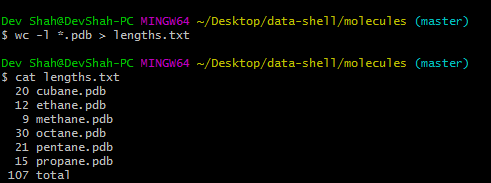
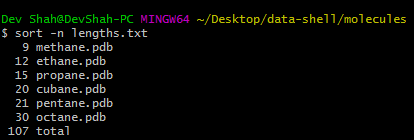
Bash [Unix shell: It is a command-line interpreter or shell that provides a traditional Unix-like command line user interface] –

1. pwd : get current directory
2. ls : to list directory contents
3. ls -l : more details of the contents
4. ls -f :
5. cd : change directory (goes back to home)
6. cd.. : goes back one level
7. curl : to get or send a webpage or API
8. mkdir : to create a new directory
9. nano/vi : to create files
   1. i: insert
   2. :wq : to get out of the file
10. cat : displays content of the file
11. rm : delete file
12. rm -r -i : to delete a folder
13. mv : move file (if in the same directory then it renames the file)
14. head : gives top results
15. wc : word count in line, word, char format
16. \* : wildcard to get matches with similar patterns written after \*
17. \*[AB].txt : Returns files having A or B in the directory

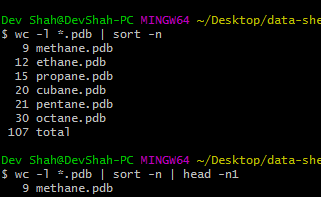
1. > : to redirect some data or answer into a file



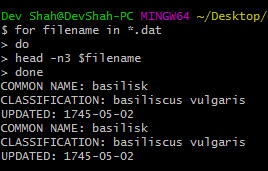
1. sort -n : sort by number



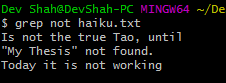
1. | (pipe) : to give multiple commands



1. For loop :-



1. bash filename.sh : To run a shell script
2. grep : to search files or strings in a file



* 1. grep -w : To get exact match of the string mentioned after grep -w
  2. grep -n -w : to get line number for the matches
  3. grep -n -w -v : to get lines which do not have the match

1. find :
2. ls -l -a : to see hidden files
3. touch : creates multiple empty files

**GIT commands** –

1. git config --global user.name “name” : to set name initially
2. git config --global user.email “email” : to set email initially
3. git config --global colors.ui auto : to set colors
4. git init : to initialize repo
5. git status : to check the status of the directory to find any files which are modified/deleted
6. git add : just like a checkbox for adding that file
   1. git add \* : add all files
   2. git add . : add all files in the current directory
7. git checkout -- “filename” : to discard changes in the file
   1. git checkout HEAD~1 : to go back one version, like the latest change never happened
   2. git checkout ‘version hash’ : to go back to the mentioned version
8. git log: Returns a log file for all changes/commits done
9. git rm “filename” : remove file
10. .gitignore : hidden git file, any files in this folder will be ignored by git
11. git commit --m “” : commit to the repo
12. git remote add origin “url” : to link with the github account and repo
    1. git remote rm origin : to remove origin path so that you can set a new one
    2. git remote add $source name $source origin : to remote add another repo
13. git push -u origin master : push all changes to the server
14. git pull : to get changes made on server to local
    1. Merges automatically if both files have changes but not on the same line else will give a conflict
15. git clone : clone entire repo/folder on local
16. git fetch $sourcename : to fetch code from another dir
17. git merge –allow-unrelated-histories $sourceBranchName : Merges changes into new repo

* Fork a fellow contributor’s code -> make changes on your own copy -> initiate a pull request to merge the changes -> hope that the person will accept the request :P
* Always pull first -> make changes -> push.