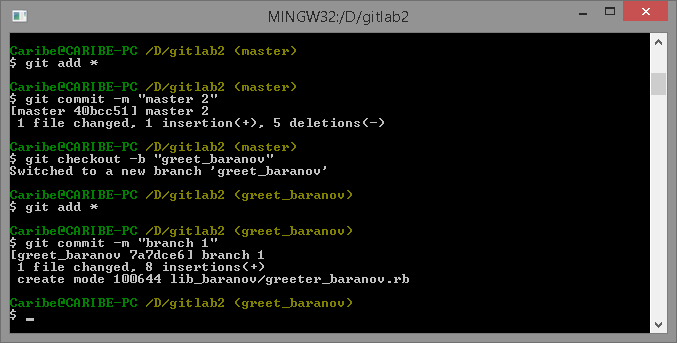
Part 6

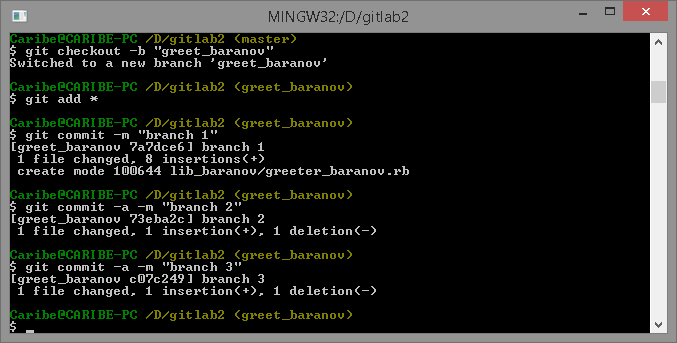
1. Создайте новую ветвь greet\_name (где name – Ваша фамилия), используя команду

git checkout -b <branchname>

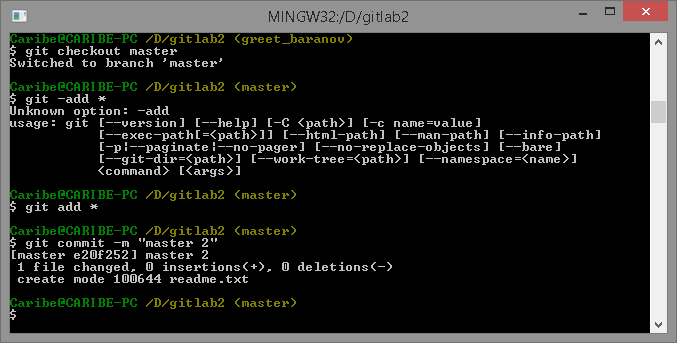
1. Добавьте новый каталог (пример lib\_name) и файл *greeter\_name.rb*

****

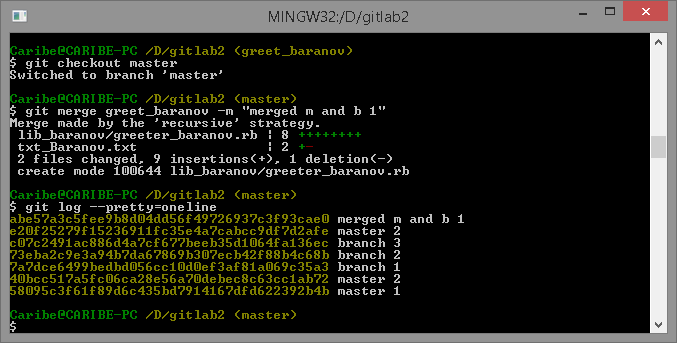
1. Сделайте 3 commits в ветку greet\_name



1. Продемонстрируйте переключение на ветку master.
2. Создайте файл README и произведите commit в master.



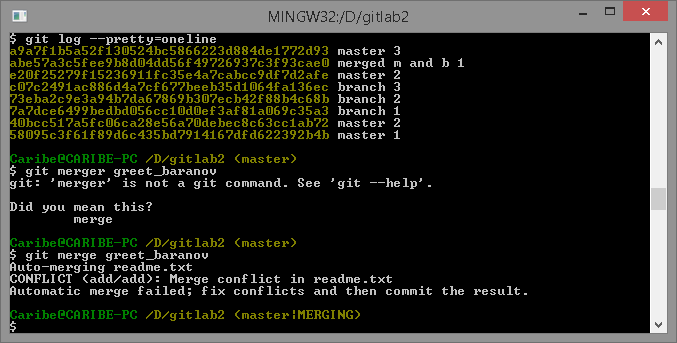
1. Произведите слияние ветвей greet и master – продемонстрируйте историю commits. (используйте опцию Fast forward).

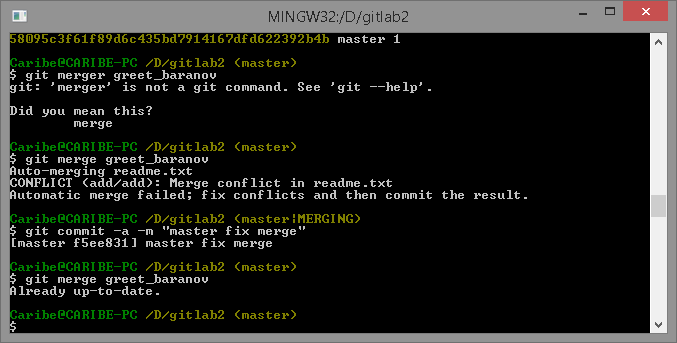


1. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов.

Часть 7.

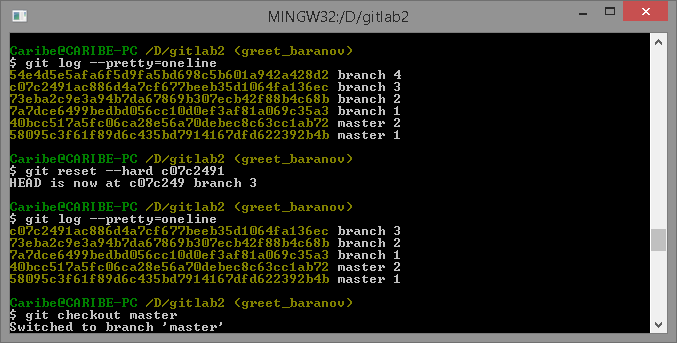
Смоделируйте искусственно конфликт, при слиянии описанном в предыдущей части.

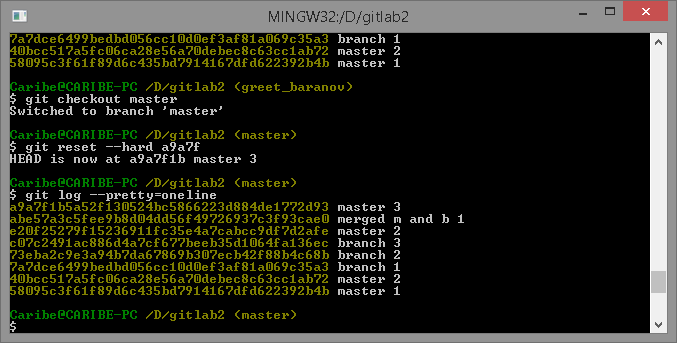




Часть 8.

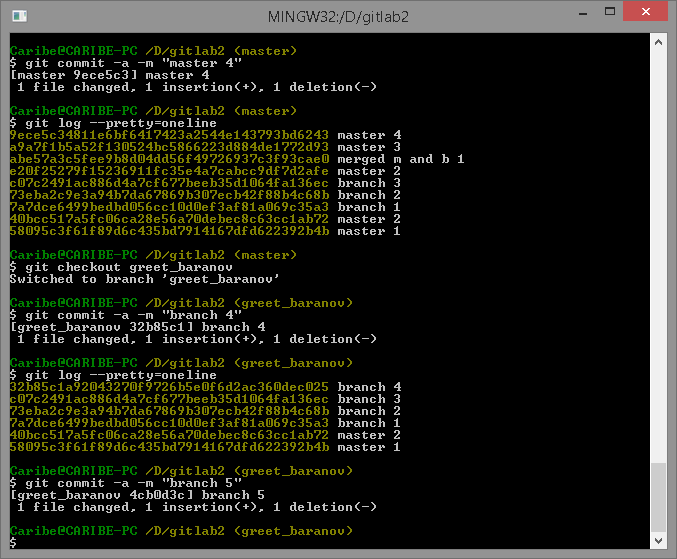
Используя команду reset верните указатель на состояние ветвей до их слияния.

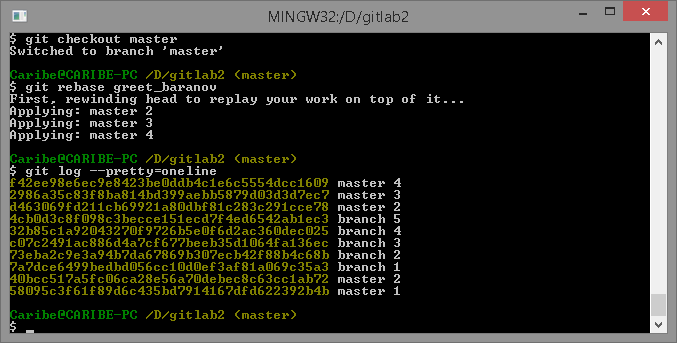




Часть 9. Проделайте шаги 3-5 в части 6. Выполните слияние изменений командой rebase. Изобразите в удобном для Вас графическом редакторе граф коммитов, где вершинами графа будут коды коммитов.

Объясните разницу между merge и rebase. На примере Ваших графов.





Merge: оставляет ответвления и почти не меняет структуру хранения данных.

Rebase: слияние, при котором 2 ветки преобразуются в одну, что более удобно для использования одним пользователем, но изменения в стуктуре негативно влияет на доступ к данным другим пользователям.

Часть 10.

1. Разбейтесь на пары.
2. Выберите в качестве общего репозитория один из двух удаленных.
3. Student1 заливает все содержимое своих папок в удаленный репозиторий.

Student 2 «забирает» проект к себе в локальный репозиторий.

1. Выполните команду git status.
2. Проведите каждый по три коммита в локальный репозиторий и «залейте изменения» в удаленный. (разрешается использование и merge и rebase).

