Q1.

```
trgl1@PUNHDCDT054 /c

$ mkdir test


trgl1@PUNHDCDT054 /c

$ cd test


trgl1@PUNHDCDT054 /c/test

$ mkdir A


trgl1@PUNHDCDT054 /c/test

$ mkdir B


trgl1@PUNHDCDT054 /c/test

$ mkdir C


trgl1@PUNHDCDT054 /c/test

$ mkdir D


trgl1@PUNHDCDT054 /c/test

$ cd A


trgl1@PUNHDCDT054 /c/test/A

$ git init --bare

Initialized empty Git repository in c:/test/A/


trgl1@PUNHDCDT054 /c/test/A (BARE:master)

$ cd ..


trgl1@PUNHDCDT054 /c/test

$ cd B
```

```
trgl1@PUNHDCDT054 /c/test/B
$ git init
Initialized empty Git repository in c:/test/B/.git/

trgl1@PUNHDCDT054 /c/test/B (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd C

trgl1@PUNHDCDT054 /c/test/C
$ git init
Initialized empty Git repository in c:/test/C/.git/

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd D

trgl1@PUNHDCDT054 /c/test/D
$ git init
Initialized empty Git repository in c:/test/D/.git/

trgl1@PUNHDCDT054 /c/test/D (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ git clone --bare /c/test/A /c/test/E
Cloning into bare repository 'c:/test/E'...
```

```
done.
```

Q2.

```
trgl1@PUNHDCDT054 /c/test
$ cd B

trgl1@PUNHDCDT054 /c/test/B (master)
$ touch info.txt

trgl1@PUNHDCDT054 /c/test/B (master)
$ echo "Project">info.txt

trgl1@PUNHDCDT054 /c/test/B (master)
$ add info.txt
bash: add: command not found

trgl1@PUNHDCDT054 /c/test/B (master)
$ git add info.txt
warning: LF will be replaced by CRLF in info.txt.
The file will have its original line endings in your working directory.

trgl1@PUNHDCDT054 /c/test/B (master)
$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
```

```
#
#       new file:   info.txt
#

trgl1@PUNHDCDT054 /c/test/B (master)
$ git commit info.txt -m 'for admin'
warning: LF will be replaced by CRLF in info.txt.
The file will have its original line endings in your working directory.
[master (root-commit) 1f2b026] for admin
warning: LF will be replaced by CRLF in info.txt.
The file will have its original line endings in your working directory.
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 info.txt


trgl1@PUNHDCDT054 /c/test
$ cd B

trgl1@PUNHDCDT054 /c/test/B (master)
$ git branch Development

trgl1@PUNHDCDT054 /c/test/B (master)
$ git branch Testing

trgl1@PUNHDCDT054 /c/test/B (master)
$ git branch
  Development
  Testing
* master
trgl1@PUNHDCDT054 /c/test/B (master)
$ git push /c/test/A Development
```

```
Total 0 (delta 0), reused 0 (delta 0)

To c:/test/A

 * [new branch]      Development -> Development


trgl1@PUNHDCDT054 /c/test/B (master)

$ git push /c/test/A Testing

Total 0 (delta 0), reused 0 (delta 0)

To c:/test/A

 * [new branch]      Testing -> Testing


trgl1@PUNHDCDT054 /c/test/B (master)

$ cd ..


trgl1@PUNHDCDT054 /c/test

$ cd A


trgl1@PUNHDCDT054 /c/test/A (BARE:master)

$ git branch

  Development

  Testing

* master
```

Q3.

```
trgl1@PUNHDCDT054 /c/test/B (master)

$ git checkout Development

Switched to branch 'Development'


trgl1@PUNHDCDT054 /c/test/B (Development)
```

```
$ touch dev1.txt

trgl1@PUNHDCDT054 /c/test/B (Development)
$ echo "f dev1">dev1.txt

trgl1@PUNHDCDT054 /c/test/B (Development)
$ git add dev1.txt
warning: LF will be replaced by CRLF in dev1.txt.
The file will have its original line endings in your working directory.

trgl1@PUNHDCDT054 /c/test/B (Development)
$ git commit dev1.txt -m 'ff dev1'
warning: LF will be replaced by CRLF in dev1.txt.
The file will have its original line endings in your working directory.
[Development e1e0dd5] ff dev1
warning: LF will be replaced by CRLF in dev1.txt.
The file will have its original line endings in your working directory.
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 dev1.txt

trgl1@PUNHDCDT054 /c/test/B (Development)
$ git checkout Testing
Switched to branch 'Testing'

trgl1@PUNHDCDT054 /c/test/B (Testing)
$ touch tes1.txt

trgl1@PUNHDCDT054 /c/test/B (Testing)
$ echo "f tes1">tes1.txt

trgl1@PUNHDCDT054 /c/test/B (Testing)
```

```
$ git add tes1.txt
warning: LF will be replaced by CRLF in tes1.txt.
The file will have its original line endings in your working directory.

trgl1@PUNHDCDT054 /c/test/B (Testing)
$ git commit tes1.txt -m 'ff tes1'
warning: LF will be replaced by CRLF in tes1.txt.
The file will have its original line endings in your working directory.
[Testing 22441af] ff tes1
warning: LF will be replaced by CRLF in tes1.txt.
The file will have its original line endings in your working directory.
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 tes1.txt

trgl1@PUNHDCDT054 /c/test/B (Testing)
$ git checkout master
Switched to branch 'master'

trgl1@PUNHDCDT054 /c/test/B (master)
$ git push /c/test/A Testing
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 273 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
To c:/test/A
   1f2b026..22441af  Testing -> Testing

trgl1@PUNHDCDT054 /c/test/B (master)
$ git push /c/test/A Development
```

```
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 276 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
To c:/test/A
   1f2b026..e1e0dd5  Development -> Development

trgl1@PUNHDCDT054 /c/test/B (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd C

trgl1@PUNHDCDT054 /c/test/C (master)
$ git pull /c/test/A Development
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From c:/test/A
 * branch            Development -> FETCH_HEAD

trgl1@PUNHDCDT054 /c/test/C (master)
$ git branch
* master

trgl1@PUNHDCDT054 /c/test/C (master)
$ ls
dev1.txt  info.txt
```

```
trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd D

trgl1@PUNHDCDT054 /c/test/D (master)
$ git pull /c/test/A Testing
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From c:/test/A
 * branch            Testing    -> FETCH_HEAD

trgl1@PUNHDCDT054 /c/test/D (master)
$ git branch
* master

trgl1@PUNHDCDT054 /c/test/D (master)
$ ls
info.txt  tes1.txt
```

Q4.

```
trgl1@PUNHDCDT054 /c/test/D (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
```

```
$ cd C

trgl1@PUNHDCDT054 /c/test/C (master)
$ touch MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/C (master)
$ echo "f MyJavaCode" >MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/C (master)
$ add MyJavaCode.txt
bash: add: command not found

trgl1@PUNHDCDT054 /c/test/C (master)
$ git add MyJavaCode.txt
warning: LF will be replaced by CRLF in MyJavaCode.txt.
The file will have its original line endings in your working direct

trgl1@PUNHDCDT054 /c/test/C (master)
$ git commit MyJavaCode.txt -m 'ff MyJavaCode'
warning: LF will be replaced by CRLF in MyJavaCode.txt.
The file will have its original line endings in your working direct
[master 43fab61] ff MyJavaCode
warning: LF will be replaced by CRLF in MyJavaCode.txt.
The file will have its original line endings in your working direct
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
```

```
$ cd D

trgl1@PUNHDCDT054 /c/test/D (master)
$ touch MyJUnitTestCase.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ echo "f MyJUnitTestCase">MyJUnitTestCase.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ git add MyJUnitTestCase.txt
warning: LF will be replaced by CRLF in MyJUnitTestCase.txt.
The file will have its original line endings in your working direct

trgl1@PUNHDCDT054 /c/test/D (master)
$ git commit MyJUnitTestCase.txt -m 'ff MyJTestUnitCase'
warning: LF will be replaced by CRLF in MyJUnitTestCase.txt.
The file will have its original line endings in your working direct
[master 2a93f69] ff MyJTestUnitCase
warning: LF will be replaced by CRLF in MyJUnitTestCase.txt.
The file will have its original line endings in your working direct
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 MyJUnitTestCase.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ ls
MyJUnitTestCase.txt  info.txt  tes1.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
```

```
$ cd C

trgl1@PUNHDCDT054 /c/test/C (master)
$ ls
MyJavaCode.txt  dev1.txt  info.txt

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd C

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd A

trgl1@PUNHDCDT054 /c/test/A (BARE:master)
$ git checkout Development
fatal: This operation must be run in a work tree

trgl1@PUNHDCDT054 /c/test/A (BARE:master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd C

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..
```

```
trgl1@PUNHDCDT054 /c/test

$  cd D


trgl1@PUNHDCDT054 /c/test/D (master)

$ git pull /c/test/C master

remote: Counting objects: 7, done.

remote: Compressing objects: 100% (4/4), done.

remote: Total 6 (delta 0), reused 0 (delta 0)

Unpacking objects: 100% (6/6), done.

From c:/test/C

 * branch            master     -> FETCH_HEAD

Merge made by the 'recursive' strategy.

 MyJavaCode.txt |    1 +

 dev1.txt       |    1 +

 2 files changed, 2 insertions(+), 0 deletions(-)

 create mode 100644 MyJavaCode.txt

 create mode 100644 dev1.txt


trgl1@PUNHDCDT054 /c/test/D (master)

$ ls

MyJUnitTestCase.txt  MyJavaCode.txt  dev1.txt  info.txt  tes1.txt
```

Q5.

```
trgl1@PUNHDCDT054 /c/test/D (master)

$ git status

# On branch master

# Changes not staged for commit:

#   (use "git add <file>..." to update what will be committed)

#   (use "git checkout -- <file>..." to discard changes in working directory)

#
```

```
#       modified:   MyJavaCode.txt
#
no changes added to commit (use "git add" and/or "git commit -a")

trgl1@PUNHDCDT054 /c/test/D (master)
$ git add MyJ
MyJUnitTestCase.txt  MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ git add MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ git commit  MyJavaCode.txt -m 'modified accidently'
[master f2f460c] modified accidently
 1 files changed, 2 insertions(+), 0 deletions(-)

trgl1@PUNHDCDT054 /c/test/D (master)
$ git push /c/test/A master
Counting objects: 13, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 1.04 KiB, done.
Total 11 (delta 3), reused 0 (delta 0)
Unpacking objects: 100% (11/11), done.
To c:/test/A
   1f2b026..f2f460c  master -> master

trgl1@PUNHDCDT054 /c/test/D (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
```

```
$ cd C

trgl1@PUNHDCDT054 /c/test/C (master)
$ git branch
* master

trgl1@PUNHDCDT054 /c/test/C (master)
$ git pull /c/test/A master
remote: Counting objects: 14, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 11 (delta 2), reused 0 (delta 0)
Unpacking objects: 100% (11/11), done.
From c:/test/A
 * branch            master     -> FETCH_HEAD
Updating 43fab61..f2f460c
Fast-forward
 MyJUnitTestCase.txt |    1 +
 MyJavaCode.txt      |    2 ++
 tes1.txt            |    1 +
 3 files changed, 4 insertions(+), 0 deletions(-)
 create mode 100644 MyJUnitTestCase.txt
 create mode 100644 tes1.txt

trgl1@PUNHDCDT054 /c/test/C (master)
$ git commit
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   MyJavaCode.txt
```

```
#
no changes added to commit (use "git add" and/or "git commit -a")

trgl1@PUNHDCDT054 /c/test/C (master)
$ git add -a
error: unknown switch `a'
usage: git add [options] [--] <filepattern>...

    -n, --dry-run         dry run
    -v, --verbose         be verbose

    -i, --interactive     interactive picking
    -p, --patch           select hunks interactively
    -e, --edit            edit current diff and apply
    -f, --force           allow adding otherwise ignored files
    -u, --update          update tracked files
    -N, --intent-to-add   record only the fact that the path will be added la

    -A, --all             add changes from all tracked and untracked files
    --refresh             don't add, only refresh the index
    --ignore-errors       just skip files which cannot be added because of er
s
    --ignore-missing      check if - even missing - files are ignored in dry



trgl1@PUNHDCDT054 /c/test/C (master)
$ git add MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/C (master)
$ git commit MyJavaCode.txt -m"removed change"
```

```
[master c020ca0] removed change
 1 files changed, 1 insertions(+), 3 deletions(-)

trgl1@PUNHDCDT054 /c/test/C (master)
$ git fetch /c/test/A master
From c:/test/A
 * branch            master     -> FETCH_HEAD

trgl1@PUNHDCDT054 /c/test/C (master)
$ mergetool
bash: mergetool: command not found

trgl1@PUNHDCDT054 /c/test/C (master)
$ git mergetool
No files need merging

trgl1@PUNHDCDT054 /c/test/C (master)
$ git status
# On branch master
nothing to commit (working directory clean)

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd D

trgl1@PUNHDCDT054 /c/test/D (master)
$ git status
# On branch master
# Changes not staged for commit:
```

```
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   MyJavaCode.txt
#
no changes added to commit (use "git add" and/or "git commit -a")

trgl1@PUNHDCDT054 /c/test/D (master)
$ git add MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ git commit MyJavaCode.txt -m 'one more'
[master 735c4be] one more
 1 files changed, 3 insertions(+), 1 deletions(-)

trgl1@PUNHDCDT054 /c/test/D (master)
$ git push /c/test/A master
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 306 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
To c:/test/A
   f2f460c..735c4be  master -> master

trgl1@PUNHDCDT054 /c/test/D (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd C
```

```
trgl1@PUNHDCDT054 /c/test/C (master)
$ git fetch  /c/test/A master
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From c:/test/A
 * branch            master     -> FETCH_HEAD

trgl1@PUNHDCDT054 /c/test/C (master)
$ git mergetool
No files need merging

trgl1@PUNHDCDT054 /c/test/C (master)
$ git merge /c/test/A master
fatal: 'c:/test/A' does not point to a commit

trgl1@PUNHDCDT054 /c/test/C (master)
$ git status
# On branch master
nothing to commit (working directory clean)

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd A

trgl1@PUNHDCDT054 /c/test/A (BARE:master)
$ ls
```

```
HEAD  branches/  config  description  hooks/  info/  objects/  refs/

trgl1@PUNHDCDT054 /c/test/A (BARE:master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd D

trgl1@PUNHDCDT054 /c/test/D (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   MyJavaCode.txt
#
no changes added to commit (use "git add" and/or "git commit -a")

trgl1@PUNHDCDT054 /c/test/D (master)
$ git add MyJavaCode.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ git commit MyJavaCode.txt -m 'second'
[master ff445d3] second
 1 files changed, 3 insertions(+), 1 deletions(-)

trgl1@PUNHDCDT054 /c/test/D (master)
$ git push  /c/test/A master
Counting objects: 5, done.
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
To c:/test/A
   735c4be..ff445d3  master -> master

trgl1@PUNHDCDT054 /c/test/D (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd C

trgl1@PUNHDCDT054 /c/test/C (master)
$ git pull  /c/test/A master
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 3), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From c:/test/A
 * branch            master     -> FETCH_HEAD
Auto-merging MyJavaCode.txt
CONFLICT (content): Merge conflict in MyJavaCode.txt
Automatic merge failed; fix conflicts and then commit the result.

trgl1@PUNHDCDT054 /c/test/C (master|MERGING)
$ git mergetool
Merging:
MyJavaCode.txt

Normal merge conflict for 'MyJavaCode.txt':
  {local}: modified file

  {remote}: modified file

Hit return to start merge resolution tool (kdiff3):

$ git commit -a

[master 946e053] Merge branch 'master' of c:/test/A
```

Q6.

```
trgl1@PUNHDCDT054 /c/test/C (master)

$ git pull  /c/test/A master

From c:/test/A

 * branch            master     -> FETCH_HEAD

Already up-to-date.


trgl1@PUNHDCDT054 /c/test/C (master)

$ git push  /c/test/E master

Counting objects: 30, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (22/22), done.

Writing objects: 100% (29/29), 2.38 KiB, done.

Total 29 (delta 10), reused 0 (delta 0)

Unpacking objects: 100% (29/29), done.

To c:/test/E

   1f2b026..946e053  master -> master


trgl1@PUNHDCDT054 /c/test/C (master)

$ touch Souce2.java
```

```
trgl1@PUNHDCDT054 /c/test/C (master)
$ echo "source" >Souce2.java


trgl1@PUNHDCDT054 /c/test/C (master)
$ add Source2.java
bash: add: command not found


trgl1@PUNHDCDT054 /c/test/C (master)
$ git add Source2.java
fatal: pathspec 'Source2.java' did not match any files


trgl1@PUNHDCDT054 /c/test/C (master)
$ git add Souce2.java
warning: LF will be replaced by CRLF in Souce2.java.
The file will have its original line endings in your working directory.


trgl1@PUNHDCDT054 /c/test/C (master)
$ git commit Souce2.java -m 'source'
warning: LF will be replaced by CRLF in Souce2.java.
The file will have its original line endings in your working directory.
[master e12fdcc] source
warning: LF will be replaced by CRLF in Souce2.java.
The file will have its original line endings in your working directory.
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 Souce2.java


trgl1@PUNHDCDT054 /c/test/C (master)
$ git push  /c/test/E master
Counting objects: 4, done.
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 273 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
To c:/test/E
   946e053..e12fdcc  master -> master

trgl1@PUNHDCDT054 /c/test/C (master)
$ cd ..

trgl1@PUNHDCDT054 /c/test
$ cd D

trgl1@PUNHDCDT054 /c/test/D (master)
$ touch HttpdTest.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ echo "source" >HttpdTest.txt

trgl1@PUNHDCDT054 /c/test/D (master)
$ git add HttpdTest.txt
warning: LF will be replaced by CRLF in HttpdTest.txt.
The file will have its original line endings in your working directory.

trgl1@PUNHDCDT054 /c/test/D (master)
$ git commit HttpdTest.txt
warning: LF will be replaced by CRLF in HttpdTest.txt.
The file will have its original line endings in your working directory.
[master 84362cd] asas
warning: LF will be replaced by CRLF in HttpdTest.txt.
The file will have its original line endings in your working directory.
```

```
 1 files changed, 1 insertions(+), 0 deletions(-)

 create mode 100644 HttpdTest.txt


trgl1@PUNHDCDT054 /c/test/D (master)

$ git push  /c/test/E master

To c:/test/E

 ! [rejected]        master -> master (non-fast-forward)

error: failed to push some refs to 'c:/test/E'

To prevent you from losing history, non-fast-forward updates were rejected

Merge the remote changes (e.g. 'git pull') before pushing again.  See the

'Note about fast-forwards' section of 'git push --help' for details.


trgl1@PUNHDCDT054 /c/test/D (master)

$ git pull  /c/test/E master

remote: Counting objects: 13, done.

remote: Compressing objects: 100% (6/6), done.

remote: Total 9 (delta 3), reused 0 (delta 0)

Unpacking objects: 100% (9/9), done.

From c:/test/E

 * branch            master     -> FETCH_HEAD

Merge made by the 'recursive' strategy.

 MyJavaCode.txt |    2 --

 Souce2.java    |    1 +

 2 files changed, 1 insertions(+), 2 deletions(-)

 create mode 100644 Souce2.java


trgl1@PUNHDCDT054 /c/test/D (master)

$ git push  /c/test/E master

Counting objects: 6, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (4/4), done.
```

```
Writing objects: 100% (4/4), 491 bytes, done.
Total 4 (delta 2), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
To c:/test/E
   e12fdcc..6d14c69  master -> master

trgl1@PUNHDCDT054 /c/test/D (master)
$ git pull  /c/test/A master
From c:/test/A
 * branch            master     -> FETCH_HEAD
Already up-to-date.

trgl1@PUNHDCDT054 /c/test/D (master)
$ git push  /c/test/A master
Counting objects: 17, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (13/13), 1.26 KiB, done.
Total 13 (delta 5), reused 0 (delta 0)
Unpacking objects: 100% (13/13), done.
To c:/test/A
   ff445d3..6d14c69  master -> master
```

Q7.

```
trgl1@PUNHDCDT054 /c/test/B (master)
$ touch b1.txt b2.txt b3.txt b4.txt

trgl1@PUNHDCDT054 /c/test/B (master)
```

```
$ git add -A

trgl1@PUNHDCDT054 /c/test/B (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   b1.txt
#       new file:   b2.txt
#       new file:   b3.txt
#       new file:   b4.txt
#

trgl1@PUNHDCDT054 /c/test/B (master)
$ git commit b1.txt -m'b1'
[master ca6946c] b1
 0 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b1.txt

trgl1@PUNHDCDT054 /c/test/B (master)
$ git commit b2.txt -m'b2'
[master dfcc03c] b2
 0 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b2.txt

trgl1@PUNHDCDT054 /c/test/B (master)
$ git commit b3.txt -m'b3'
[master d615cc2] b3
 0 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b3.txt
```

```
trgl1@PUNHDCDT054 /c/test/B (master)
$ git commit b4.txt -m'b4'
[master b4f0049] b4
 0 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b4.txt

trgl1@PUNHDCDT054 /c/test/B (master)
$ git stutus
git: 'stutus' is not a git command. See 'git --help'.

Did you mean this?
        status

trgl1@PUNHDCDT054 /c/test/B (master)
$ git status
# On branch master
nothing to commit (working directory clean)

trgl1@PUNHDCDT054 /c/test/B (master)
$ git log --pretty=format:"%h %s" HEAD~3..
b4f0049 b4
d615cc2 b3
dfcc03c b2

trgl1@PUNHDCDT054 /c/test/B (master)
$ git rebase -i HEAD~3
Successfully rebased and updated refs/heads/master.

trgl1@PUNHDCDT054 /c/test/B (master)
$ git log --pretty=format:"%h %s" HEAD~3..
```

```
e7b4071 b2

2af7904 b4

be51188 b3
```

change file in txt

____

```
pick dfcc03c b2

pick d615cc2 b3

pick b4f0049 b4


# Rebase ca6946c..b4f0049 onto ca6946c

#

# Commands:

#  p, pick = use commit

#  r, reword = use commit, but edit the commit message

#  e, edit = use commit, but stop for amending

#  s, squash = use commit, but meld into previous commit

#  f, fixup = like "squash", but discard this commit's log message

#  x, exec = run command (the rest of the line) using shell

#

# If you remove a line here THAT COMMIT WILL BE LOST.

# However, if you remove everything, the rebase will be aborted.
```

```
pick d615cc2 b3

pick b4f0049 b4

pick dfcc03c b2

# Rebase ca6946c..b4f0049 onto ca6946c

#

# Commands:
```

```
#  p, pick = use commit
#  r, reword = use commit, but edit the commit message
#  e, edit = use commit, but stop for amending
#  s, squash = use commit, but meld into previous commit
#  f, fixup = like "squash", but discard this commit's log message
#  x, exec = run command (the rest of the line) using shell
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
```