/* GIT-GITHUB (chp1-GIT)

GIT - It is a version control system i.e.
the way to save the history of changes
made in your file.

#ALL THESE ARE TO BE WRITTEN IN cmd OR GITBASH.
#REPOSITORY - It is a project that can be tracked
with help of Git.
#To clear the entire command terminal press cmd+k

1. To check VERSION - git --version
   of GIT

2.To set CONFIG i.e our NAME and EMAIL -

   i. NAME - git config --global user.name "Rohit Pandey"
   ii. EMAIL -git config --global user.email "alpharohit777@gmail.com"

3. TO check whether NAME or EMAIL is properly set -

   i. NAME - git config --global user.name
   ii. EMAIL -git config --global user.email

4. To edit NAME or EMAIL - git config --global --edit
   given at first.

5. To make NEW FOLDER and to CHANGE DIRECTORY -

   i. MAKE NEW FOLDER - mkdir foldername
   ii. CHANGE DIRECTORY - cd directory or foldername

6. To make a folder - git init
   in Git repository

7. To know the current status   - git status
   or changes in your repository.

Note : A. Inside Git we first need to
          track a file before commiting it.

       B. In stagging area we hold changes
           before comitting it.

       C. Flow of file in GIT -

```
  DIRECTORY(FOLDER) ----------> STAGGING AREA ----------
>REPOSITORY
```

8. To Track a file and ADD - git add Sum.java(filename)
   it in a stagging area

9. To commit a file in repo -

```
git commit -m "initial commit"
              OR,
git add. -It is used to commit all existing files
```

10. To see a detailed info - git log
    about comitted file

BRANCH CONCEPT.

A GIT consists a head branch which is called as the master branch from which we can create multiple branches.

In production their are multiple branches in which people create new branches to work on their feature.

As a result the feature on which we are working remains their in the new branch and the other features doesn't have any impact of that uncomplete feature.

```
  11. To go/move to different   -   git checkout hascode
      (commit/changes/branches)
      to work.


  12. To go back to  - git checkout master
      total changes


  13. To check your branch - git branch


  14. To create a new branch - git branch branchname


  15.To make a new branch     -  git checkout -b rohit/multiply
     and checkout it at
     the same.

     where rohit/multiply is new branch name



  16.Merging files from  -  git merge rohit/multiply (branchname)
    another branch to one
    branch.



  NOTE: To ignore files from getting  into git
        ecosystem or getting tracked by git we
        create a file called (.GITIGNORE) and
        store those files in it.

        (.GITIGNORE)can also be stored inside
        (.GITIGNORE) to ignore it.

 */
//see lectures from  22:00
```

/* GIT-GITHUB (chp2-GITHUB)

GITHUB - It is platform that help people solve problem by building software together.

1. STEPS TO PUSH A LOCAL REPOSITORY TO REMOTE GIT REPOSITORY.

STEP1: change directory or folder name.

```
|| cd directory name ||
```

STEP2: To make a Local repo folder

```
|| git init ||
```

STEP3: Then add all files of directory to stagging area.

```
|| git add index.html ||
```

STEP4: To commit files from stagging area.

```
|| git commit -m "initial commit" ||
```

STEP5: To add remote origin or git repository where you want to push your local repository.

```
|| git remote add origin repolink ||
```

STEP6: To push our files in master or main branch in Git repository.

```
|| git branch -M master/main ||
```

STEP7: To push all final changes to file master or main.

```
|| git push -u origin master/main||
```

NOTE: To know our current origin - git remote -v

2. OPENSOURCE CONTRIBUTION STEPS...

A.FORK
B.CLONE
C.CHANGES
D.PUSH
E.PULL REQUEST.

A. FORK - It makes a copy of repository of another person in your github.

B. CLONE - To clone all files from github to make changes.

```
command -  cd ..
           git clone <repo urllink>
```

C. To see the changes we have done - cat filename

D. After that work on that clone file then add it to stagging area and then commit it in a new branch created by you , not in master branch. And then push it to your GitHub.

E. PULL REQUEST - GO TO CONTRIBUTE
AND CLICK ON OPEN PULL REQUEST.
AND THEN CREATE A PULL REQUEST.

REVIEW CHANGES THEN APPROVE IT,
THEN MERGE PULL REQUEST

*/