1. Используйте git status, чтобы узнать, на какой ветке вы находитесь.

```
lina@LaptopLii:/home/git$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
```

sosss

2. Как выглядит git log?

```
lina@LaptopLii:/home/git$ git log
commit cc3a32e9606338b332ee829ca4249c4b3f991eea (HEAD -> main)
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Dec 4 21:40:56 2025 +0700

    Replace git.c with labGit.c

commit 3a4a0421aae8278eccc03416bd648e6f56b45282 (origin/main)
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Nov 27 10:46:19 2025 +0700

:...skipping...
commit cc3a32e9606338b332ee829ca4249c4b3f991eea (HEAD -> main)
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Dec 4 21:40:56 2025 +0700

    Replace git.c with labGit.c
```

3. Создайте файл sort.c и вставьте туда код функции любой сортировки (только ф-ия сортировки)

```c
#include <stdio.h>
#include <stdlib.h>

int comp(const void *a, const void *b) {
    return (*(int *)a - *(int *)b);
}

int main() {
    int arr[] = {5, 2, 3, 1, 4};
    int n = sizeof(arr) / sizeof(arr[0]);

    // Sort the array arr
    qsort(arr, n, sizeof(arr[0]), comp);

    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

4. Как сейчас выглядит вывод git status?

```
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working direct
        deleted:    labGit.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        sort.c
```

5. Добавьте файл в область stage (add)

git add .

6. Как сейчас выглядит git status?

```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:    sort.c
        deleted:    labGit.c
```

7.Закоммитить файл в репозиторий

```
lina@LaptopLii:/home/git$ git commit -m "Комиит сортировки"
[main b711632] Комиит сортировки
 2 files changed, 18 insertions(+)
 create mode 100644  sort.c
 delete mode 100644 labGit.c
```

8. Как сейчас выглядит git status?

```
lina@LaptopLii:/home/git$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

9. Добавить комментарий с любым текстом в этот же файл

```
lina@LaptopLii:/home/git$ echo "// This is a bubble sort implementation" >> sort.c
lina@LaptopLii:/home/git$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        sort.c

nothing added to commit but untracked files present (use "git add" to track)
lina@LaptopLii:/home/git$
```

10. Как сейчас выглядит git status?

```
lina@LaptopLii:/home/git$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        sort.c

nothing added to commit but untracked files present (use "git add" to track)
```

11. Добавьте (add) изменение файла
git add .
12. Как сейчас выглядит git status?

```
lina@LaptopLii:/home/git$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   sort.c
```

13. Измените файл еще раз (можно добавить еще комментарий или убрать старый)
14. Сделайте коммит

```
lina@LaptopLii:/home/git$ git commit -m "еще повторный коммит"
[main 7700b45] еще повторный коммит
 1 file changed, 1 insertion(+)
 create mode 100644 sort.c
```

15. Как сейчас выглядит status? Журнал (log)?

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   sort.c

no changes added to commit (use "git add" and/or "git commit -a")
```

```
commit 7700b4591ca8da0512d0cba1dca0e07960747f2f (HEAD -> main)
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Dec 4 21:59:43 2025 +0700

    еще повторный коммит

commit b711632091dce70124628ea9477728785b3947d7
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Dec 4 21:50:13 2025 +0700

    Комиит сортировки

commit cc3a32e9606338b332ee829ca4249c4b3f991eea
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
```

16. Добавьте в stage и закоммитьте последнее изменение

```
lina@LaptopLii:/home/git$ git commit -m "коммит"
[main eb50d8a] коммит
 1 file changed, 1 insertion(+)
```

17. Запушим на удаленный репо (git push)

1. Переключитесь на ветку mybranch. В ней будет файл sort.c из предыдущих шагов с функцией сортировки

```
lina@LaptopLii:/home/git/lab9$ git checkout -b mybranch
\Switched to a new branch 'mybranch'
```

2. Перезапишите содержимое в sort.c добавив функцию main(), в которой будет объявлен массив из нескольких чисел (пример int a[] = {4, 2, 0};) и вызвана функция сортировки для этого массива.

3. Что вам говорит git diff?

```
lina@LaptopLii:/home/git/lab9$ git diff
diff --git a/lab9/ sort.c b/lab9/ sort.c
index ca04e9d..694fee2 100644
--- a/lab9/ sort.c
+++ b/lab9/ sort.c
@@ -1,18 +1,46 @@
 #include <stdio.h>
-#include <stdlib.h>

-int comp(const void *a, const void *b) {
-    return (*(int *)a - *(int *)b);
+// Быстрая сортировка
+void quickSort(int arr[], int low, int high) {
+    if (low < high) {
+        int pivot = arr[high];
:...skipping...
diff --git a/lab9/ sort.c b/lab9/ sort.c
index ca04e9d..694fee2 100644
--- a/lab9/ sort.c
+++ b/lab9/ sort.c
@@ -1,18 +1,46 @@
 #include <stdio.h>
-#include <stdlib.h>
```

4. Что вам говорит git diff --staged? Пустой?

5. Добавьте в staged файл sort.c
6. Что вам говорит git diff?
7. Что вам говорит git diff --staged?

```
lina@LaptopLii:/home/git/lab9$ git diff --staged
lina@LaptopLii:/home/git/lab9$ git add .
lina@LaptopLii:/home/git/lab9$ git diff
lina@LaptopLii:/home/git/lab9$ git diff --staged\
>
diff --git a/lab9/ sort.c b/lab9/ sort.c
index ca04e9d..694fee2 100644
--- a/lab9/ sort.c
+++ b/lab9/ sort.c
@@ -1,18 +1,46 @@
 #include <stdio.h>
-#include <stdlib.h>

-int comp(const void *a, const void *b) {
-    return (*(int *)a - *(int *)b);
+// Быстрая сортировка
+void quickSort(int arr[], int low, int high) {
+    if (low < high) {
+        int pivot = arr[high];
+        int i = (low - 1);
+
+        for (int j = low; j <= high - 1; j++) {
+            if (arr[j] < pivot) {
+                i++;
+                int temp = arr[i];
+                arr[i] = arr[j];
+                arr[j] = temp;
```

8. Удалите любое из чисел в массиве в sort.c:

9. Что вам говорит git diff?
10. Что вам говорит git diff --staged?

```
diff --git a/lab9/ sort.c b/lab9/ sort.c
index ca04e9d..694fee2 100644
--- a/lab9/ sort.c
+++ b/lab9/ sort.c
@@ -1,18 +1,46 @@
 #include <stdio.h>
-#include <stdlib.h>

-int comp(const void *a, const void *b) {
-    return (*(int *)a - *(int *)b);
+// Быстрая сортировка
+void quickSort(int arr[], int low, int high) {
+    if (low < high) {
+        int pivot = arr[high];
+        int i = (low - 1);
+
+        for (int j = low; j <= high - 1; j++) {
```

11. Объясните, что происходит
12. Запустите git status и обратите внимание, что sort.c присутствует дважды в выводе.

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:    sort.c
```

13. Запустите git restore --staged sort.c, чтобы отменить индексацию изменения
14.Что вам теперь говорит git status?

```
lina@LaptopLii:/home/git/lab9$ git restore --staged sort.c
lina@LaptopLii:/home/git/lab9$ git status
On branch mybranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:    sort.c

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:    sort.c
        deleted:    sort.c
```

14. Индексируйте изменение (add) и сделайте коммит

```
lina@LaptopLii:/home/git/lab9$ git add sort.c
lina@LaptopLii:/home/git/lab9$ git commit -m "коммит после изменений"
[mybranch 4ea3f4a] коммит после изменений
 2 files changed, 39 insertions(+), 13 deletions(-)
 delete mode 100644 lab9/sort.c
lina@LaptopLii:/home/git/lab9$
```

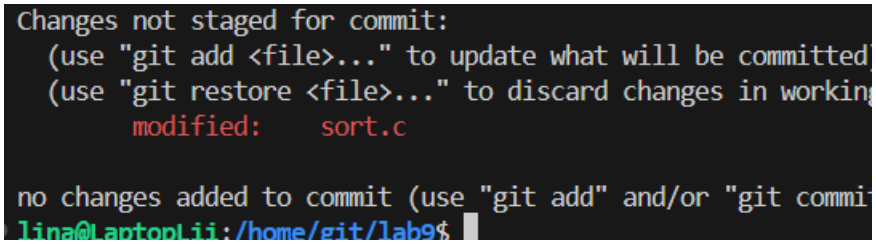
15. Как выглядит журнал?

```
commit 4ea3f4ae80e6b25a3ef7d9c6b269d666eb7d4939 (HEAD -> mybranch)
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Dec 4 22:56:02 2025 +0700

    коммит после изменений

commit b91ea8e7e4f976130b64035bf6ba30cb918226a6 (origin/main, main)
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Dec 4 22:13:20 2025 +0700

    папка с файлами лаб9

commit eb50d8a6bb75cc66756dd4f81a75ad0822c15b74
Author: alkaa0l0 <alinagumenyuk07@gmail.com>
Date:   Thu Dec 4 22:11:07 2025 +0700

    коммит
:
```

16. Добавьте в sort.c в main() printf(“hello git\n”);.

```
    printf("Sorted array: ");
    printf("hello git\n");
```

17. Каково содержимое sort.c?
18. Что нам говорит git status?



19. Запустите git restore sort.c
20. Каково содержимое sort.c?

```
lina@LaptopLii:/home/git/lab9$ git restore sort.c
lina@LaptopLii:/home/git/lab9$ cat sort.c
#include <stdio.h>

// Быстрая сортировка
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pivot = arr[high];
        int i = (low - 1);

        for (int j = low; j <= high - 1; j++) {
            if (arr[j] < pivot) {
                i++;
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
        int temp = arr[i + 1];
        arr[i + 1] = arr[high];
        arr[high] = temp;
        int pi = i + 1;

        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {
    int numbers[] = {64, 34, 25, 12, 22, 90};

    printf("Sorted array: ");
    printf("hello git\n");
```

21. Что нам говорит git status?

```
lina@LaptopLii:/home/git/lab9$ git status
On branch mybranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   sort.c

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    sort.c
```

22. Запушить на удаленный репо ветку.

```
lina@LaptopLii:/home/git/lab9$ git push -u origin mybranch
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 770 bytes | 770.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'mybranch' on GitHub by visiting:
remote:      https://github.com/alkaa0l0/Programming-C_projects/pull/new/mybranch
remote:
To https://github.com/alkaa0l0/Programming-C_projects.git
 * [new branch]      mybranch -> mybranch
branch 'mybranch' set up to track 'origin/mybranch'.
lina@LaptopLii:/home/git/lab9$
```

Теперь поиграемся с ветками и ff-merge Помните: 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. Создать файл greeting.txt, проиндексировать его и закоммитить с сообщением “Add file greeting.txt”.
2. Добавить в этот файл слово hello, индексируем и коммитим с текстом "Add content to greeting.txt"

```
lina@LaptopLii:/home/git/lab9$ git add greeting.txt
lina@LaptopLii:/home/git/lab9$ git commit -m "Add file greeting.txt"
[mybranch da2a984] Add file greeting.txt
 2 files changed, 47 insertions(+)
 create mode 100644 lab9/greeting.txt
 create mode 100644 lab9/sort.c
lina@LaptopLii:/home/git/lab9$
```

3. Создайте ветку с именем feature/uppercase (да, feature/uppercase — это совершенно допустимое имя ветки и общепринятое соглашение).
4. Переключитесь на эту ветку

5. Каков вывод git status?

```
lina@LaptopLii:/home/git/lab9$ git branch feature/uppercase
lina@LaptopLii:/home/git/lab9$ git checkout feature/uppercase
Switched to branch 'feature/uppercase'
lina@LaptopLii:/home/git/lab9$ git status
On branch feature/uppercase
nothing to commit, working tree clean
lina@LaptopLii:/home/git/lab9$
```

6. Отредактируйте greeting.txt, чтобы он содержал приветствие в верхнем регистре (HELLO)

7. Добавьте файл greeting.txt и закоммитьте

```
lina@LaptopLii:/home/git/lab9$ git add greeting.txt
lina@LaptopLii:/home/git/lab9$ git commit -m "commit hello txt"
[feature/uppercase ca3ebd6] commit hello txt
 1 file changed, 1 insertion(+)
lina@LaptopLii:/home/git/lab9$
```

8. Каков вывод git branch?

```
lina@LaptopLii:/home/git/lab9$ git branch
  * feature/uppercase
    main
    mybranch
```

9. Каков вывод git log --oneline --graph –all

```
lina@LaptopLii:/home/git/lab9$ git log --oneline --graph --all
* ca3ebd6 (HEAD -> feature/uppercase) commit hello txt
* f63502f (main) Add file greeting.txt
| * da2a984 (mybranch) Add file greeting.txt
| * 4ea3f4a (origin/mybranch) коммит после изменений
|/
* b91ea8e (origin/main) папка с файлами лаб9
* eb50d8a коммит
* 7700b45 еще повторный коммит
* b711632 Комиит сортировки
* cc3a32e Replace git.c with labGit.c
* 3a4a042 first commit
* 81d284d first commit
lina@LaptopLii:/home/git/lab9$
```

10.Переключитесь на главную ветку

11.Используйте cat, чтобы увидеть содержимое файла greetings.txt

```
lina@LaptopLii:/home/git/lab9$ cat greeting.txt
```

13. Сравните ветки

```
"HELLO"lina@LaptopLii:/home/git/lab9$ git diff main..feature/uppercase
diff --git a/lab9/greeting.txt b/lab9/greeting.txt
index e69de29..e27d4b8 100644
--- a/lab9/greeting.txt
+++ b/lab9/greeting.txt
@@ -0,0 +1 @@
+"HELLO"
\ No newline at end of file
lina@LaptopLii:/home/git/lab9$
```

14. Объедините ветки

```
lina@LaptopLii:/home/git/lab9$ git merge feature/uppercase
Updating f63502f..ca3ebd6
Fast-forward
 lab9/greeting.txt | 1 +
 1 file changed, 1 insertion(+)
lina@LaptopLii:/home/git/lab9$
```

15. Используйте cat, чтобы увидеть содержимое файла greetings.txt

```
lina@LaptopLii:/home/git/lab9$ cat greeting.txt
"HELLO"lina@LaptopLii:/home/git/lab9$
```

16. Удалите ветку с заглавными буквами (feature/uppercase)

```
"HELLO"lina@LaptopLii:/home/git/lab9$ git branch -d feature/uppercase
Deleted branch feature/uppercase (was ca3ebd6).
```

17. Смержить ветку mybranch в master (git merge)

```
lina@LaptopLii:/home/git/lab9$ git commit -m "Merge branch 'mybranch' into main"
[main 3535577] Merge branch 'mybranch' into main
lina@LaptopLii:/home/git/lab9$ git merge mybranch
Already up to date.
```

18. Что выводит git log --oneline --graph –all?

```
lina@LaptopLii:/home/git/lab9$ git log --oneline --graph --all
*   3535577 (HEAD -> main) Merge branch 'mybranch' into main
|\
| * da2a984 (mybranch) Add file greeting.txt
| * 4ea3f4a (origin/mybranch) коммит после изменений
* | ad1e685 changes
* | ca3ebd6 commit hello txt
* | f63502f (origin/main) Add file greeting.txt
|/
* b91ea8e папка с файлами лаб9
* eb50d8a коммит
* 7700b45 еще повторный коммит
* b711632 Комиит сортировки
* cc3a32e Replace git.c with labGit.c
* 3a4a042 first commit
* 81d284d first commit
```

19. Запушить изменения ветки master на удаленный репо.

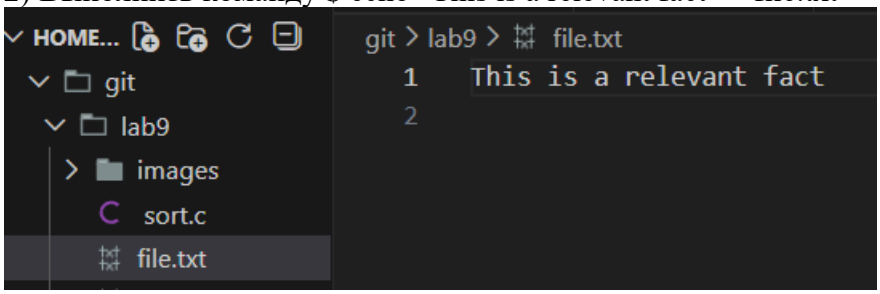
20. Запушить документ с результатами вашей работы

Оценка 5:

1) Создать ветку branch1, переключиться на нее

```
lina@LaptopLii:/home/git/lab9$ git branch branch1
lina@LaptopLii:/home/git/lab9$ git checkout branch1
Switched to branch 'branch1'
```

2) Выполнить команду $ echo "This is a relevant fact" > file.txt



3) Закоммитить это изменение

```
lina@LaptopLii:/home/git/lab9$ git commit  -m "commit"
[branch1 ace5501] commit
 3 files changed, 5 insertions(+)
 create mode 100644 lab9/README.md
 create mode 100644 lab9/file.txt
 create mode 100644 lab9/images/image.png
```

4) Переключиться на главную ветку и выполнить команду echo "This is an indispensable truth!" > file.txt

```
lina@LaptopLii:/home/git/lab9$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
lina@LaptopLii:/home/git/lab9$ echo "This is an indispensable truth!" > file.txt
lina@LaptopLii:/home/git/lab9$
```

5) Закоммитить изменения в master

```
lina@LaptopLii:/home/git/lab9$ git commit -m "commit"
[main a8b56ab] commit
 1 file changed, 1 insertion(+)
 create mode 100644 lab9/file.txt
```

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

```
lina@LaptopLii:/home/git/lab9$ git log --oneline --graph --all
* a8b56ab (HEAD -> main) commit
| * ace5501 (branch1) commit
|/
*   3535577 (origin/main) Merge branch 'mybranch' into main
|\
| * da2a984 (mybranch) Add file greeting.txt
:...skipping...
* a8b56ab (HEAD -> main) commit
| * ace5501 (branch1) commit
|/
*   3535577 (origin/main) Merge branch 'mybranch' into main
|\
| * da2a984 (mybranch) Add file greeting.txt
| * 4ea3f4a (origin/mybranch) коммит после изменений
:...skipping...
* a8b56ab (HEAD -> main) commit
```

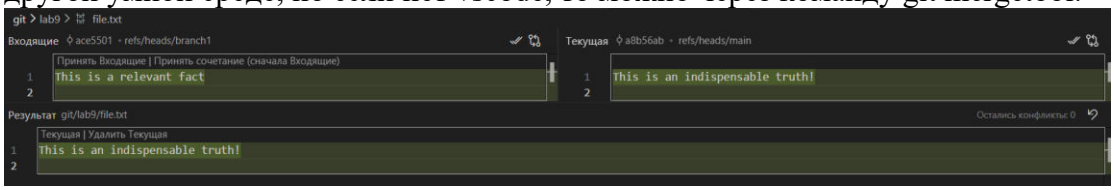
7) Использовать команду git merge чтобы смержить ветку branch1 в master (получим конфликт это норм)

```
lina@LaptopLii:/home/git/lab9$ git merge branch1
Auto-merging lab9/file.txt
CONFLICT (add/add): Merge conflict in lab9/file.txt
Automatic merge failed; fix conflicts and then commit the result.
```

8) Что показывает git status?

```
lina@LaptopLii:/home/git/lab9$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
        new file:   README.md
        new file:   images/image.png

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both added:      file.txt
```

9) Посмотреть содержимое файла file.txt и в любом любимом текстовом редакторе исправьте конфликт. Желательно делать это в VSCode или любой другой умной среде, но если нет vscode, то можно через команду git mergetool.



10) Что показывает git log --oneline –graph?

```
lina@LaptopLii:/home/git/lab9$ git log --oneline --graph
* a8b56ab (HEAD -> main) commit
*   3535577 (origin/main) Merge branch 'mybranch' into main
|\
| * da2a984 (mybranch) Add file greeting.txt
| * 4ea3f4a (origin/mybranch) коммит после изменений
* | ad1e685 changes
* | ca3ebd6 commit hello txt
* | f63502f Add file greeting.txt
|/
* b91ea8e папка с файлами лаб9
* eb50d8a коммит
* 7700b45 еще повторный коммит
* b711632 Комиит сортировки
* cc3a32e Replace git.c with labGit.c
* 3a4a042 first commit
* 81d284d first commit
lina@LaptopLii:/home/git/lab9$
```

11) Запушить изменения

Починим Merge конфликты для сортировки MergeSort на python. Содержимое base.py, lefty.py и righty.py находится в конце этого документа.
1) Находясь в ветке master создадим файл mergesort.py с содержимым из base.py (см. приложение)
2) 2) Проиндексируем файл и закоммитим

```
lina@LaptopLii:/home/git/lab9$ cd ./MergeSort/
lina@LaptopLii:/home/git/lab9/MergeSort$ git add mergesort.py
lina@LaptopLii:/home/git/lab9/MergeSort$ git commit -m "commit merpy"
[main 66af139] commit merpy
 4 files changed, 8 insertions(+)
 create mode 100644 lab9/MergeSort/mergesort.py
 create mode 100644 lab9/README.md
 create mode 100644 lab9/images/image.png
lina@LaptopLii:/home/git/lab9/MergeSort$
```

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

```
lina@LaptopLii:/home/git/lab9/MergeSort$ git branch Mergesort-Impl
lina@LaptopLii:/home/git/lab9/MergeSort$ git checkout M
error: pathspec 'M' did not match any file(s) known to git
lina@LaptopLii:/home/git/lab9/MergeSort$ git checkout Mergesort-Impl
Switched to branch 'Mergesort-Impl'
```

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

```
lina@LaptopLii:/home/git/lab9/MergeSort$ git add mergesort.py
lina@LaptopLii:/home/git/lab9/MergeSort$ git commit -m "change code"
[Mergesort-Impl a9fca78] change code
 1 file changed, 35 insertions(+)
lina@LaptopLii:/home/git/lab9/MergeSort$
```

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

```
lina@LaptopLii:/home/git/lab9/MergeSort$ git add mergesort.py
lina@LaptopLii:/home/git/lab9/MergeSort$ git commit -m "change code main"
[main d75cbe1] change code main
 1 file changed, 35 insertions(+)
```

8) 8) Что показывает git log --oneline –graph?

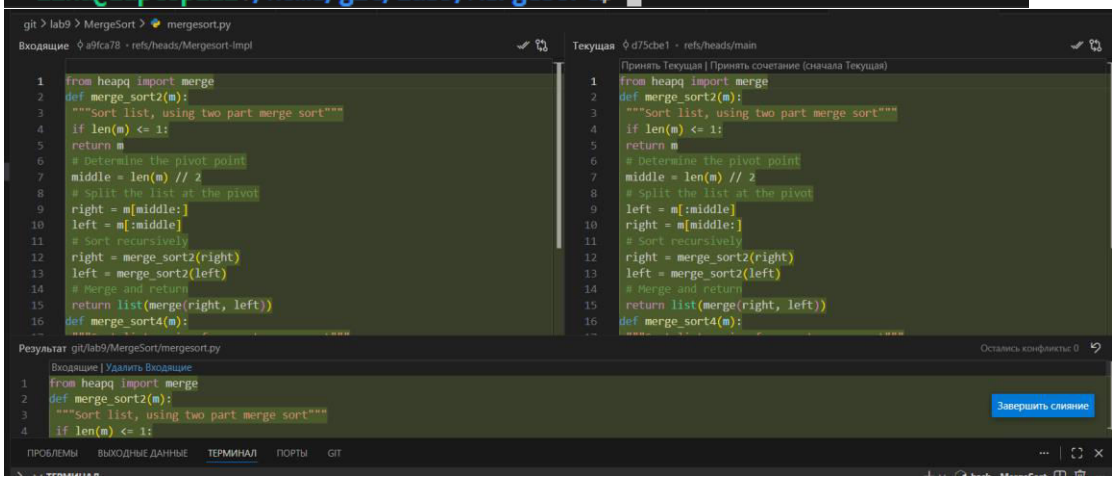
```
lina@LaptopLii:/home/git/lab9/MergeSort$ git log --onel
* d75cbe1 (HEAD -> main) change code main
* 66af139 commit merpy
* a8b56ab (origin/main) commit
*   3535577 Merge branch 'mybranch' into main
|\
| * da2a984 (mybranch) Add file greeting.txt
| * 4ea3f4a (origin/mybranch) коммит после изменений
* | ad1e685 changes
* | ca3ebd6 commit hello txt
* | f63502f Add file greeting.txt
|/
* b91ea8e папка с файлами лаб9
* eb50d8a коммит
* 7700b45 еще повторный коммит
* b711632 Комиит сортировки
* cc3a32e Replace git.c with labGit.c
* 3a4a042 first commit
* 81d284d first commit
```

9) 9) Что показывает git branch?

```
lina@LaptopLii:/home/git/lab9/MergeSort$ git branch
  Mergesort-Impl
  branch1
* main
  mybranch
```

10) 10) Необходимо смержить Mergesort-Impl в master.

```
lina@LaptopLii:/home/git/lab9/MergeSort$ git merge Mergesort-Impl
Auto-merging lab9/MergeSort/mergesort.py
CONFLICT (content): Merge conflict in lab9/MergeSort/mergesort.py
Automatic merge failed; fix conflicts and then commit the result.
lina@LaptopLii:/home/git/lab9/MergeSort$
```



11) 11) После исправления всех merge конфликтов запушить в master изменения.

12) 12) Запушить документ с результатами вашей работы