[CS200]-STT II **Fall 2020-21**

Assignment: III

Name: Shubham Gupta RollNo: 11941140 Email: shubhamgupta@iitbhilai.ac.in

Collaborators Names: Aayush Deshmukh(11940010), Santaz Sahiti(11940230)

Question 3,
We are given the case of the initial commit tree and need to convert this to the final git commit tree. The git command used in this transformation will be:

-> git reset –hard

This command is extremely useful as it removes git commits in a manner in which they never existed. We can see that commits B1, B2 , B3, B4, B5, B6 are removed from the commit tree as if they had never existed. Here this reset command comes to our rescue.
Post, removing of commits, we will just add commit H1 to the master and we are done.
The following demonstration of git flow will show the conversion:
The initial commmit tree has been built then and the HEAD of branch ”long” is at B6 and
C1 is the first commit,

-> git reset –hard HEAD∼6

(This command will ensure you will remove 6 commits from the commit that the HEAD is pointing. On completetion of this commit, the HEAD of the branch ”long” is pointing at C1 and has removed commits B1, B2 , B3, B4, B5, B6)
Using checkout to switch to master and the HEAD will point to master,

-> git checkout master

The ”long” branch is at C1 and we need same branch for H1, so we will delete the branch

-> git branch -d ”long”

Making a new branch with the name ”long” commit on that branch with H1,

-> git checkout -b ”long”
-> git commit -m ”H1”

On making of the commit of H1, we are done.
You can verify that on completion of this git work flow, we have obtained the final git commit tree.