Учреждение образования «Полоцкий государственный университет»

Кафедра технологий программирования

Отчёт по курсу «Основы git» на stepik.org

Дисциплина «Программирование для интернет» (практическая часть курса)

Выполнил: 17-ИТ-4 Трубкин А.Д.

Проверил: преподаватель-стажер

Виноградова А.Д.,

старший преподаватель

Деканова М.В.

Полоцк, 2020

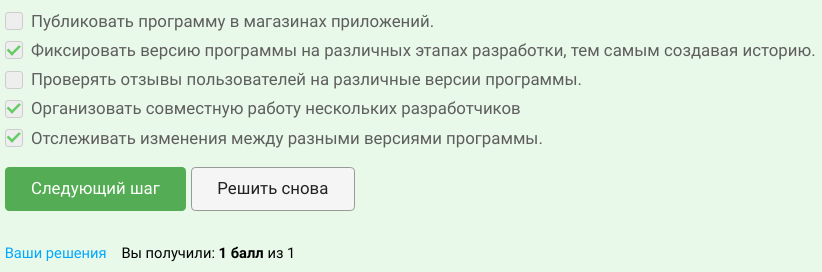
**Модуль 1 Введение**

**Раздел 1.1 Введение - Теория**

**Задача 1.**

Условие. Система контроля версий позволяет?

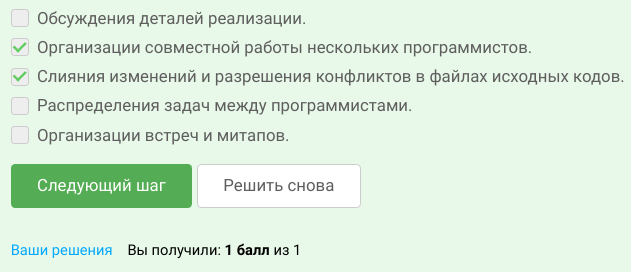
Результат:



**Задача 2.**

Условие. В случае, когда программистов несколько, система контроля версий выполняет функцию

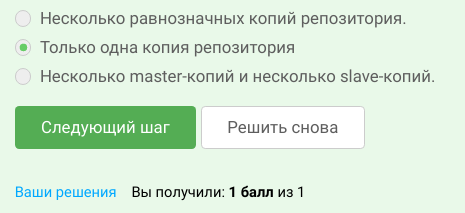
Результат:



**Задача 3.**

Условие. В централизованной системе контроля версий есть

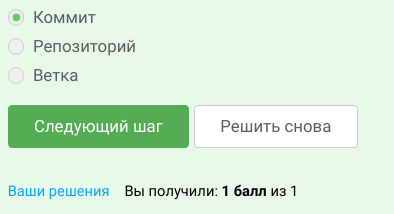
Результат:



**Задача 4.**

Условие. В Минимальной "единицей" в git (исходя из предыдущего видео) является

Результат:



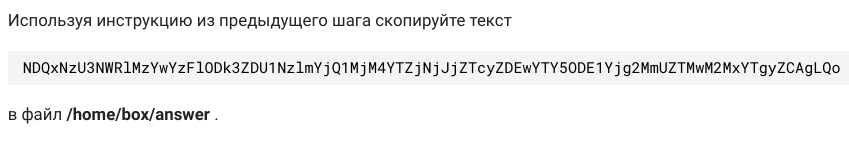
**Раздел 1.2 Как решать задачи Linux Challenge**

**Задача 1.**

Условие. Скопируйте содержимое файла /home/box/condition в /home/box/answer.

Результат: **cp /home/box/condition /home/box/answer.**

**Задача 2.**

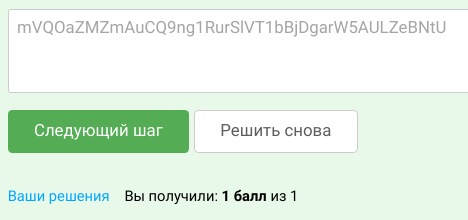
Условие. 

Результат:**echo "NDQxNzU3NWRlMzYwYzFlODk3ZDU1NzlmYjQ1MjM4YTZjNjJjZTcyZDEwYTY5ODE1Yjg2MmUZTMwM2MxYTgyZCAgLQo" >> answer**

**Задача 3.**

Условие.Скопируйте содержимое файла **/home/box/for\_next\_task**  из предыдущего задания в поле ниже..

Результат:



**Модуль 2 Использование существующих репозиториев**

**Раздел 2.1 Использование существующих репозиториев - Простые задачи**

**Задача 1.**

Условие. Склонируйте репозиторий по адресу ﻿﻿<https://github.com/OSLL/git_course_example_repo> в папку REPO, находящуюся в домашней директории текущего пользователя. В результате, содержимое репозитория должно находиться в REPO.

Результат: git clone ﻿<https://github.com/OSLL/git_course_example_repo>  REPO

**Задача 2.**

Условие. Репозиторий <https://github.com/OSLL/git_course_example_repo>  склонирован в папку $HOME/REPO . Настройте git так, чтобы имя git-пользователя было GitMegauser, а e-mail git\_super@example.com

Результат:

git config --global user.name "GitMegauser"

git config --global user.email git\_super@example.com

**Раздел 2.2 Использование существующих репозиториев - Сложные задачи**

**Задача 1.**

Условие. Репозиторий <https://github.com/OSLL/git_course_example_repo> склонирован в папку $HOME/REPO. Настройте его локальную конфигурацию так, чтобы можно было выполнять команду push на сервер, не вводя логин и пароль github аккаунта. Результат: Управляемая куча, Стек.

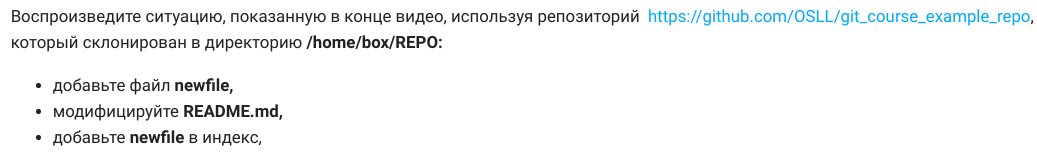
Результат: change the line "url" to git@github.com:OSLL/git\_course\_example\_repo.git

**Модуль 3 Обзор текущего состояние локального репозитория**

**Раздел 3.1 Обзор текущего состояние локального репозитория - Теория**

**Задача 1.**

Условие.



Результат:

cd REPO

touch newfile

vim README.md

git add newfile README.md

**Раздел 3.2 Обзор текущего состояние локального репозитория - Простые задачи**

**Задача 1.**

Условие. Выберите команды, с помощью которых можно найти коммит, содержащий некоторый искомый файл.

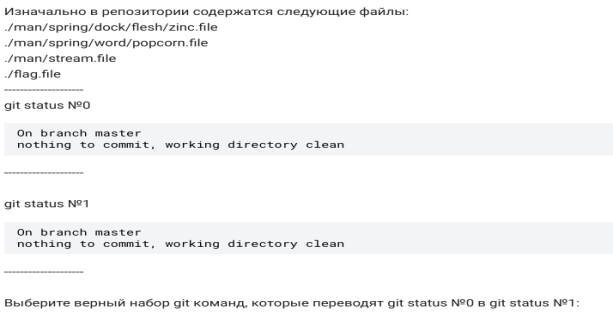
Результат:

git log –summary

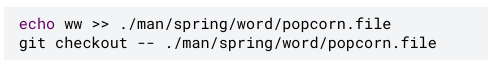
git log -- stat

**Задача 2.**

Условие.



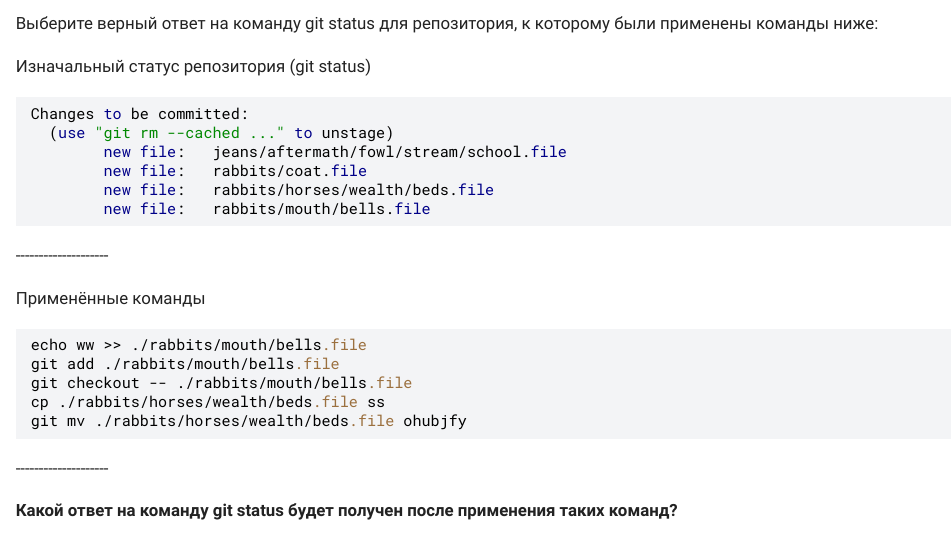
Результат:

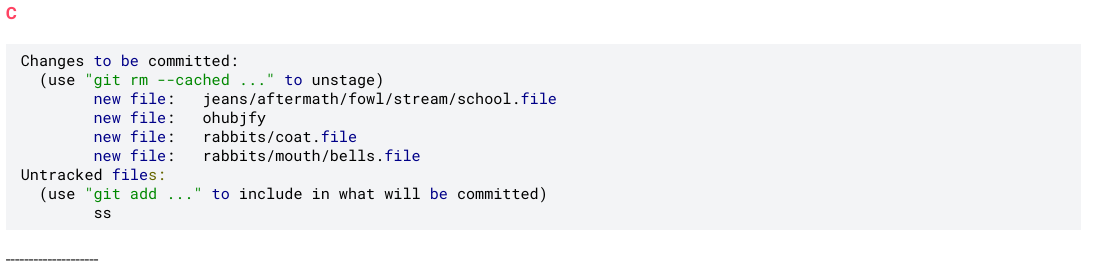


**Раздел 3.3 Обзор текущего состояние локального репозитория - Сложные задачи**

**Задача 1.**

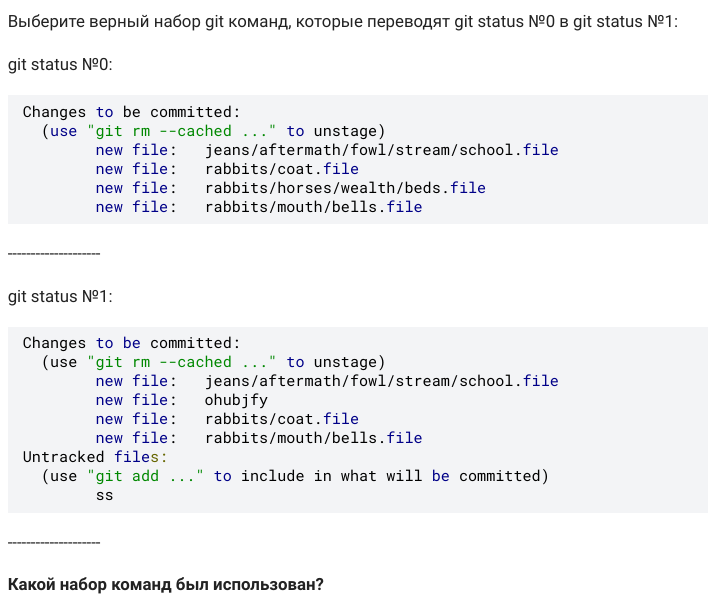
Условие.

Результат:

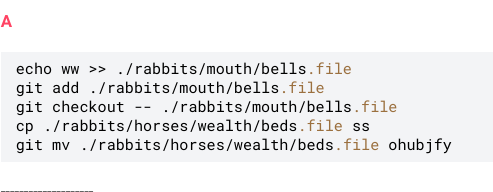


**Задача 2.**

Условие.

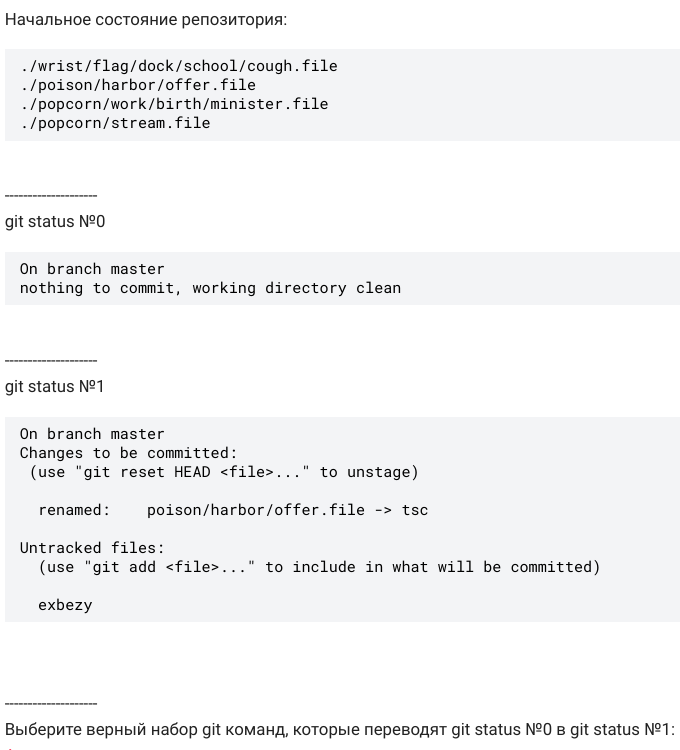


Результат:

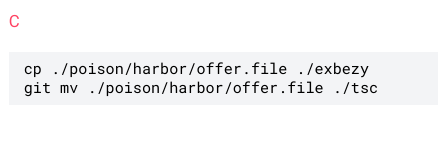


**Задача 3.**

Условие.



Результат:



**Модуль 4 Коммит изменений**

**Раздел 4.1 Коммит изменений - Теория**

**Задача 1.**

Условие. У вас есть склонированный локально репозиторий  <https://github.com/OSLL/git_course_example_repo> в ﻿/home/box/REPO. Сделайте коммит с сообщением "Git rules!"

Результат: git commit --amend -m "Git rules!"

**Раздел 4.2 Коммит изменений (индекс) - Простые задачи**

**Задача 1.**

Условие. Отметьте команды, которые создают хэш:

Результат: git commit

**Задача 2.**

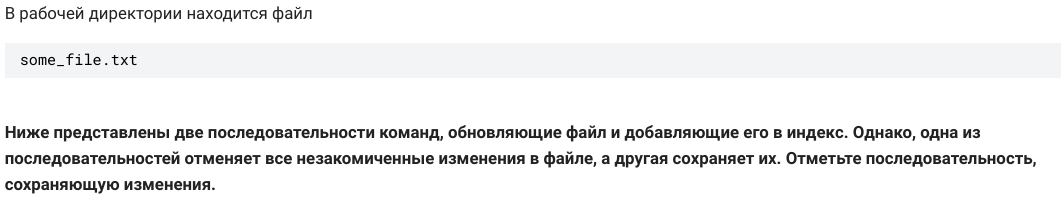
Условие. Отметьте команды, которые создают снимок файлов из индекса и обновляют состояние локального репозитория.

Результат: git commit

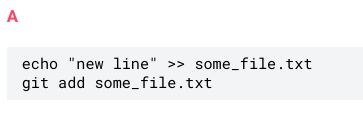
**Раздел 4.3 Коммит изменений (индекс) - Сложные задачи**

**Задача 1.**

Условие.

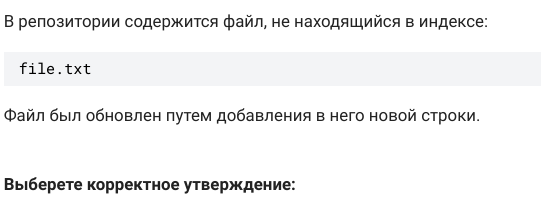


Результат:

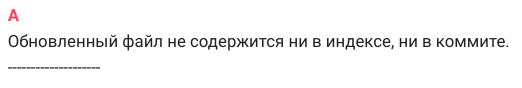


**Задача 2.**

Условие.



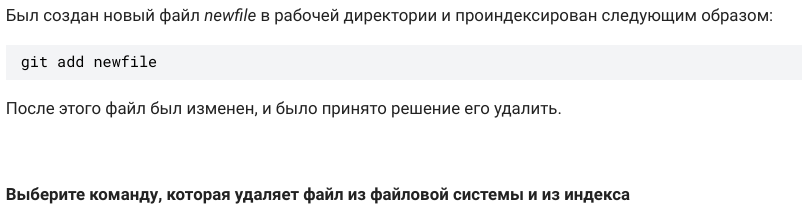
Результат:



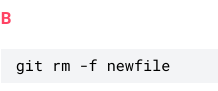
**Раздел 4.4 Коммит изменений - Простые задачи**

**Задача 1.**

Условие.

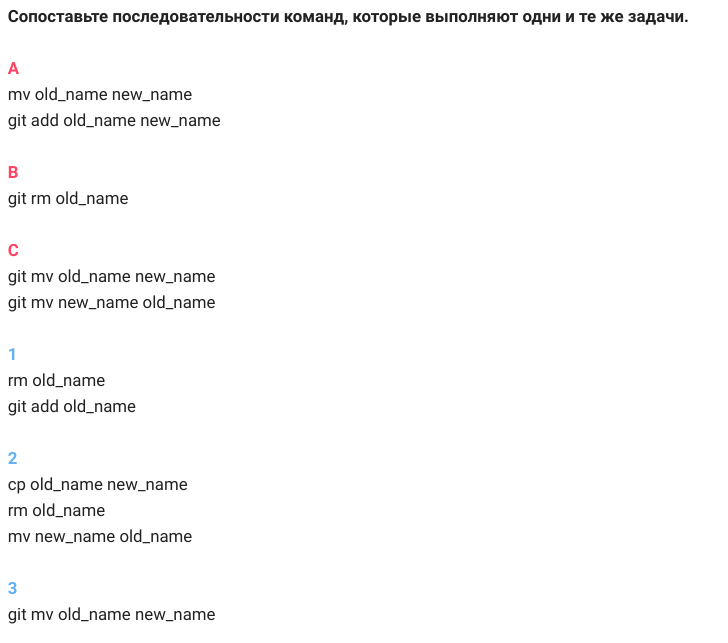


Результат:



**Задача 2.**

Условие.

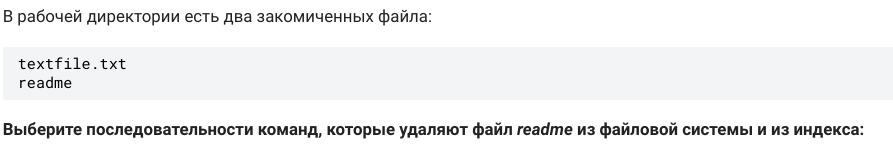


Результат: B1,C2,A3

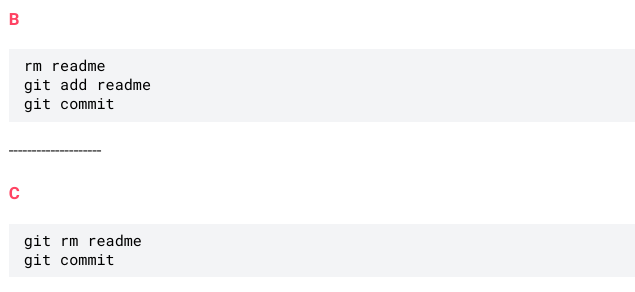
**Раздел 4.5 Коммит изменений - Сложные задачи**

**Задача 1.**

Условие.

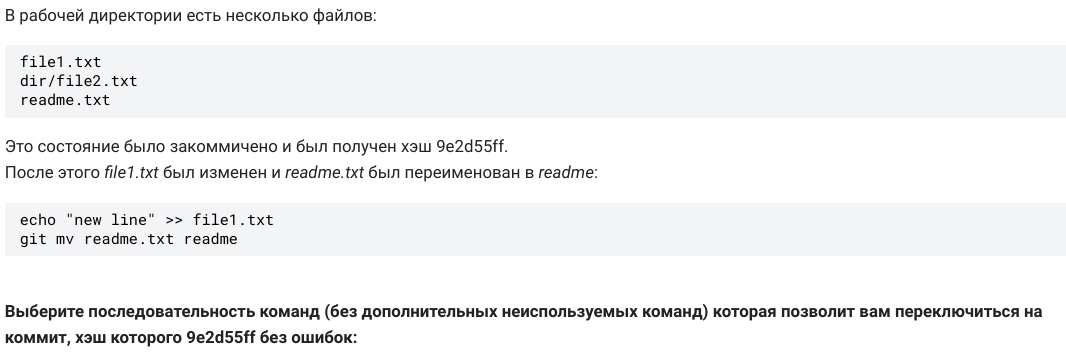


Результат:

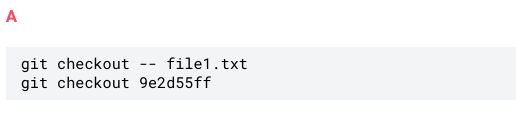


**Задача 2.**

Условие.



Результат:



**Модуль 5 Создание веток**

**Раздел 5.1 Создание веток - Теория**

**Задача 1.**

Условие. Вам дана виртуальная машина, где в директории /home/box/REPO ﻿размещена локальная копия репозитория ﻿  <https://github.com/OSLL/git_course_example_repo> . На основе ветки master создайте новую ветку под названием  "git\_rules"﻿.

Результат:

cd /home/box/REPO

git branch git\_rules

**Задача 2.**

Условие. Вам дана виртуальная машина, где в директории /home/box/REPO размещена локальная копия репозитория   <https://github.com/OSLL/git_course_example_repo> . ﻿Кто-то создал в репозитории очень много веток и одна из них содержит в названии 5555. Определите полное название этой ветки с помощью git branch и запишите его в файл /home/box/answer .

Результат:

git branch --list '\*5555\*' > /home/box/answer

**Раздел 5.2 Создание веток - Простые задачи**

**Задача 1.**

Условие. Выберите команды, которыми можно отобразить список веток удаленного репозитория.

Результат:

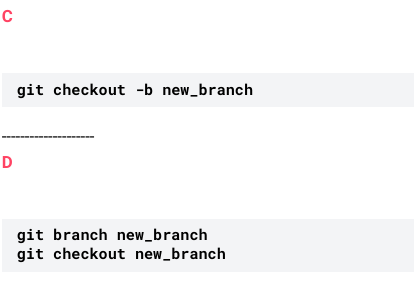
git branch --all

git branch -r

**Задача 2.**

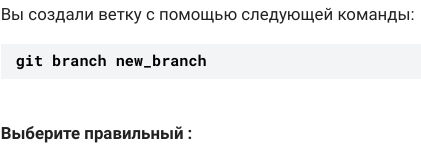
Условие. Выберите правильный набор команд (или отдельные команды), которые создают ветку и переключают рабочее дерево на нее. В данном задании несколько правильных вариантов ответа.

Результат:

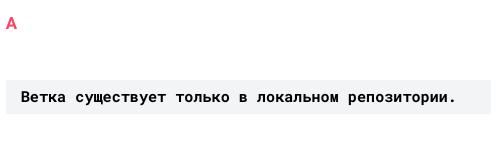


**Задача 3.**

Условие.



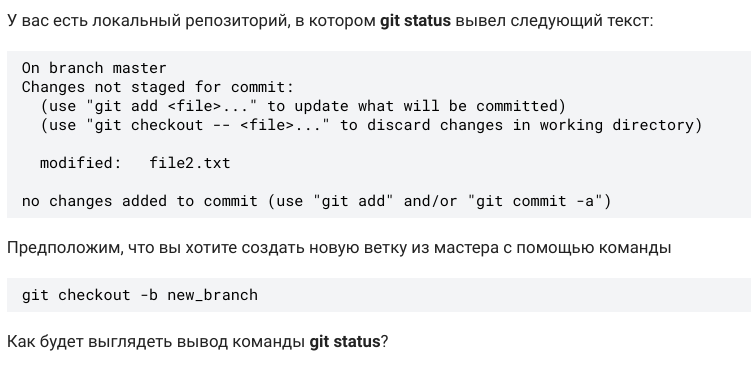
Результат:

****

**Раздел 5.3 Создание веток - Сложные задачи**

**Задача 1.**

Условие.



Результат:



**Модуль 6 Переход между коммитами**

**Раздел 6.1 Переход между коммитами - Теория**

**Задача 1.**

Условие. У вас есть локальный репозиторий  в /home/box/REPO. Найдите  в истории коммит с сообщением "Git rules!" и напишите его хэш (полностью) в файл /home/box/answer.

Результат: git log --grep="Git rules!" --pretty=format:%H > /home/box/answer

**Задача 2.**

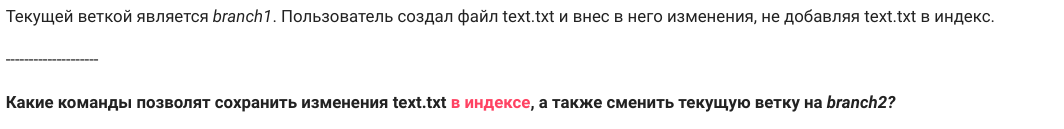
Условие. Аналогично предыдущему заданию, у вас есть локальный репозиторий  в /home/box/REPO. Требуется найти коммит, сообщение которого содержит "Git rules!",  и переключиться к этому коммиту.

Результат: git checkout $(git log --grep="Git rules!" --format=format:%H)

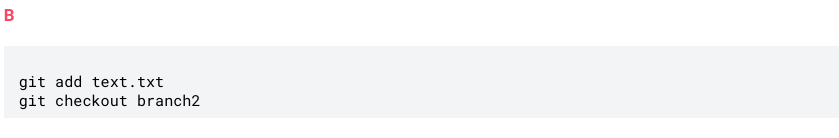
**Раздел 6.2 Переход между коммитами - Простые задачи**

**Задача 1.**

Условие.

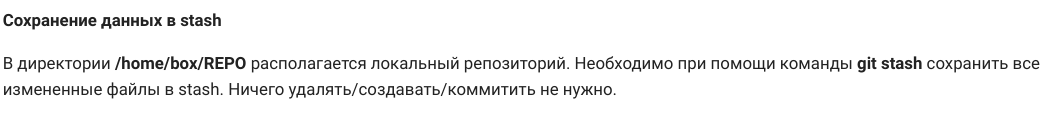


Результат:



**Задача 2.**

Условие.



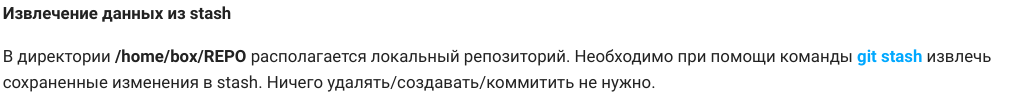
Результат:

cd REPO

git stash

**Задача 3.**

Условие.



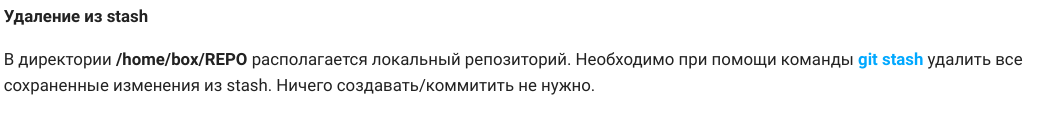
Результат:

cd REPO

git stash pop

**Задача 4.**

Условие.



Результат:

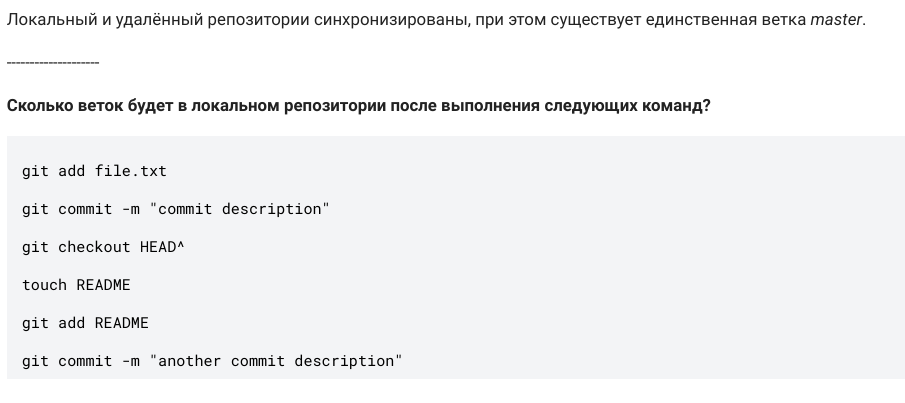
cd REPO

git stash clear

**Раздел 6.3 Переход между коммитами - Сложные задачи**

**Задача 1.**

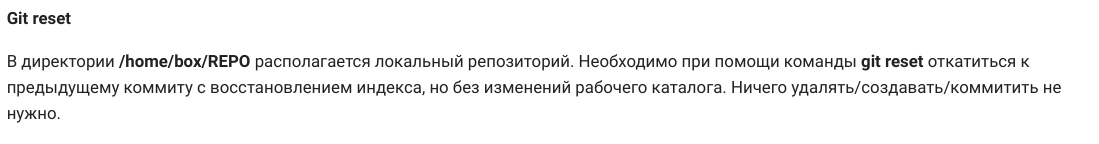
Условие.



Результат: 1

**Задача 2.**

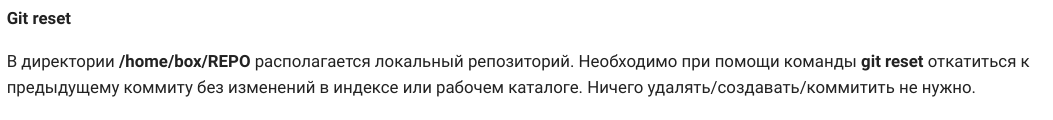
Условие.



Результат: git reset -- soft @~

**Задача 3.**

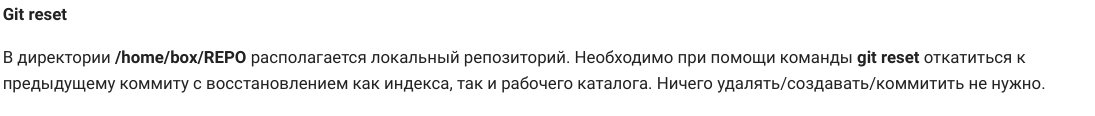
Условие.



Результат: git reset --soft HEAD~

**Задача 4.**

Условие.



Результат: git reset --hard HEAD~

**Модуль 7 git pull и git push**

**Раздел 7.1 git pull и git push - Простые задачи**

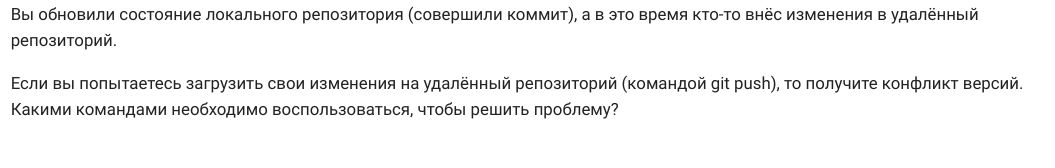
**Задача 1.**

Условие. Состояние локального и удалённого репозиториев различаются. Если необходимо скачать только новые файлы с удалённого репозитория, не обновляя существующие на локальном, то подходит команда:

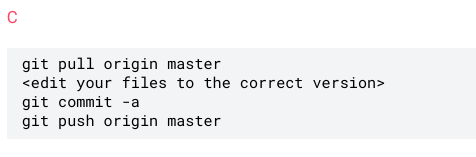
Результат: git fetch

**Задача 2.**

Условие.

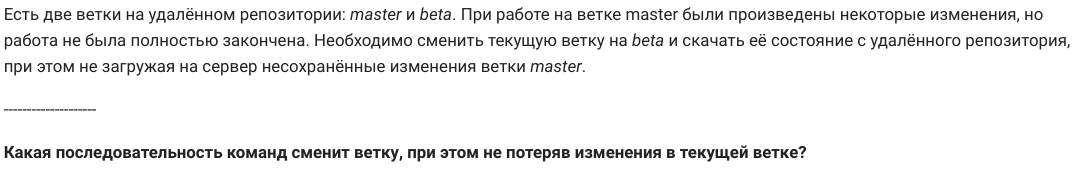


Результат:

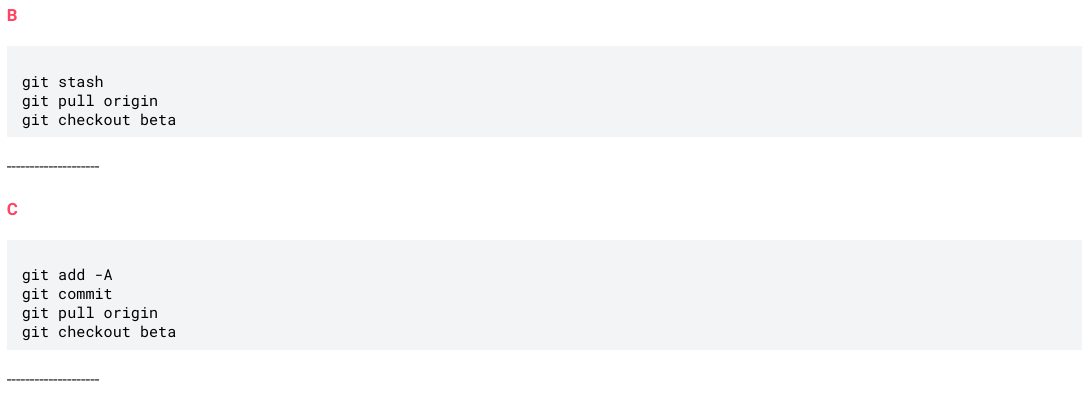


**Задача 3.**

Условие.



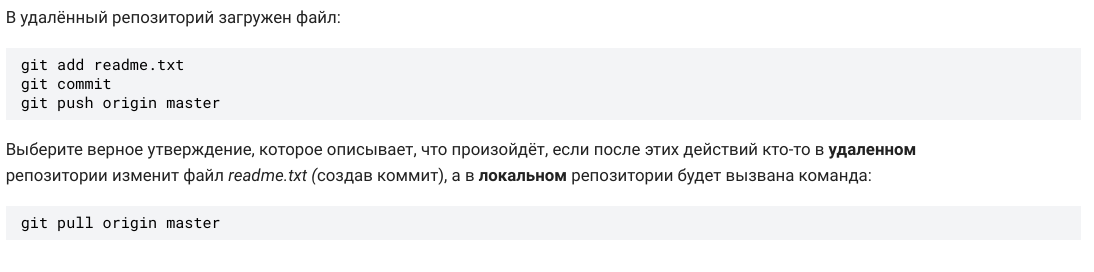
Результат:

****

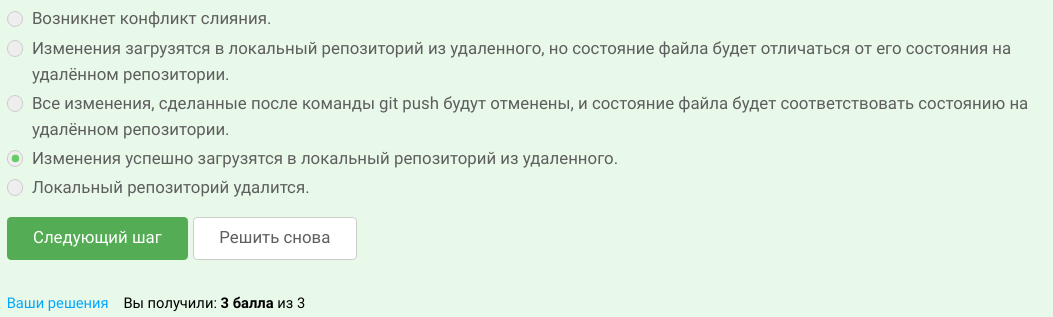
**Раздел 7.2 git pull и git push - Сложные задачи**

**Задача 1.**

Условие.

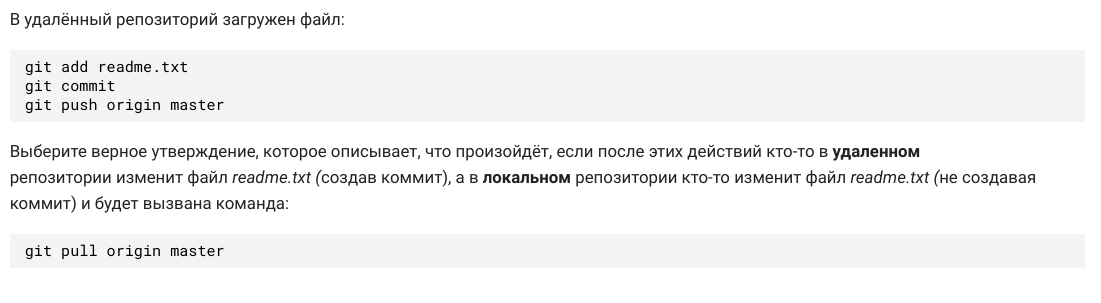


Результат:

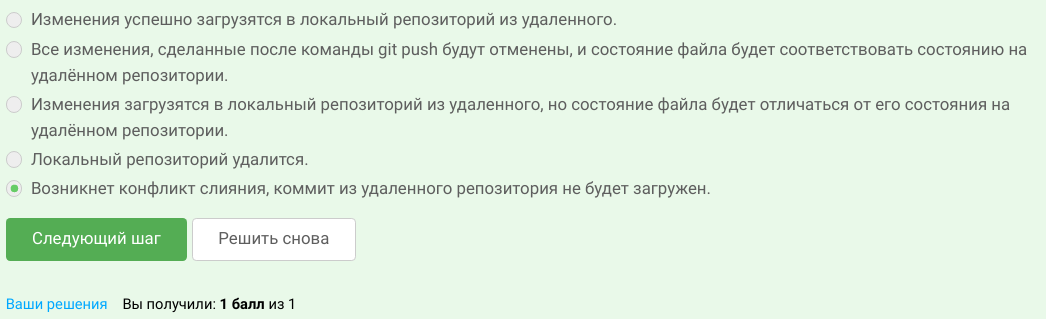


**Задача 2.**

Условие.



Результат:

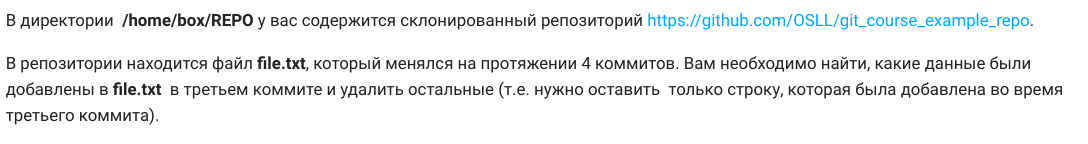
****

**Модуль 8 git diff**

**Раздел 8.1 git diff - Теория**

**Задача 1.**

Условие.

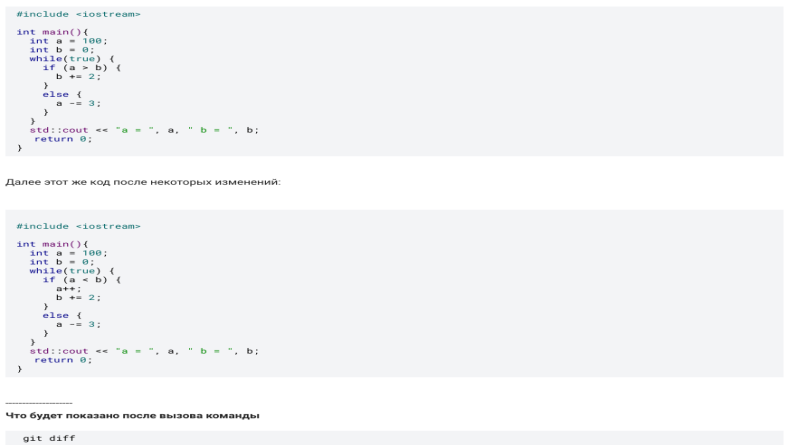


Результат: git diff HEAD^^ HEAD^ | grep -P '^\+\w+$' | sed 's/+//' > file.txt

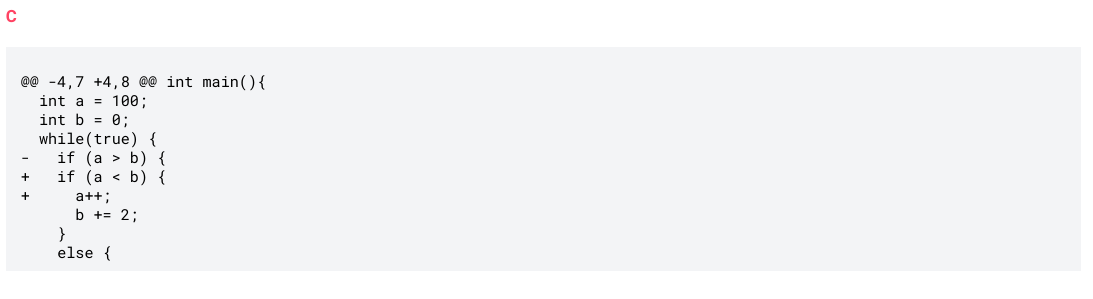
**Раздел 8.2 git diff - Простые задачи**

**Задача 1.**

Условие.



Результат:

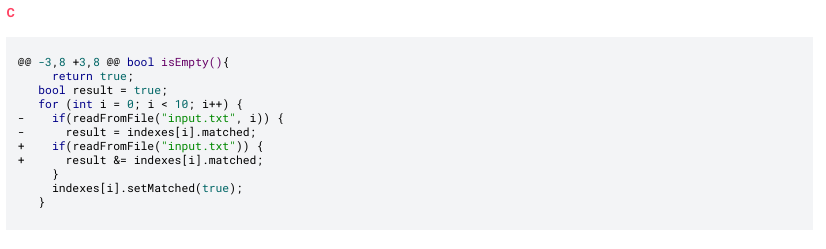


**Задача 2.**

Условие.



Результат:



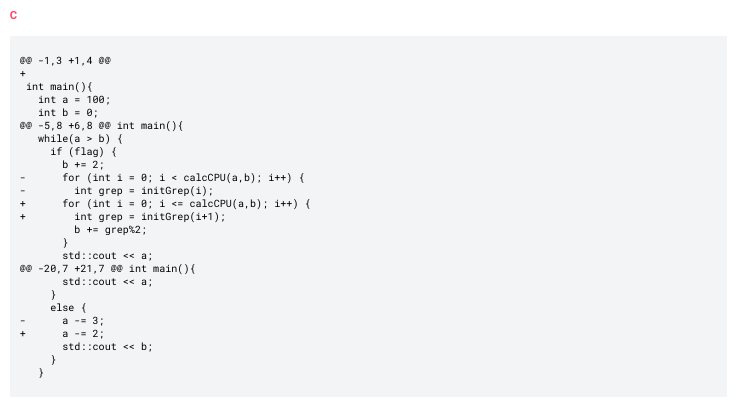
**Раздел 8.3 git diff - Сложные задачи**

**Задача 1.**

Условие.

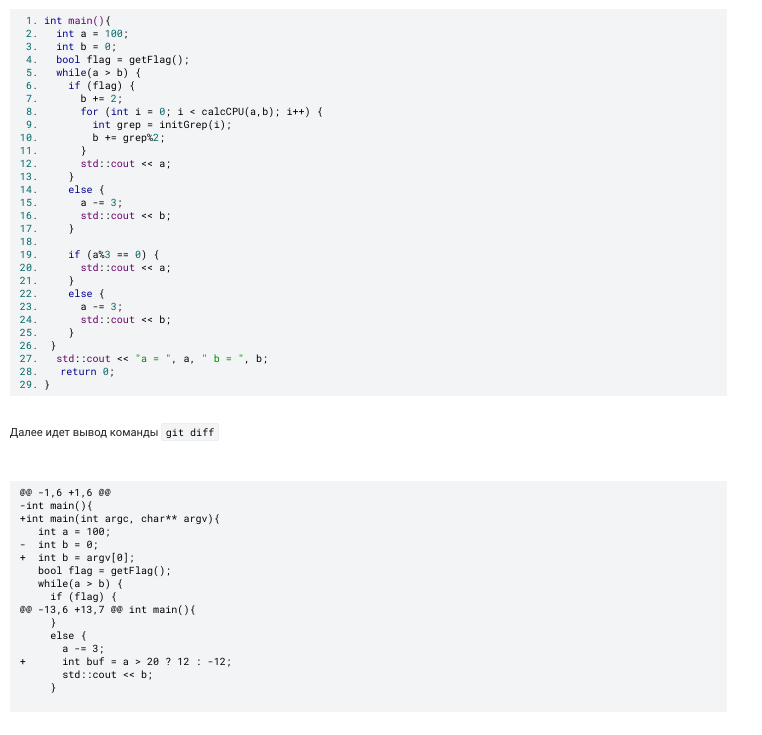


Результат:



**Задача 2.**

Условие.



Результат:

****