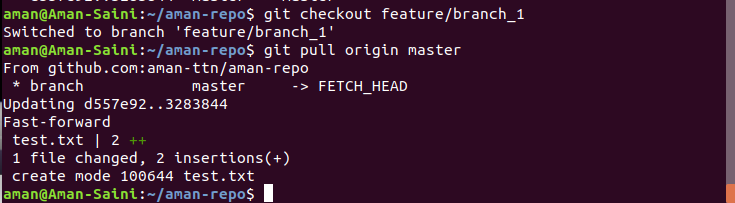
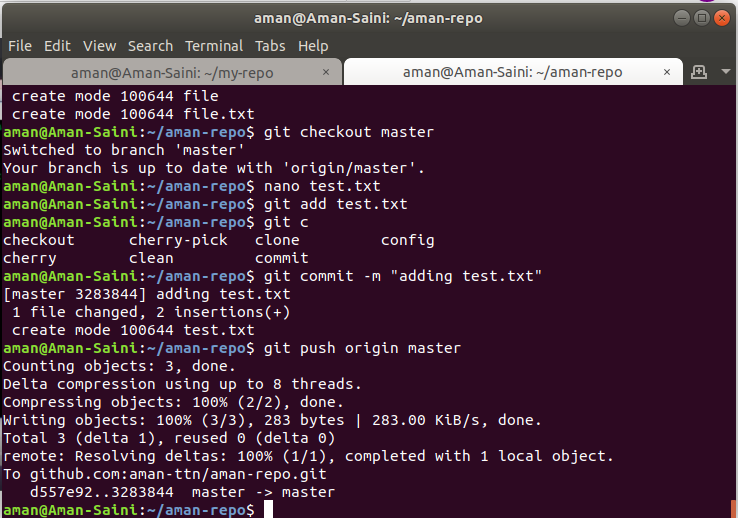
Q 9)-Clone the repository

SS-



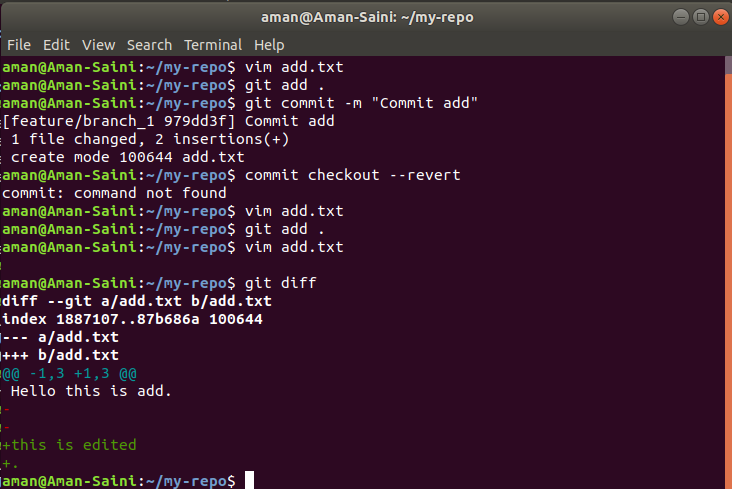
Q 10)-Add changes to one of the copies and pull the changes in the other.

SS-



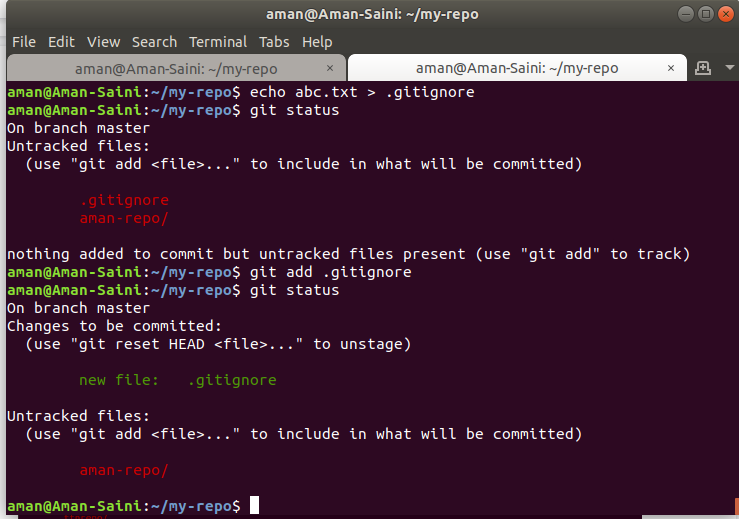
Q 11)-Check differences between a file and its staged version

SS-



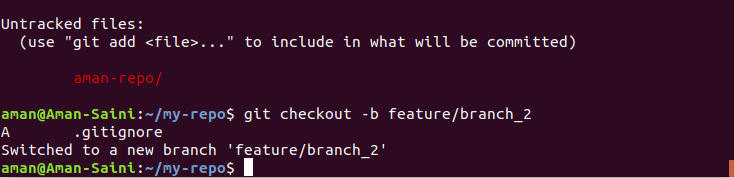
Q 12)-Ignore a few files to be checked in

SS-



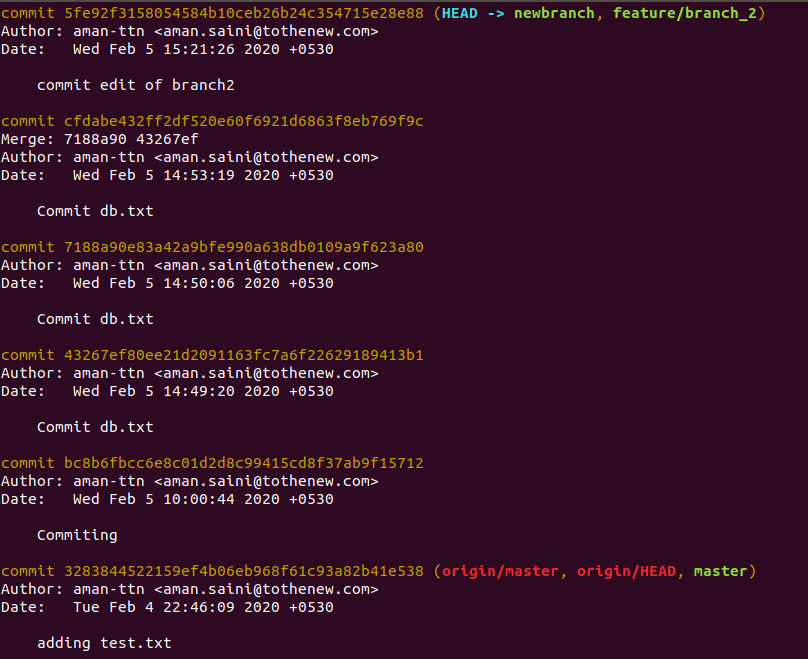
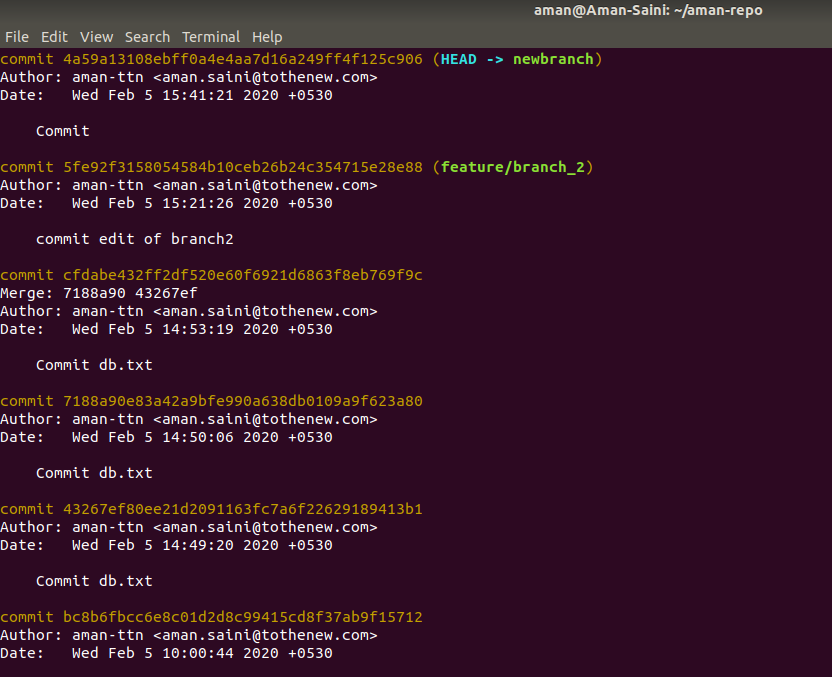
Q 13)-Create a new branch.

SS-



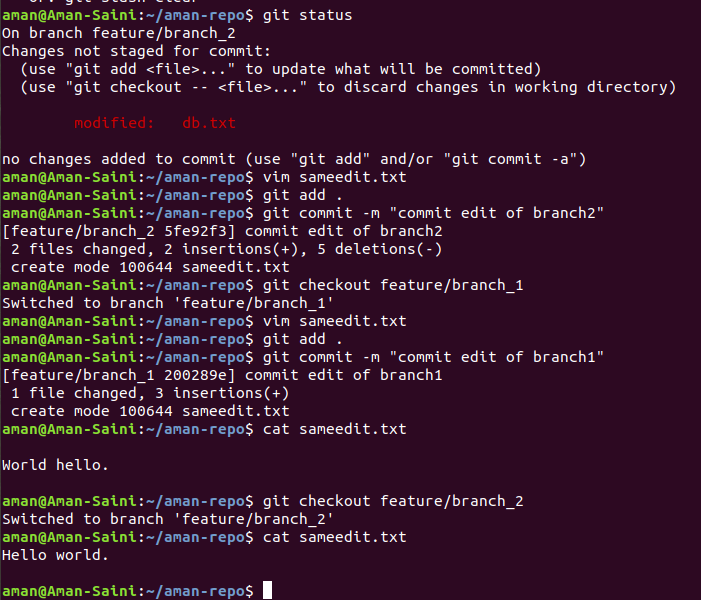
Q 14)-Diverge them with commits.

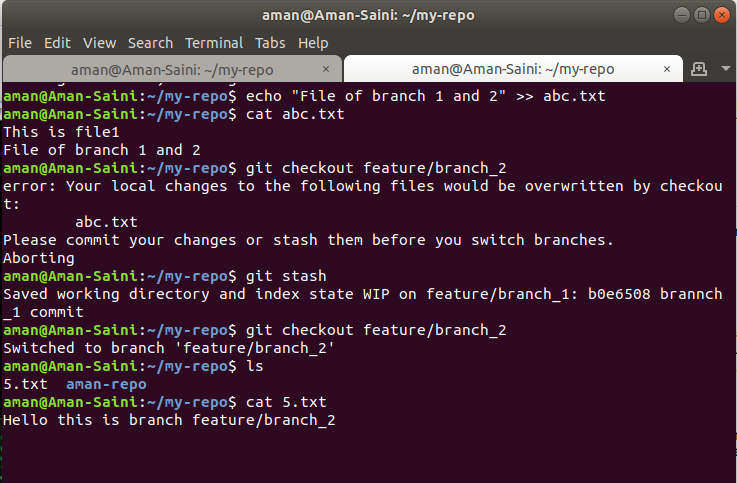
SS-



Q 15)-Edit the same file at the same line on both branches and commit

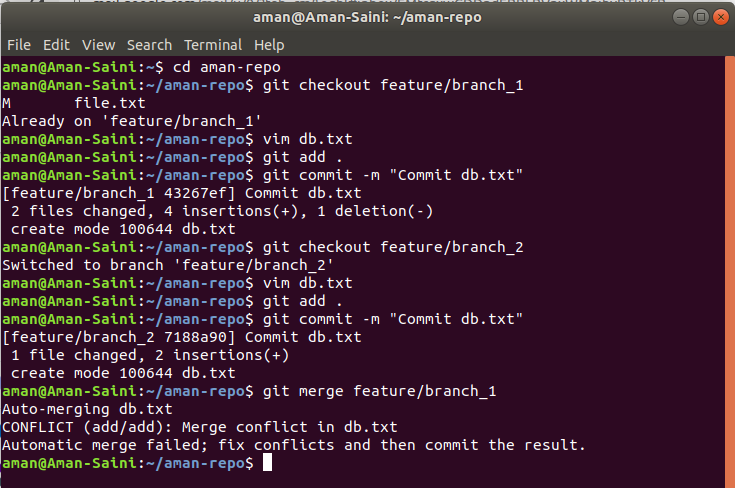
SS-



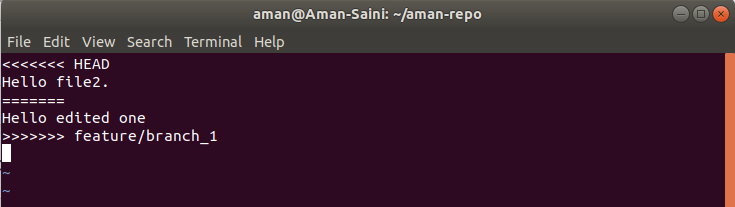


Q 16)-Try merging and resolve merge conflicts

SS-

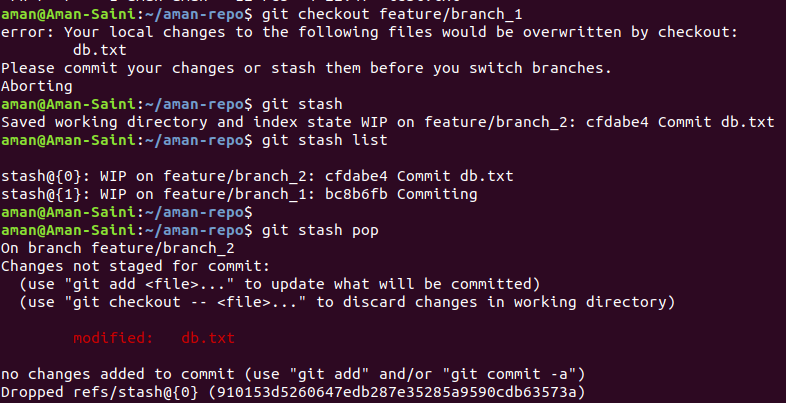


Conflict can removed by editing content of one file-



Q 17)-Stash the changes and pop them

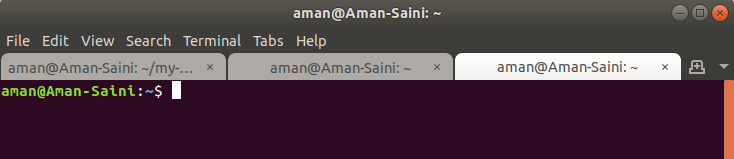
SS-



Q 18)-Add the following code to your .bashrc file : color\_prompt="yes"  
parse\_git\_branch() {  
git branch 2> /dev/null | sed -e '/^[^\*]/d' -e 's/\* \(.\*\)/(\1)/'  
}  
if [ "$color\_prompt" = yes ]; then  
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse\_git\_branch)\[\033[00m\]\$ '  
else  
PS1='\u@\h:\W $(parse\_git\_branch)\$ '  
fi  
unset color\_prompt force\_color\_prompt

SS-

Before adding-



After adding-

