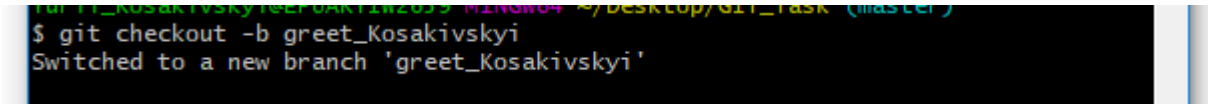
Основы работы с ветвями в GIT. Основы слияния. Разрешение конфликтов.

Часть 1. Создание ветвей и слияние.

1. Создайте новую ветвь greet_name (где name – Ваша фамилия), используя команду

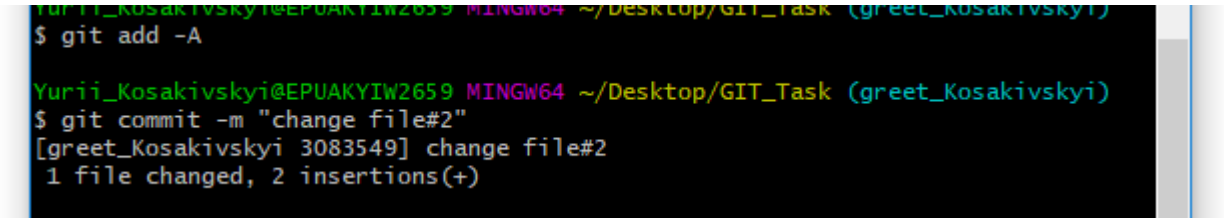
```
git checkout -b <branchname>
```



Добавьте новый каталог (пример lib_name) и файл *greeter_name.rb* с содержимым:

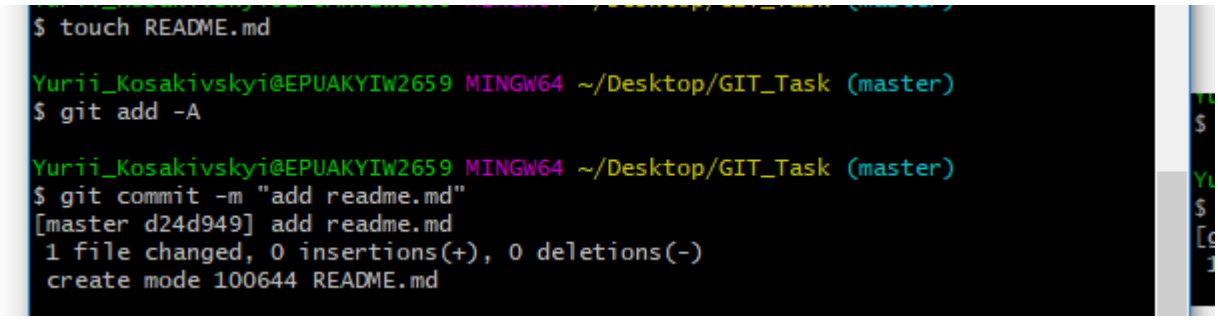
```
class Greeter
  def initialize(who)
    @who = who
  end
  def greet
    "Hello, #{@who}"
  end
end
```

2. Сделайте 3 commits в ветку greet_name



3. Продемонстрируйте переключение на ветку master.

4. Создайте файл README и произведите commit в master.



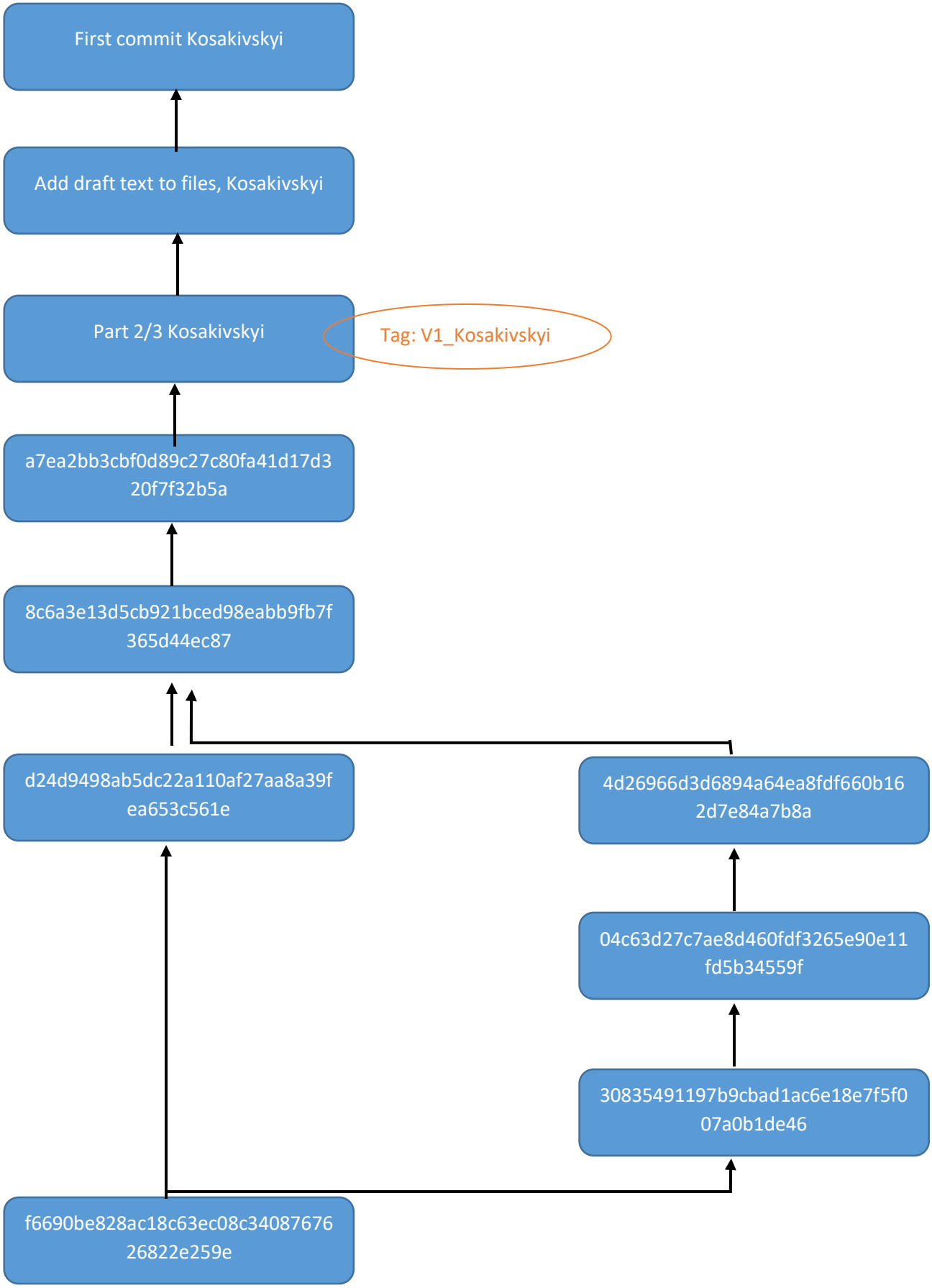
5. Произведите слияние ветвей greet и master – продемонстрируйте историю commits.

```
$ git merge greet_Kosakivskyi
Merge made by the 'recursive' strategy.
 lib_Kosakivskyi/greeter_Kosakivskyi.rb | 10 ++++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 lib_Kosakivskyi/greeter_Kosakivskyi.rb
```

```
commit f6690be828ac18c63ec08c3408767626822e259e (HEAD -> master)
Merge: d24d949 3083549
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 16:13:16 2019 +0300

    Merge branch 'greet_Kosakivskyi'

commit d24d9498ab5dc22a110af27aa8a39fea653c561e
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 16:12:41 2019 +0300

    add readme.md

commit 30835491197b9cbad1ac6e18e7f5f007a0b1de46 (greet_Kosakivskyi)
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 16:09:47 2019 +0300

    change file#2

commit 04c63d27c7ae8d460fdf3265e90e11fd5b34559f
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 16:09:27 2019 +0300

    change file

commit 4d26966d3d6894a64ea8fdf660b162d7e84a7b8a
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 16:04:00 2019 +0300

    first commit on new branch

commit 8f054fdf51708ba352d9ab4216cbde6a41242bae
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 15:53:11 2019 +0300

:
```

```
    Revert "Revert "Revert "part 5/5 Kosakivskyi""

    This reverts commit 326729fb096b68d936b04b0de36467a502eb00e6.

commit 326729fb096b68d936b04b0de36467a502eb00e6
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 15:52:13 2019 +0300

    Revert "Revert "part 5/5 Kosakivskyi""

    This reverts commit bdb58a1e55e361a00ac6bf3a7fde52a4ed3f72ff.

commit bdb58a1e55e361a00ac6bf3a7fde52a4ed3f72ff
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 15:51:03 2019 +0300

commit 8c6a3e13d5cb921bced98eabb9fb7f365d44ec87
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 15:50:54 2019 +0300

    part 5/5 Kosakivskyi

commit a7ea2bb3cbf0d89c27c80fa41d17d320f7f32b5a
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 15:33:57 2019 +0300

    2bd file changed

commit 7f7dd5b3dea2aea326665c2e73d4af227a3f5bc1
Author: Yurii_tmp <example@mail.com>
Date:   Fri May 24 15:33:30 2019 +0300
:
commit f6690be828ac18c63ec08c3408767626822e259e
```

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


First commit Kosakivskyi
Add draft text to files, Kosakivskyi
Part 2/3 Kosakivskyi
Tag: V1_Kosakivskyi
a7ea2bb3cbf0d89c27c80fa41d17d320f7f32b5a
8c6a3e13d5cb921bced98eabb9fb7f365d44ec87
d24d9498ab5dc22a110af27aa8a39fea653c561e
4d26966d3d6894a64ea8fdf660b162d7e84a7b8a
04c63d27c7ae8d460fdf3265e90e11fd5b34559f
30835491197b9cbad1ac6e18e7f5f007a0b1de46
f6690be828ac18c63ec08c3408767626822e259e

Часть 2.

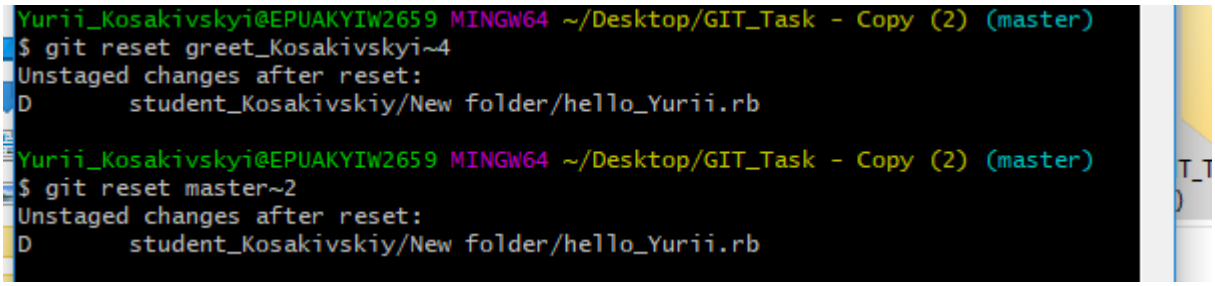
Смоделируйте искусственно конфликт, при слиянии описанном в предыдущей части.

Конфликт можно легко создать изменив один и тот же файл на разніх ветвях.

```
Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (master)
$ git checkout
greet_Kosakivskyi   master              V1_Kosakivskyi
HEAD                ORIG_HEAD

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (master)
$ git checkout greet_Kosakivskyi
Switched to branch 'greet_Kosakivskyi'

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (greet_Kosakivs
kyi)
$ git add -A

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (greet_Kosakivs
kyi)
$ git commit -m "ds"
[greet_Kosakivskyi 2be1929] ds
 1 file changed, 2 insertions(+), 1 deletion(-)

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (greet_Kosakivs
kyi)
$ git checkout master
Switched to branch 'master'

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (master)
$ git add -A

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (master)
$ git commit -m"sdf"
[master c0147a1] sdf
 1 file changed, 2 insertions(+)

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (master)
$ git merge greet_Kosakivskyi
Auto-merging student_Kosakivskiy/hello_Yurii2.rb
CONFLICT (content): Merge conflict in student_Kosakivskiy/hello_Yurii2.rb
Automatic merge failed; fix conflicts and then commit the result.

Yurii_Kosakivskyi@EPUAKYIW2659 MINGW64 ~/Desktop/GIT_Task - Copy (master|MERGING
)
$
```

Часть 3.

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



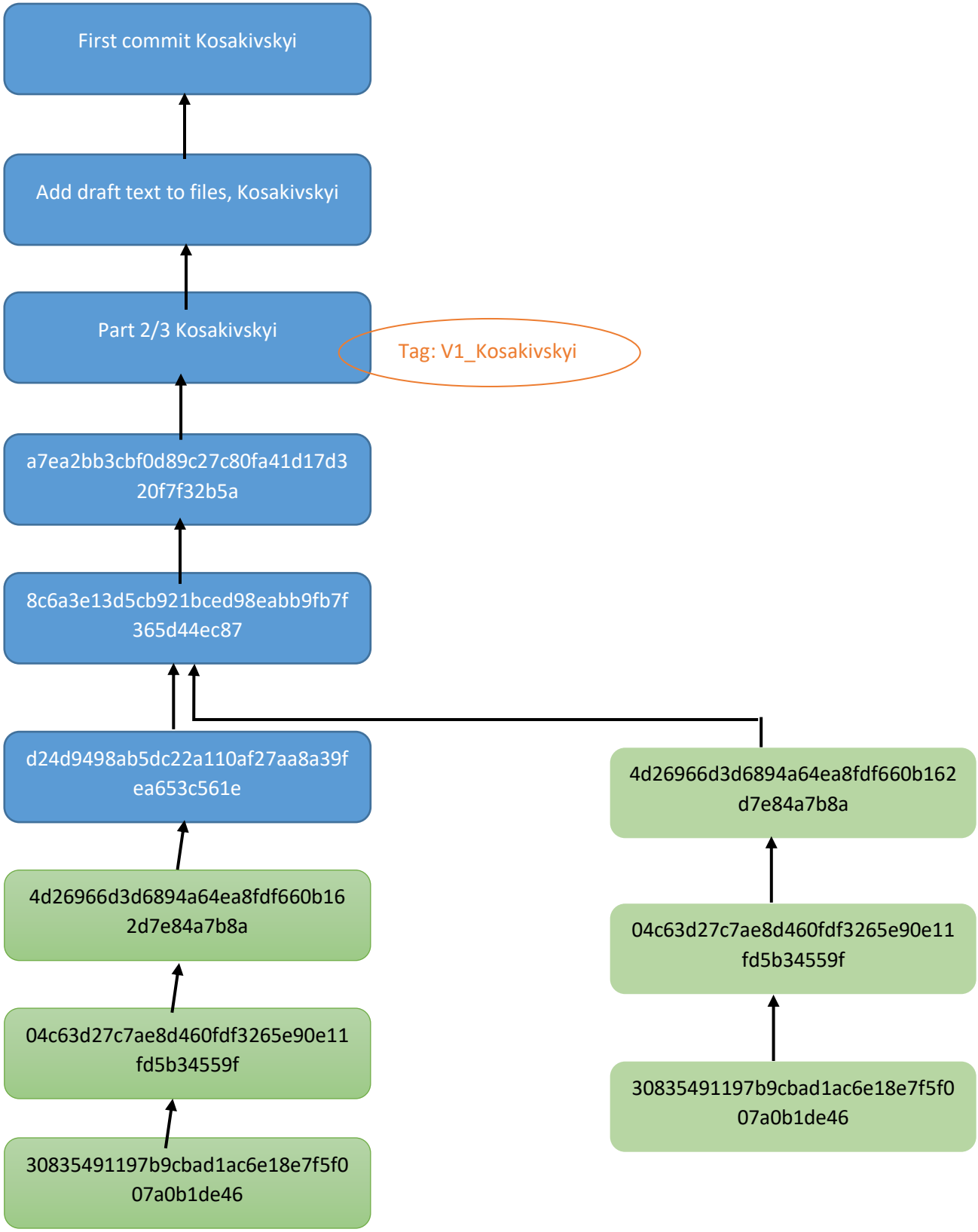
Часть 4.

Проделайте шаги 3-5 в части 1. 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.

График на следующей странице,

Насколько я понимаю после команды merge будет новый комит с изменениями с двух веток, и обе ветки будут ссылатся на етот комит. А rebase – копирует комиты с одной ветки и вставляет их на другую. Будет выглядить так как будто бы комиты были сделаны последовательно.


First commit Kosakivskyi
Add draft text to files, Kosakivskyi
Part 2/3 Kosakivskyi
Tag: V1_Kosakivskyi
a7ea2bb3cbf0d89c27c80fa41d17d320f7f32b5a
8c6a3e13d5cb921bced98eabb9fb7f365d44ec87
d24d9498ab5dc22a110af27aa8a39fea653c561e
4d26966d3d6894a64ea8fdf660b162d7e84a7b8a
04c63d27c7ae8d460fdf3265e90e11fd5b34559f
30835491197b9cbad1ac6e18e7f5f007a0b1de46
4d26966d3d6894a64ea8fdf660b162d7e84a7b8a
04c63d27c7ae8d460fdf3265e90e11fd5b34559f
30835491197b9cbad1ac6e18e7f5f007a0b1de46

Для начала работы создайте/зарегистрируйтесь (в случае отсутствия на текущий момент) на удаленном сервисе (типа *github*) *,* т.е. инициализируйте удаленный репозиторий.

Часть 5.

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.

4. Выполните команду git status.
5. Проведите каждый по три коммита в локальный репозиторий и «залейте изменения» в удаленный.