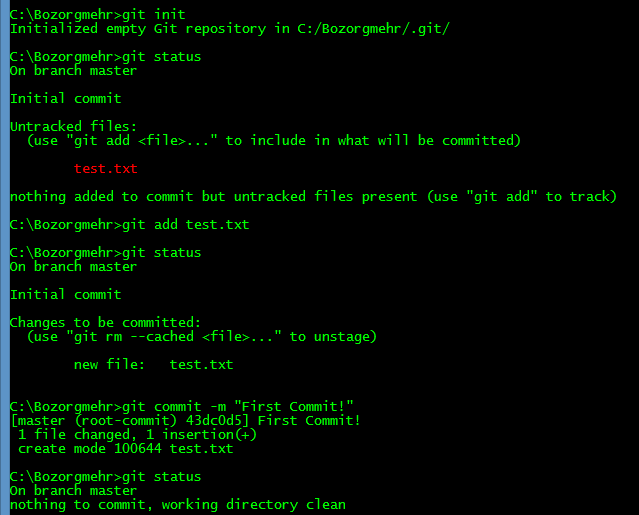
**Software Engineering 2:**

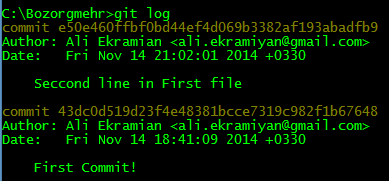
**Git Homework:**

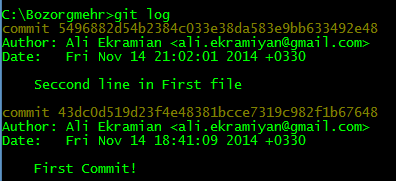
**Ali Ekramian (9012606007)**

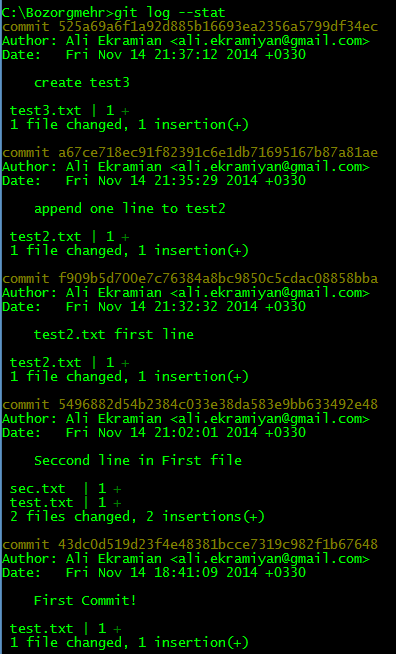
**Git repository address:**

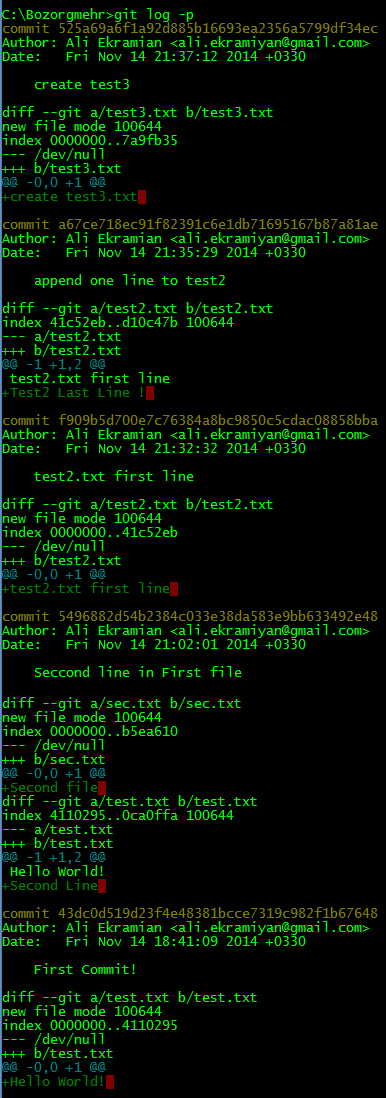
**https://github.com/ekramian/Bozorgmehr.git**

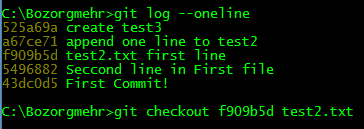
**A.**First when we initial an empty repository, the ‘test.txt’ is an untracked file which git doesn’t follow its changes.  
When we run “git add test.txt” then we add ‘test.txt’ to staging area and we tell git that we want to include this file to our repository and git follow its changes.  
After commit the staging area, git add a snapshot of added files to repository history and working directory would be clean of any uncommitted changes.

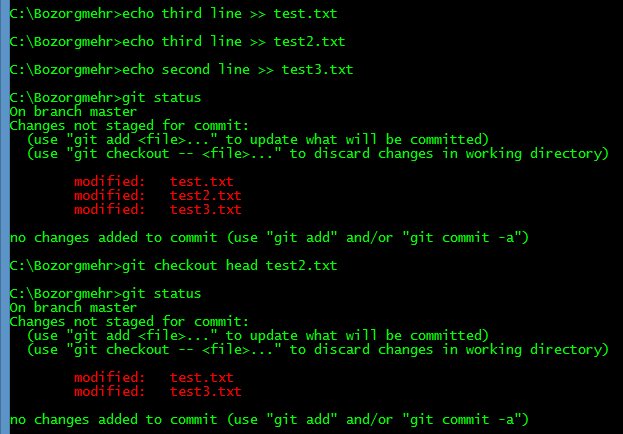
**B.  
**  
We have 4 line for each commit, display commit ID, Author, commit Date and commit message.

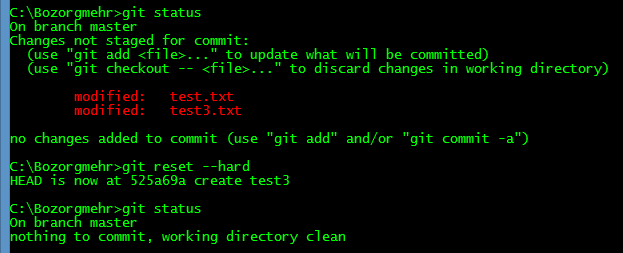
**C.**The last commit is replaced with a new commit that have the same snapshot of working directory of removed commit including ‘sec.txt’ file. This new commit have the same author, date and message (we can change message when we run ‘git commit --amend’) but it has a different commit ID (so it’s new commit and the last commit is deleted).

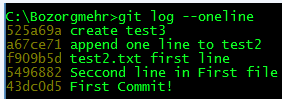
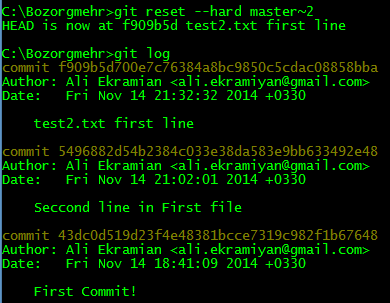
**D.**We use ‘git log --stat’ or ‘git log -p’ for most detailed view.  




**E1.**We use ‘git checkout’ command:  


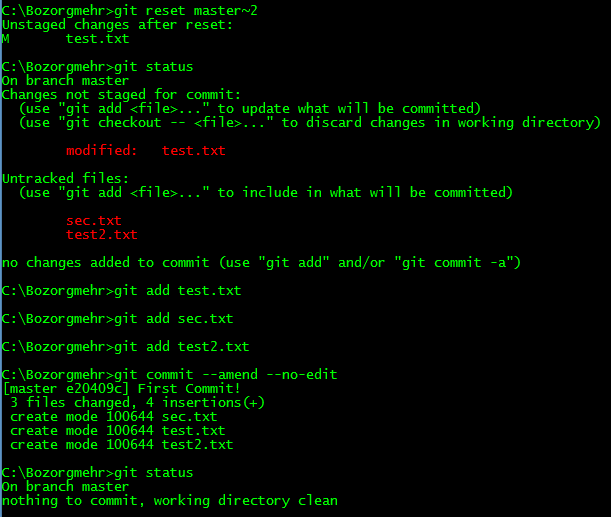
**E2.**We use ‘git checkout head tets2.txt’  


**F.**We run ‘git reset --hard’  


**G.**We run “git reset --hard master~2”. Thins remove two last commits.  
  


**H.**Run:

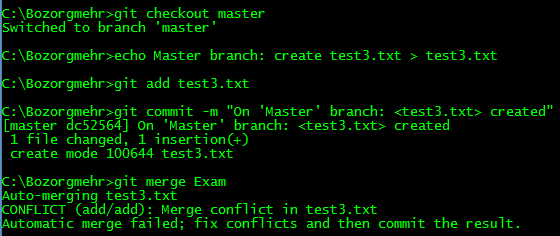
git reset master~2  
git add test.txt  
git add test2.txt  
git add sec.txt  
git commit –amend –no-edit

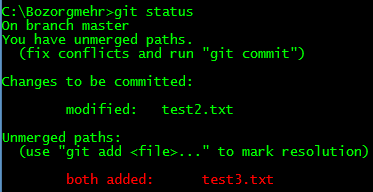
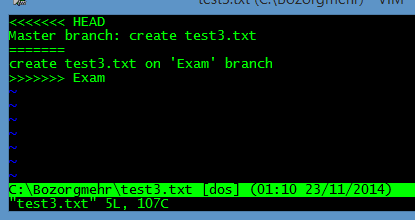
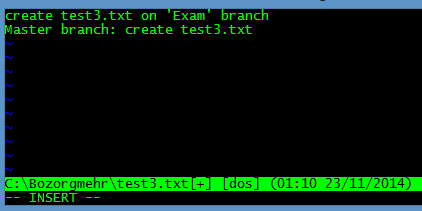


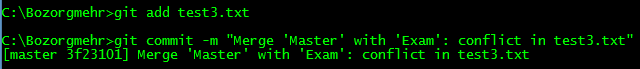
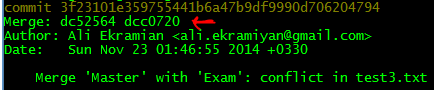
**I.**

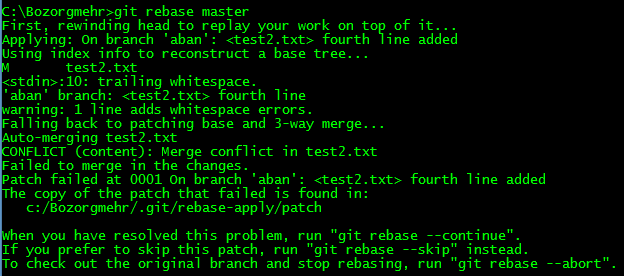
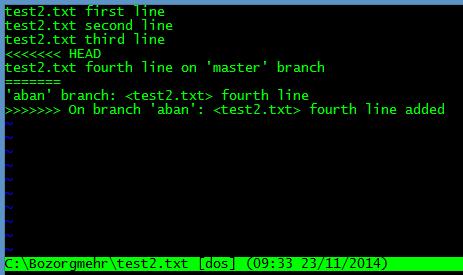
git branch

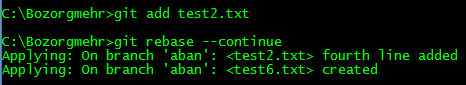
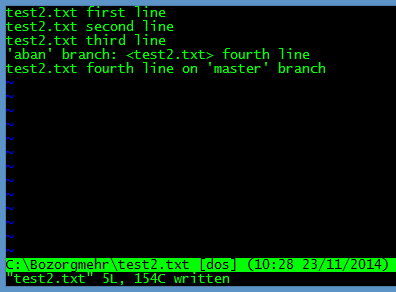


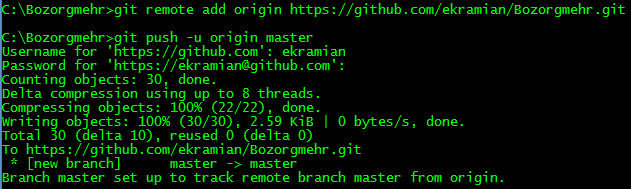
**J.**

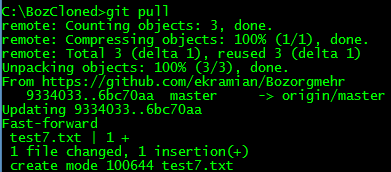
We have merge conflict in test3.txt. For continue merging, we should resolve conflicts: we just open files, edit it, and add it to staging area.  
  
  


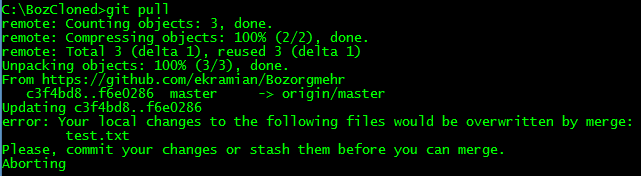
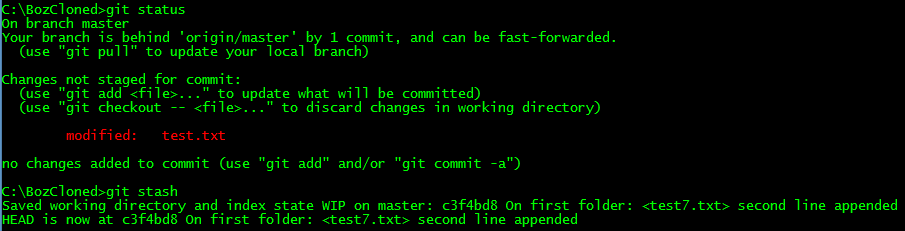
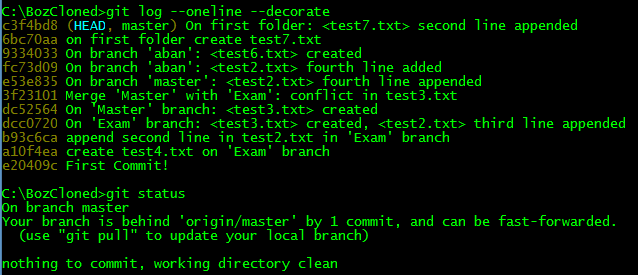
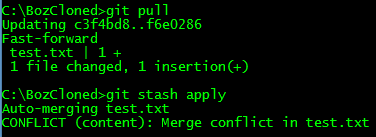
  
After merging, when we check commits log, we see that last commit is the merge commit:  


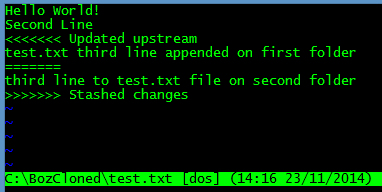
**K.**We get conflict message: “CONFLICT (content): Merge conflict in test2.txt”. We edit ‘test2.txt’ to resolve conflict, add it to staging area and run ‘git rebase --continue’  
  




**L.**We use ‘git remote …’ command to add connection to github repository, and we use ‘git push …’ to save changes on github server.  


**M.**After running ‘git pull’, test7.txt will create.  


**N.**On first run on git pull, git abort pulling because of same change on same file on different repositories.  
After run ‘git stash apply’, again we get merge conflict message in test.txt. we should solve conflicts and commit it.  
  
  
  
  



**O.**