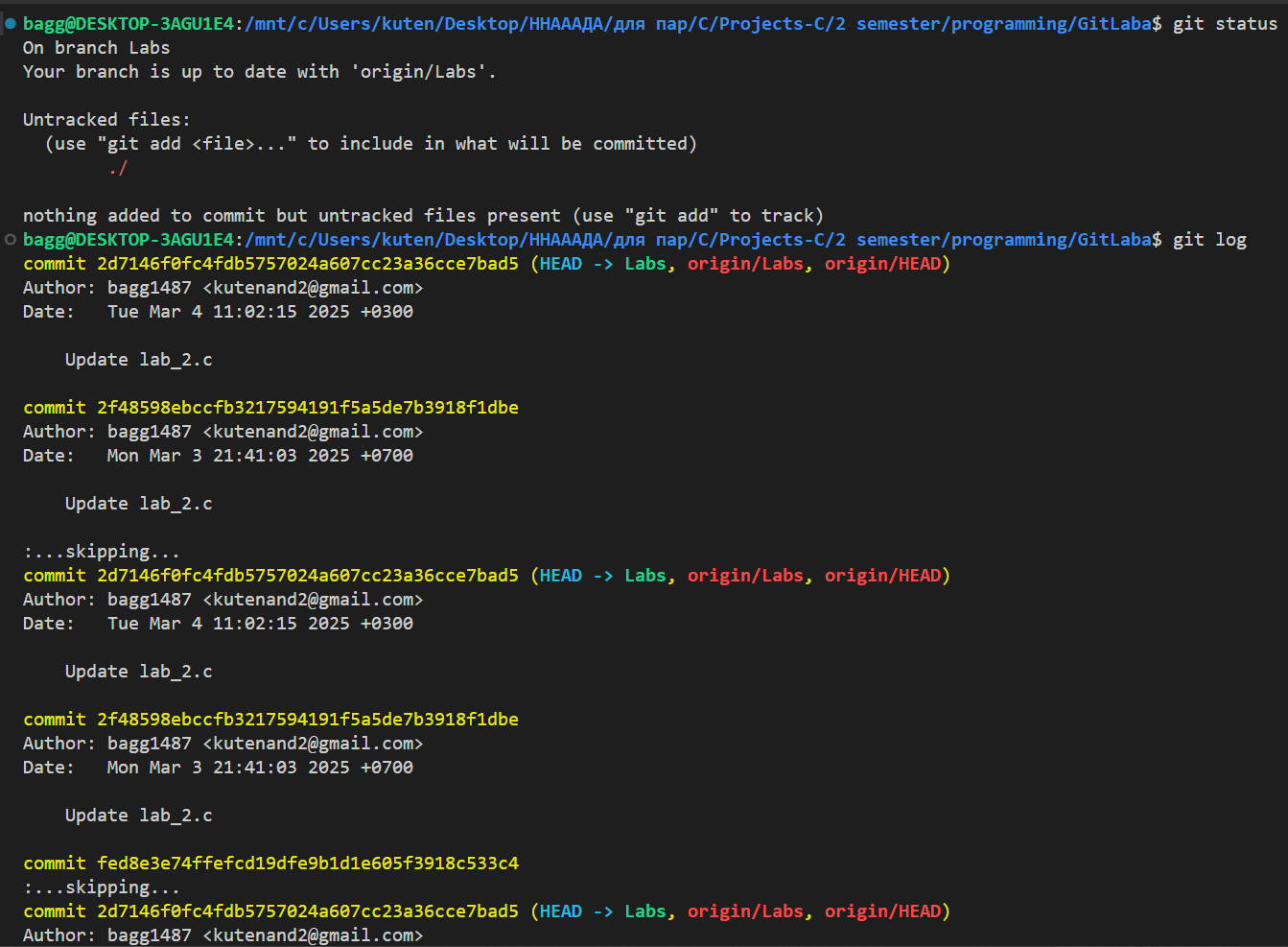
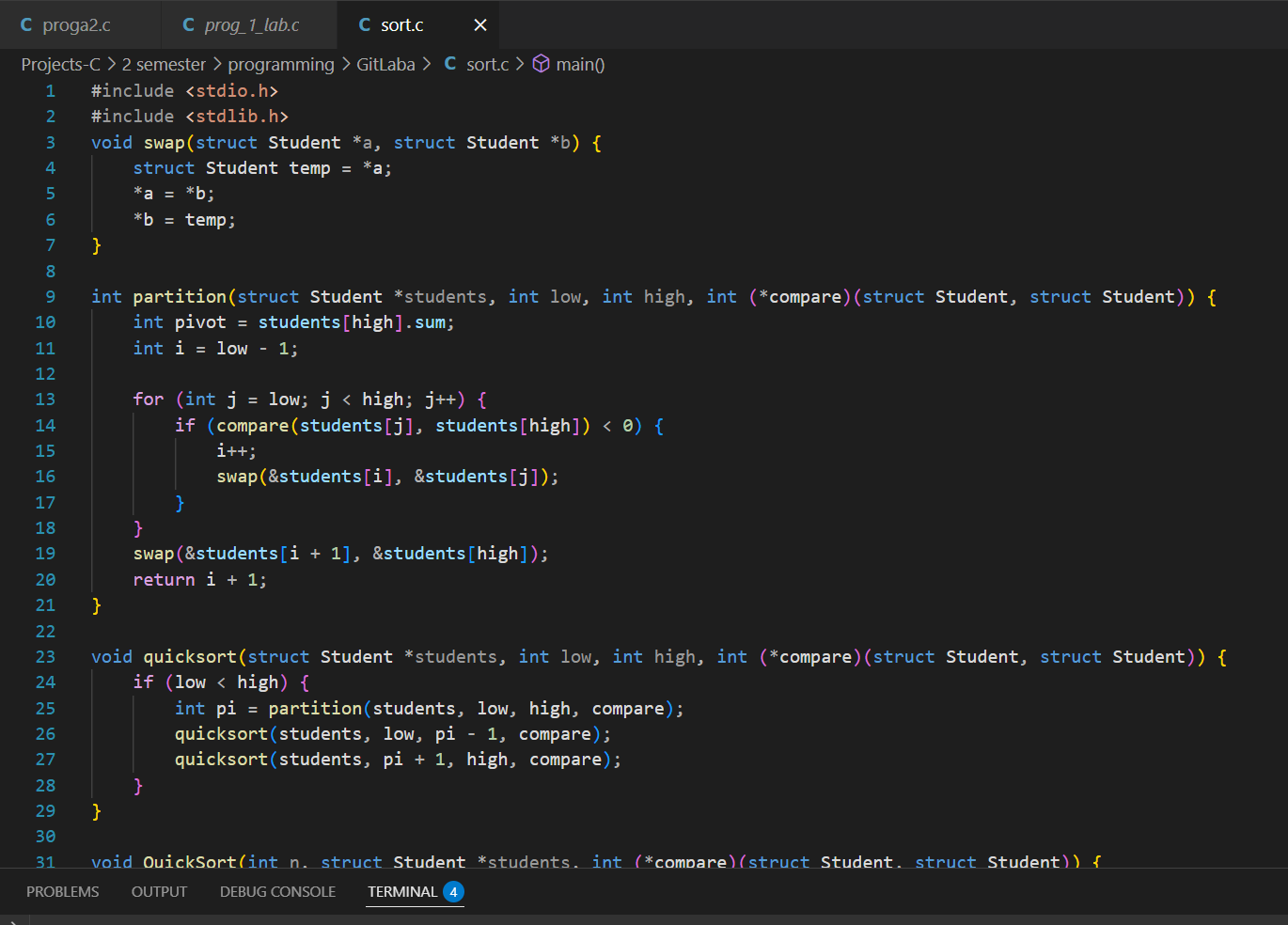
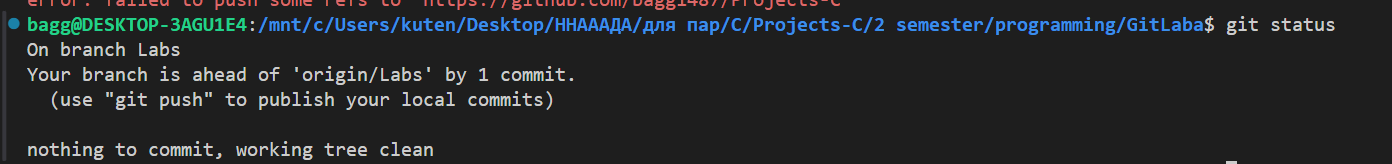
1,2)Использовал **git status**, чтобы узнать, на какой ветке я нахожусь. Посмотрел, как выглядит **git log.**



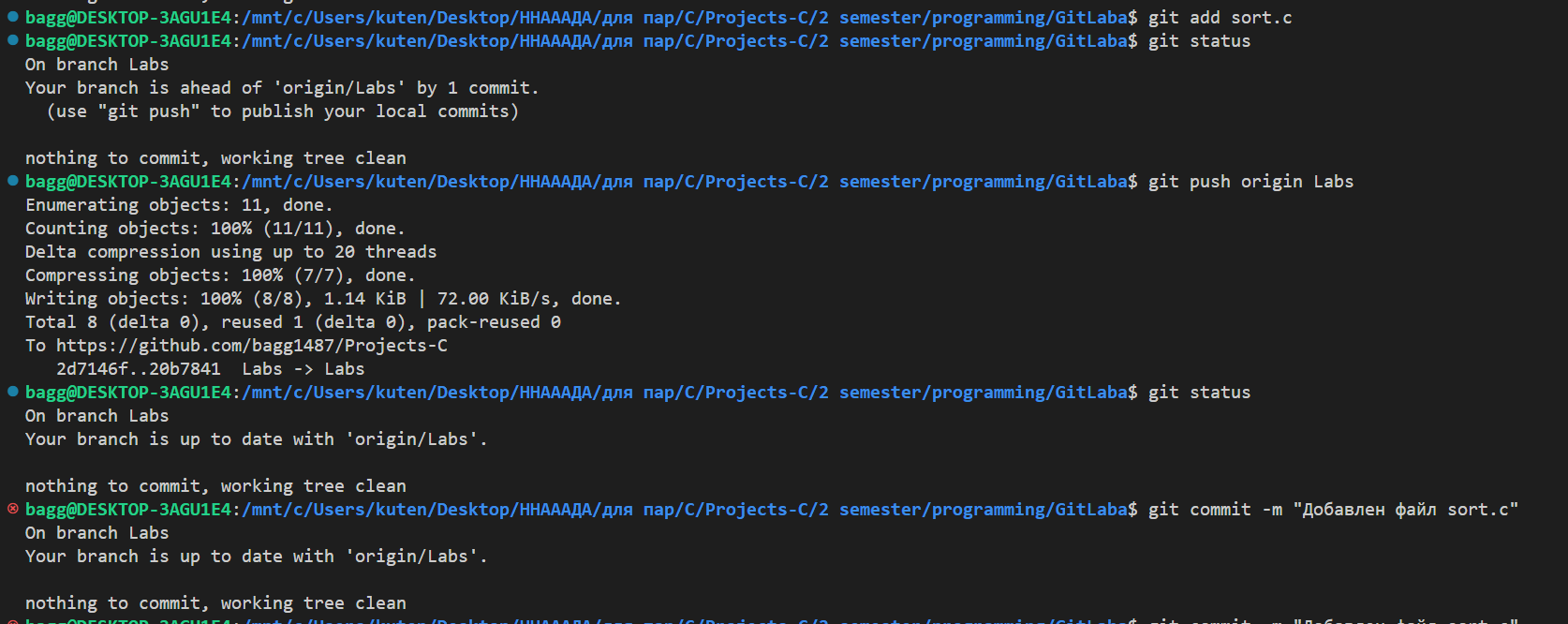
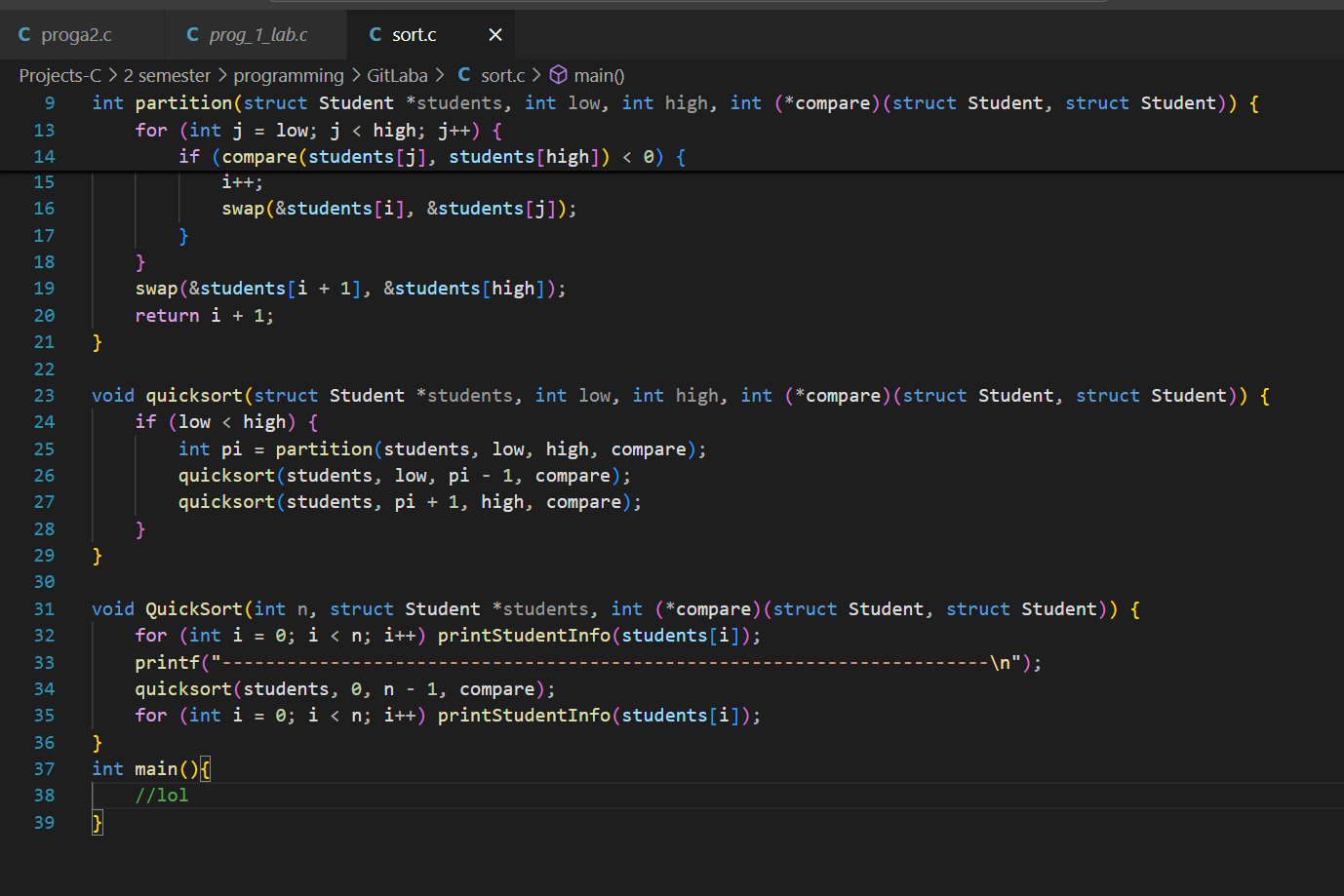
3) Создал файл sort.c и вставил туда код функции сортировки Хоара.

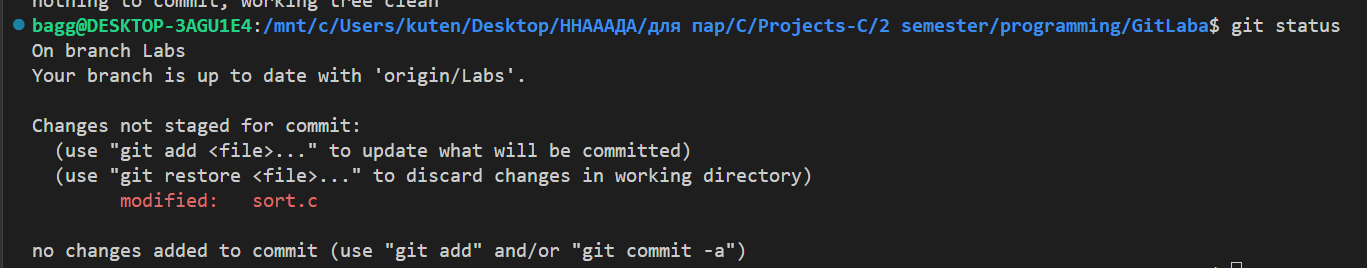


4) Посмотрел, как выглядит **git status.**

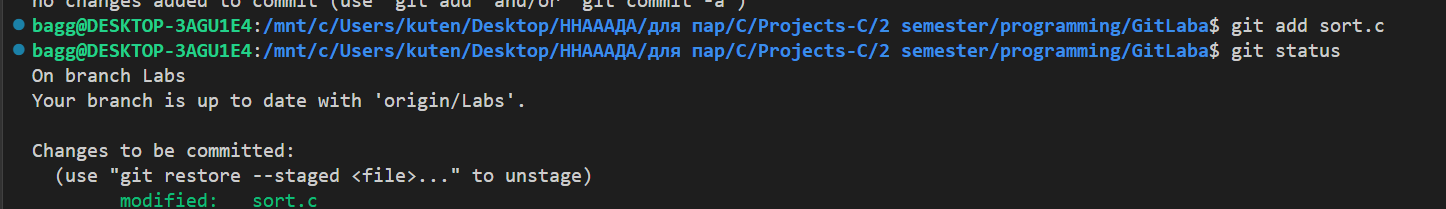
****

5,6,7,8) Добавил файл в файл в область stage (add) и посмотрел **git status,** закоммитив его.

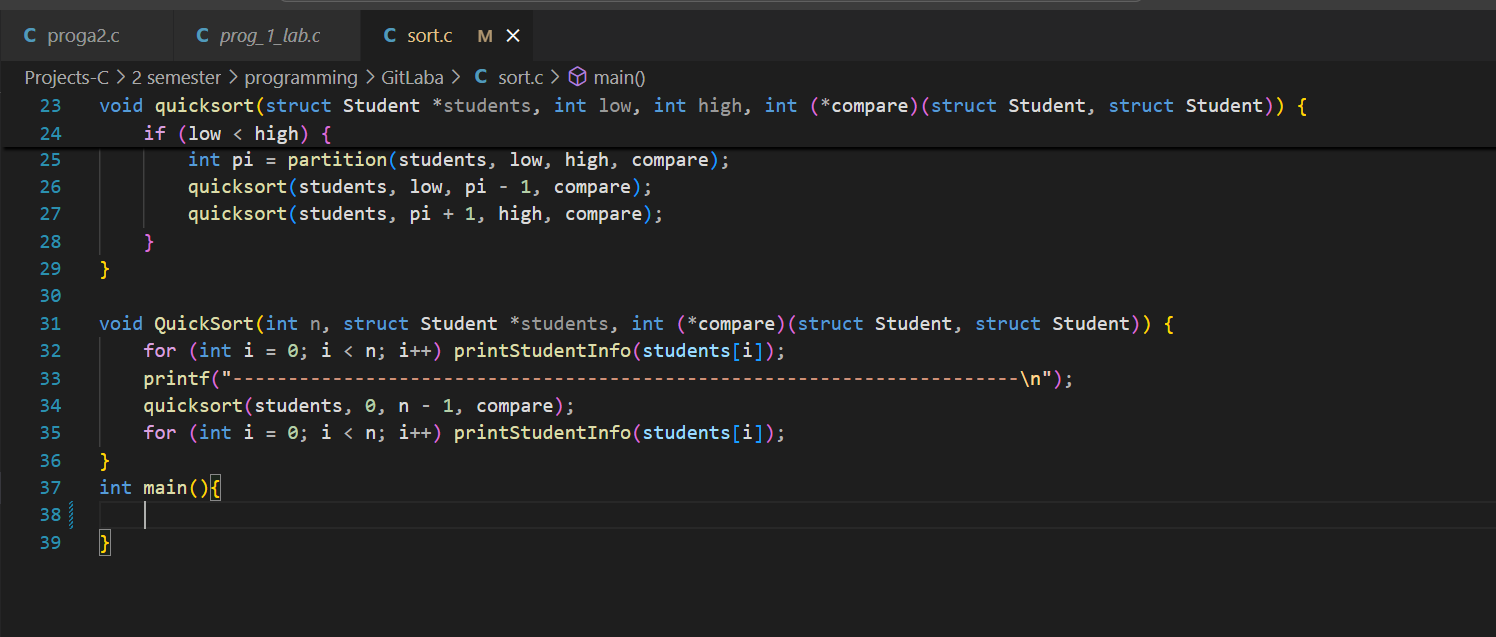
9,10) Добавил комментарий в код и посмотрел **git status.  
**

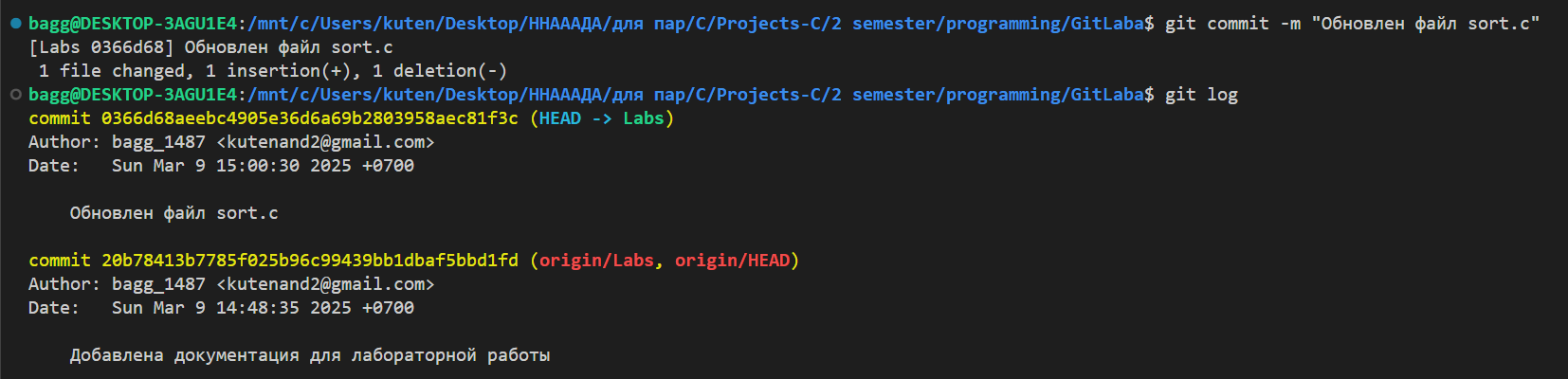


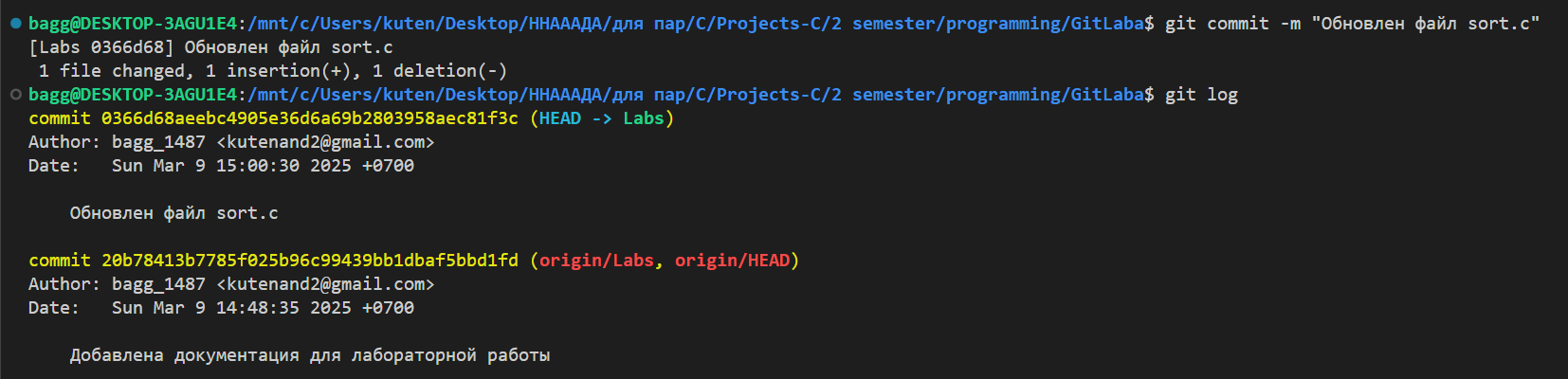
11,12) Добавил файл в файл в область stage (add) и посмотрел **git status.**



13,14,15) Изменил файл, коммитнул и посмотрел журнал (log).

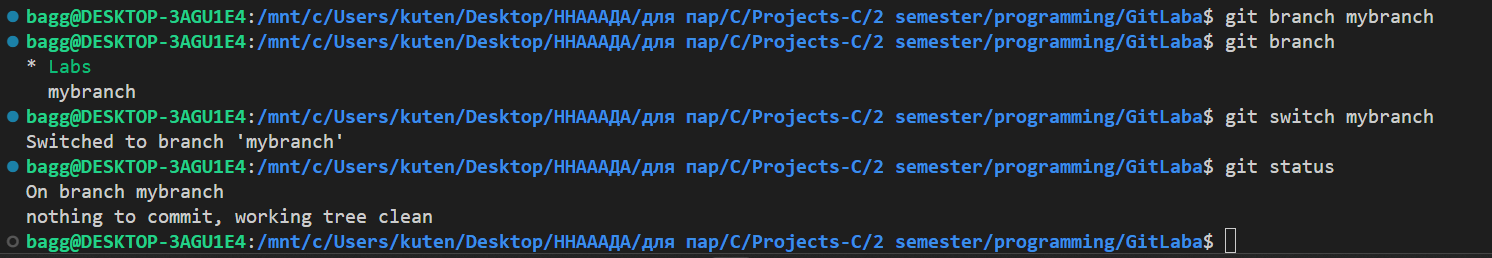


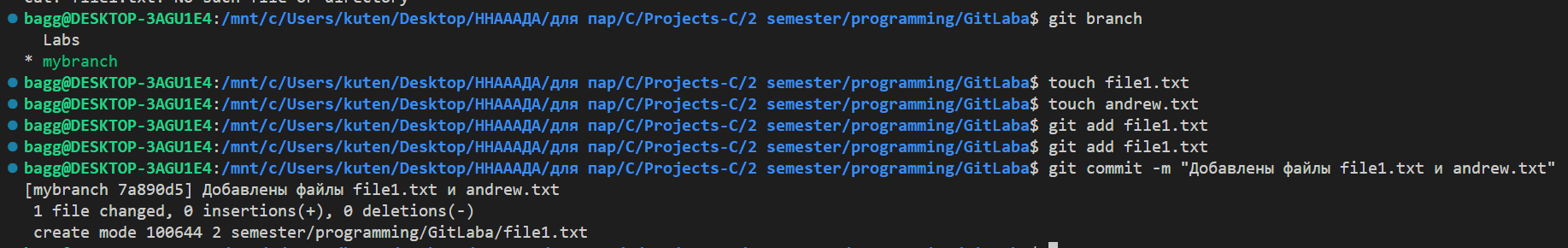


16,17) Добавил в **stage** и закоммитил последнее изменение. 

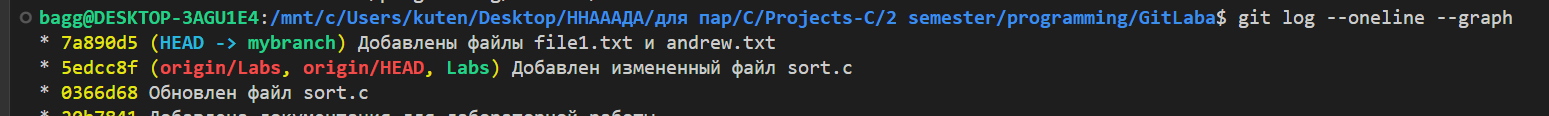
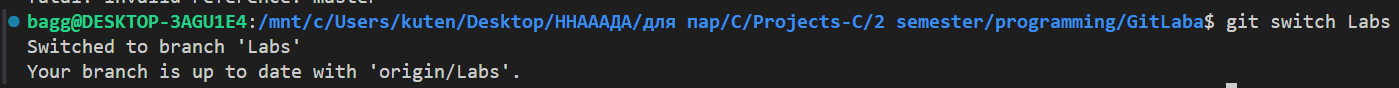
Работа с ветками.

1,2,3,4) Использовал **git branch mybranch**, чтобы создать новую ветку с именем **mybranch.** Снова использовал **git branch**, чтобы увидеть новую созданную ветку. Использовал **git switch mybranch**, чтобы переключиться на новую ветку. Вывод поменялся (вместо **on branch Labs** стало **on branch mybranch**).

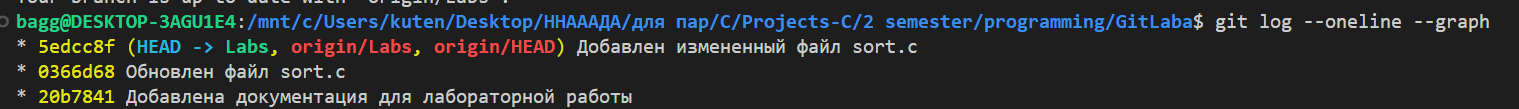


5,6,7) Убедитесь, что нахожусь на своей ветке **mybranch**, прежде чем продолжить. Создал файл с именем **file1.txt** и своим именем. Добавил файл и закоммитил это изменение 

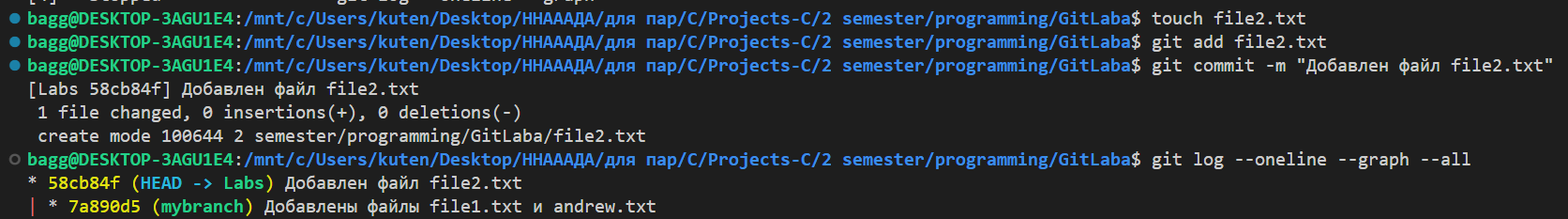
8,9) Использовал **git log --oneline --graph**, чтобы увидеть, что ветка указывает на новый коммит. Вернулся к ветке с именем **master**.

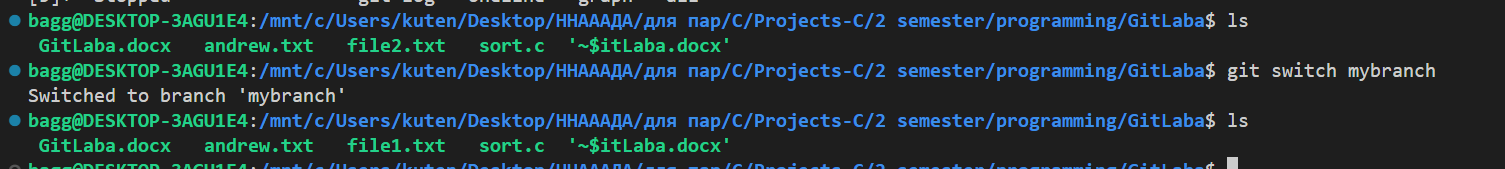
 

10) Использовал **git log --oneline --graph.** Поменялось то, что нет изменений, происходивших внутри другой ветки.

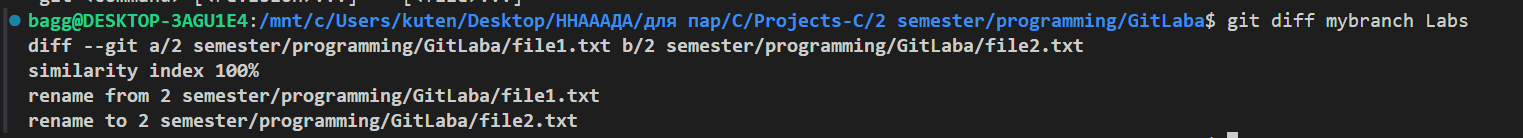


11,12) Создал новый файл с именем **file2.tx**t и закоммитил его. Использовал **git log --oneline --graph --all**, чтобы увидеть, что ветка указывает на новый коммит, и что теперь у двух веток разные коммиты.

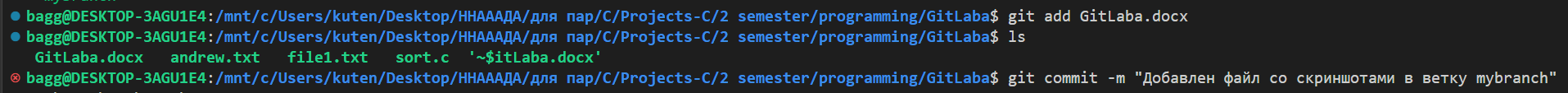


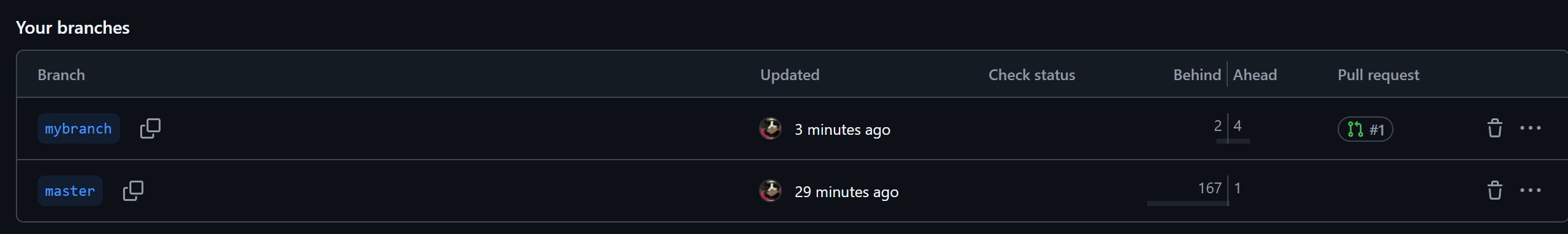
13,14) Переключился на ветку **mybranch.** Файл пропал.

15) Использовад **git diff mybranch master**, чтобы увидеть разницу между двумя ветками.



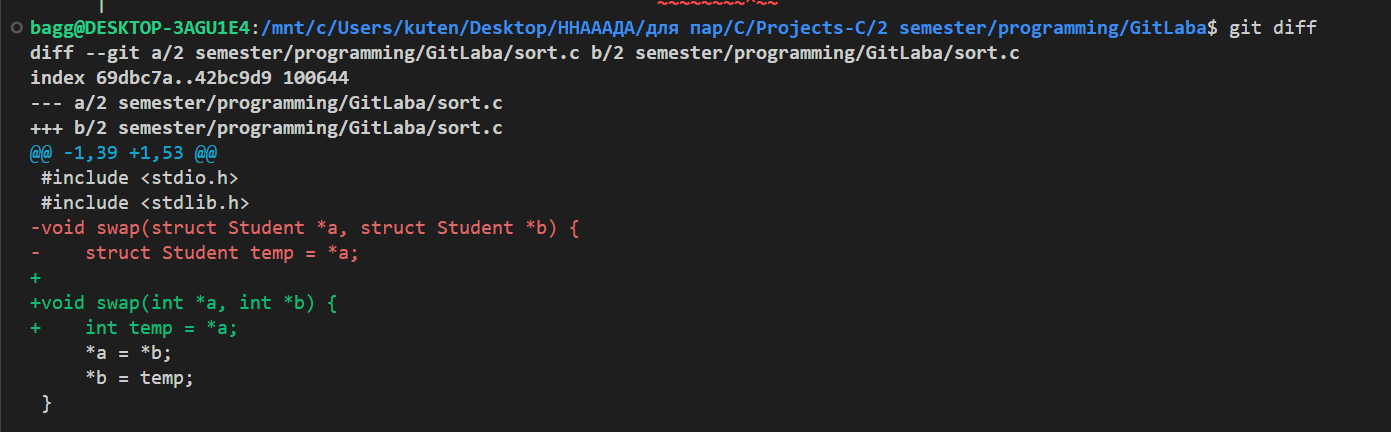
16) Добавил текстовый документ со скриншотами в ветку **mybranch**. Закоммил и запушил на удаленный репо ветку **mybranch** (**git push -u origin mybranch**).



17) Убедился, что в github.com две ветки **master** и **mybranch**. 

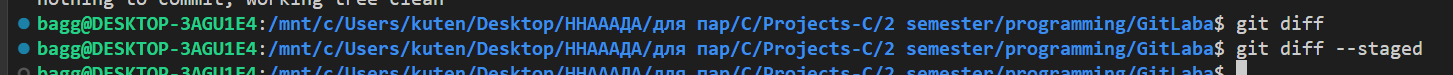
**Оценка 4:**

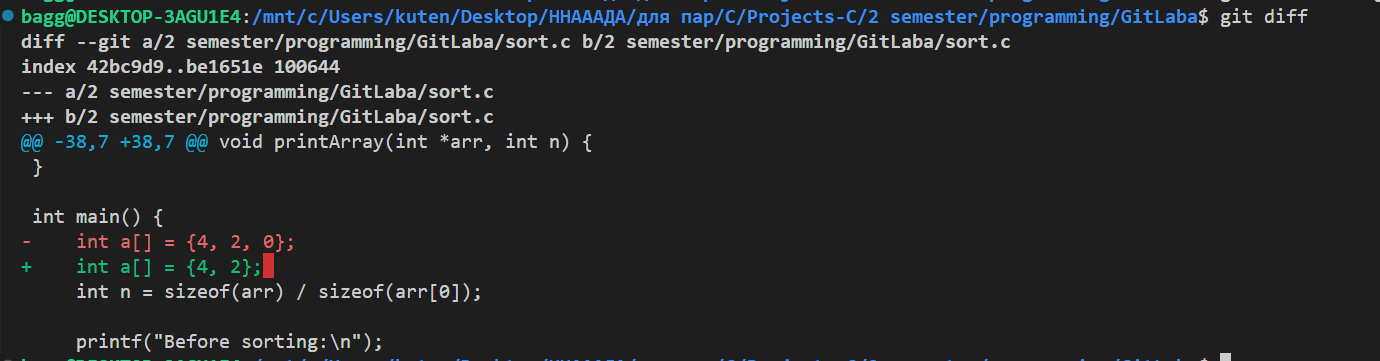
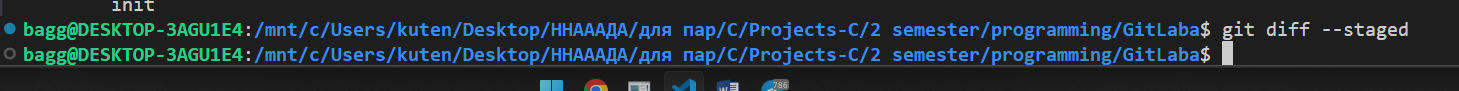
1,2) Переключился на ветку **mybranch**. Перезапиcал содержимое в **sort.c** добавив функцию **main(),** в которой будет объявлен массив из нескольких чисел и вызвана функция сортировки для этого массива.

3,4) **git diff** показывает изменения, внесенные в файл. git diff --staged пустой.

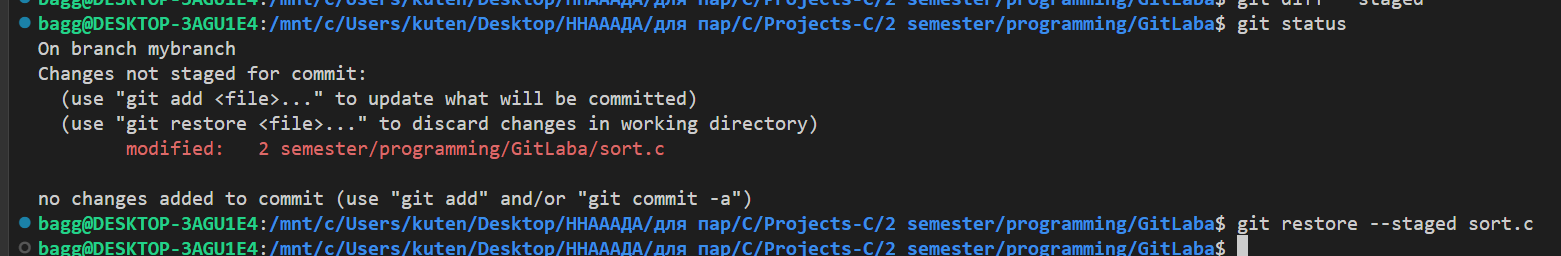
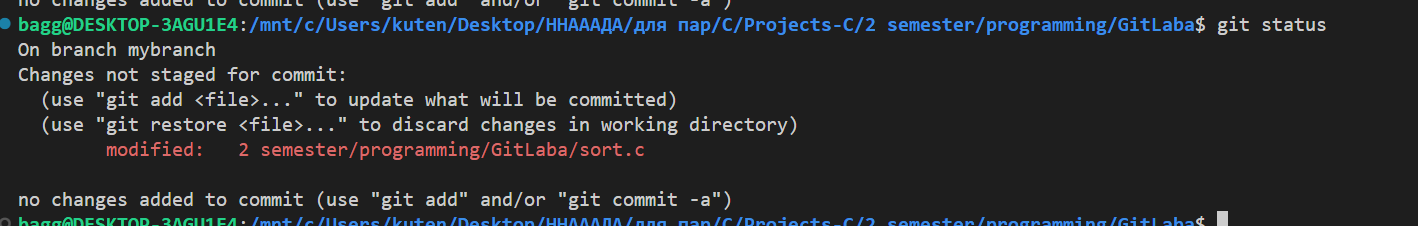


5,6,7) Добавил в **staged** файл **sort.c**. **git diff** и **git diff –staged** ничего не выводят.

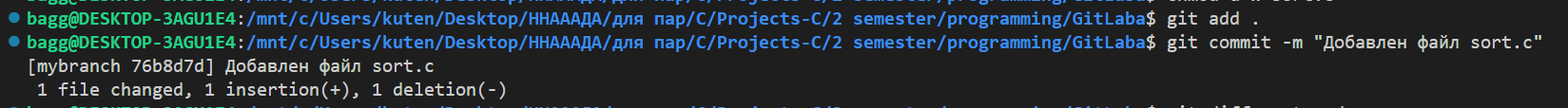
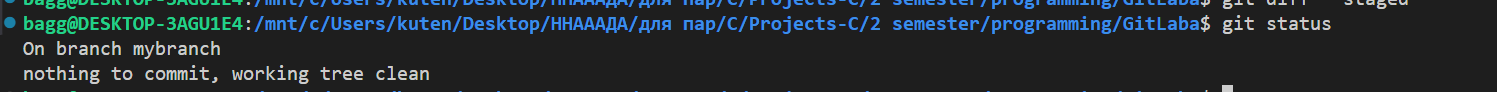


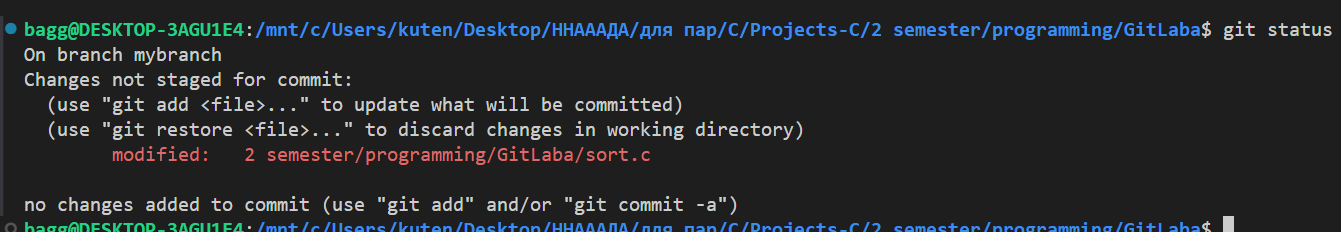
8,9,10,11) Удалил число из массива в sort.c. git diff показывает последнее изменение файла. Git diff –staged пуст.  

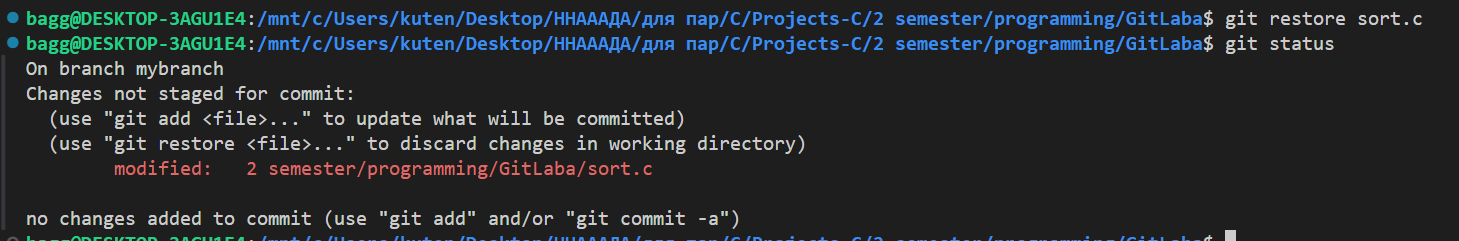
12,13,14) Запустил git status. Запустил git restore --staged sort.c. git status показывает тоже самое.

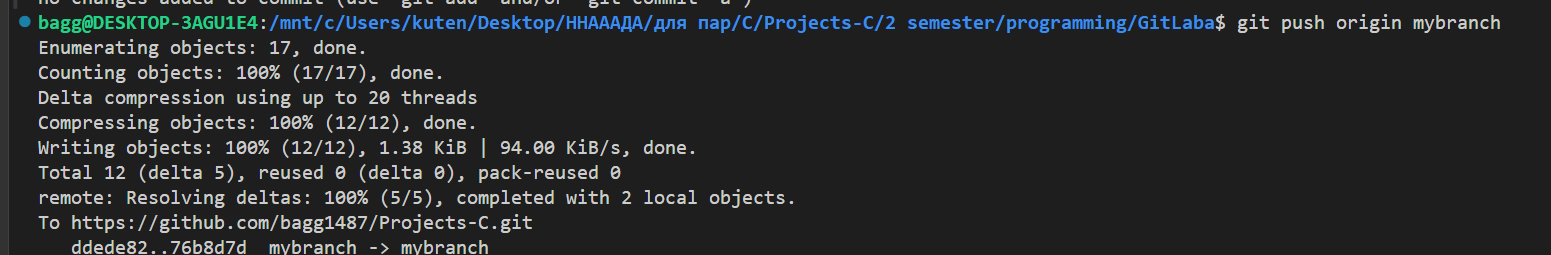
15,16)Сделал Add и коммит. Журнал.

17,18,19) Добавил в sort.c в main() printf(“hello git\n”);.sort.c содержит функцию сортировки и вывод ее работы на экран. Git status говорит нам следующее: 

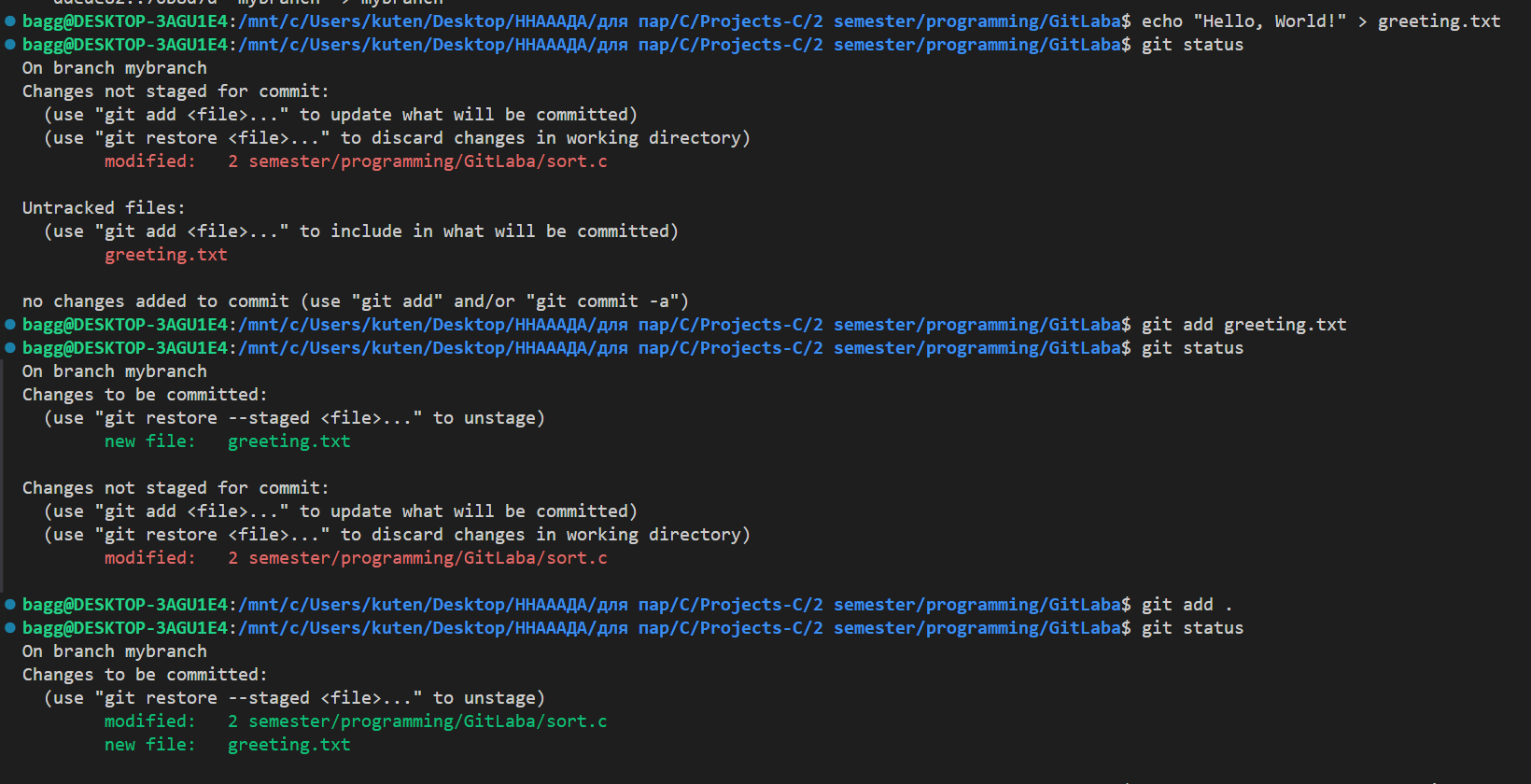
20,21,22) запустил git restore sort.c. содержимое и git status не изменились. 

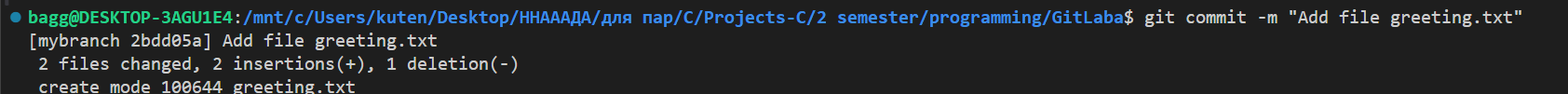
23)запушил ветку на реп.

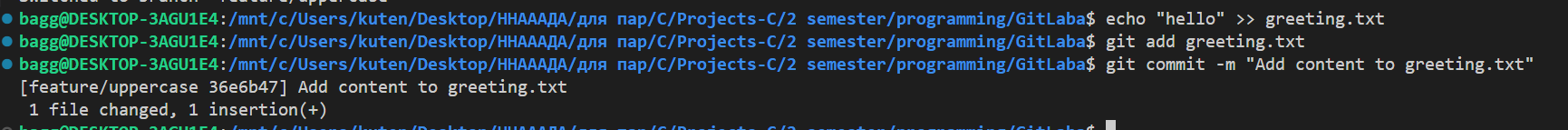


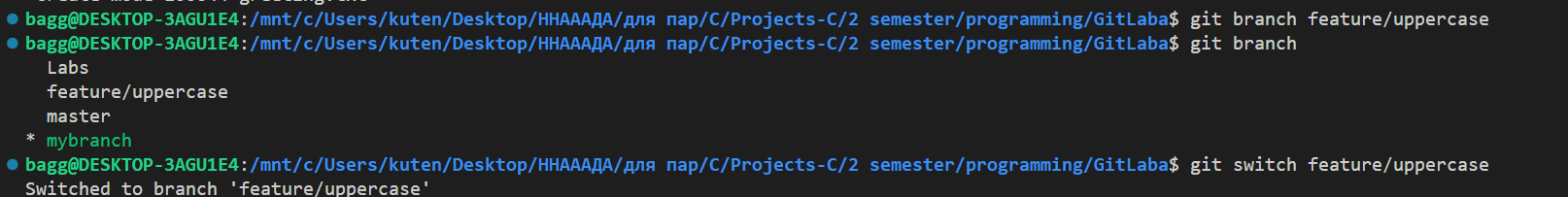
Ветки и ff merge

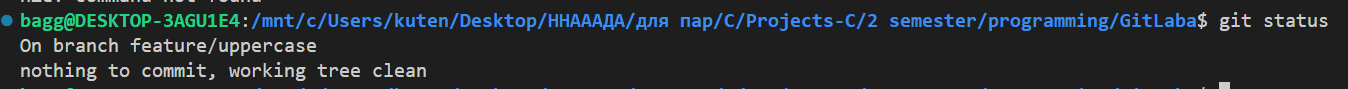
1. 1,2) Создал файл greeting.txt, проиндексировал его и закоммитил с сообщением “Add file greeting.txt”. Добавил в этот файл слово hello, индексирую и коммичу с текстом "Add content to greeting.txt"





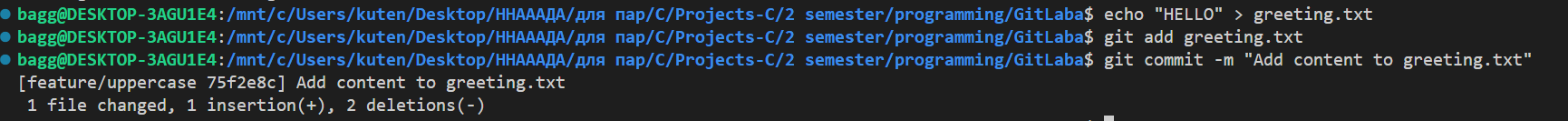
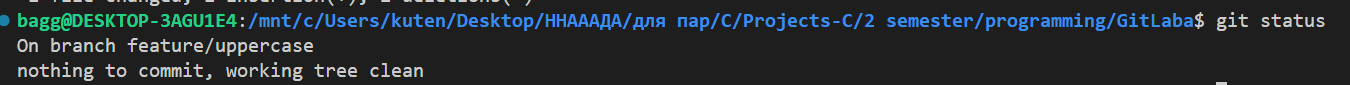


3,4,5) Создал ветку с именем feature/uppercase Переключился на эту ветку 

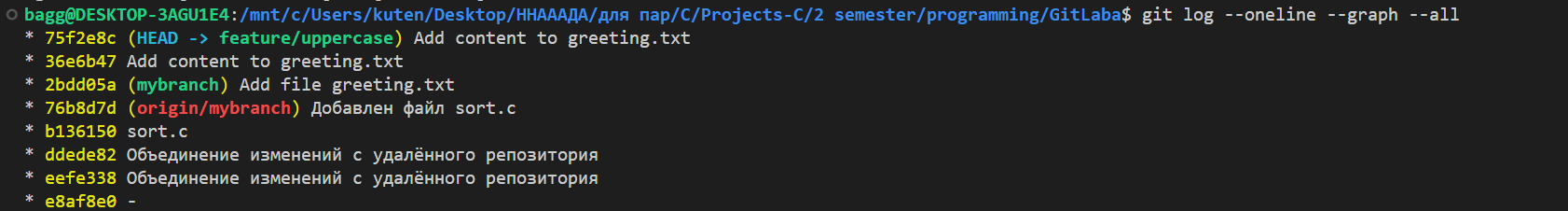


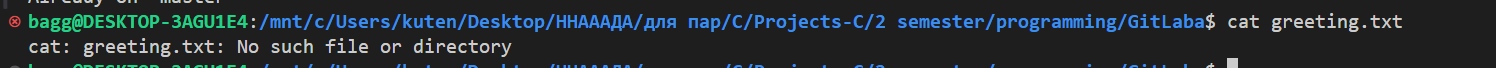
6,7,8) Отредактировал greeting.txt, чтобы он содержал приветствие в верхнем регистре

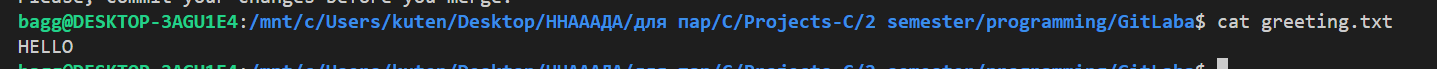
Добавил файл greeting.txt и закоммитил

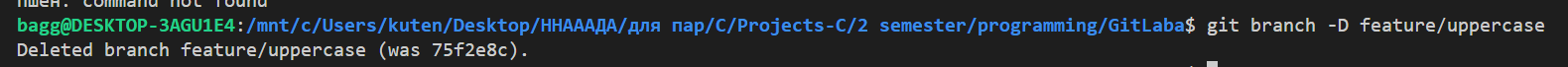
9) git log --oneline --graph –all

10,11) Переключил на главную веткуИспользовал cat, чтобы увидеть содержимое файла greetings.txt

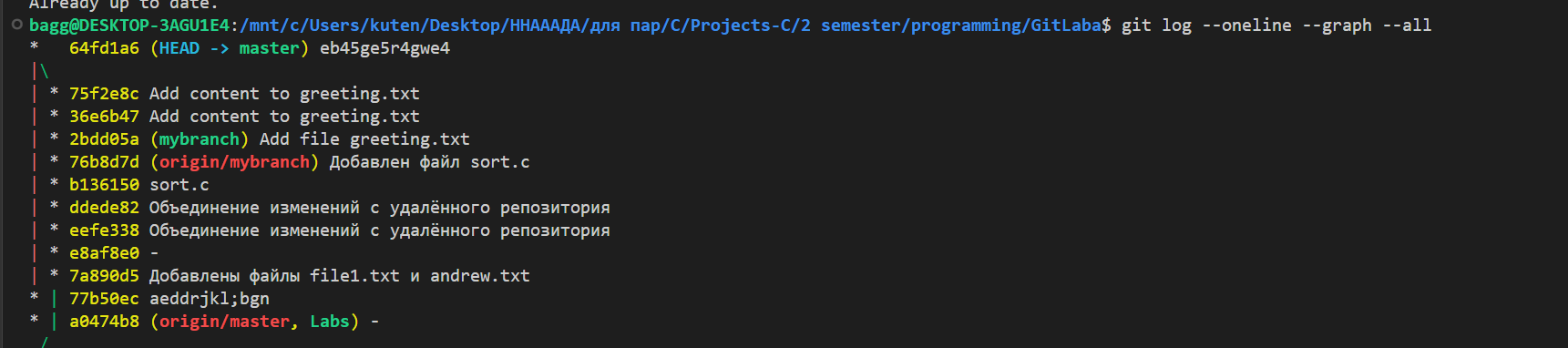


12,13,14) Объединил ветки и посмотрел содержимое файла  


15)удалил ветку feature/uppercase

16,17) Смержил ветку mybranch в master.git log --oneline --graph –all

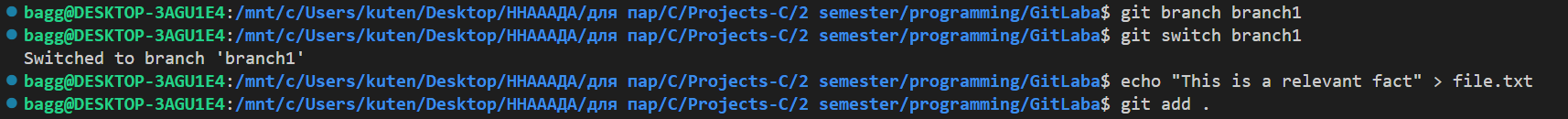
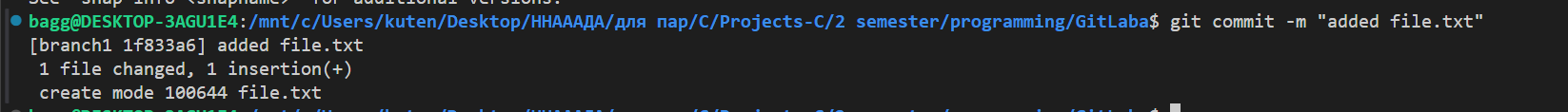




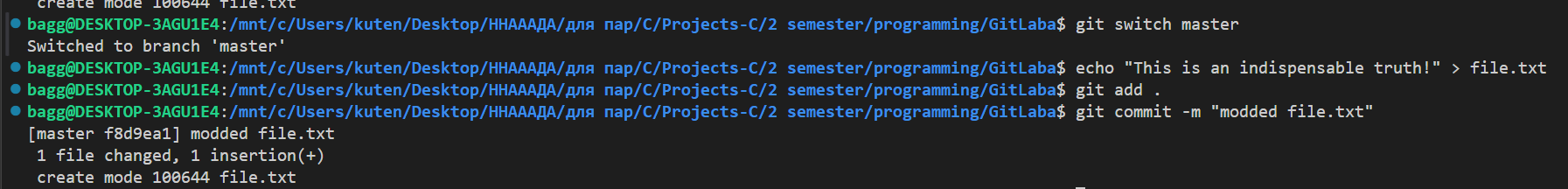
18,19) запушил документ и изменения.

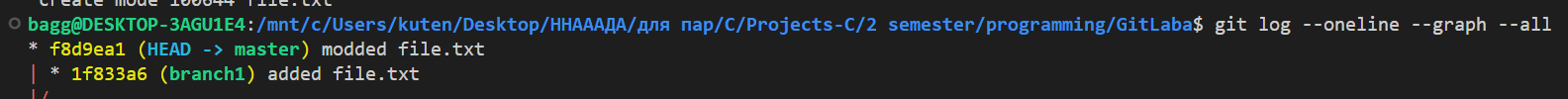
Оценка 5:

1,2,3) Создать ветку branch1, переключиться на нее Выполнить команду   
$ echo "This is a relevant fact" > file.txt Закоммитить это изменение

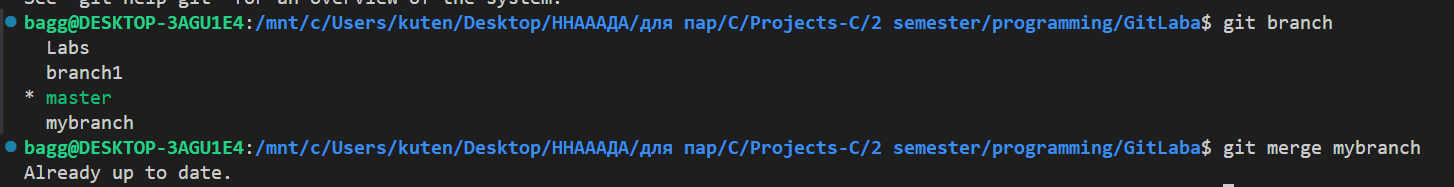
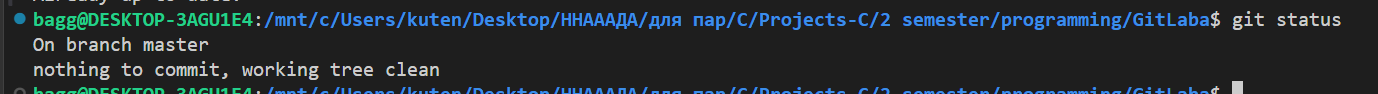
 

4,5,6) Переключиться на главную ветку и выполнить команду  
echo "This is an indispensable truth!" > file.txt Закоммитить изменения в master

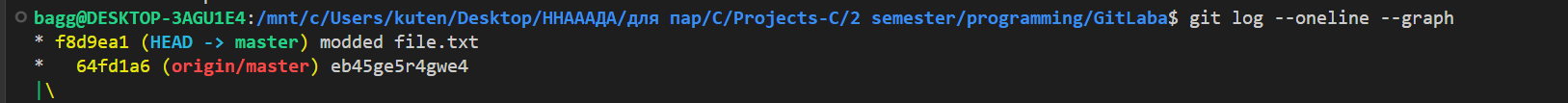




7,8) Использовать команду git merge чтобы смержить ветку branch1 в master

9) Посмотреть содержимое файла file.txt (никакого конфликта не было)

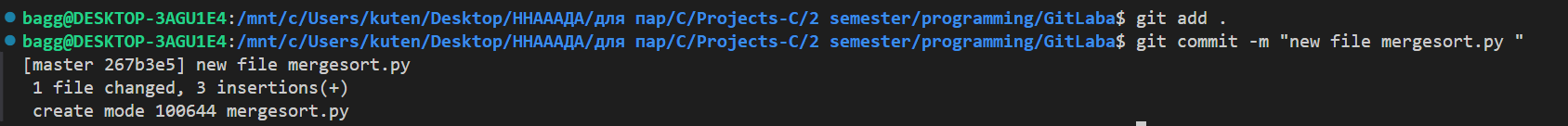
10) git log --oneline –graph 

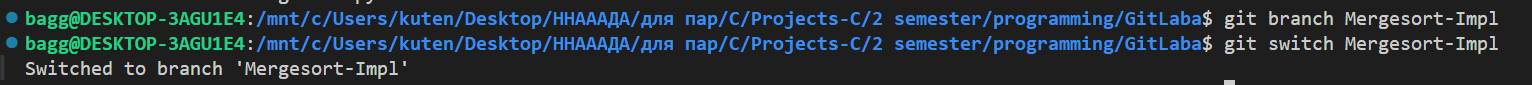
11)запушил изменения

Починим Merge конфликты для сортировки MergeSort на python.

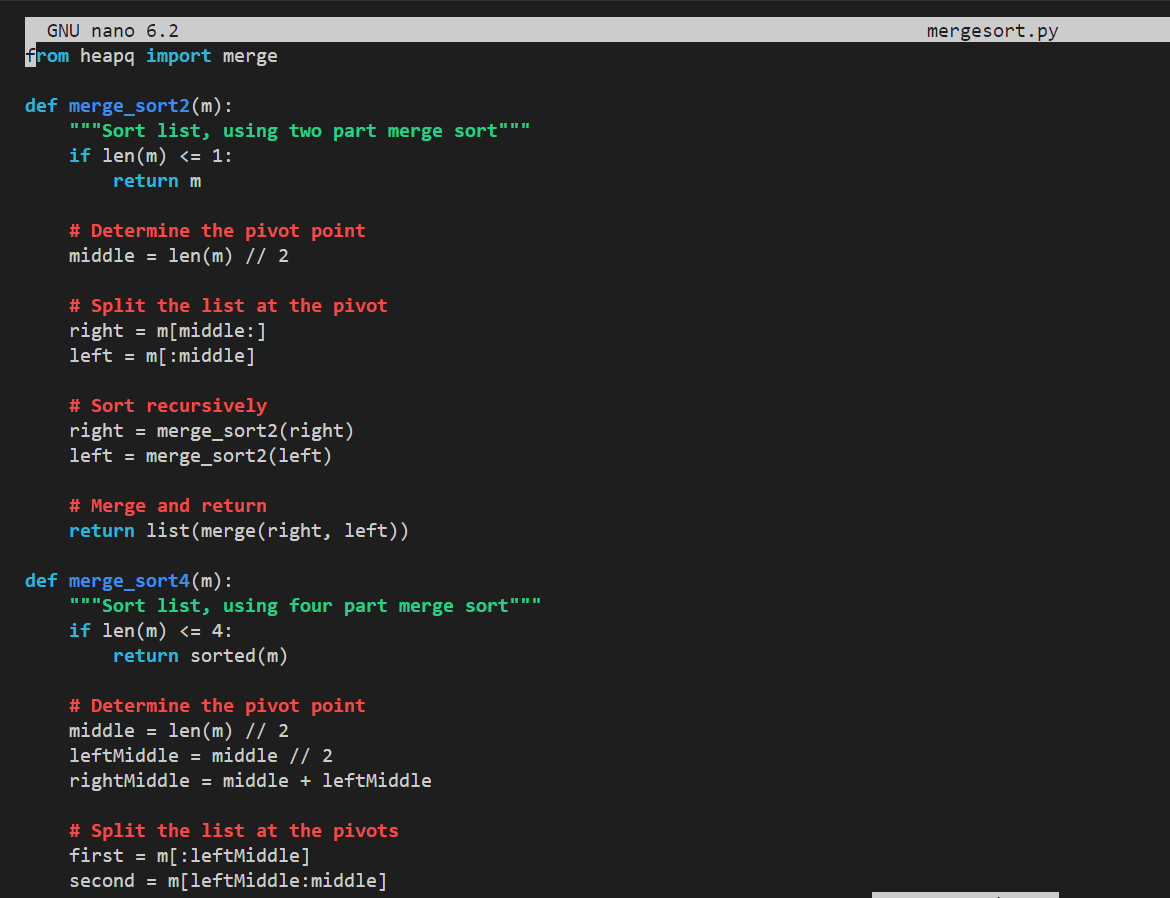
1. 1,2) Находясь в ветке master создадим файл mergesort.py с содержимым из **base.py** Проиндексируем файл и закоммитим

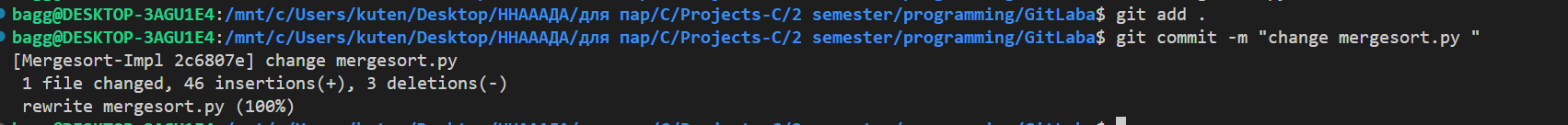




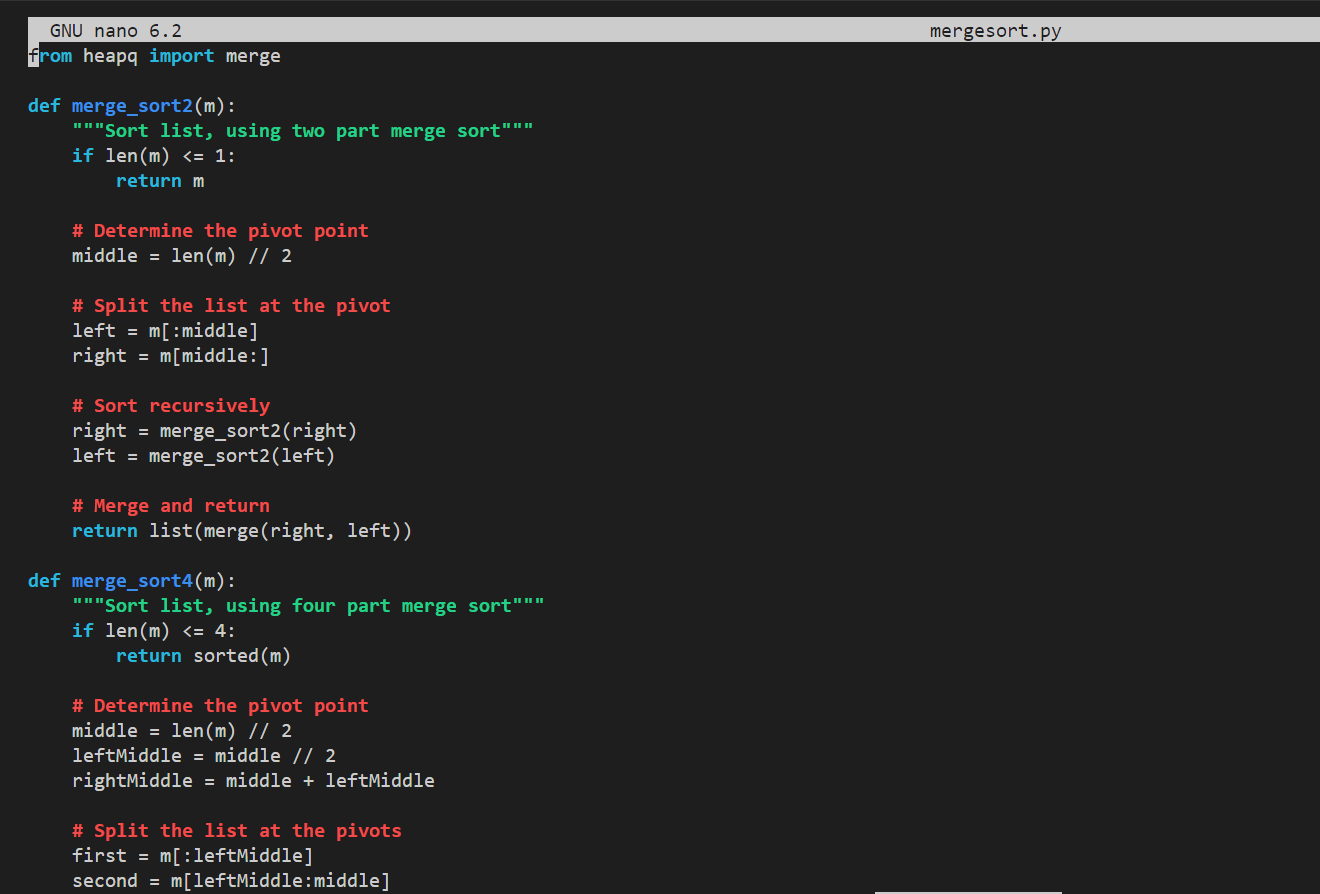
3) Создадим новую ветку Mergesort-Impl и переключимся на нее. 

4,5) Содержимое файла mergesort.py заменим на код из righty.py Коммитим изменения

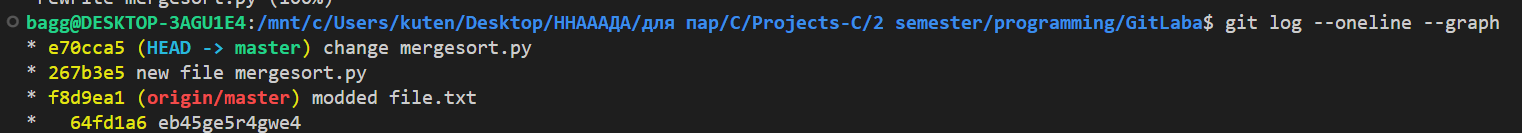




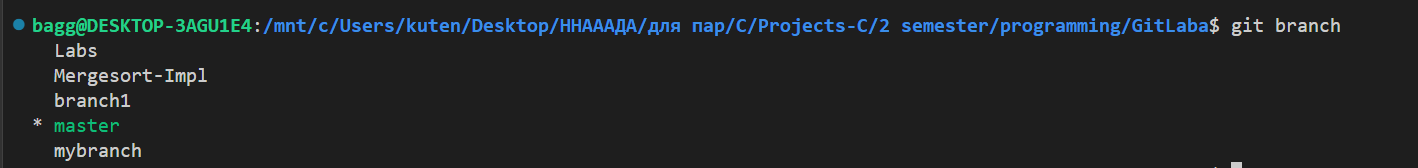
6,7) Переключаемся на master и меняем все содержимое mergesort.py на lefty.py Коммитим изменения.



8) git log --oneline –graph



9)git branch



10,11,12)смержил, исправил, запушил.

