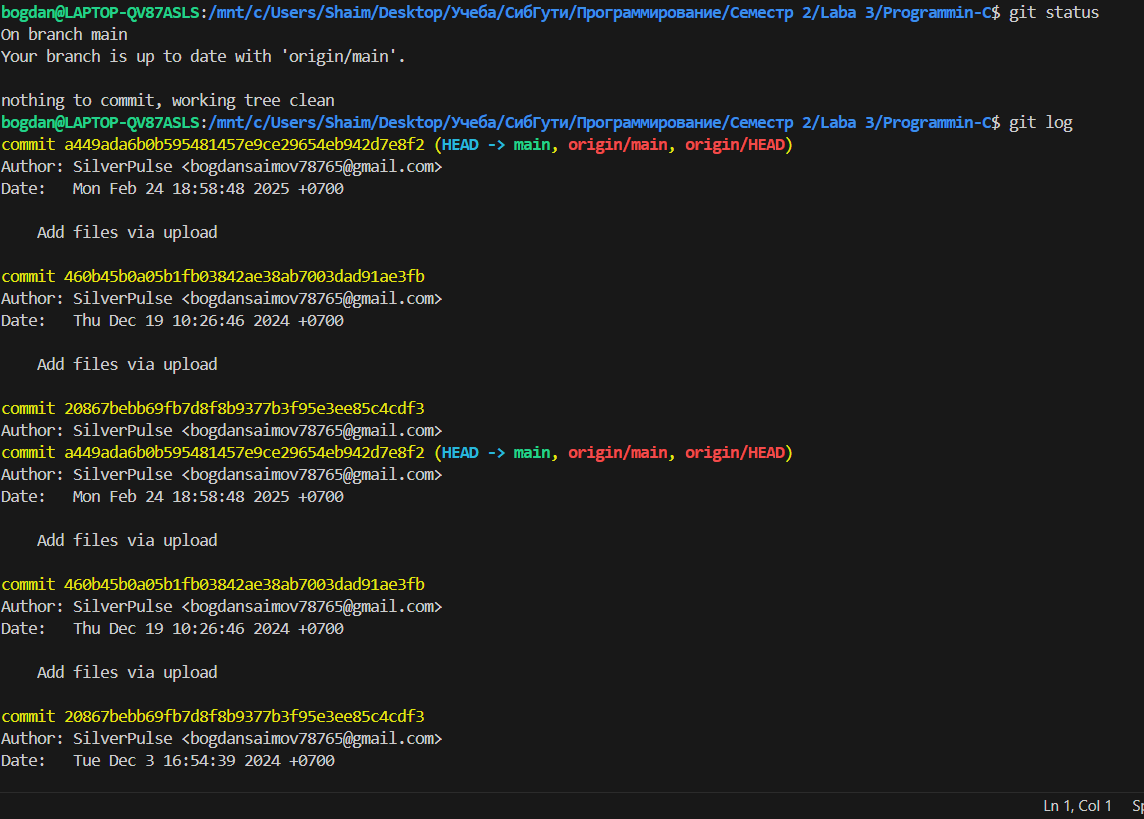
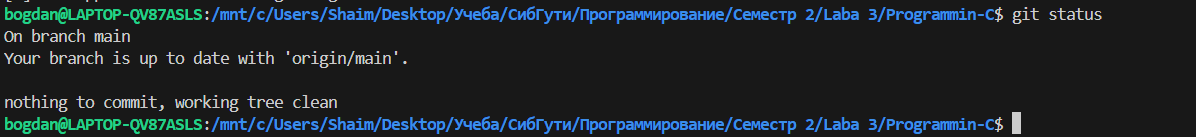
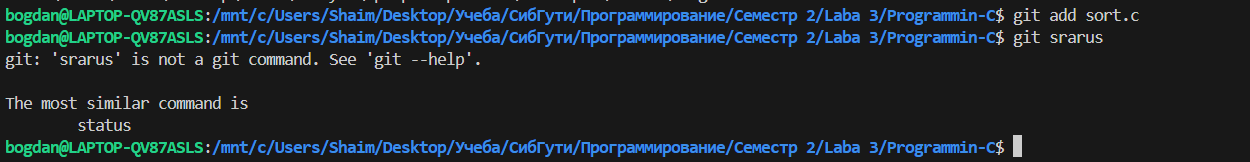
1,2

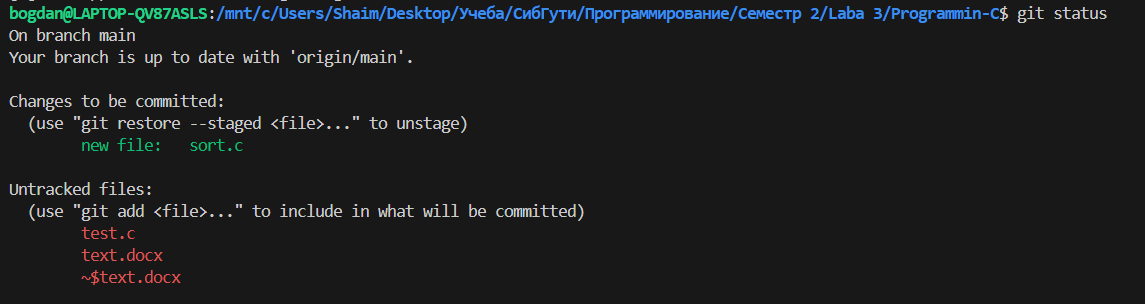


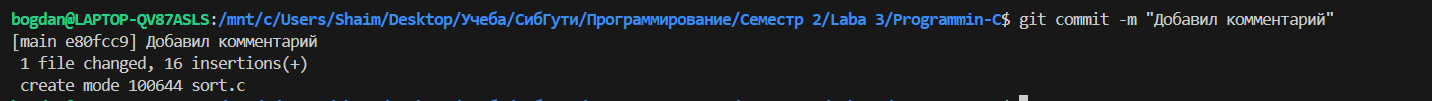
5



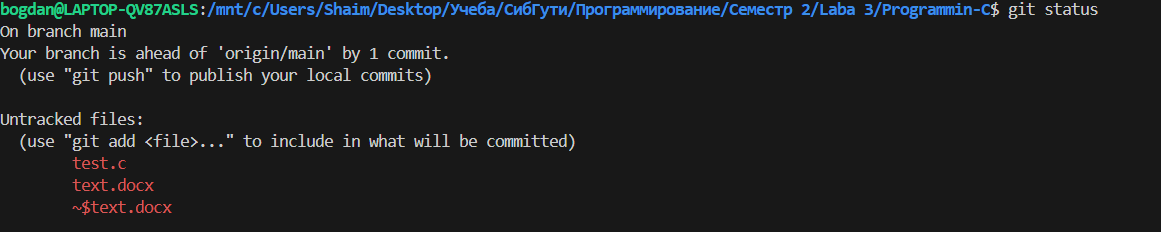
6



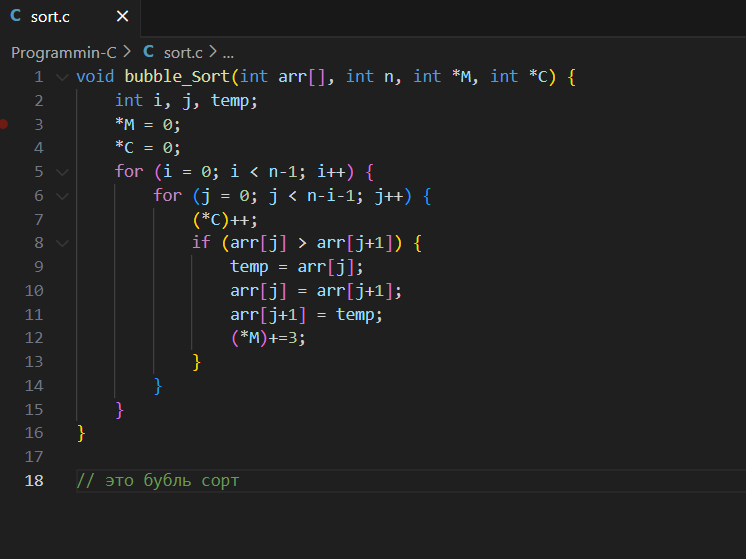


7

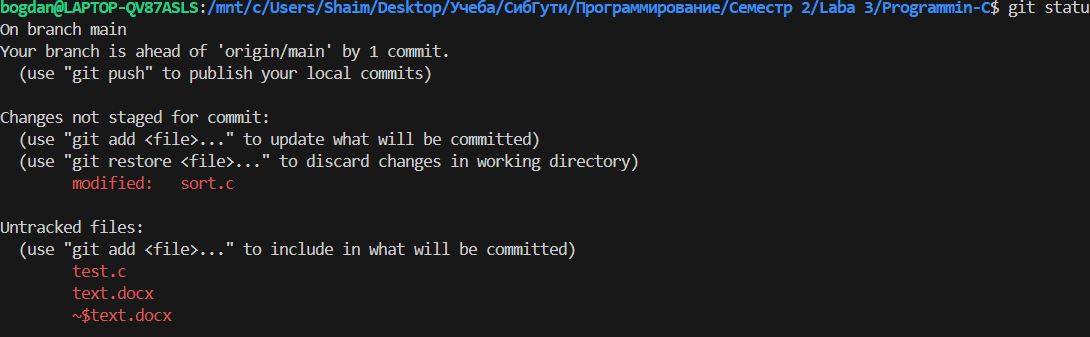
8



9 добавил комментарий по заданию.



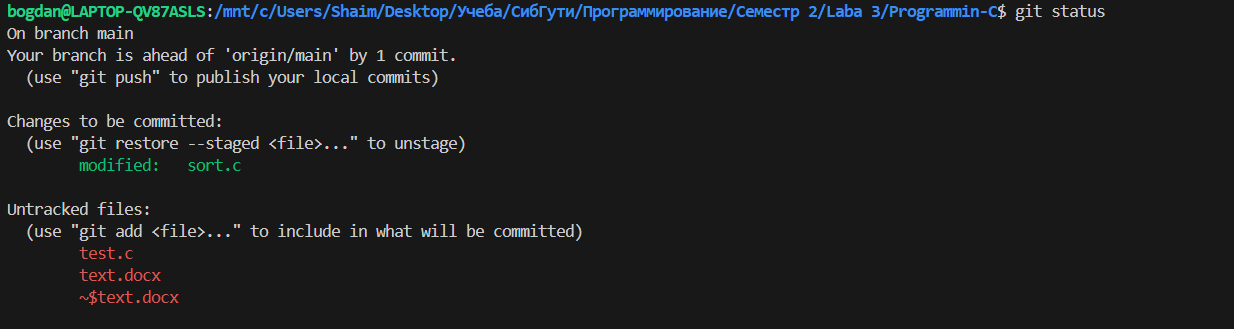
10

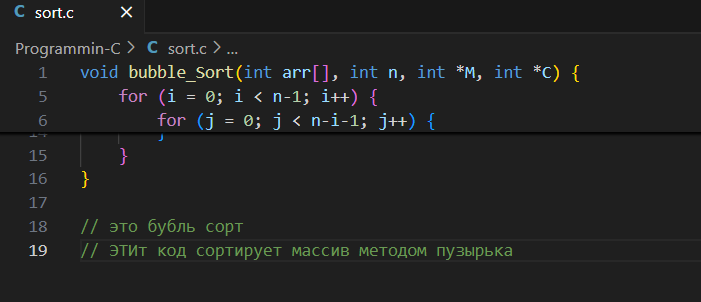


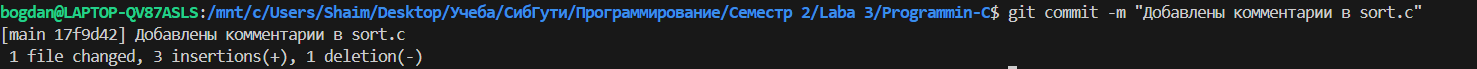
11



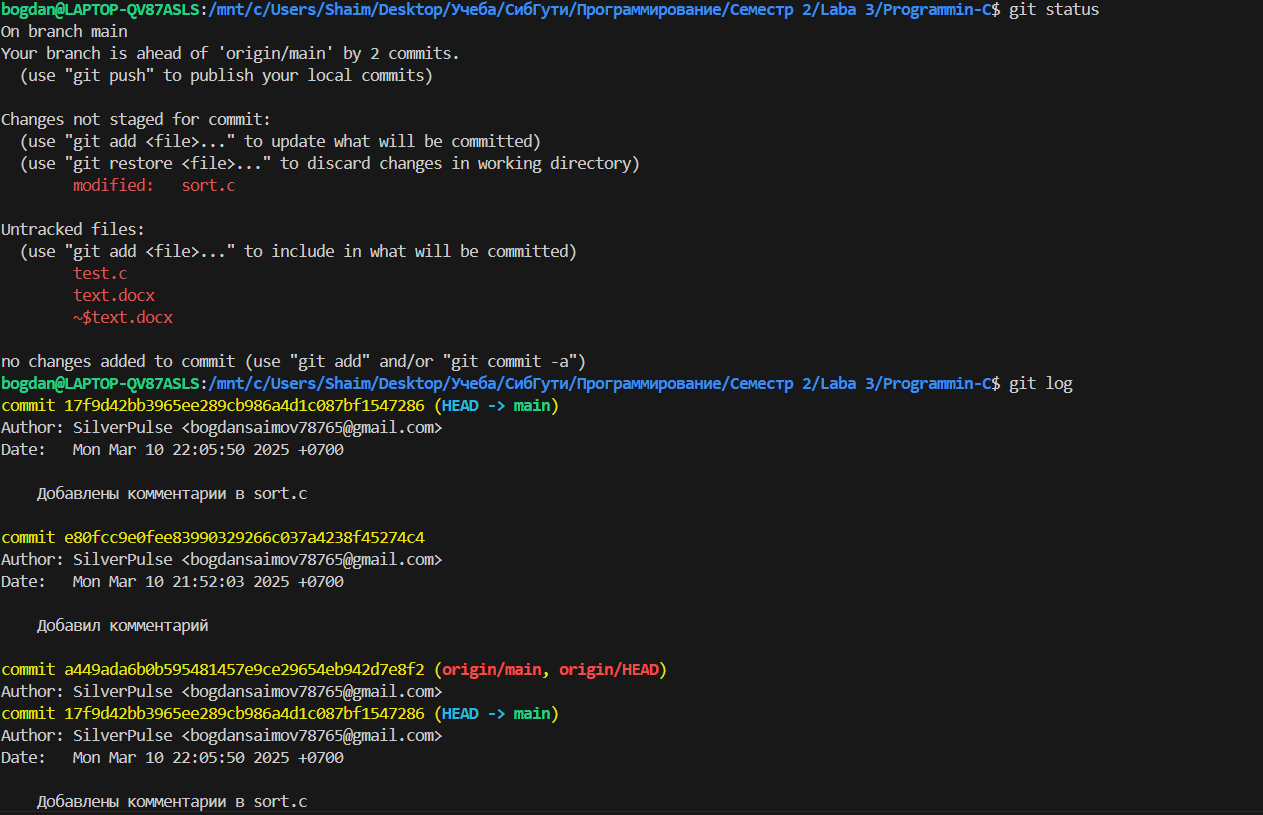
12



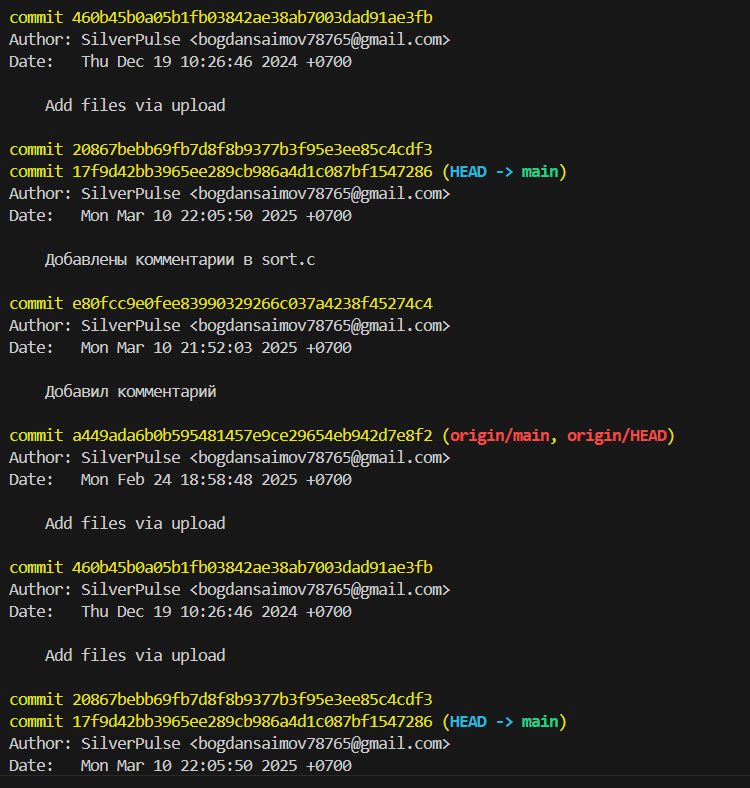
13 Ещё добавил комментарий  


14

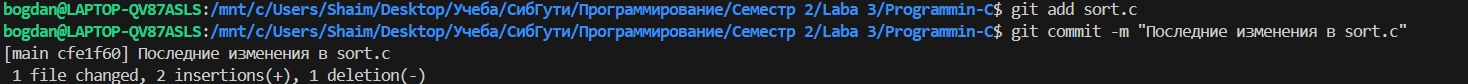
15





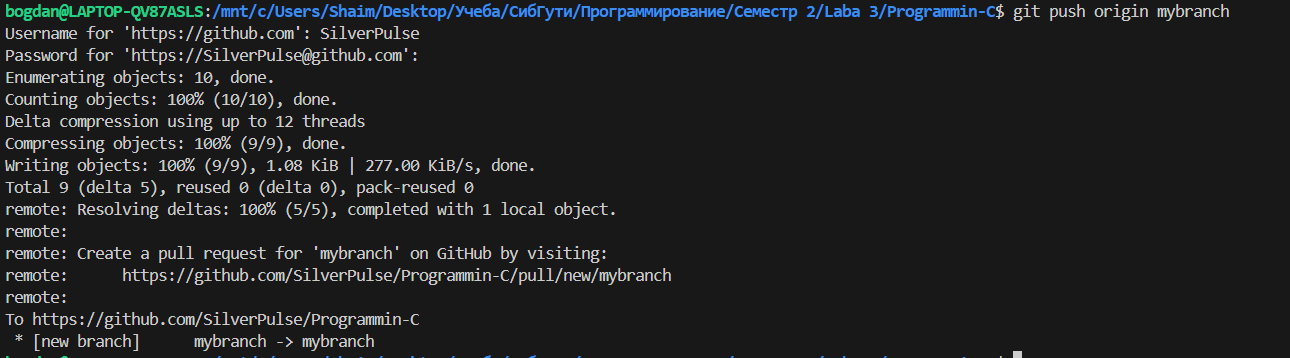


16



17 создадим ветку и запушим её c помощью временного токена вместо пароля



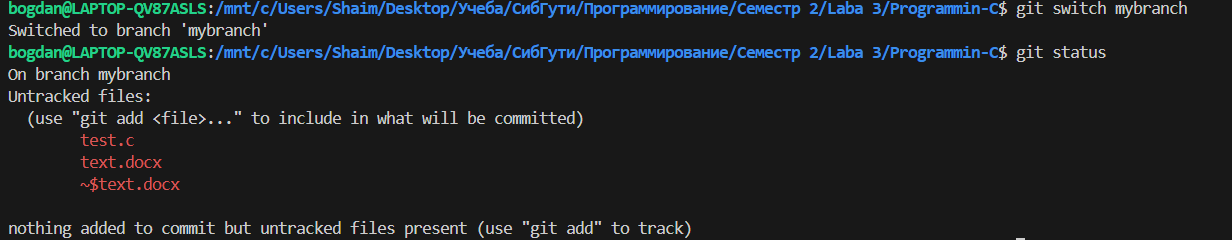


Теперь немного поиграем с ветками (это по заданию, честно)

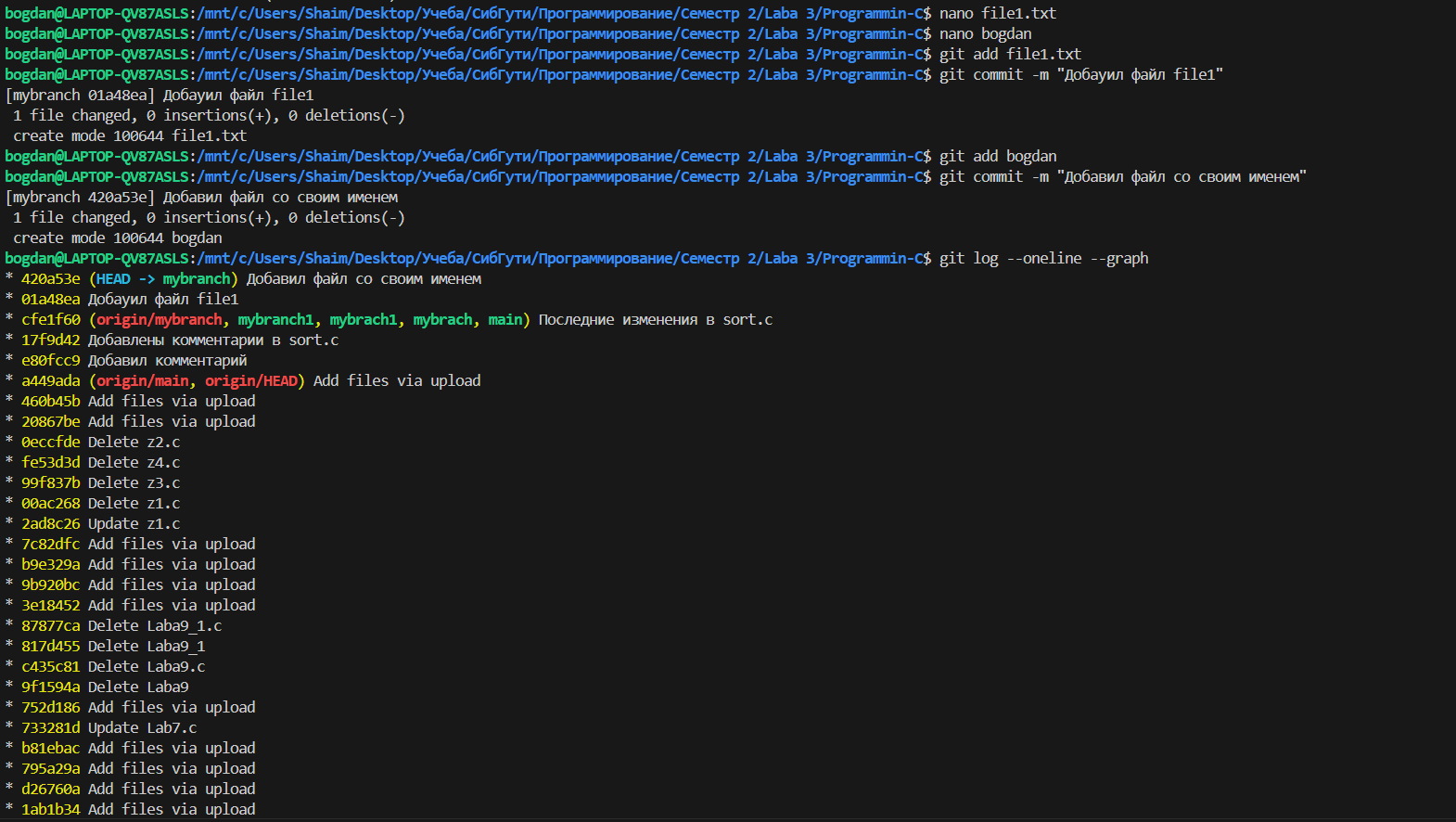
1,2



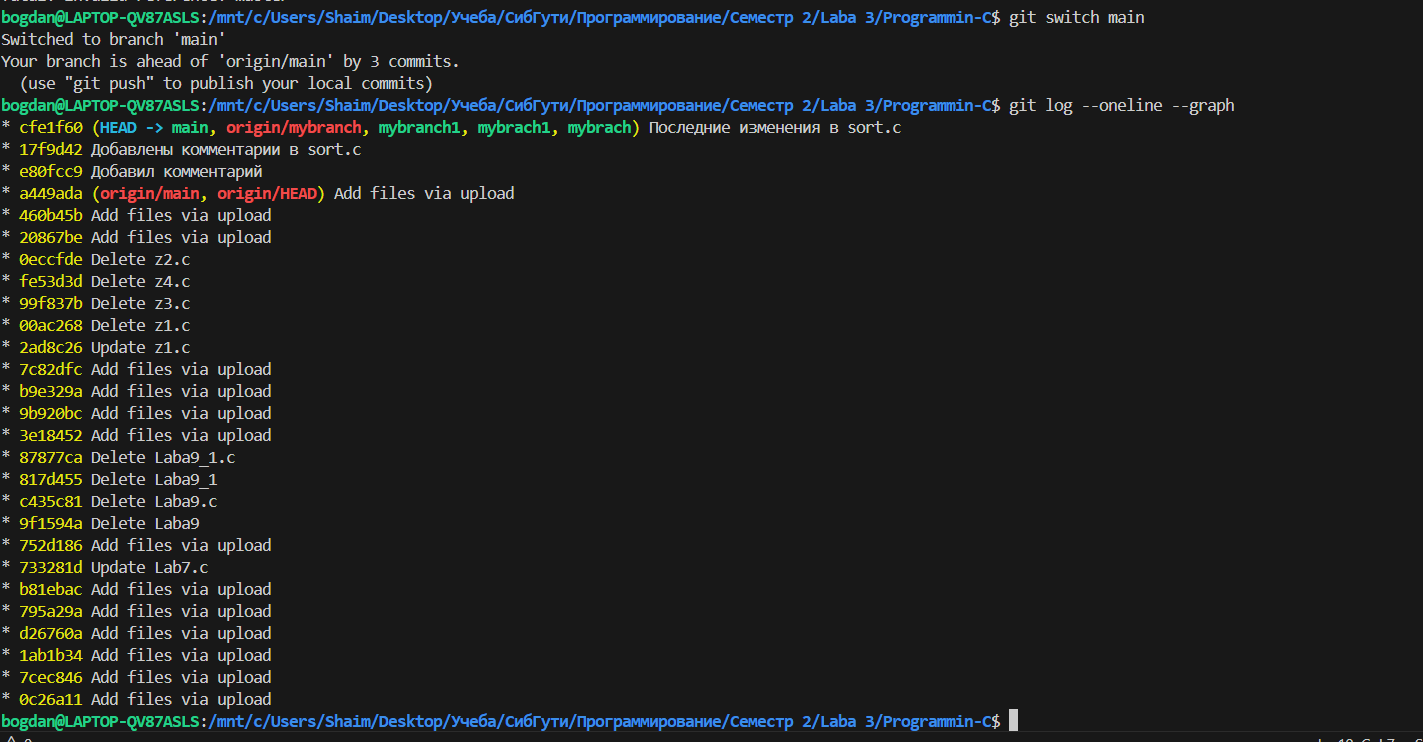
3,4,5 Переключаемся на ветку и проверяем результат



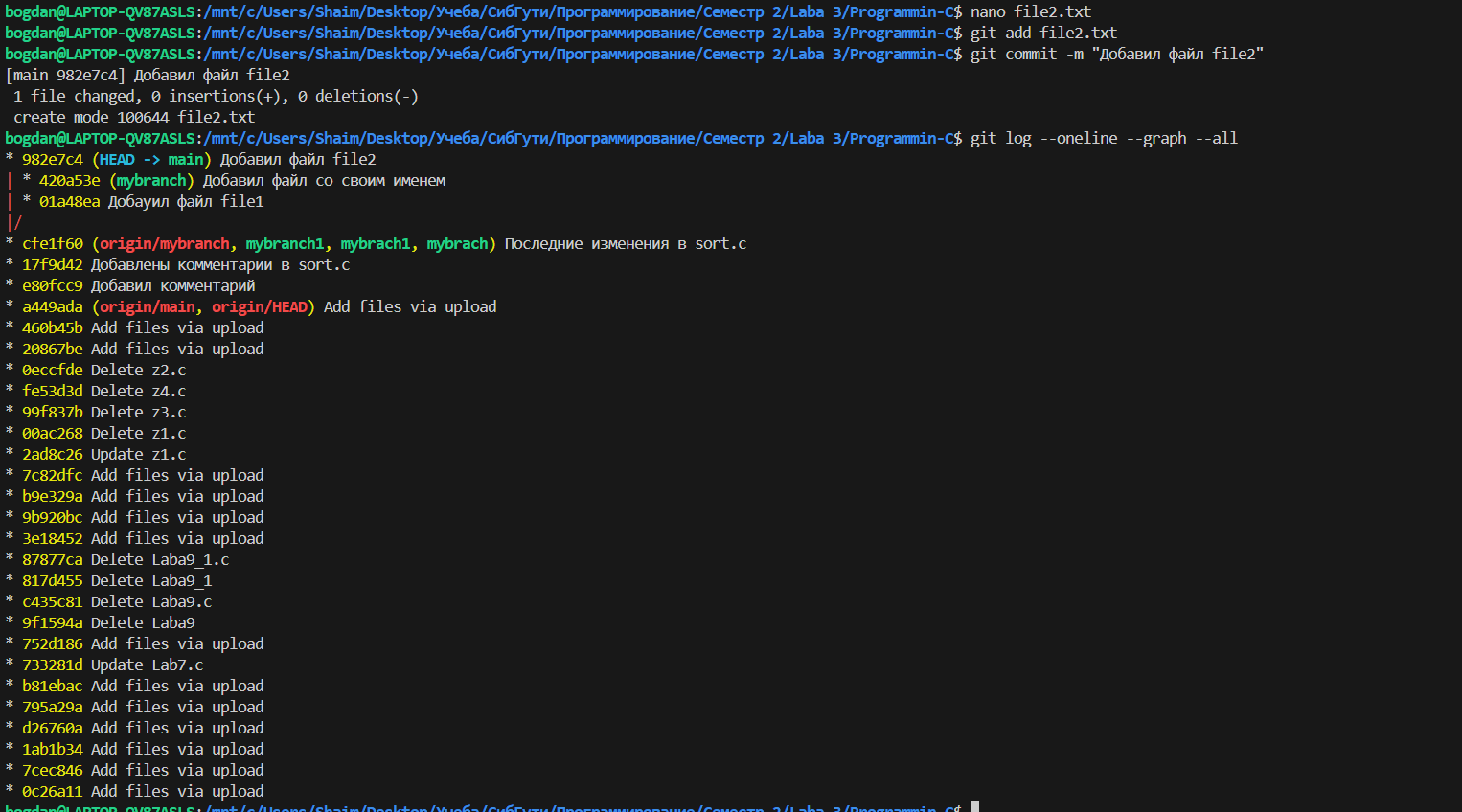
6,7,8 Создаем файл с именем file1.txt и своим именем. Добавляем файл, коммитим. Используя git log –oneline –graph,чтобы увидеть что наша ветка указывает на новый коммит.



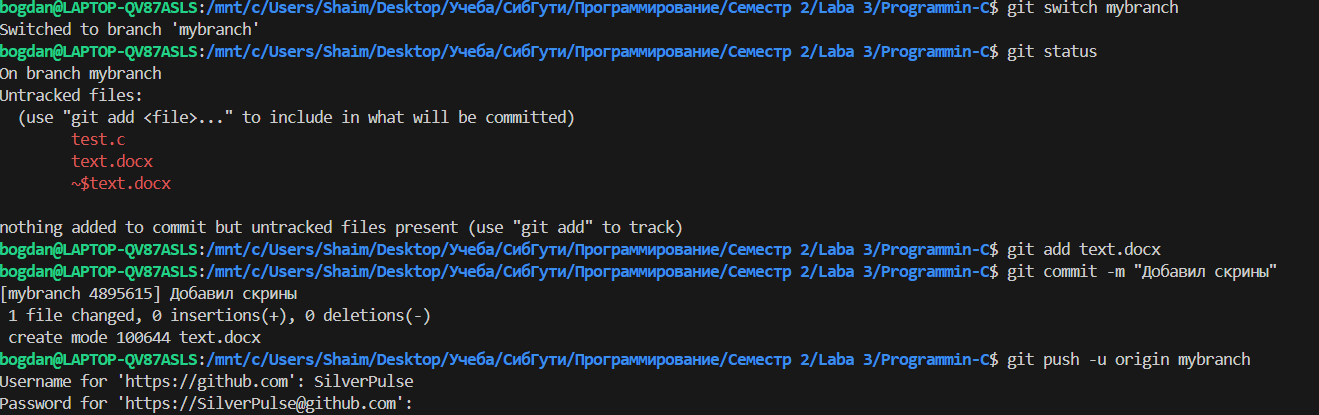
9,10 Возвращаемся в main. Добавились комментарии в main

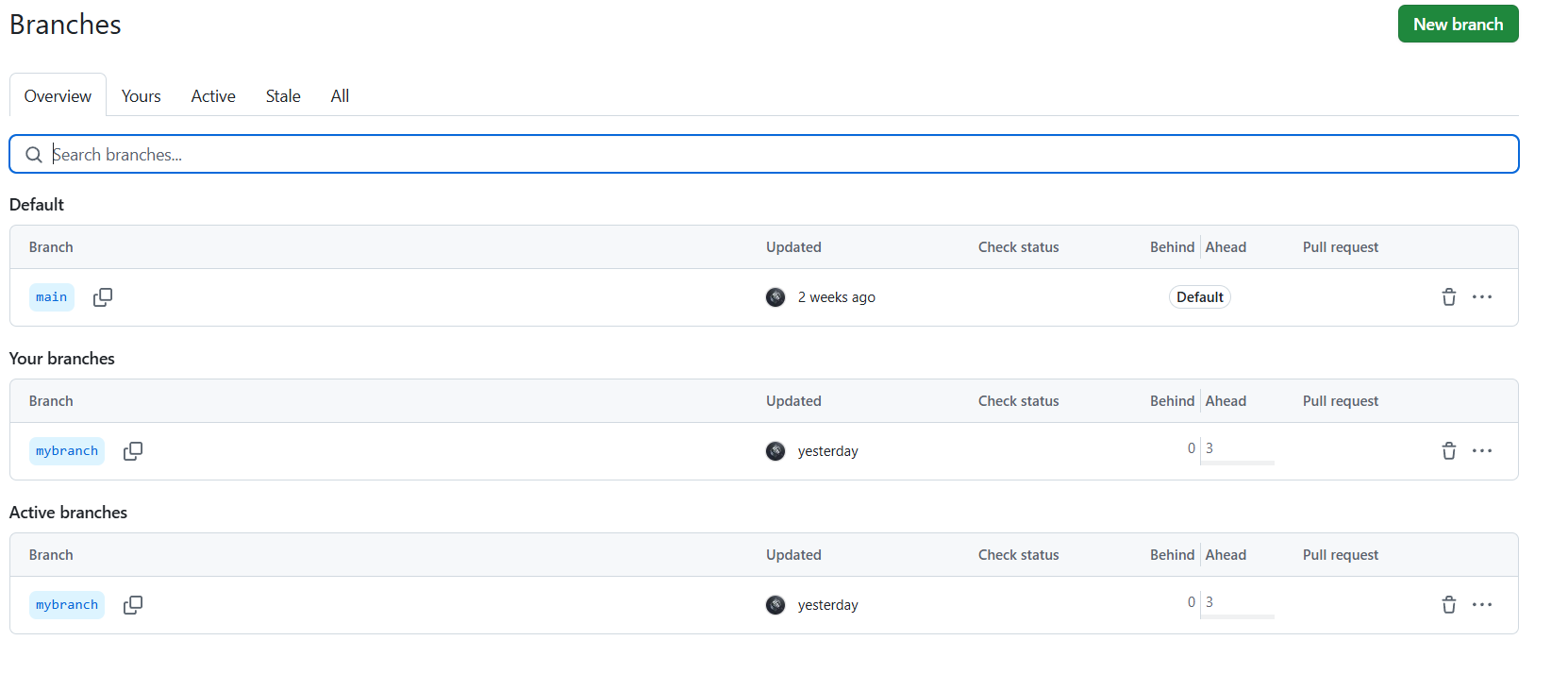


11,12 Создали файл file2.txt добавляем и коммитим. Используя git log –oneline –graph,чтобы увидеть что наша ветка указывает на новый коммит, и что у двух веток разные коммиты.

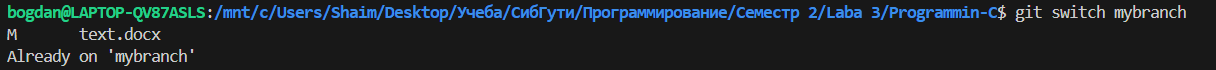


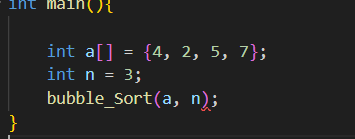
14 Если перейти в другую ветку, то файл останется в ветке main.

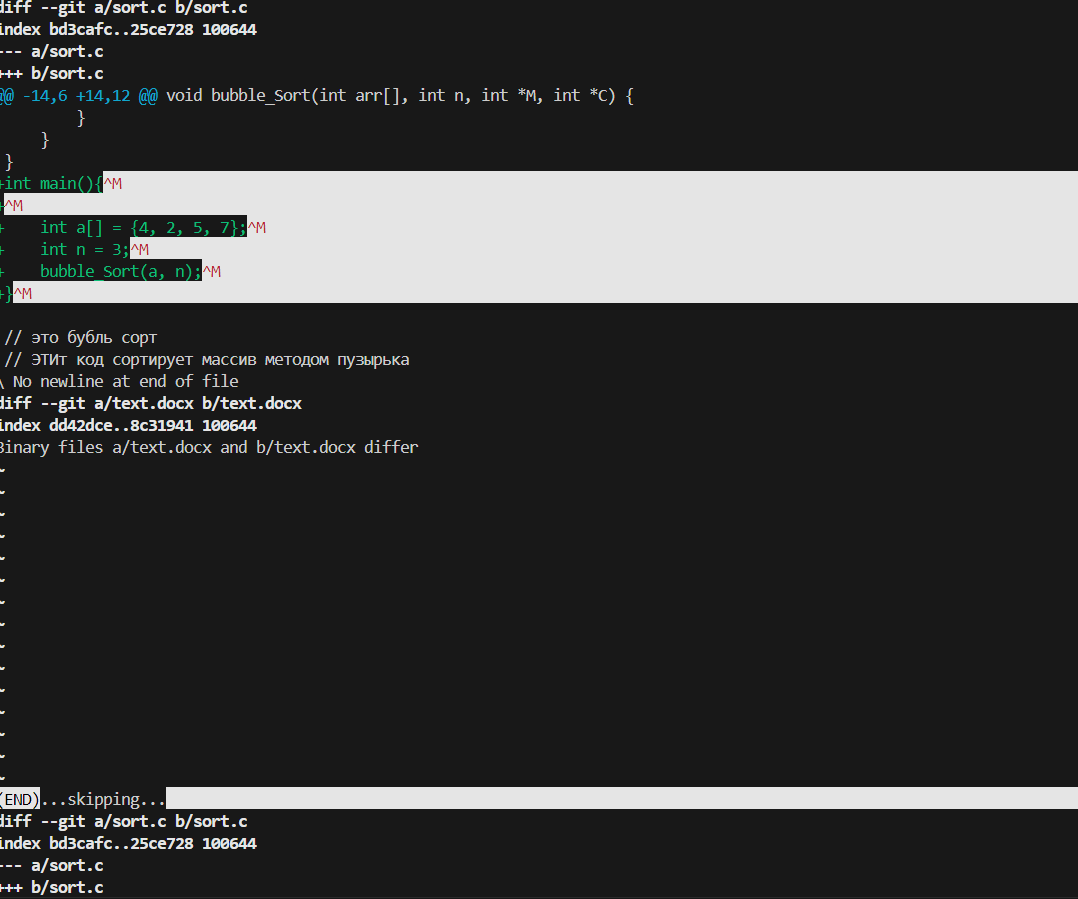
15,1Перешел в mybranch из main.Добавим ворд файл text.docx в ветку mybranch, для этого коммитим и пушим на удаленный репо.  
  


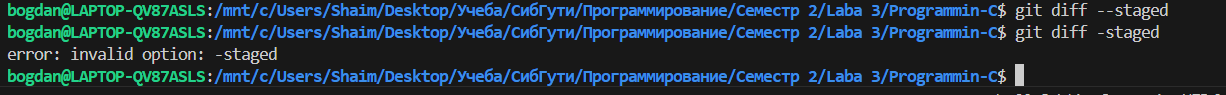
Теперь проверим ветки на гит хабе, всё чётко  


На четыре!!!  
1 переключаемся на ветку mybranch



2 добавил функцию с массивом и вызвал функцию  


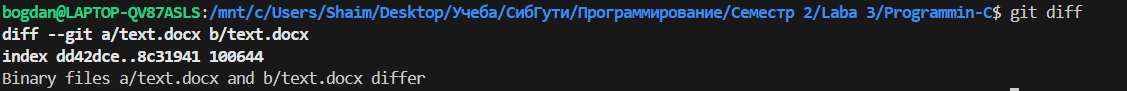
3 git diff результаты  
****

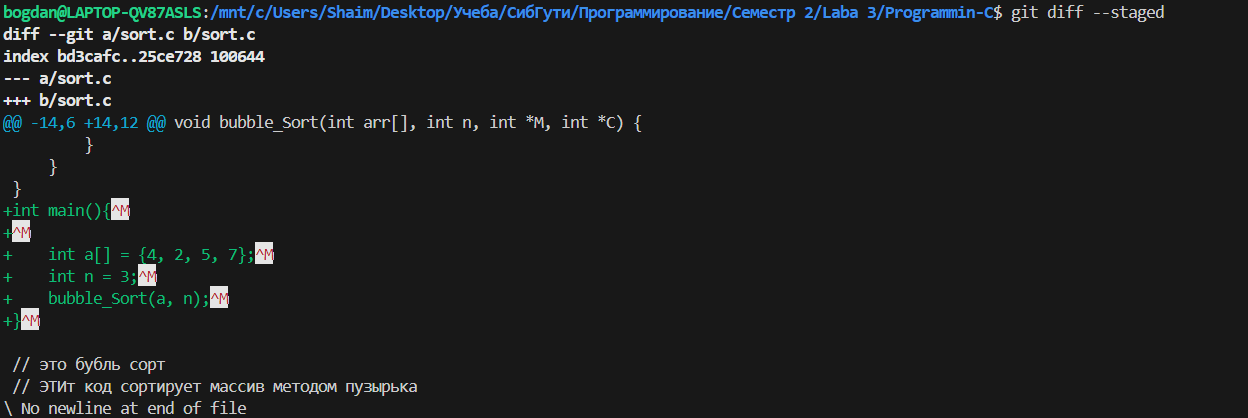
4 git diff –staged ничего не дает   


5 добавил в staged файл sort.c

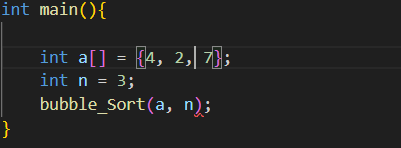


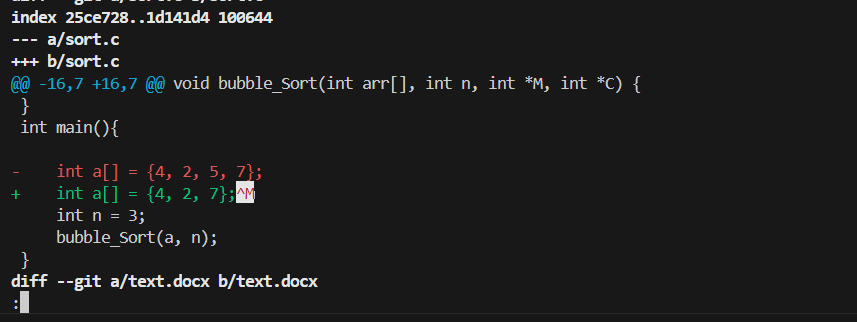
6 git diff пустой



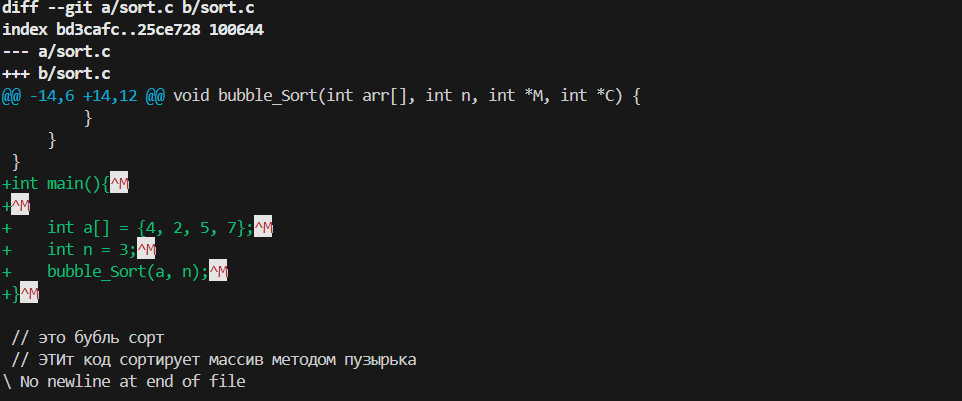
7 git diff –staged выдает те же изменения  


8 удалили элемент



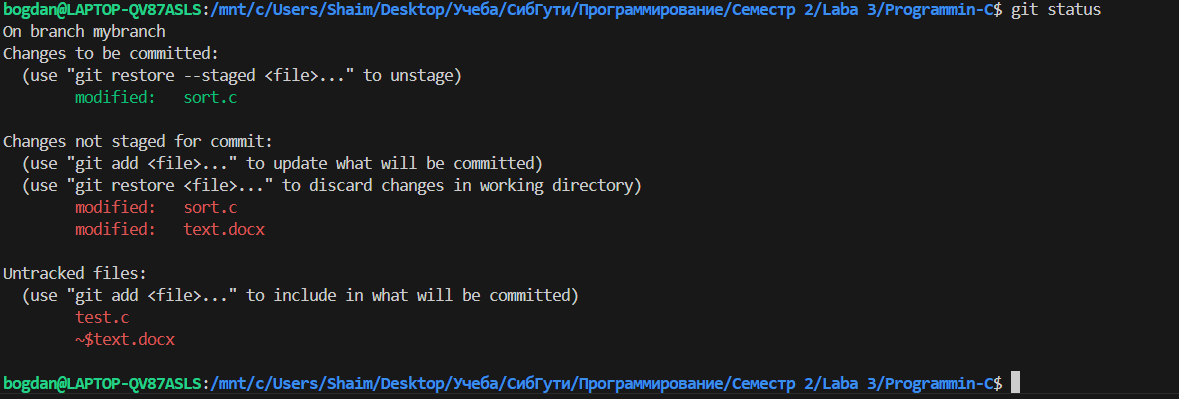
9 git diff выводит следующее. Указывает на изменение  


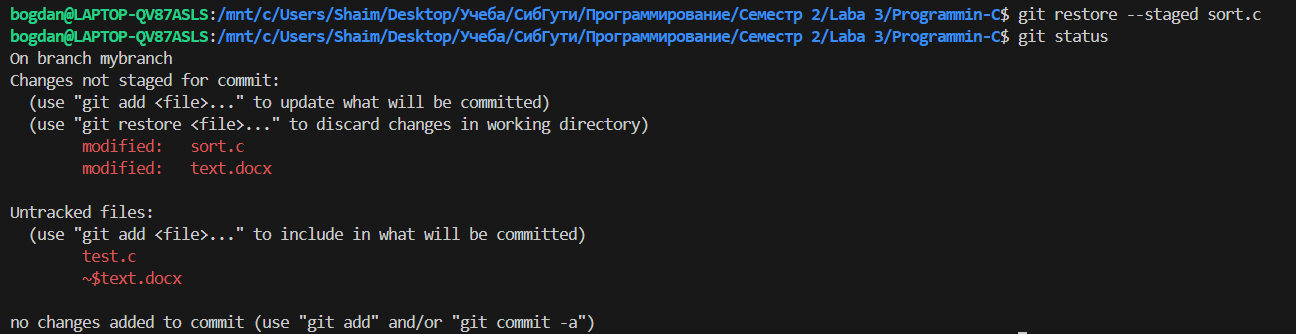
10 git diff –staged говорит

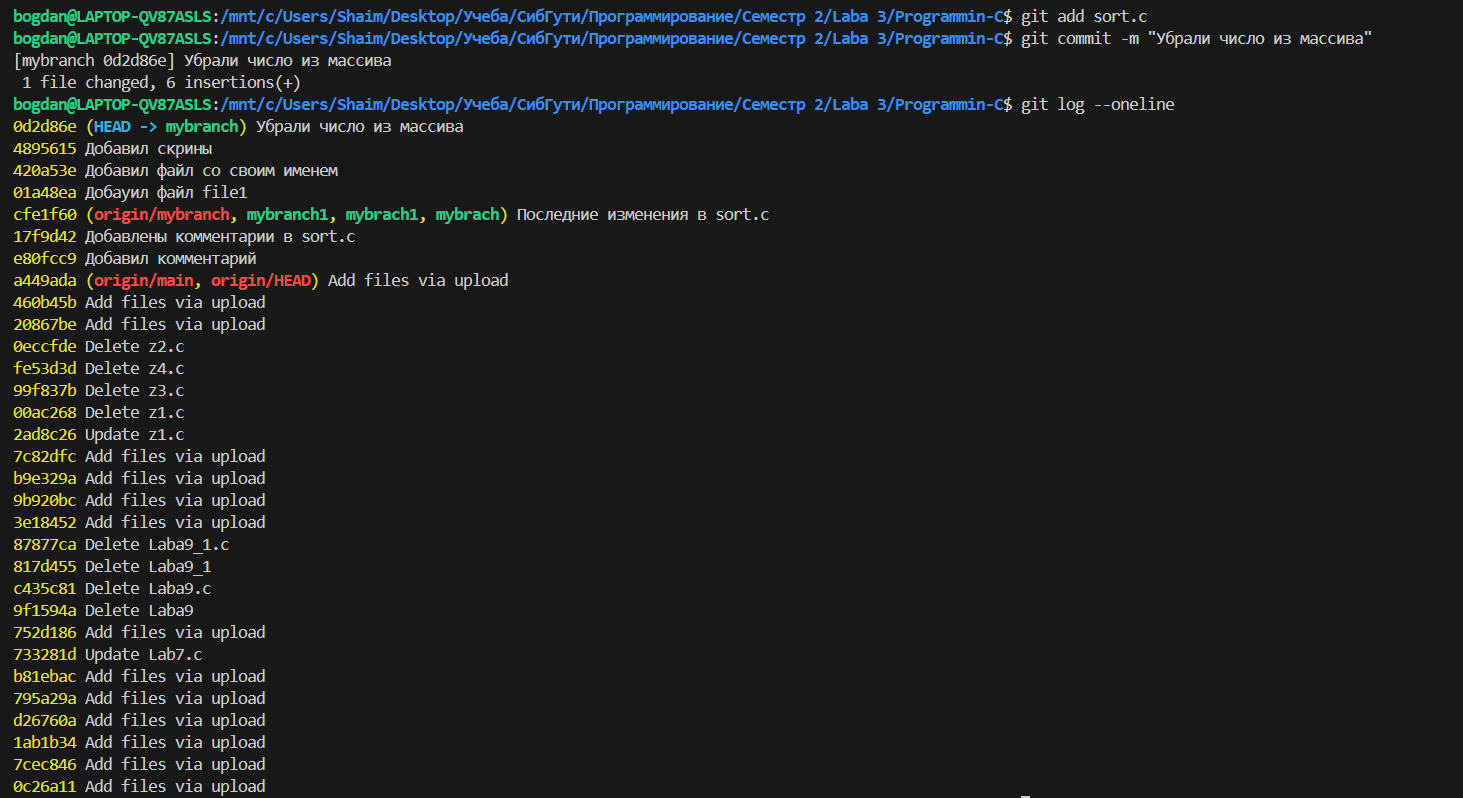


11 git diff – staged указывает нам изменения предыдущие, ведь мы не сделали git add sort.c

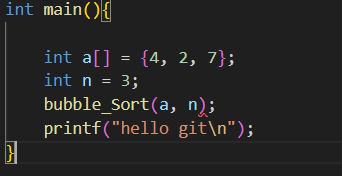
Git diff показывает изменения в рабочей директории, которые еще не добавлены в staging area. А git diff –staged показывает изменения в добавленных файлах

12 sort.c присутствует дважды в выводе при git status. И text.docx потому что я его редачу параллельно выполнению лабы  


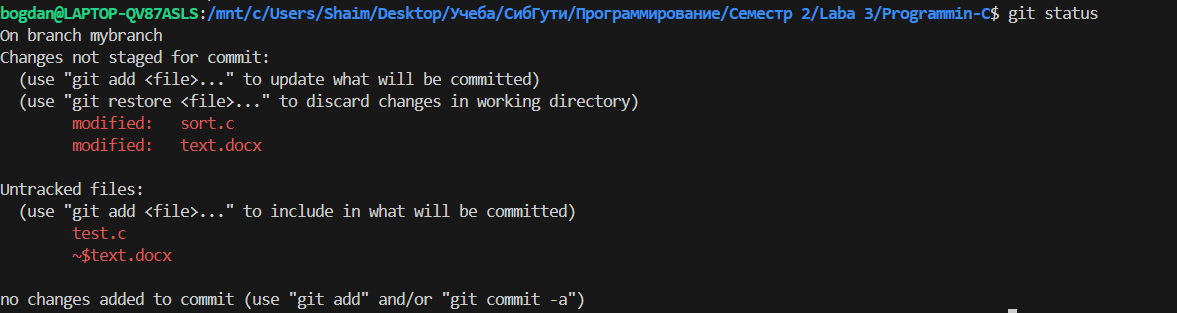
13,14 git status повествует о том,что sort.c находится только в разделе Changes not staged for commit  


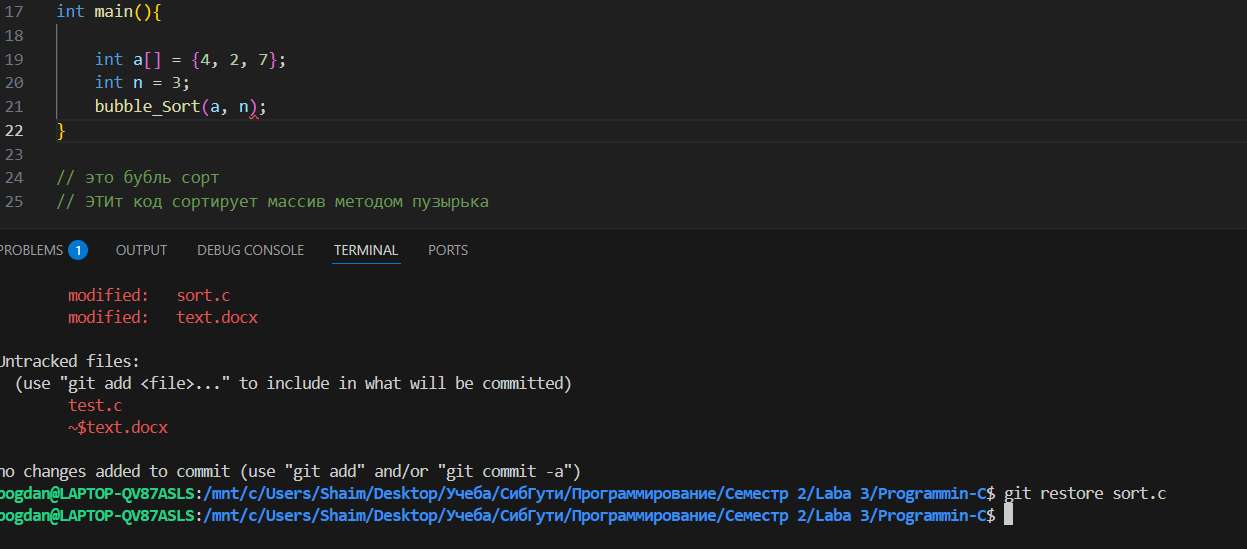
15,16 После добавления изменения в staging area(add) git log теперь выглядит так:  


17,18 добавил hello git\n

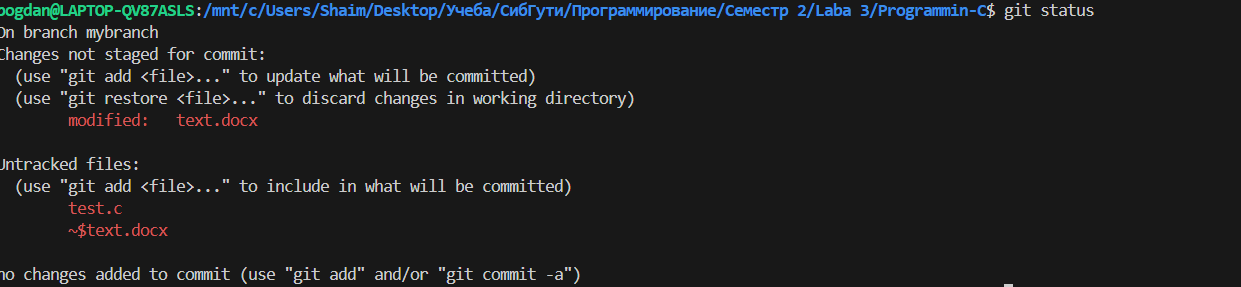


19 с помощью git status файл изменен но не добавлен в staging area



20,21 после git restore sort.c у нас пропала ласт строка которую мы только писали. То есть откатились на шаг назад  


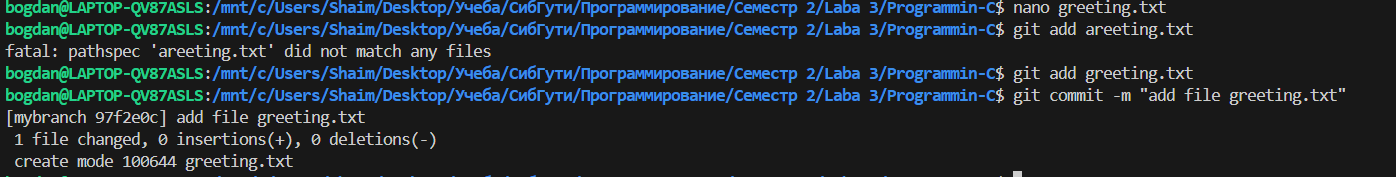
22

Git status выводит следующее  


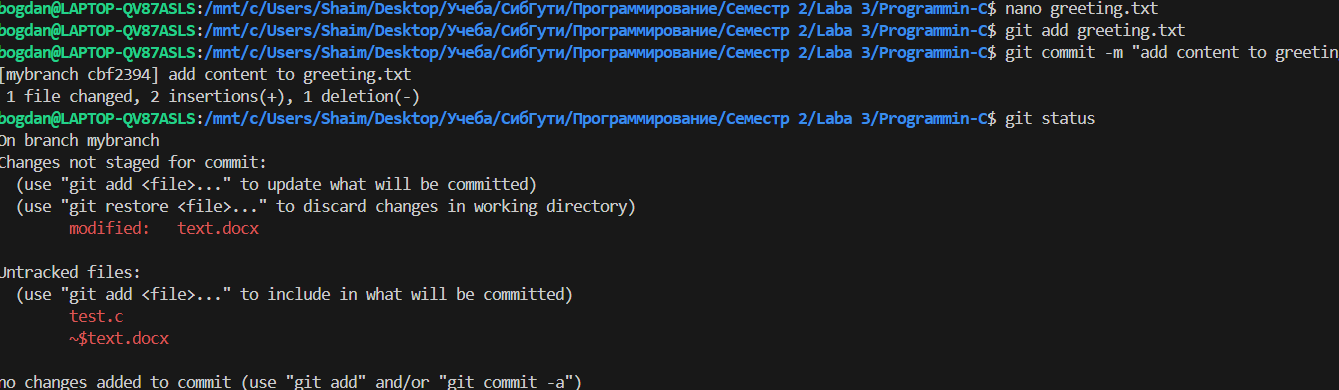
Пушим на удаленную ветку репо

Теперь поиграемся с ветками и ff-merge

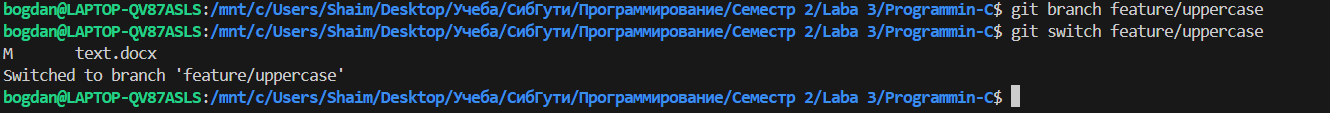
когда вы хотите обновить ветку так, чтобы она также имела все изменения из другой ветки, используйте команду 'git merge [имя ветки]', где [имя ветки] - ветка, из которой мы хотим смержить наши изменения.

1,2 создаем добавляем greeting.txt  


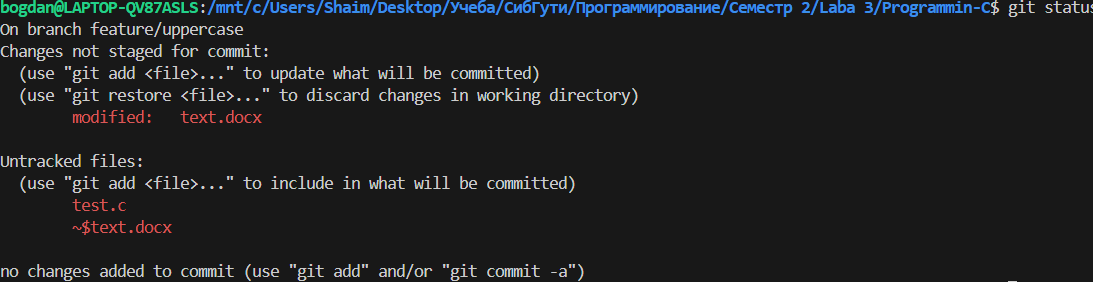
3 написали hello в txt файле, добавили в область, закомитили.



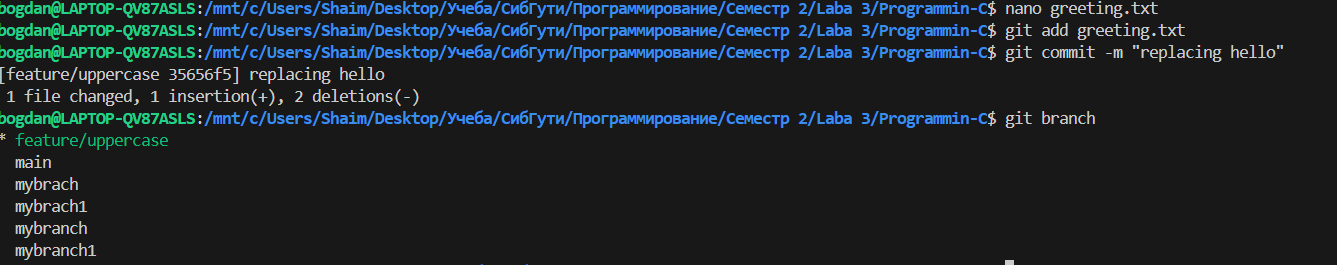
4 Создали ветку с именем feature/uppercase



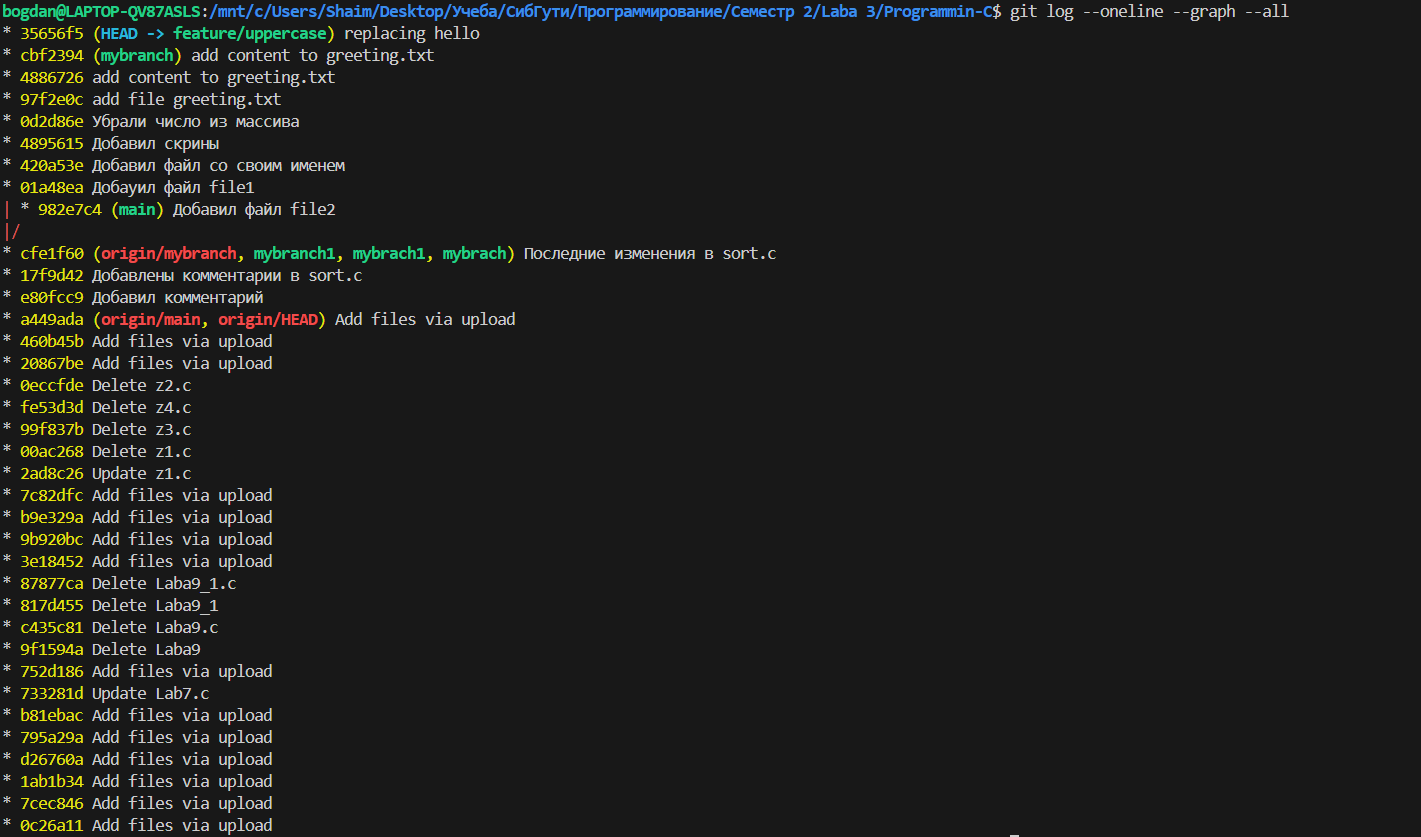
5 git status вывод



6,7,8 поменяли hello -> HELLO закомитли, добавили. Проверили ветку

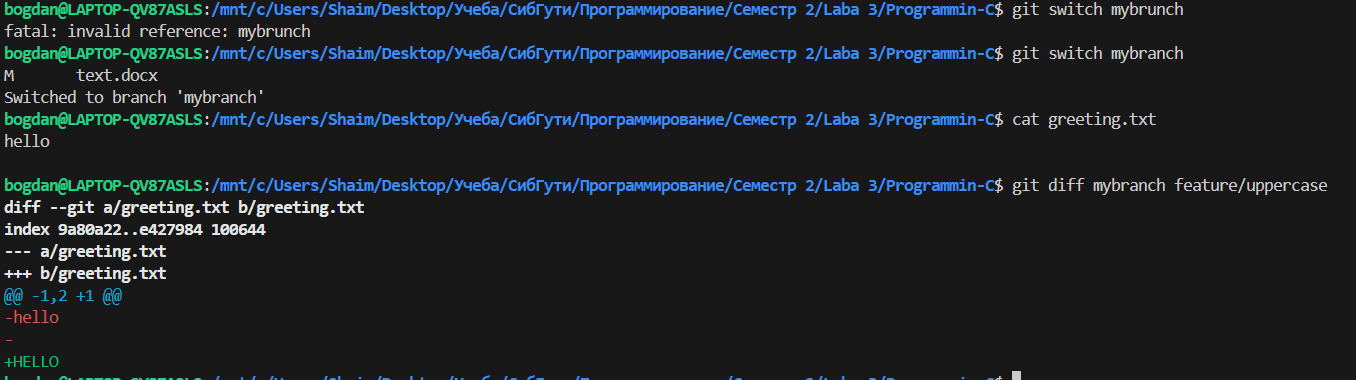


9 результат git log --oneline --graph –all

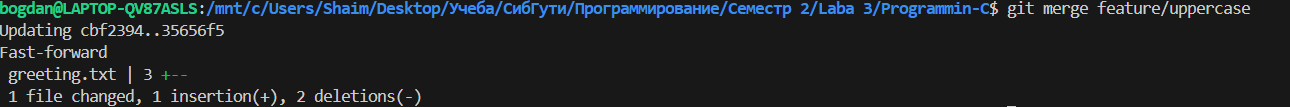


10,11,12 Переключились на главную ветку

Увидели содержимое файла greetings.txt, сравнили

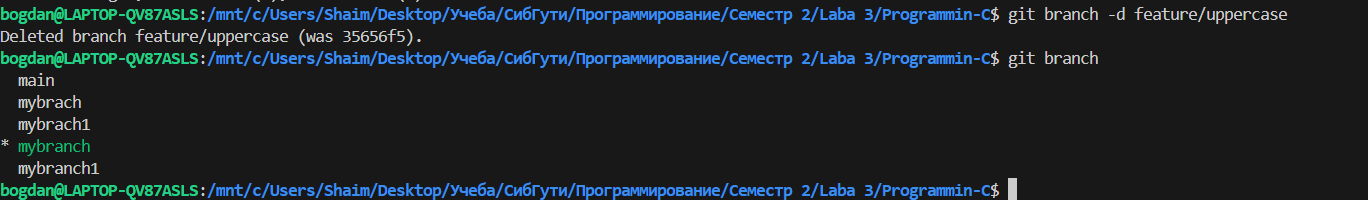


13 объединяем mybranch и feature/uppercase



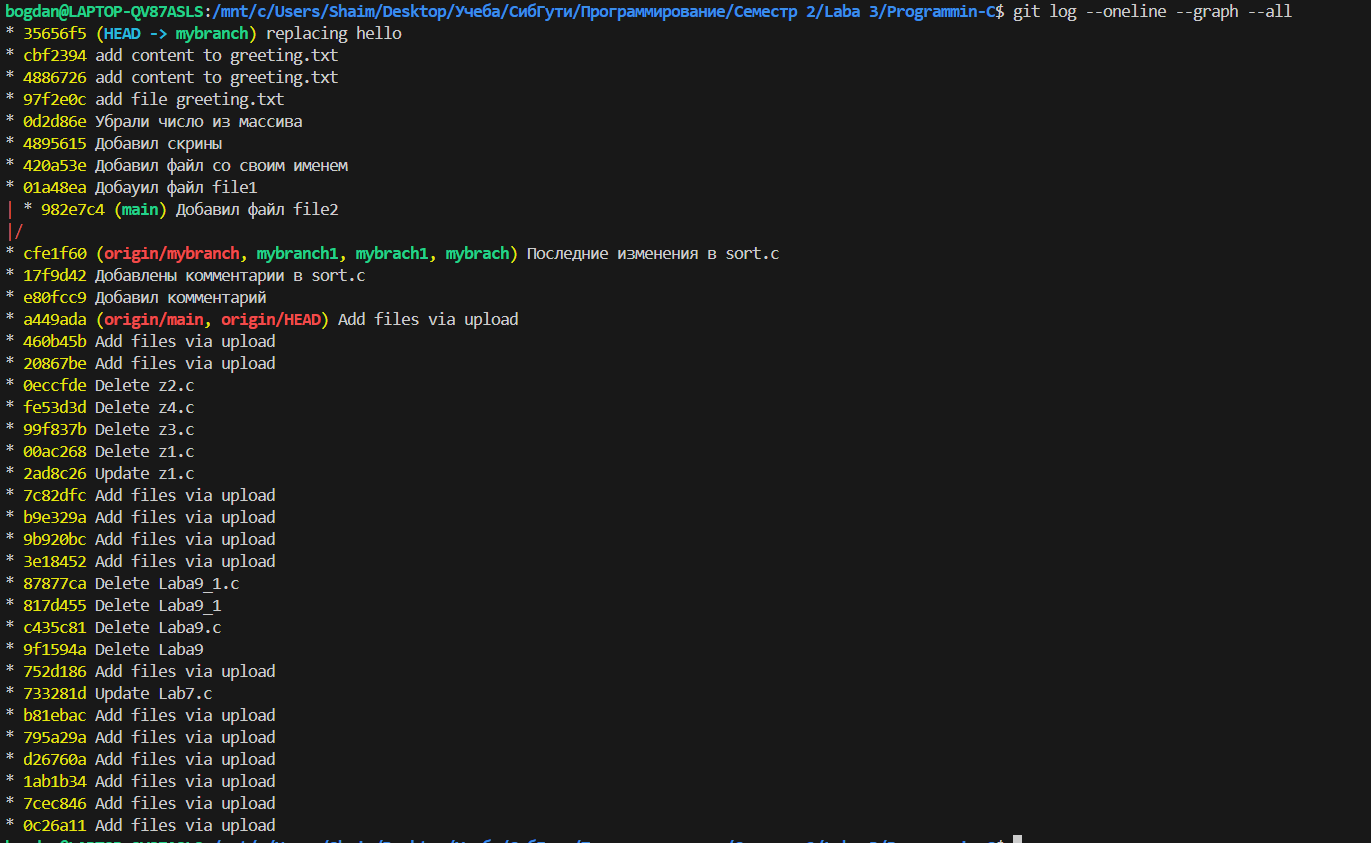
14 содержимое файла

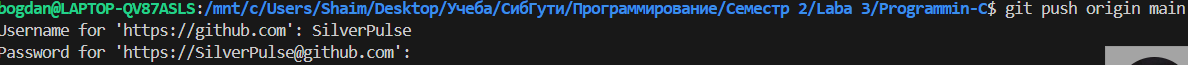
15 удаляем ветку, проверяем что получилось



16 смержили ветку  

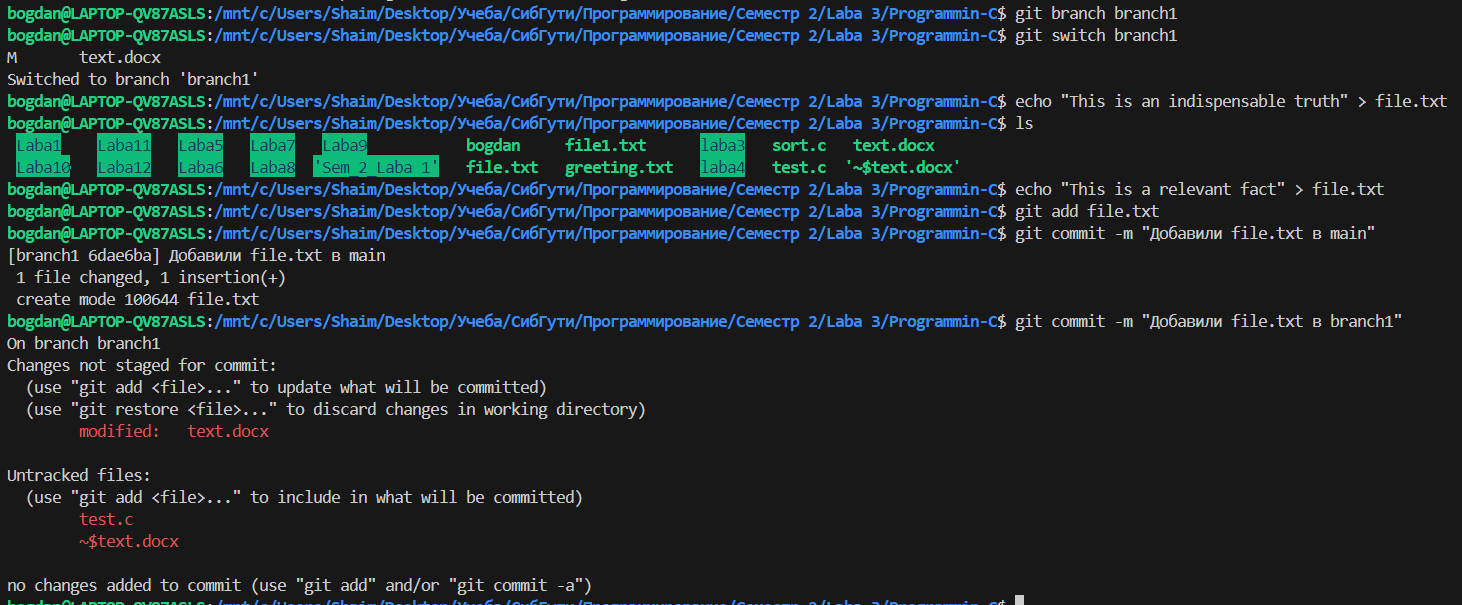

17 git log –oneline –graph –all



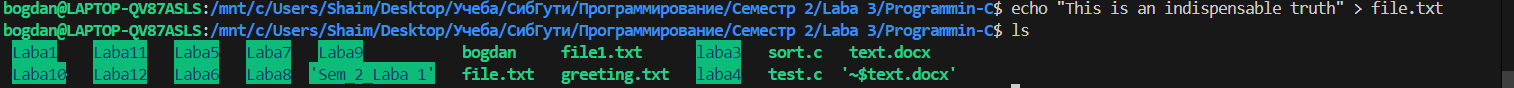
Запушим изменения ветки main на удаленный репо  
пушим сам документ test.docx на удаленный репо  


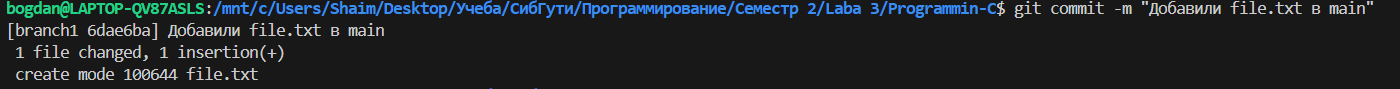
Na 5!!

1. Создаем ветку branch1, переключаемся на нее
2. Выполним команду   
   $ echo "This is a relevant fact" > file.txt
3. Закоммитим это изменение

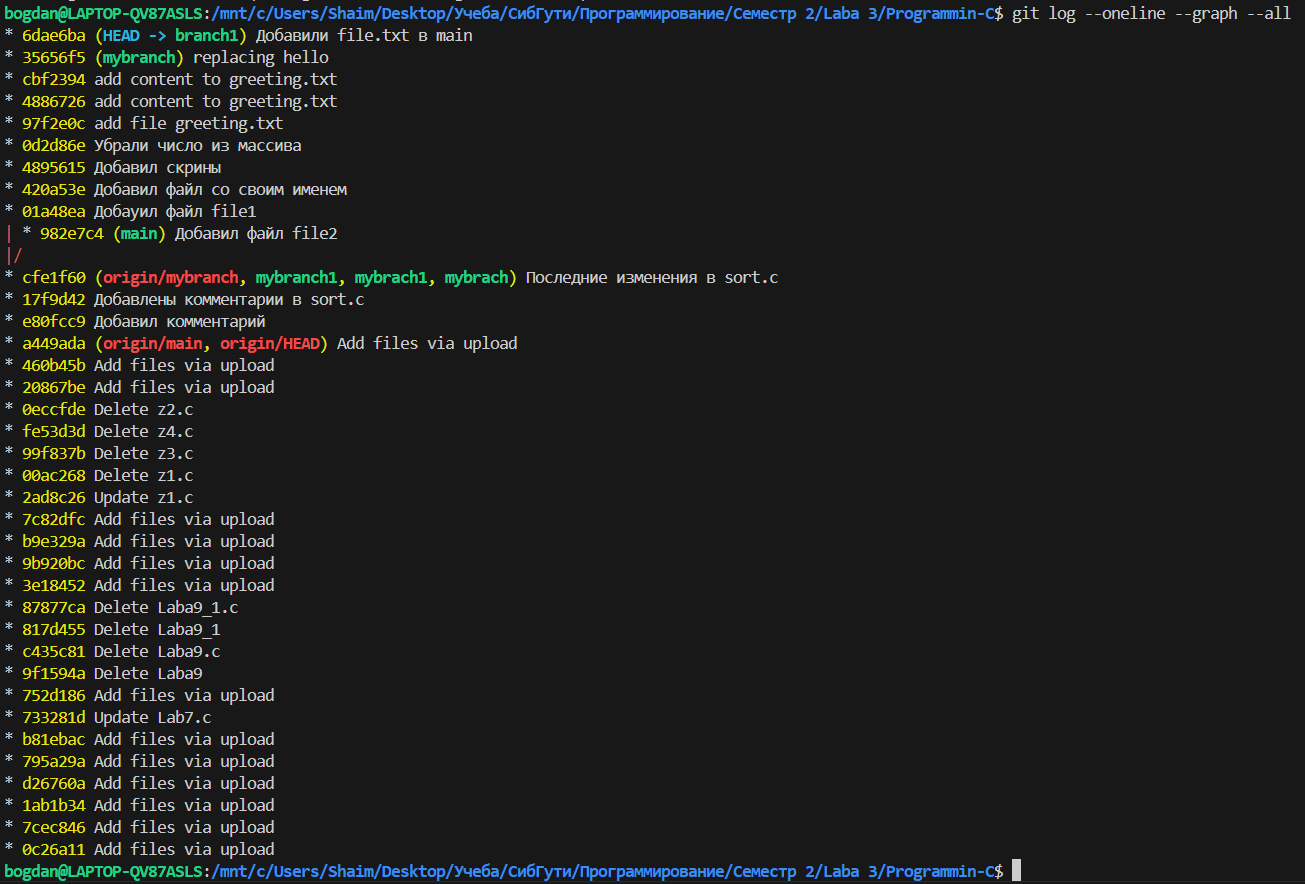


1. Переключимся на главную ветку и выполним команду  
   echo "This is an indispensable truth!" > file.txt



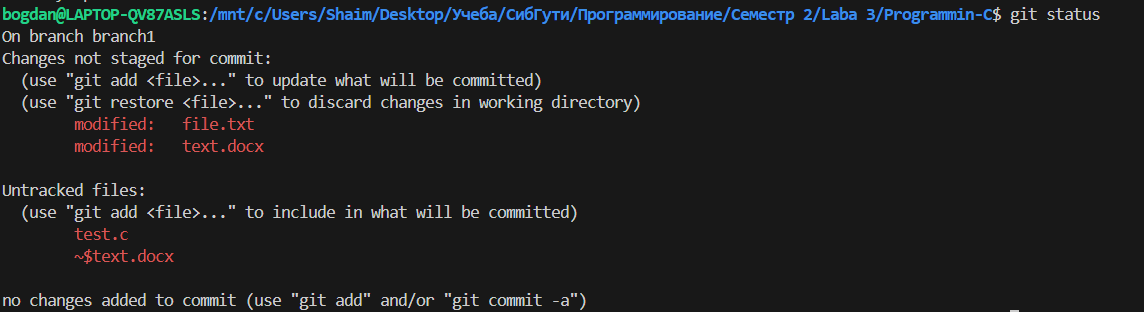
5 добавили файл в main  


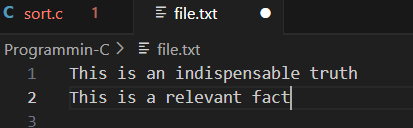
6 вывод git log --oneline --graph –all



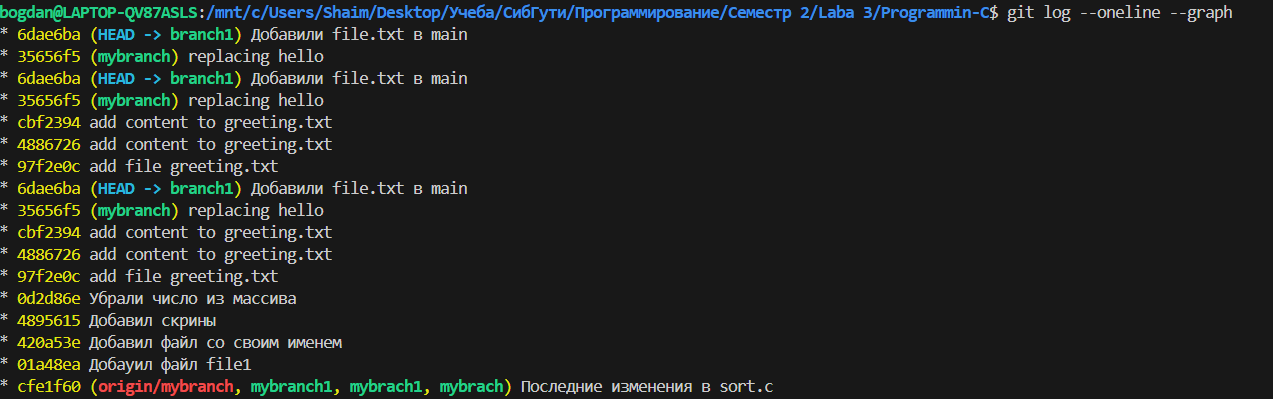
7 Смержили ветку



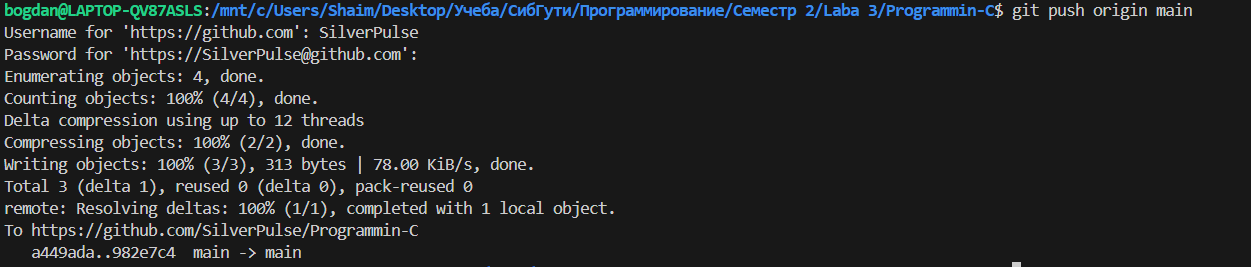
8 наш gitstatus  


9 file.txt содержимое  


10 git log --oneline –graph

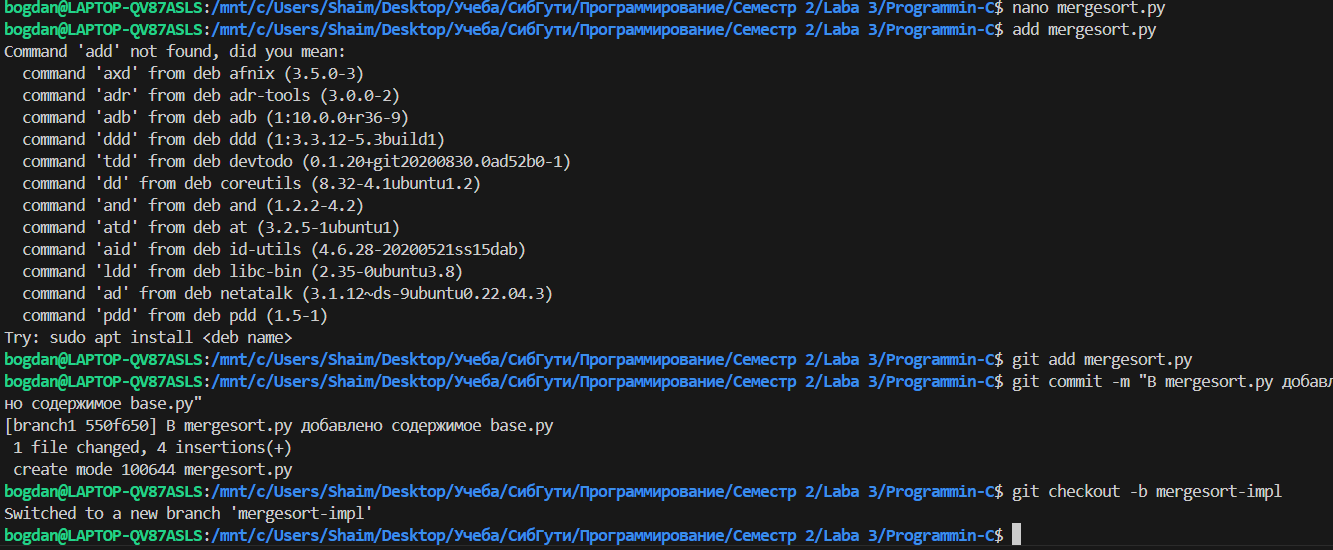


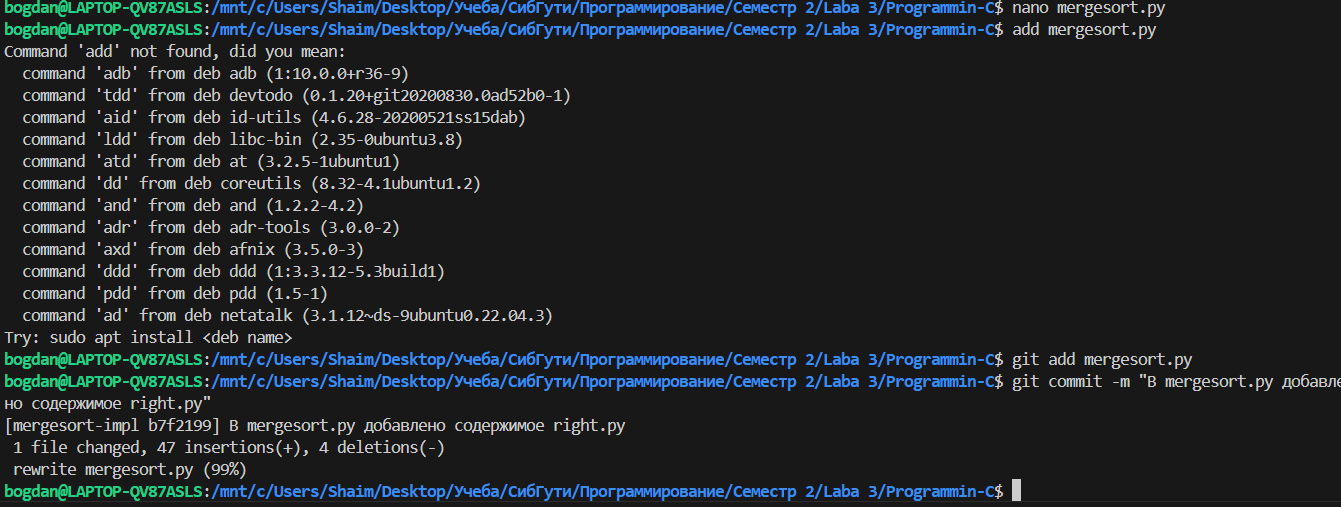
11 пушим изменения



Починим Merge конфликты для сортировки MergeSort на python. **Содержимое base.py, lefty.py и righty.py**

Создаем mergesort.py с содержимым base.py добавляем коммитим создаем ветку mergesort-impl



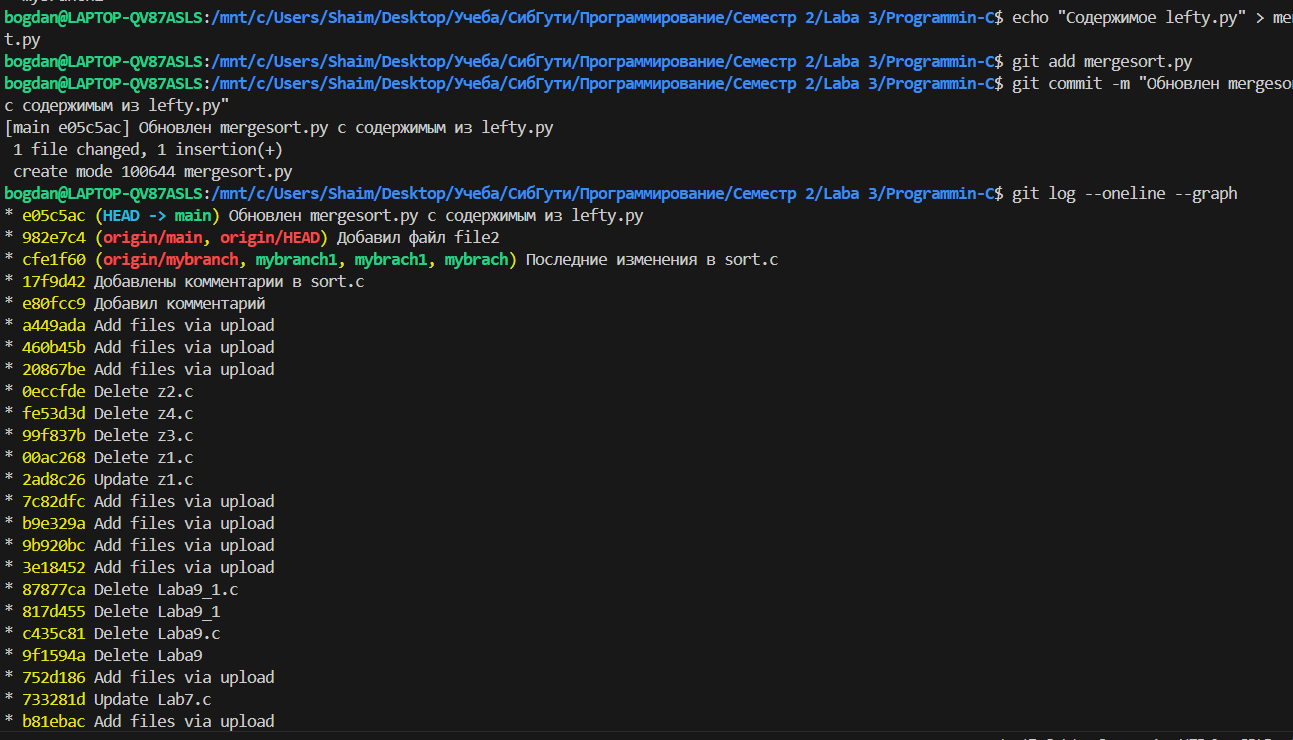
Меняем содержимое mergesort.py py с содержимым base.py на right.py  


6-8

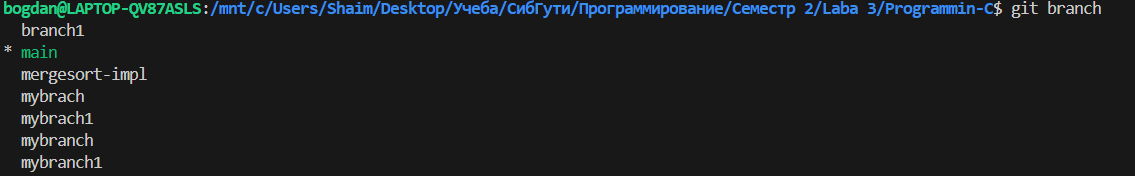
Переключаемся на master и меняем все содержимое mergesort.py на lefty.py

Коммитим изменения.

git log --oneline –graph



9 в git branch мы в main



10