PART 1
Created repo, cloned on local: **git clone https://github.com/Antines/ITMO_ScientificPython_2024**

PART 2
1) Created HW1 branch with HW1 dic, created 3 files:
git checkout -b HW1 -> mkdir HW1 | cd HW1 -> touch hw1.txt test_revert.txt test_revert_merge.txt

2) Added, commited, pushed:
git add . | git commit -m "created 3 files" | git push origin HW1

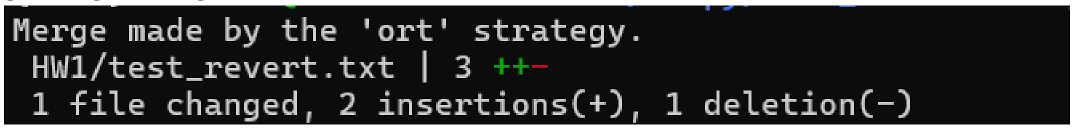
3) Created testing branch, changed “hw1.txt” on HW1, added committed pushed

git branch testing
git add .
git commit -m "Change hw1.txt"
git push -u origin HW1

4) Checkout testing, changed “test_revert.txt”, added committed pushed:

git checkout testing
git add .
git commit -m "Change test_revert.txt"
git push -u origin testing

5) Merged testing into HW1, added committed pushed
git checkout HW1
git merge testing



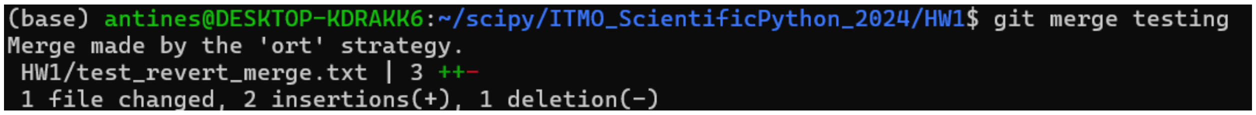
```
Merge made by the 'ort' strategy.
 HW1/test_revert.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

Now, HW1 bracnh contains first 2 updated files (1st from direct change, 2nd from merge testing, while testing contains only 2nd one changed
git add .
git commit -m "merged testing into HW1"
git push -u origin HW1

6) Reweted 1 step back state, changed 3rd file in testing, added committed pushed

git revert -m 1 HEAD
git checkout testing
git add .
git commit -m "changed 3rd file"
git push origin testing

7) Merged once again:



```
(base) antines@DESKTOP-KDRAKK6:~/scipy/ITMO_ScientificPython_2024/HW1$ git merge testing
Merge made by the 'ort' strategy.
 HW1/test_revert_merge.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

2nd merge led to a situation, where the 2nd file "test_revert.txt" didn't changed after merging with testing, as it should have been, since it was committed and pushed in the testing branch before. It might have been due to git considered the changes from the testing branch and attempted to apply them to HW1. However, since the merge commit was reverted in step back, the changes from the testing branch were essentially re-applied to the state before the original merge.

To make it right, the branch was reseted (**git reset –hard:** discarded changes in both the working directory and staging area), ensuring that HW1 branch is reset to the state of the remote HW1

git reset --hard origin/HW1
git merge testing