lab0 实验文档

文档问题

两条命令的区别: git branch 只显示本地分支, 而git branch显示所有分支，包括本地分支和远程分支（remotes/...）

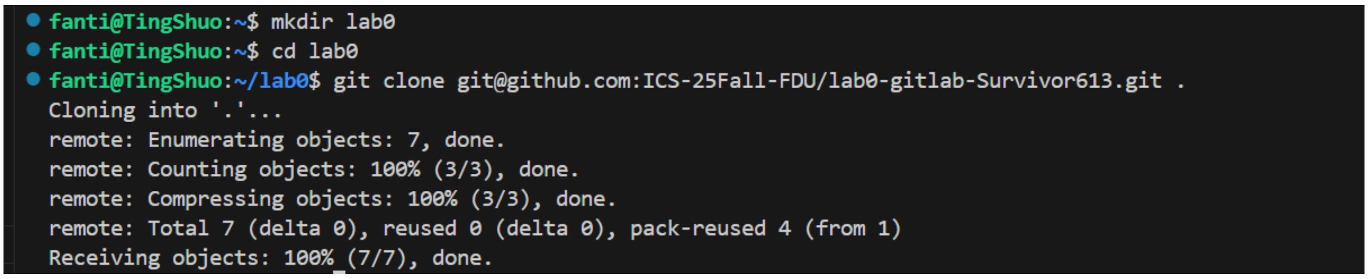
网页阅读问题

通过阅览`Commit Message 规范`, 我认识到：git提供了强大的版本控制，可以直接生成change log，同时commit能够让使用者清晰了解commit类型和具体改动内容，有助于快速筛选出自己需要的版本。

通过阅览语义化版本, 我认识到：语义化版本控制规范中，主版本号，次版本号，修订号各自的作用，在发布新版本时应该遵循的规范，以及作为发布者和使用者的一些良好实践。而Git版本控制则使我们在git commit的时候能清楚的标明我们所做的代码修改/补充是服务于哪一个大版本。

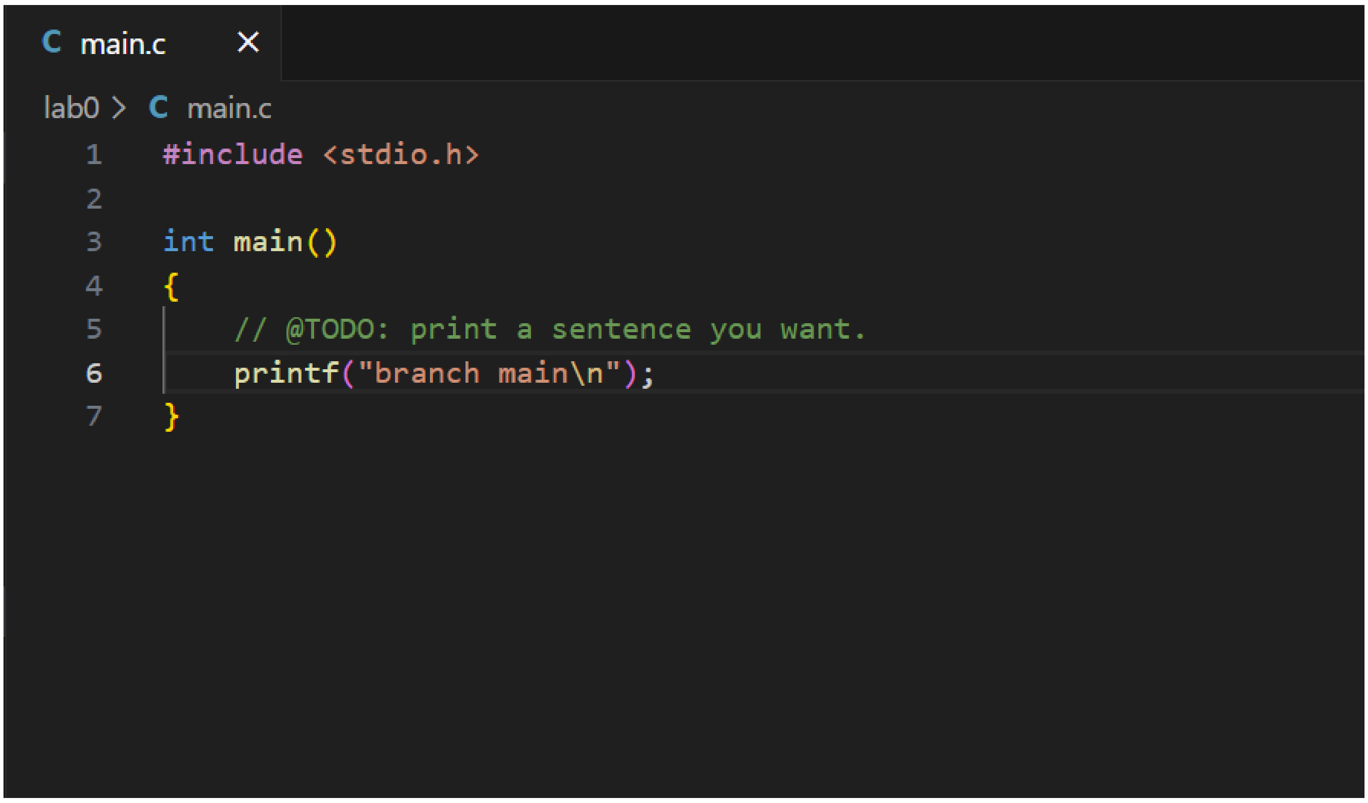
实验步骤

1. git clone

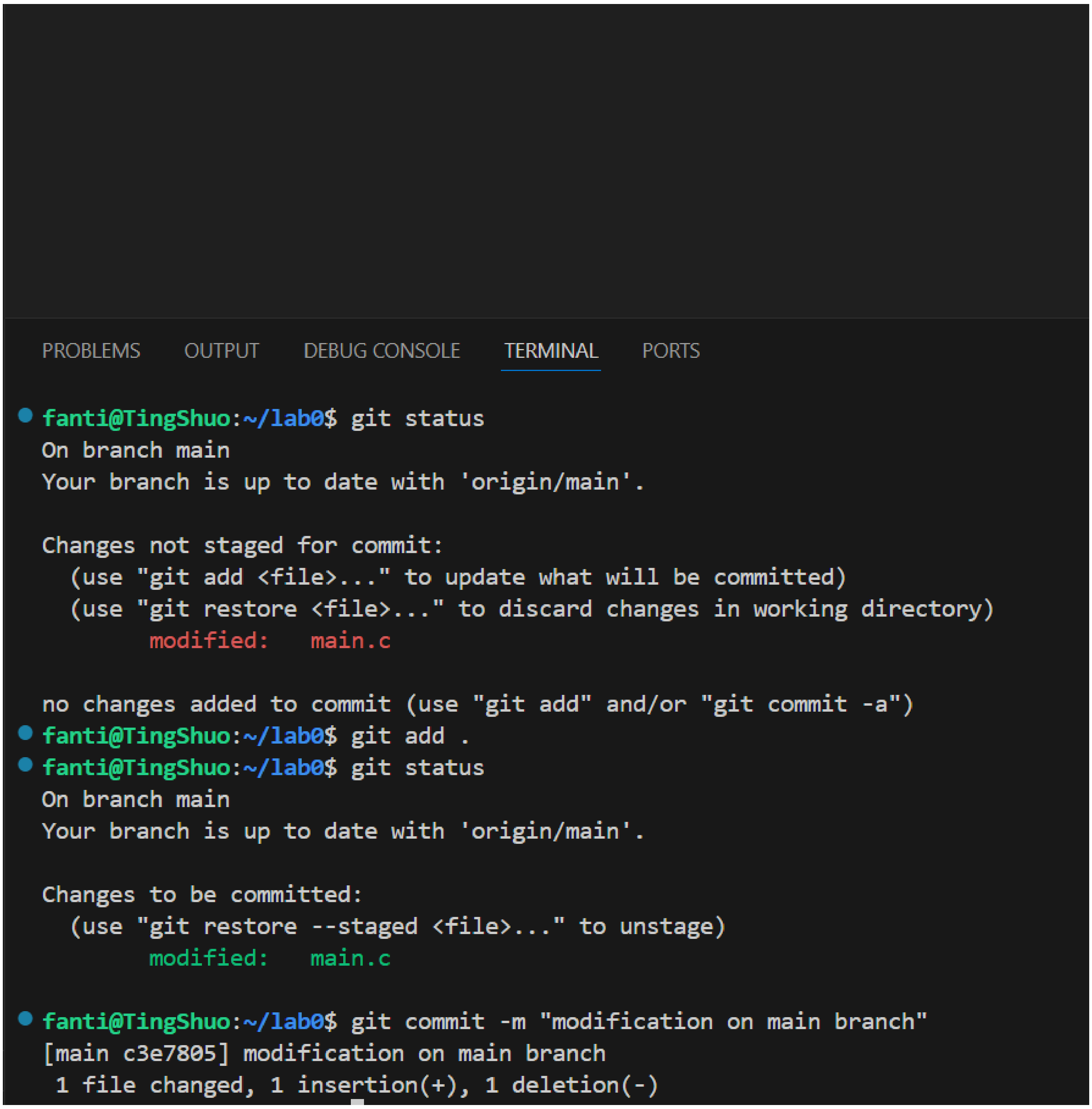


```
fanti@TingShuo:~$ mkdir lab0
fanti@TingShuo:~$ cd lab0
fanti@TingShuo:~/lab0$ git clone git@github.com:ICS-25Fall-FDU/lab0-gitlab-Survivor613.git .
Cloning into '.'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 4 (from 1)
Receiving objects: 100% (7/7), done.
```

2. 修改main分支并commit

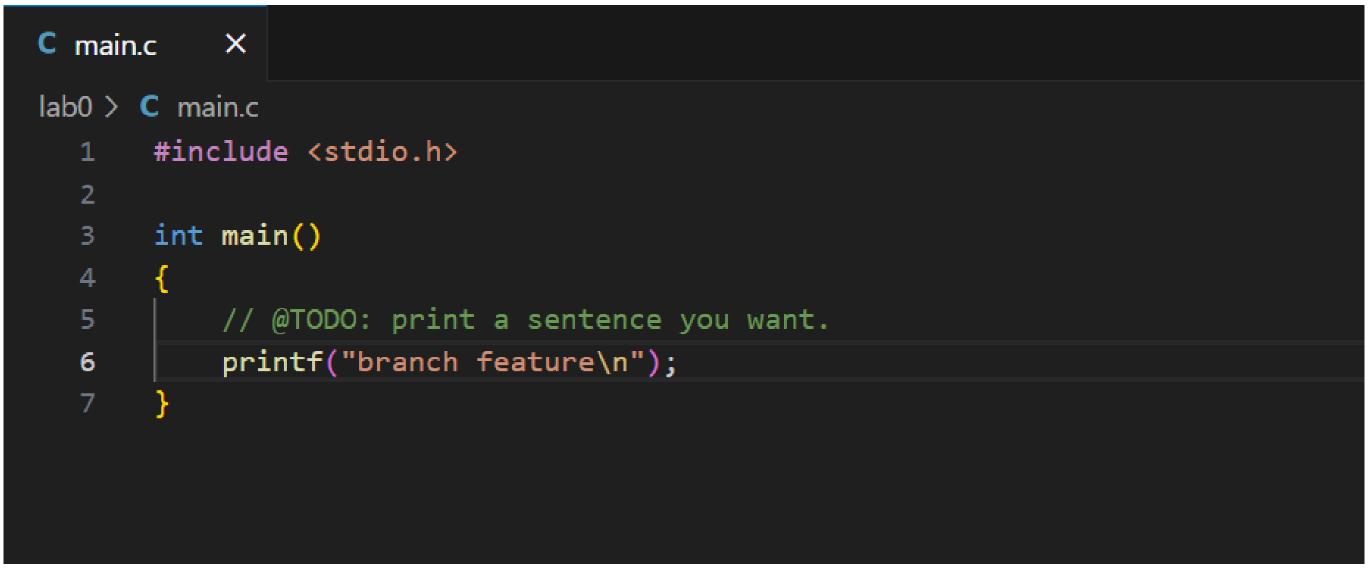


```c
#include <stdio.h>

int main()
{
    // @TODO: print a sentence you want.
    printf("branch main\n");
}
```

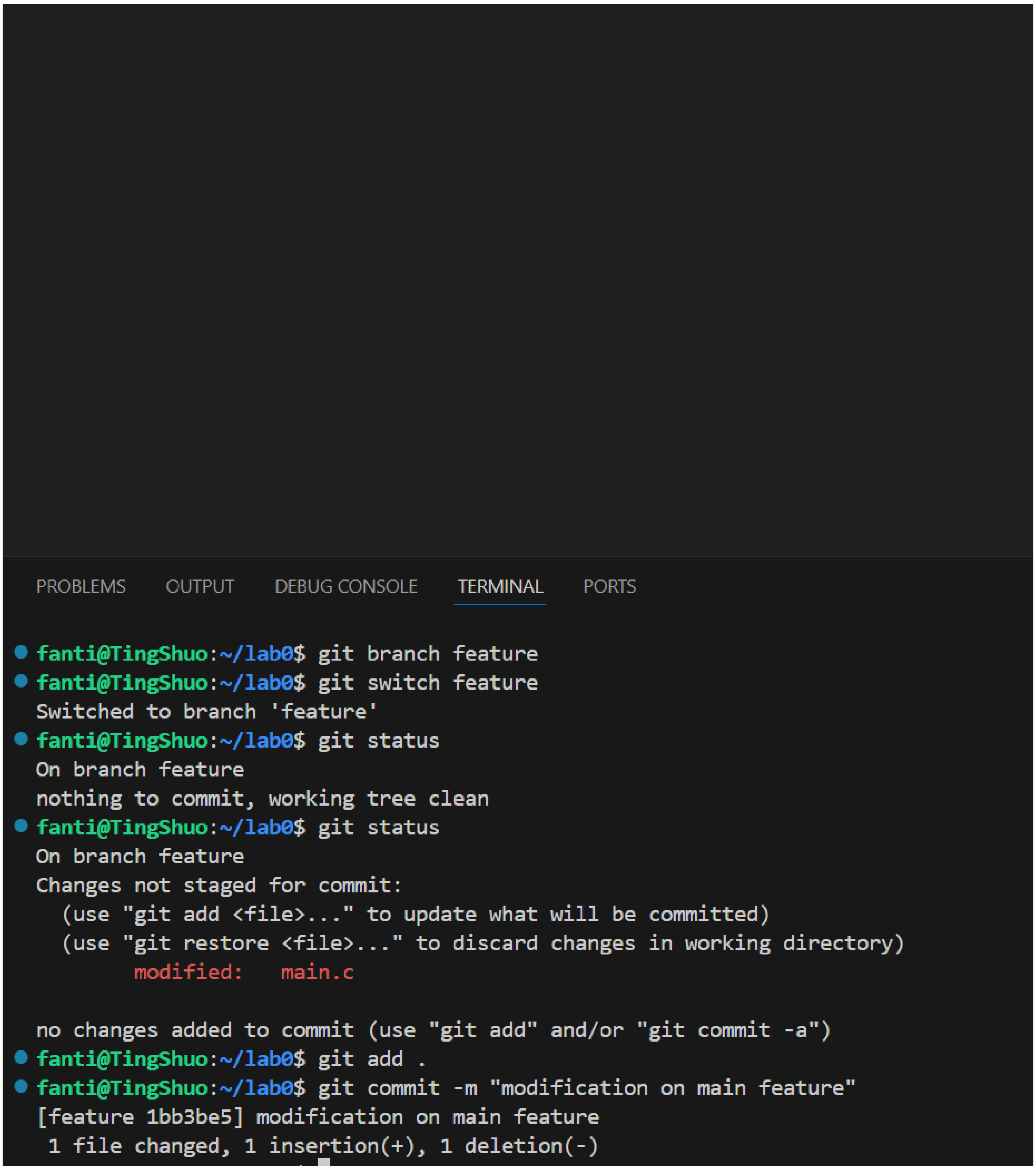


```
fanti@TingShuo:~/lab0$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.c

no changes added to commit (use "git add" and/or "git commit -a")
fanti@TingShuo:~/lab0$ git add .
fanti@TingShuo:~/lab0$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   main.c

fanti@TingShuo:~/lab0$ git commit -m "modification on main branch"
[main c3e7805] modification on main branch
 1 file changed, 1 insertion(+), 1 deletion(-)
```

3. 创建feature分支，修改并commit

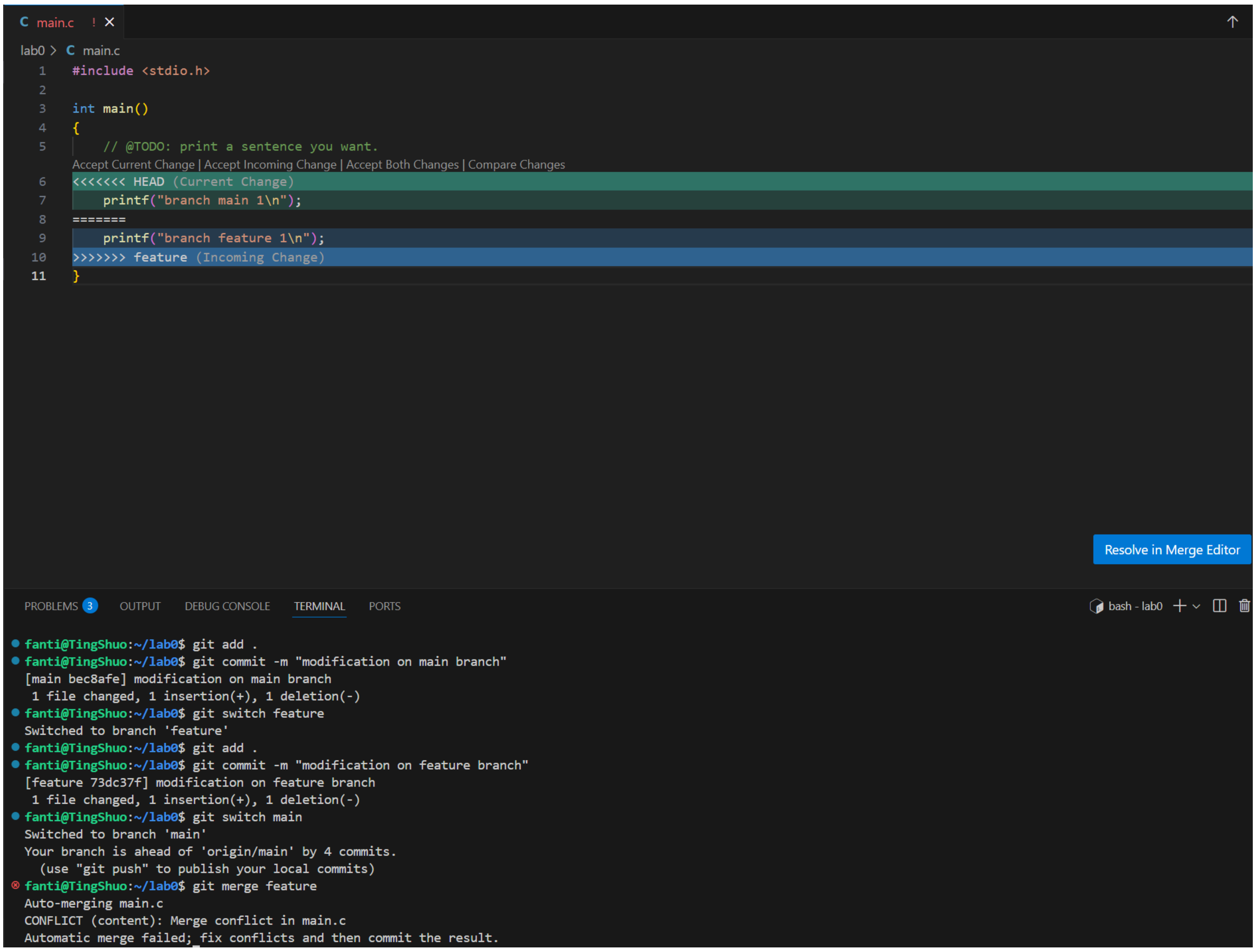


```c
#include <stdio.h>

int main()
{
    // @TODO: print a sentence you want.
    printf("branch feature\n");
}
```



```
● fanti@TingShuo:~/lab0$ git branch feature
● fanti@TingShuo:~/lab0$ git switch feature
  Switched to branch 'feature'
● fanti@TingShuo:~/lab0$ git status
  On branch feature
  nothing to commit, working tree clean
● fanti@TingShuo:~/lab0$ git status
  On branch feature
  Changes not staged for commit:
    (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
          modified:   main.c

  no changes added to commit (use "git add" and/or "git commit -a")
● fanti@TingShuo:~/lab0$ git add .
● fanti@TingShuo:~/lab0$ git commit -m "modification on main feature"
  [feature 1bb3be5] modification on main feature
   1 file changed, 1 insertion(+), 1 deletion(-)
```

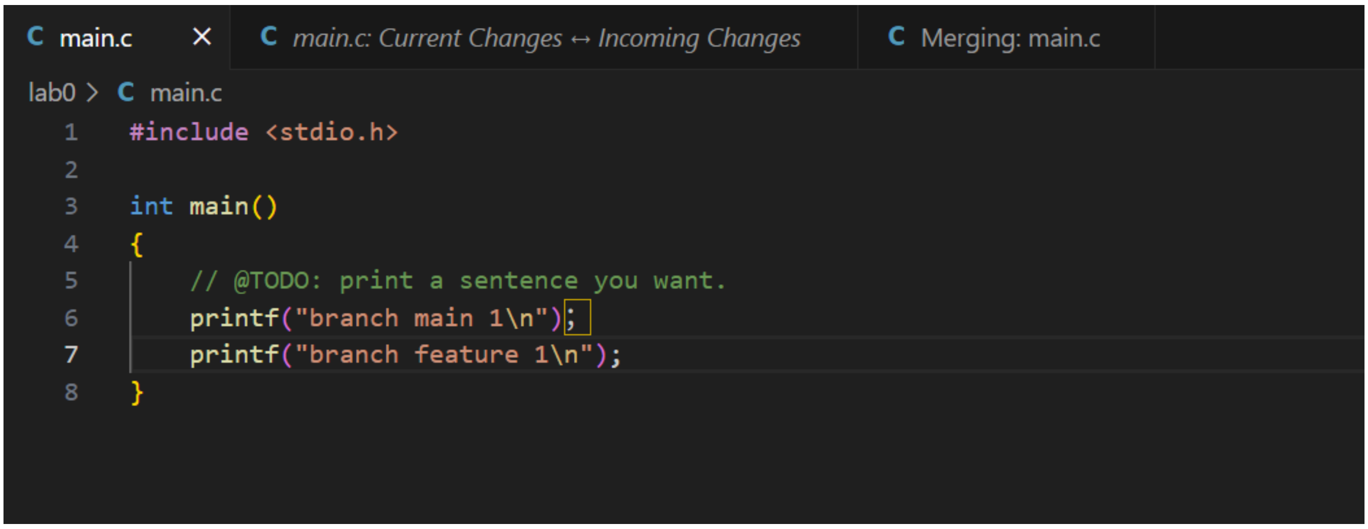
4.merge时出现冲突



```c
#include <stdio.h>

int main()
{
    // @TODO: print a sentence you want.
<<<<<<< HEAD (Current Change)
    printf("branch main 1\n");
=======
    printf("branch feature 1\n");
>>>>>>> feature (Incoming Change)
}
```

```
fanti@TingShuo:~/lab0$ git add .
fanti@TingShuo:~/lab0$ git commit -m "modification on main branch"
[main bec8afe] modification on main branch
 1 file changed, 1 insertion(+), 1 deletion(-)
fanti@TingShuo:~/lab0$ git switch feature
Switched to branch 'feature'
fanti@TingShuo:~/lab0$ git add .
fanti@TingShuo:~/lab0$ git commit -m "modification on feature branch"
[feature 73dc37f] modification on feature branch
 1 file changed, 1 insertion(+), 1 deletion(-)
fanti@TingShuo:~/lab0$ git switch main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)
fanti@TingShuo:~/lab0$ git merge feature
Auto-merging main.c
CONFLICT (content): Merge conflict in main.c
Automatic merge failed; fix conflicts and then commit the result.
```

5.解决冲突（Accept Both Changes）



```c
#include <stdio.h>

int main()
{
    // @TODO: print a sentence you want.
    printf("branch main 1\n");
    printf("branch feature 1\n");
}
```

补充说明，由于整个git版本控制中包含试错环节，如果助教需要查看本项目历史版本信息，请以最新提交的commit为标准