Practical 9 Software engineering

Aim: To perform manual and automated testing using CASE tool using sample GUI.

Theory:

1. **Automated Testing:** It is a technique where tester write scripts by own and uses suitable software or automation Tool to test the software. It is Automation Process of a manual Process. Allows To execute repetitive task without intervention of manual tester.
2. **Manual Vs Automated Testing:**
    1. **MANUAL TESTING:** Not accurate at all times due to human error therefore less reliable. Heavy investment for human resources. Time consuming. Allows for careful human observation. Initial investment is lower which in turns lowers the returns.
    2. **AUTOMATED TESTING:** Since is performed by third party tools and or Scripts therefore More reliable. Investment for tools rather than for human resources. Faster than manual testing as it is executed by tools. Practical option when test cases are run repeatedly. Higher initial investment hence higher rate of return.

3.Automation Testing Process:

1. Test Tool Selection: There will be some criteria for selection of tool. Majority of the Criteria includes: Do we have skilled resource to allocate for automation tasks, Budget constraints, Do the tool satisfy our needs?

2. Define Scope of Automation: This includes few basic points such as: Framework, should be support automation scripts, less maintenance must be there, High return on Investment, Not much complex test cases.

3. Planning Design and Development: For this we need to install particular frameworks or libraries, start designing and developing the test cases such as NUnit, Junit, Qunit or required software automation tool.

4. Test Execution: Final Execution of test cases will take place in this phase and it depends on language such as For.NET, will be using Junit, For JavaScript, we ill be Using Qunit or jasmine etc.

5.Maintainence: Creation of reports generated after Tests and that should be documented so as to refer that in future for next iterations.

4.Popular Automation Tools:

1. Selenium

2. QTP

3.Sikuli

4. Appium

5.Jmeter

6.Test NG

7. Zephyr

8.UFT.

Advantages of Automation Testing:

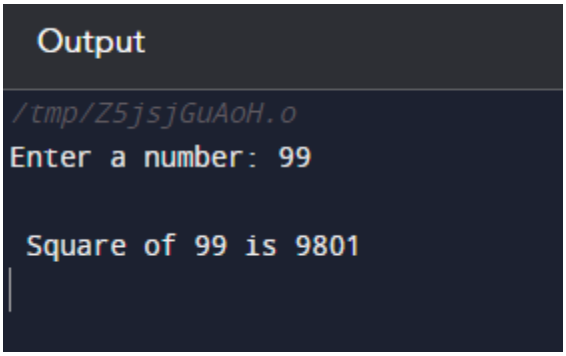
1. Simplifies Test Cases Execution
2. Improves Reliability
3. Increases amount of test Coverage
4. Minimizing Human Interaction
5. Saves Time and money
6. Test Results are made public.

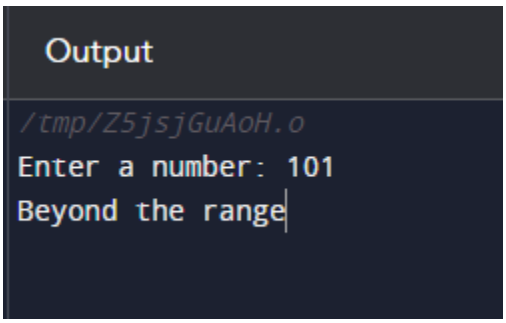
Code:

```
#include <stdio.h>

int

main()

{

   int n, res;

   printf("Enter a number: ");

   scanf("%d", &n);

   if (n >= 1 && n <= 100)

   {

         res = n * n;

         printf("\n Square of %d is %d\n", n, res);
```

```
    }
    else if (n<= 0 || n > 100)
            printf("Beyond the range");

    return 0;
}
```

OUTPUT:





Sample GUI: The Graphical User Interface is the form of user interface that allows user to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of

text-based user interfaces, typed command labels or text navigation. GUI'S were introduced in reaction to the perceived steep learning curve of command line interfaces which require commands to be typed on a computer keyboard.

GUI uses the combination of technologies and device to provide a platform that users can interact with, for the tasks of gathering and producing information. A series of elements conforming have evolved to represent information stored in computers. This makes it easier for people with few computer skills to work with and use computer software.

List of the GUI'S is: The article explains the most common elements of visual language interfaces found in the WIMP (window, icon, menu, pointer) paradigm although many also used at other GUI's post WIMP interfaces. These elements are usually embodied in an interface using a widget toolkit or desktop environment.

Conclusion: In these practical I have learned about Automated test cases using case tool using sample GUI.

Practical 10 Software engineering

Aim: To Study and execute Version Control using Subversion

Theory:

Version Control System (VCS) is a software that helps software developers to work together and maintain a complete history of their work.

Following are the goals of a Version Control System.

- Allow developers to work simultaneously.
- Do not overwrite each other's changes.
- Maintain history of every version of everything.

A VCS is divided into two categories.

- Centralized Version Control System (CVCS), and
- Distributed/Decentralized Version Control System (DVCS).

In this tutorial, we will concentrate only on the Centralized Version Control System and especially **Subversion**. Subversion falls under centralized version control system, meaning that it uses central server to store all files and enables team collaboration.

Version Control Terminologies

Let us start by discussing some of the terms that we will be using in this tutorial.

- **Repository:** A repository is the heart of any version control system. It is the central place where developers store all their work. Repository not only stores files but also the history. Repository is accessed over a network, acting as a server and version control tool acting as a client. Clients can connect to the repository, and then they can store/retrieve their changes to/from repository. By storing changes, a client makes these changes available to other people and by retrieving changes, a client takes other people's changes as a working copy.
- **Trunk:** The trunk is a directory where all the main development happens and is usually checked out by developers to work on the project.
- **Tags** : The tags directory is used to store named snapshots of the project. Tag operation allows to give descriptive and memorable names to specific version in the repository.

  For example, LAST_STABLE_CODE_BEFORE_EMAIL_SUPPORT is more memorable than

Repository UUID: 7ceef8cb-3799-40dd-a067-c216ec2e5247 and

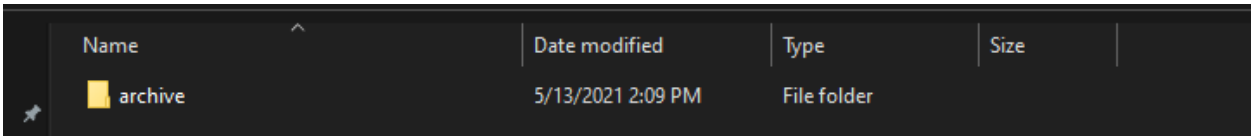
Revision: 13

- **Branches:** Branch operation is used to create another line of development. It is useful when you want your development process to fork off into two different directions. For example, when you release version 5.0, you might want to create a branch so that development of 6.0 features can be kept separate from 5.0 bug-fixes.
- **Working copy:** Working copy is a snapshot of the repository. The repository is shared by all the teams, but people do not modify it directly. Instead each developer checks out the working copy. The working copy is a private workplace where developers can do their work remaining isolated from the rest of the team.
- **Commit changes:** Commit is a process of storing changes from private workplace to central server. After commit, changes are made available to all the team. Other developers can retrieve these changes by updating their working copy. Commit is an atomic operation. Either the whole commit succeeds or is rolled back. Users never see half finished commit.

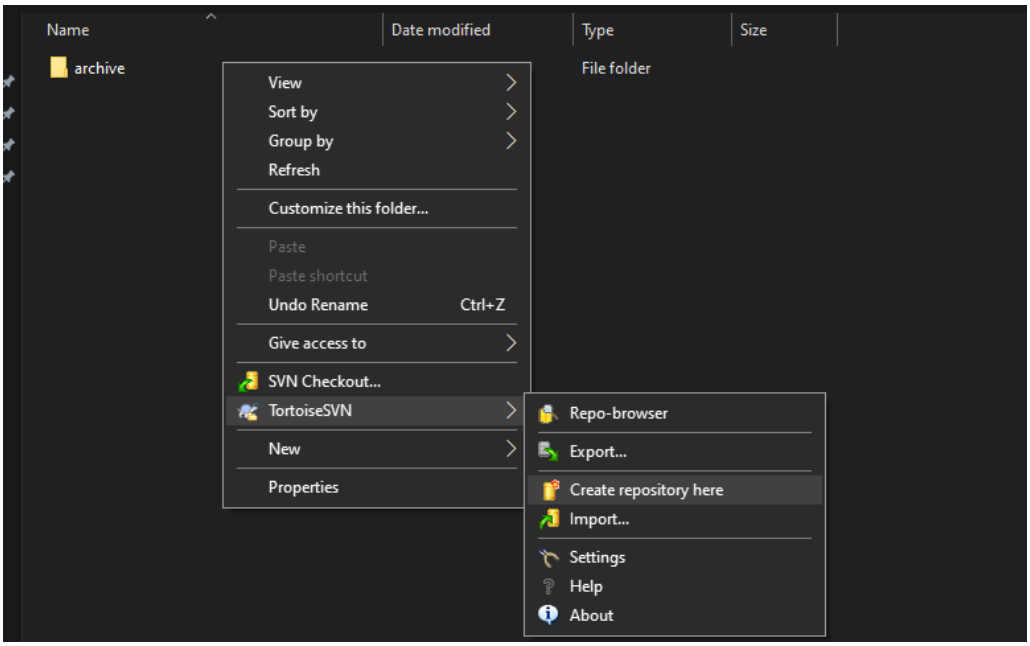
Steps

Step 1 create a folder

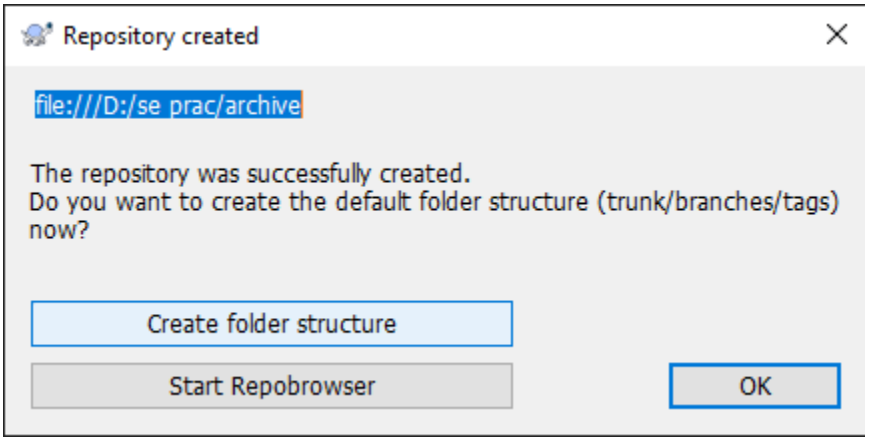
Step 2 create a sub folder named archive



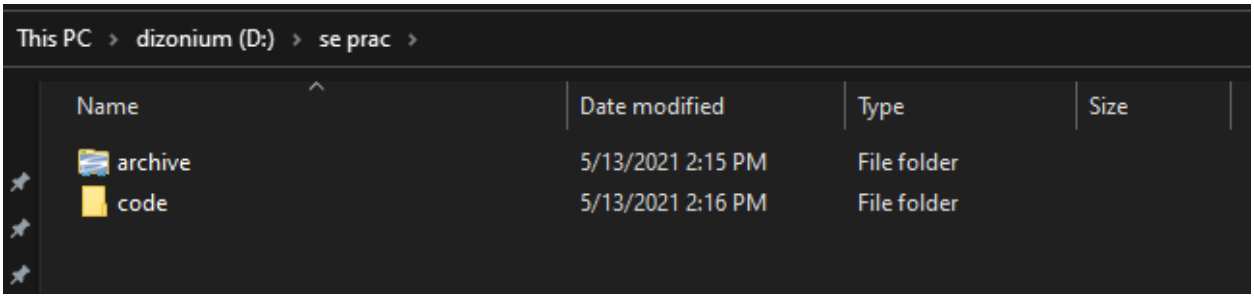
Step 3 right click on it and select create repository here



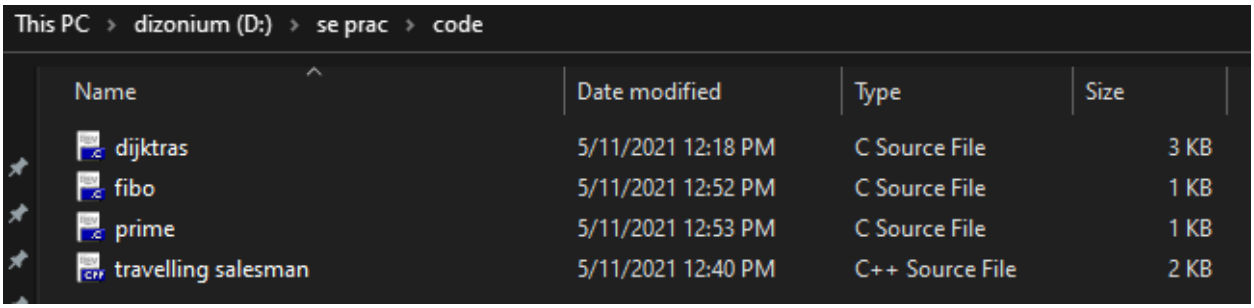
Step 4 click on create folder structure



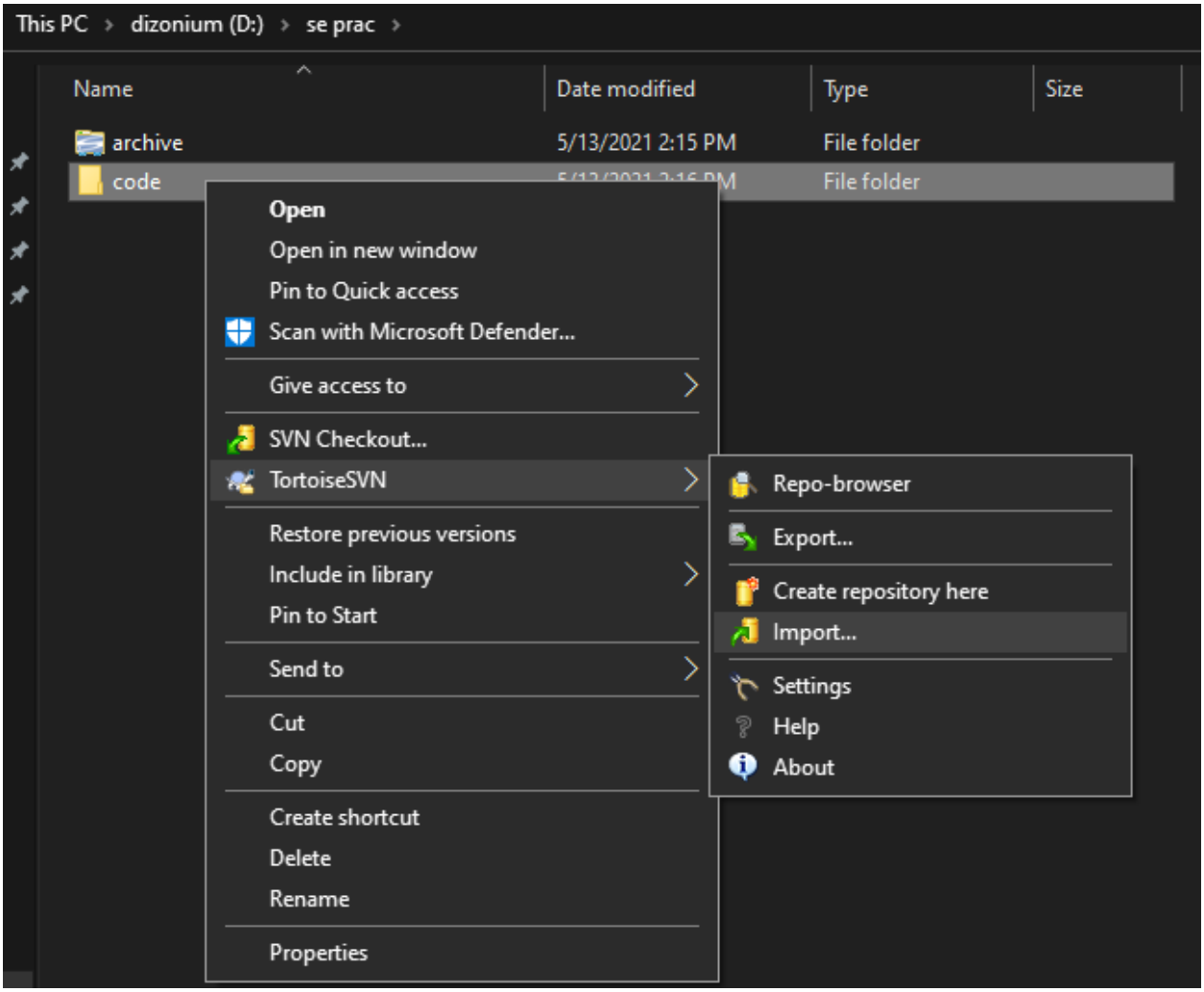
Step 5 create new folder code



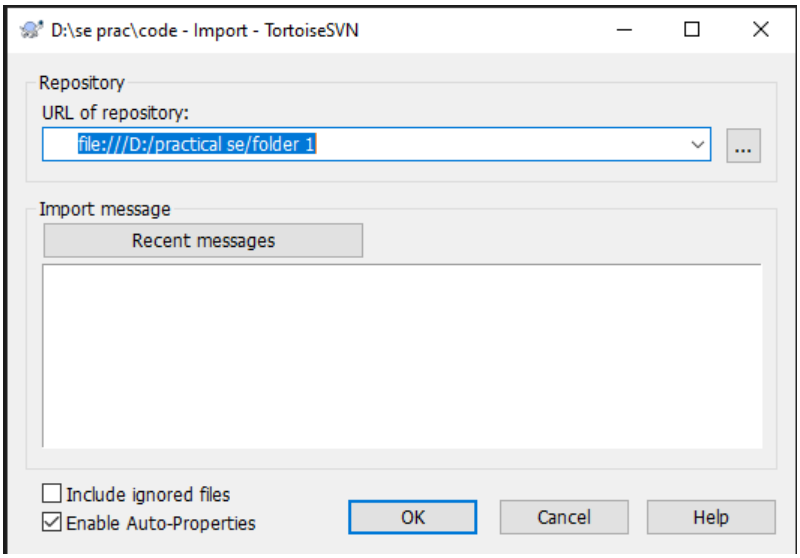
Step 6 add some code in that folder



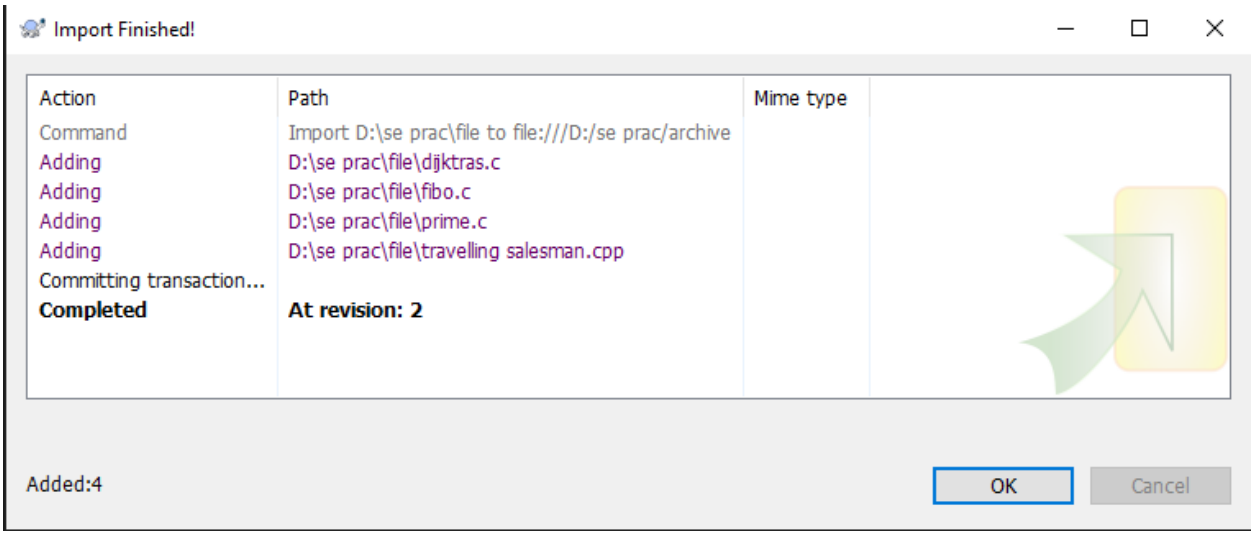
Step 7 import this folder named code



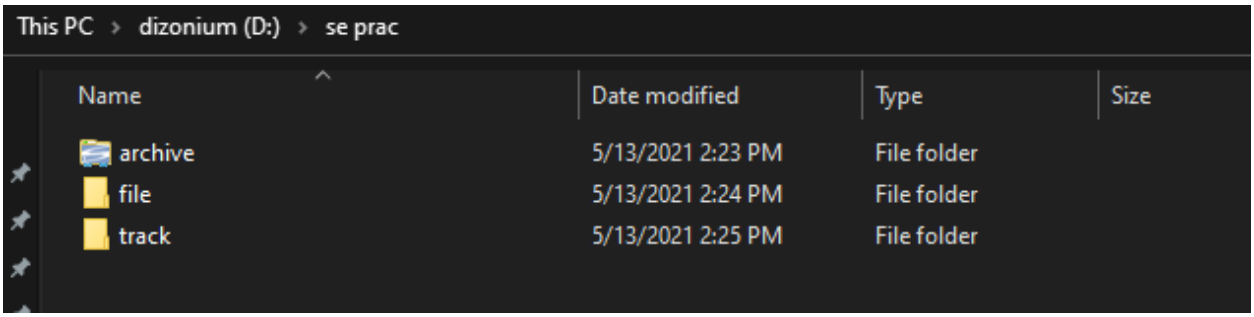
Step 8 import



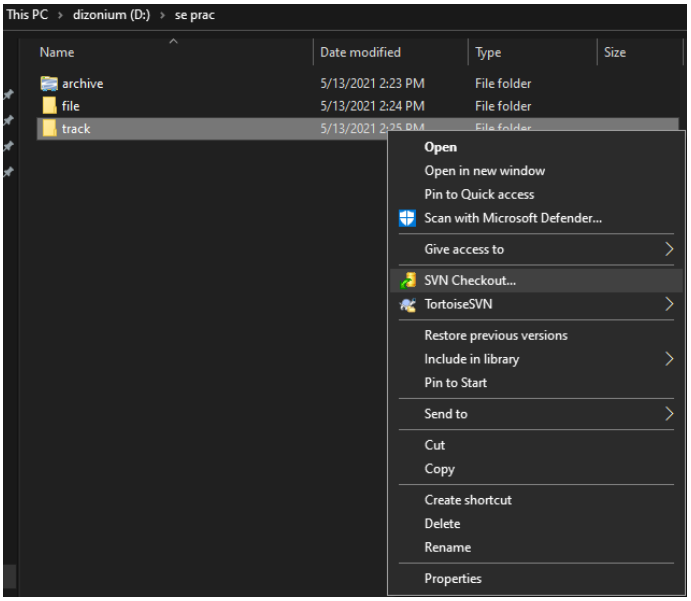
Step 9



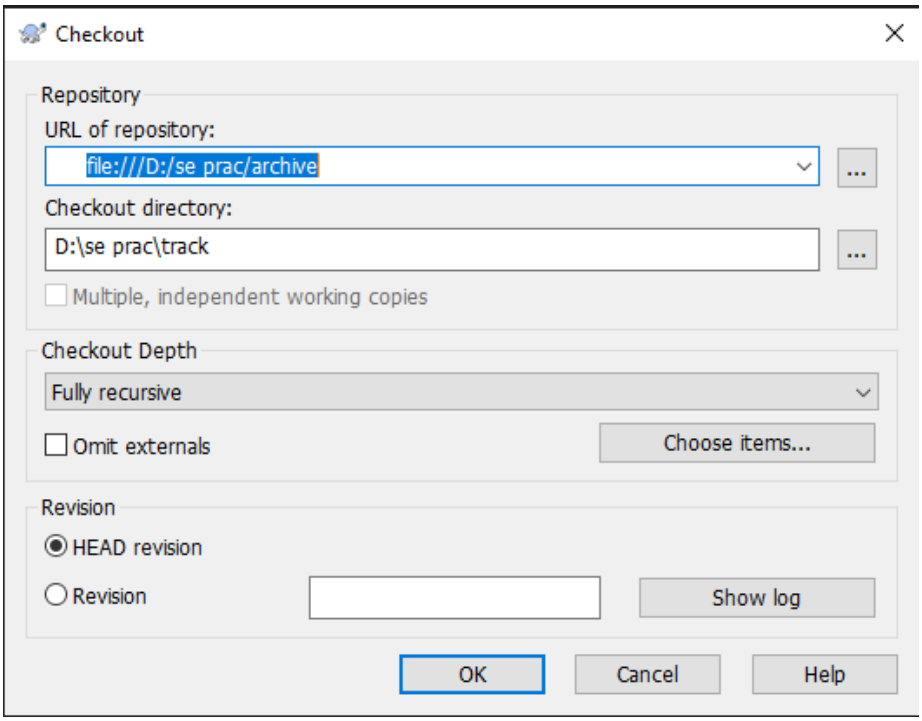
Step 10 create a track folder



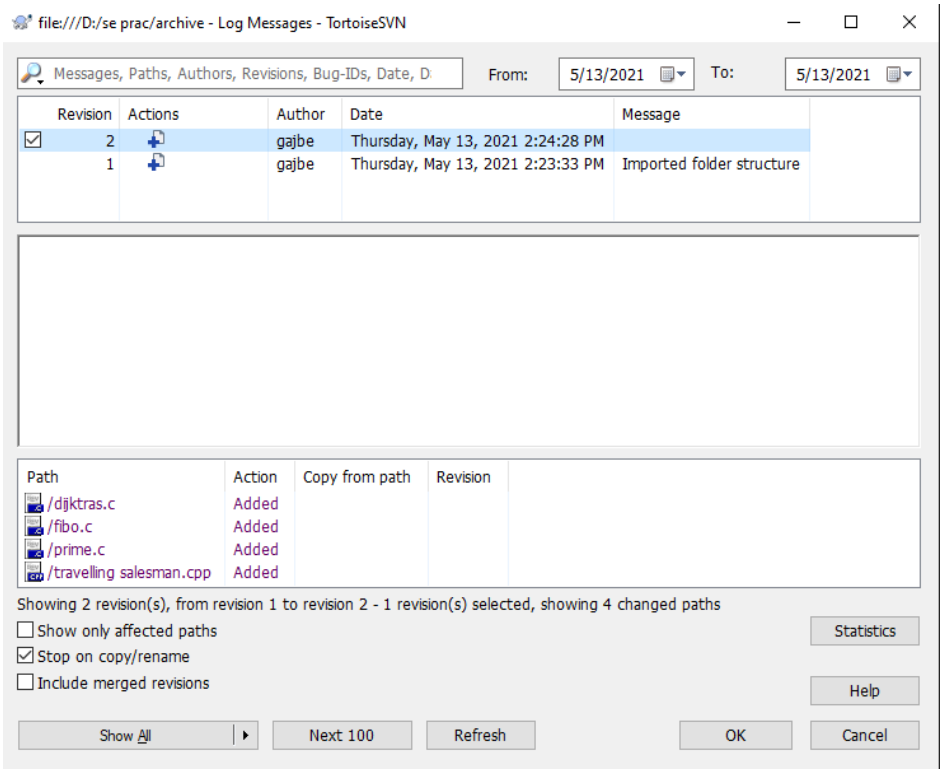
Step 11 click on svn checkout



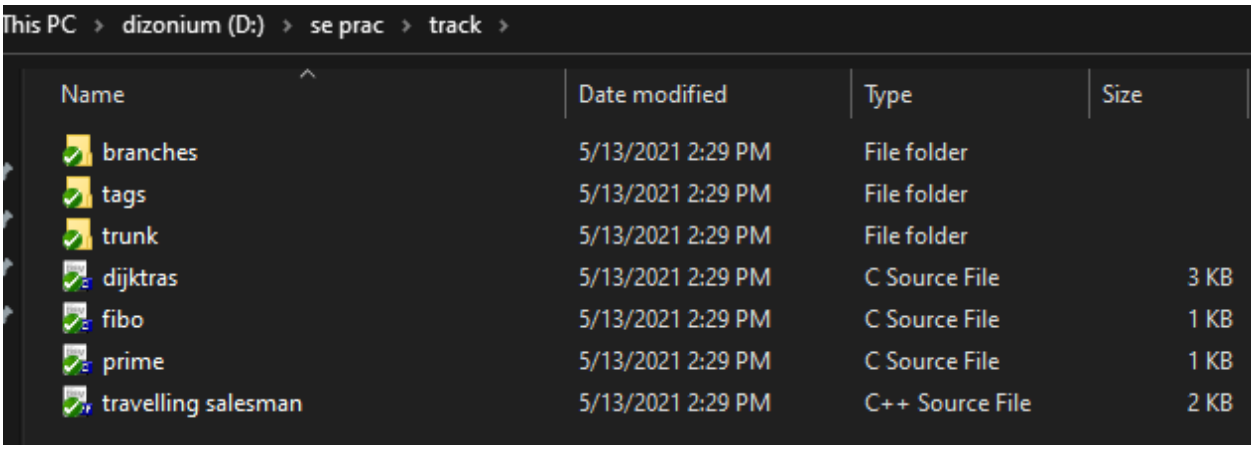
Step 12 click on ok



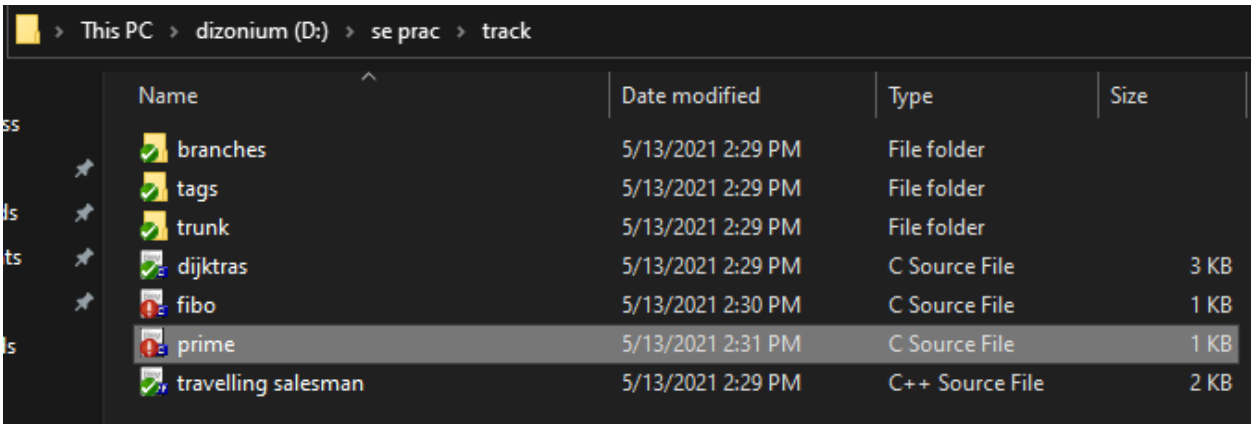
Screen shot of the logs



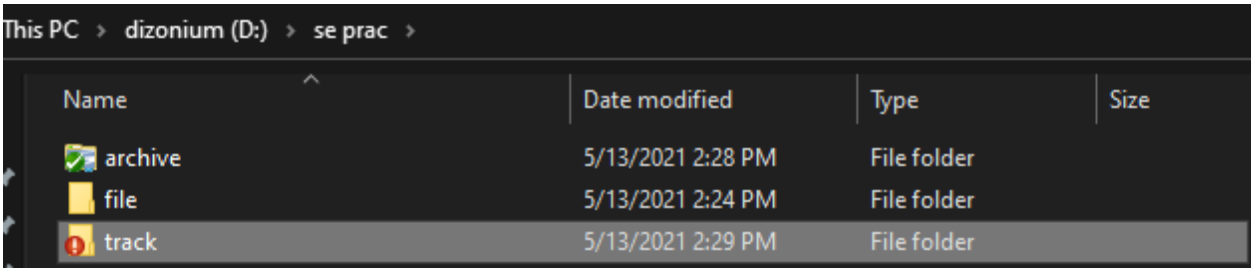
Step 13 go inside folder track



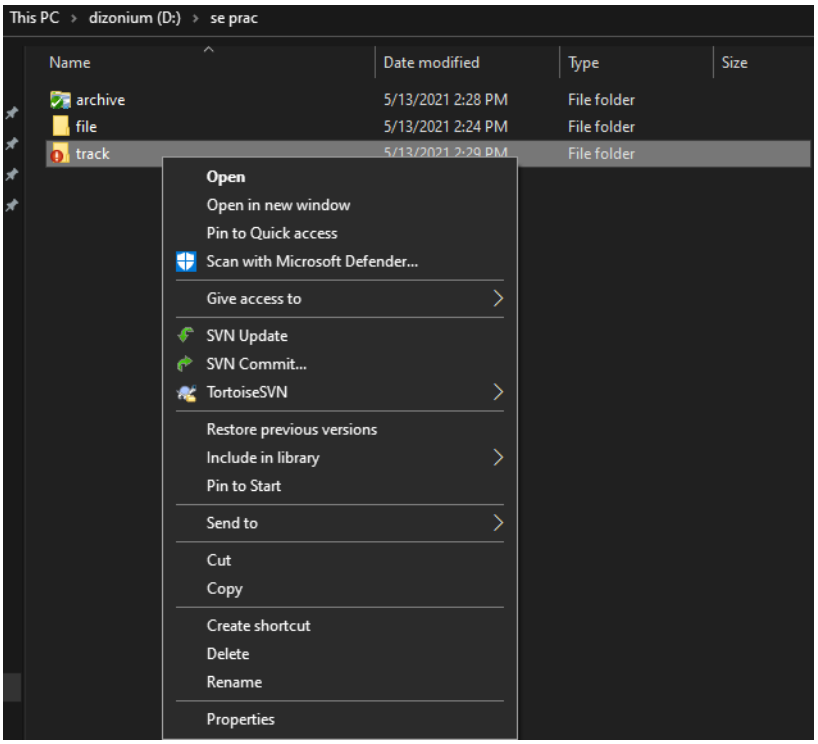
Step 14 modify any two files



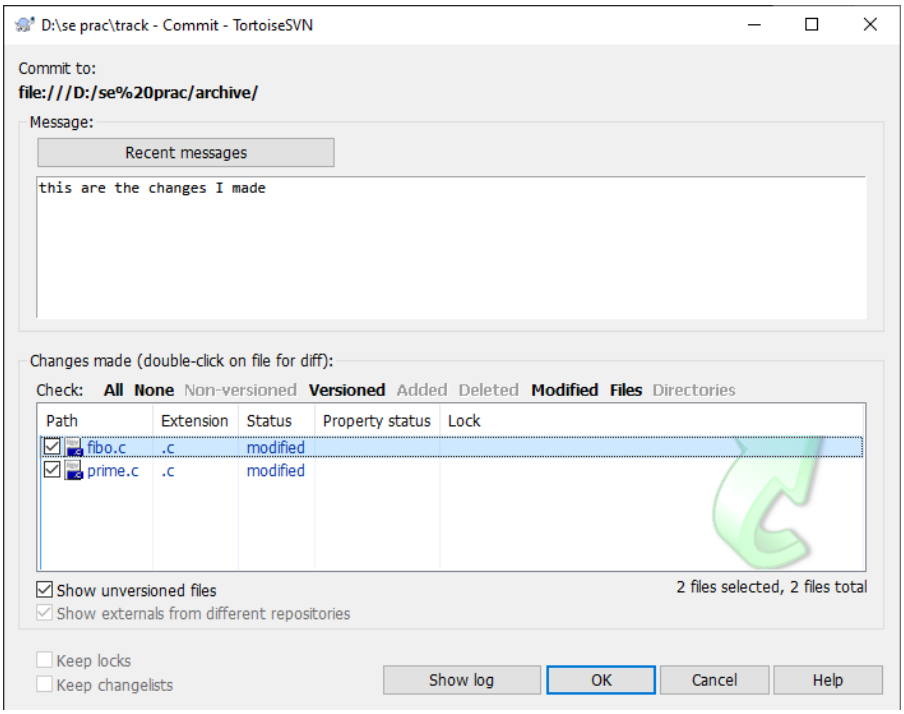
Changes made the exclamation mark is sceen……



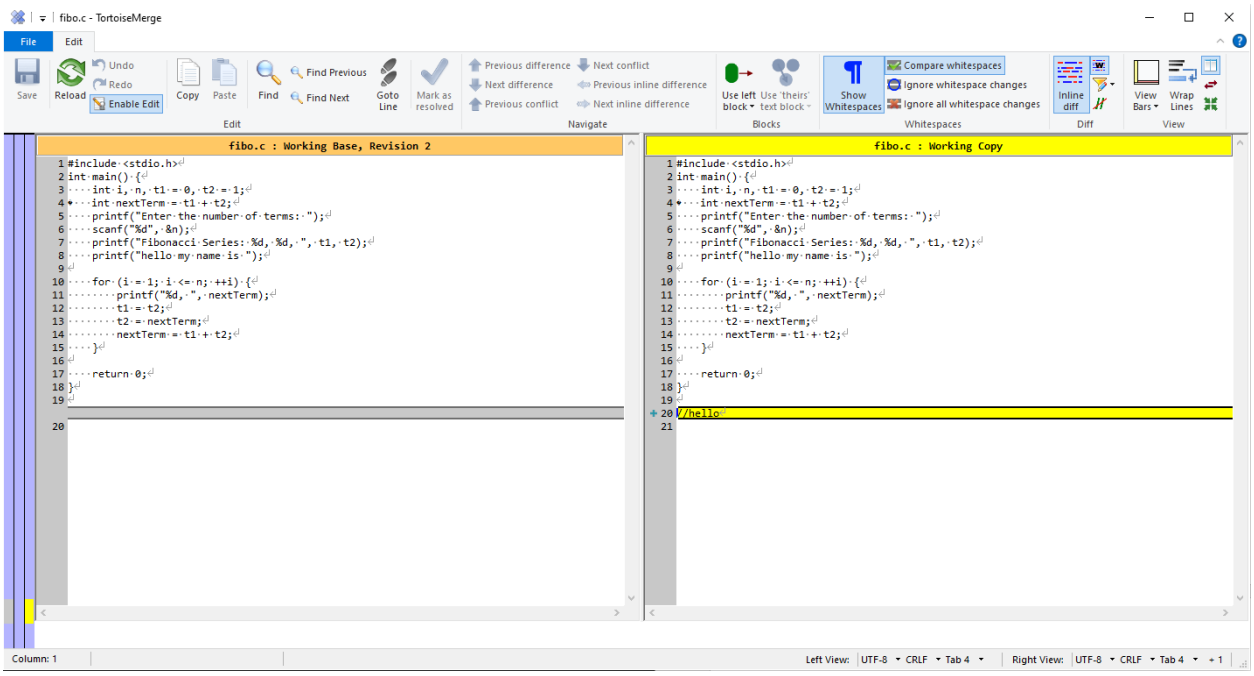
Step 15 commit



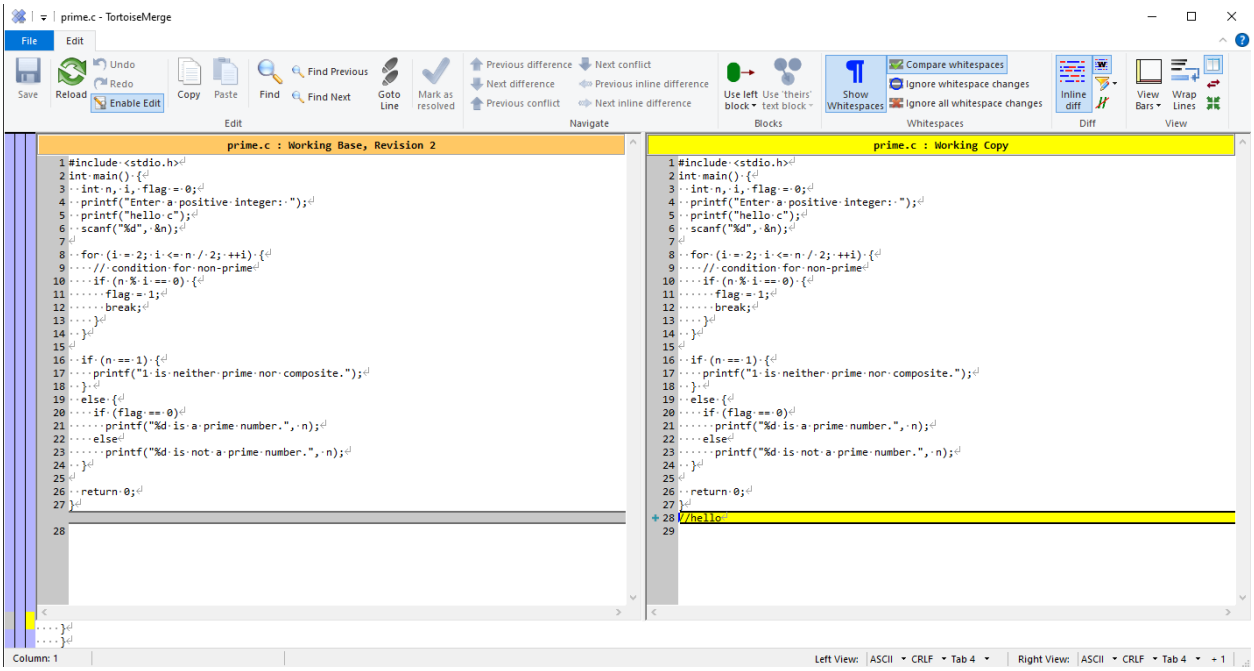
After that



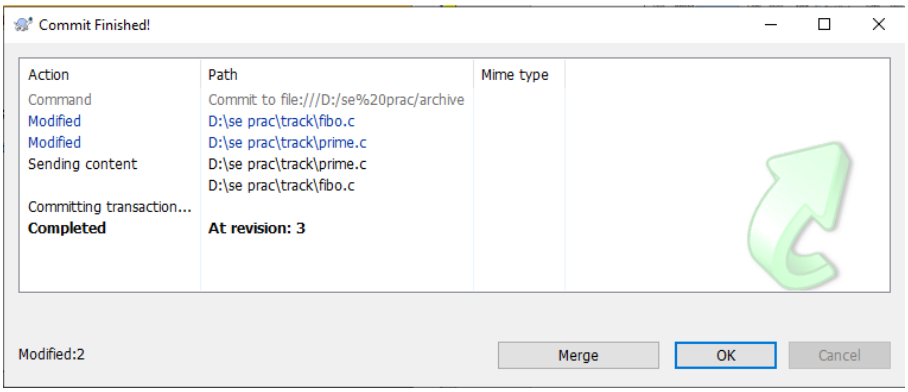
Modifications in fibo:



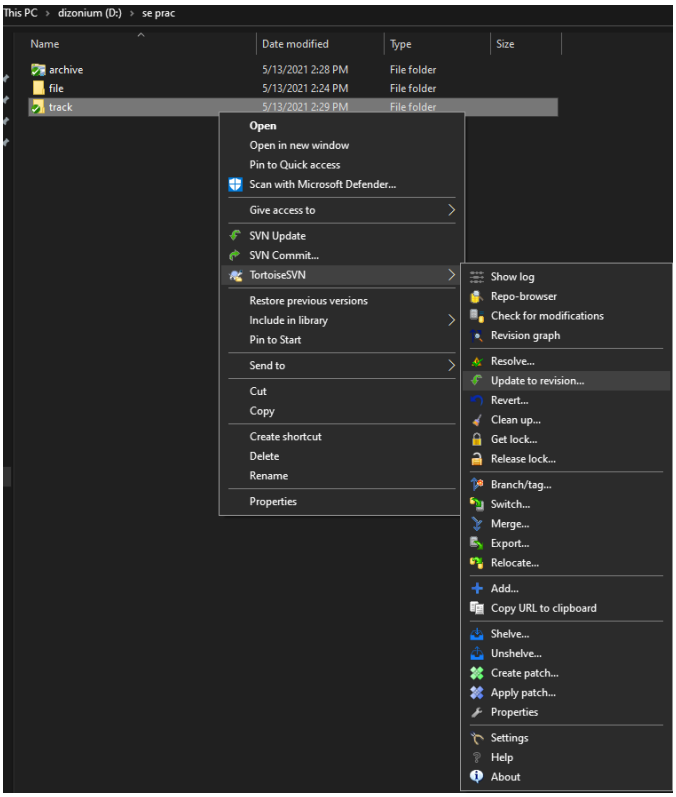
Modifications in prime:



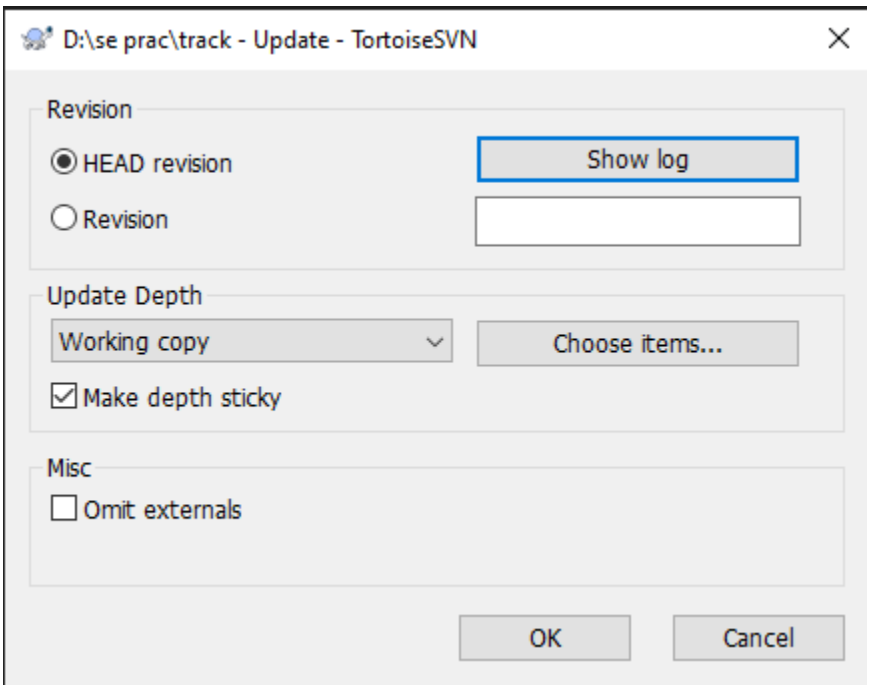
Step 16 commite

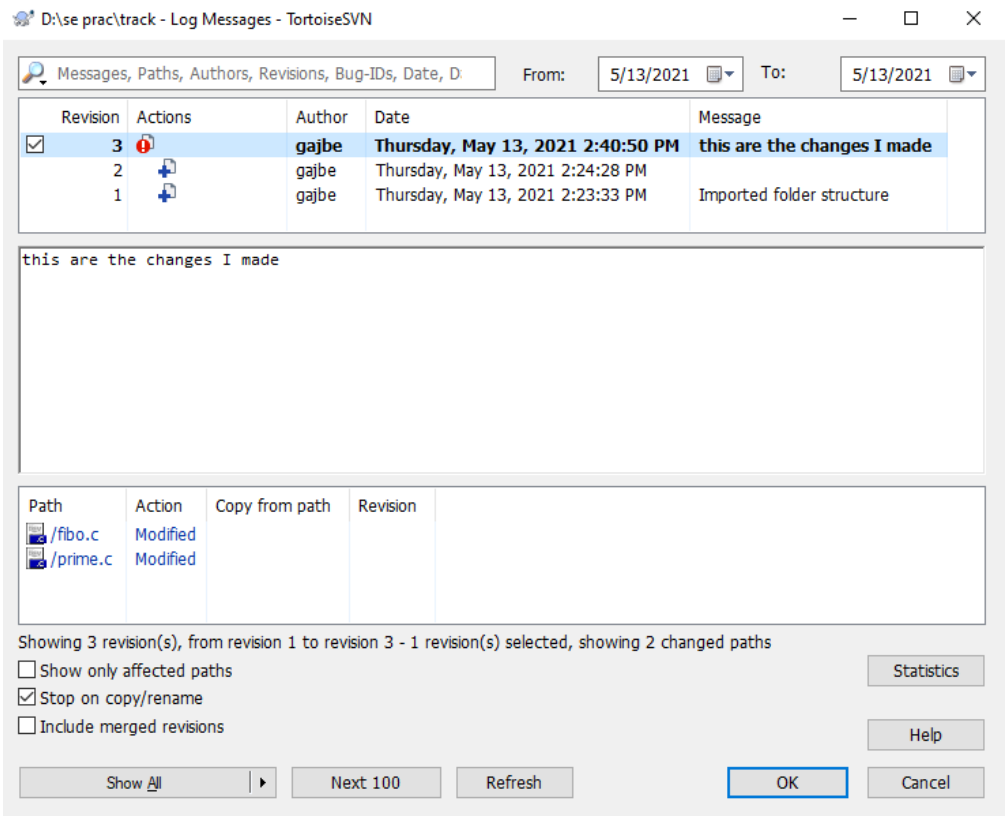


Step 17 Update to revision

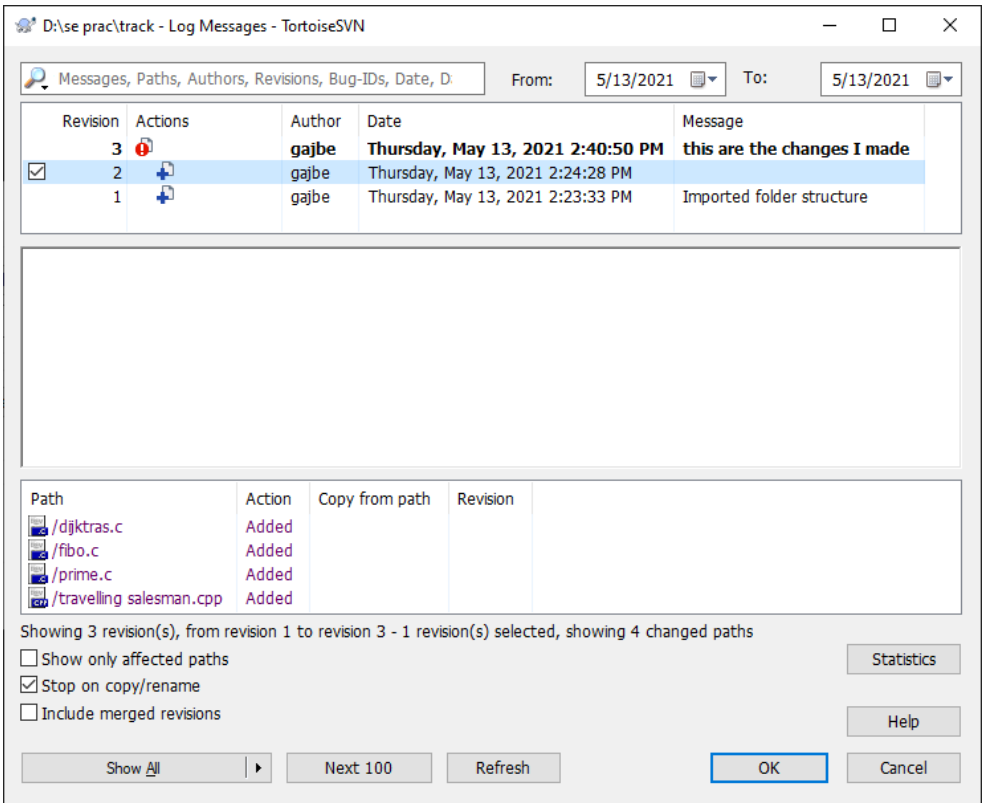


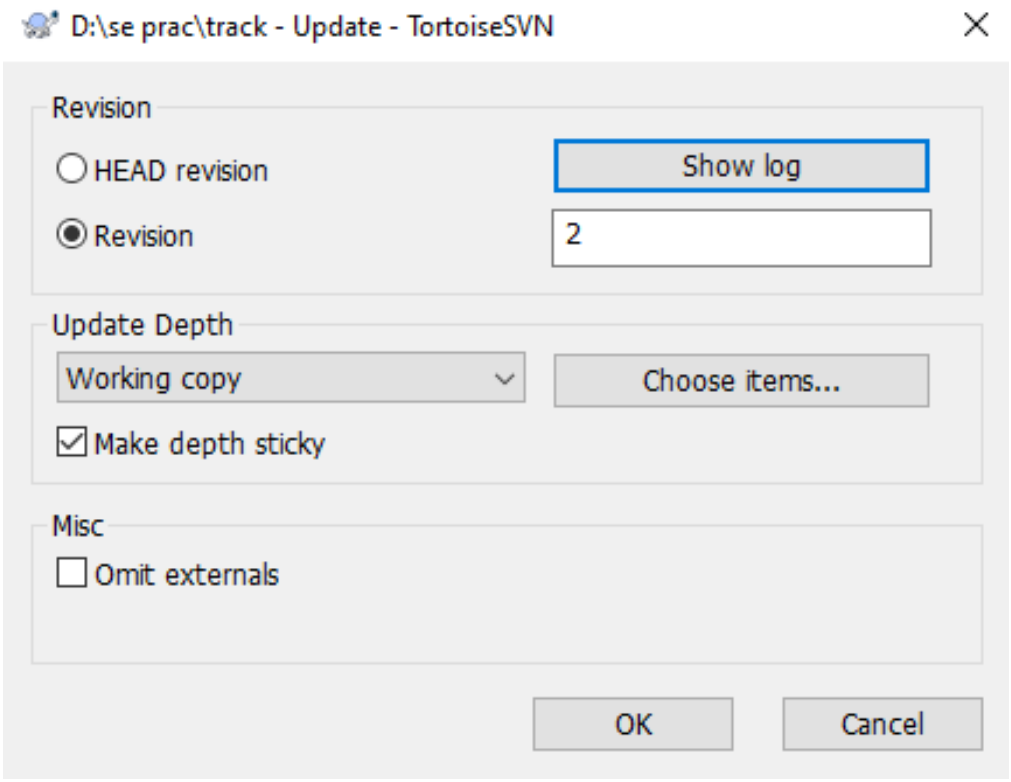
Step 18 view



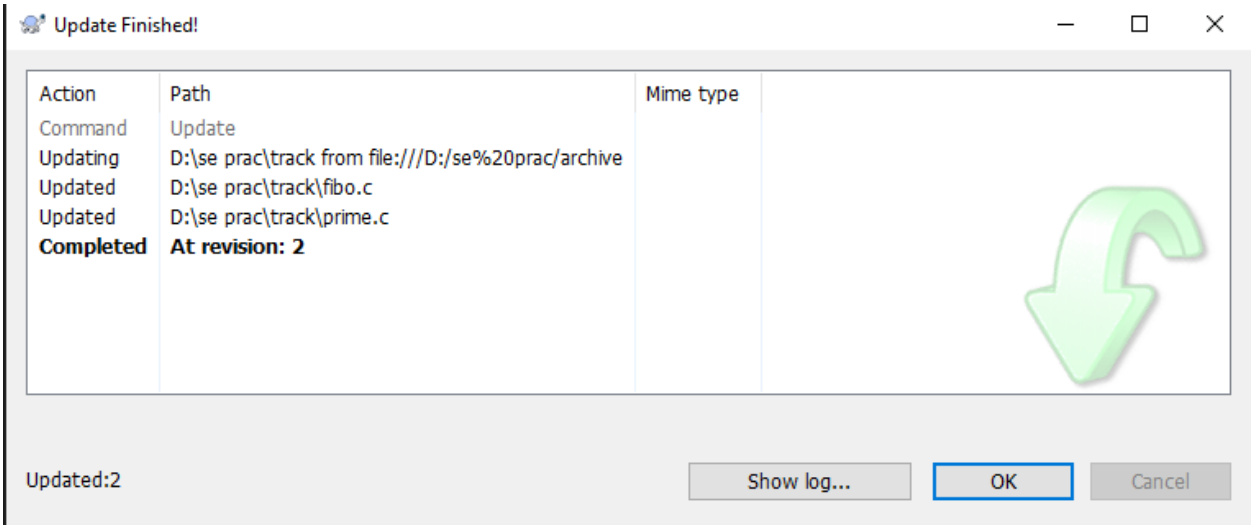


Step 19 click on Revision 2





Step 20 update complete



Conclusion: Successfully implemented the SVN commit functionality and changed the version control

By Atharva Gajbe(6ctb-142)