User Manual

C Path Finder

Sajed Jalil 12/14/18 Software Project

Contents

User Manual

C Path Finder is a unit test case generator intended for C programmers. Therefore, it is assumed that the user has experience of running general IDE.

Minimum System Requirements

Processor - Dual core 2.0 GHz Processor

Ram - 2 GB DDR3

JDK/JRE – 10 or later versions

Hard Disk – Minimum 50 MB available Hard Drive for Installation

Recommended System Requirements

Processor – Intel Core i7 (6^{th} Generation or upwards)

Ram - 8 GB DDR3

JDK/JRE – 10 or later versions

Hard Disk – 100 MB available Hard Drive for Installation

Downloading the installer

Interactive windows Installer can be downloaded from https://github.com/Sajed49/C-Path-Finder/releases/tag/1.0.0.

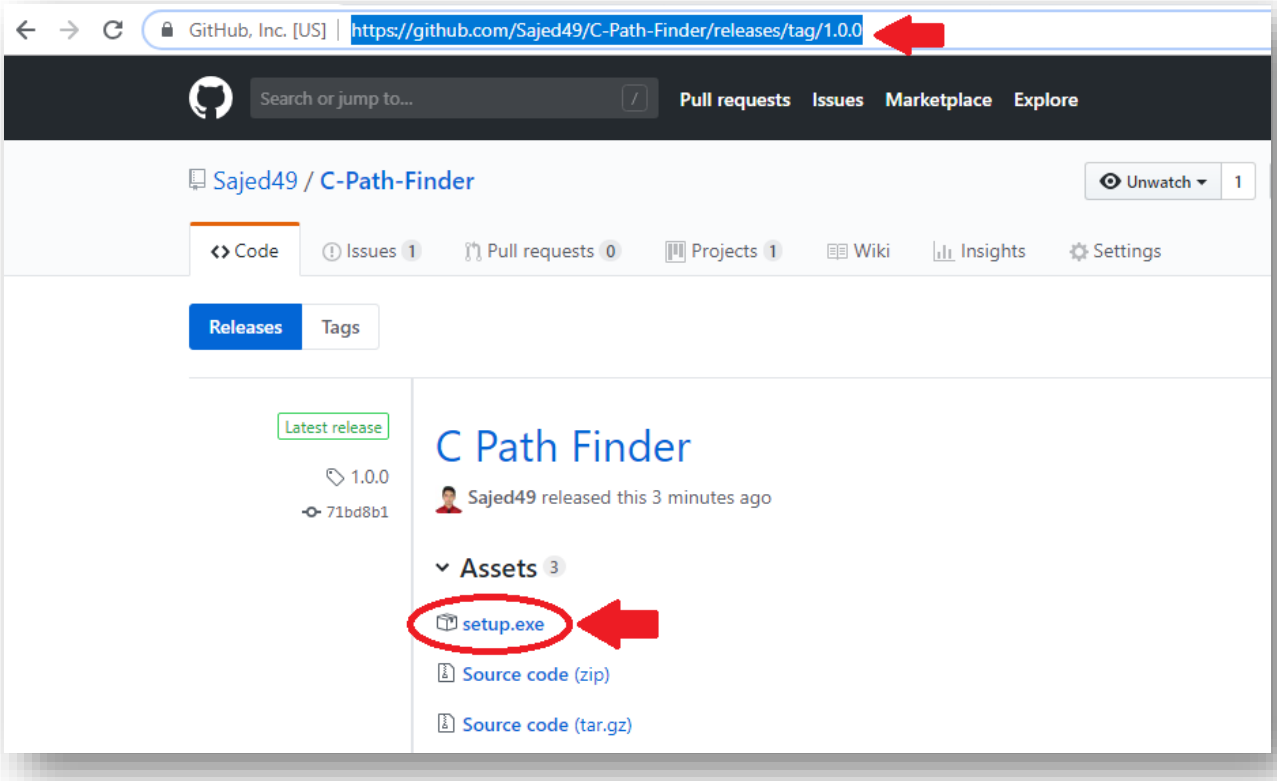

Figure 1: Download from website

Running the Installer

The downloaded installer can be run as windows executable.

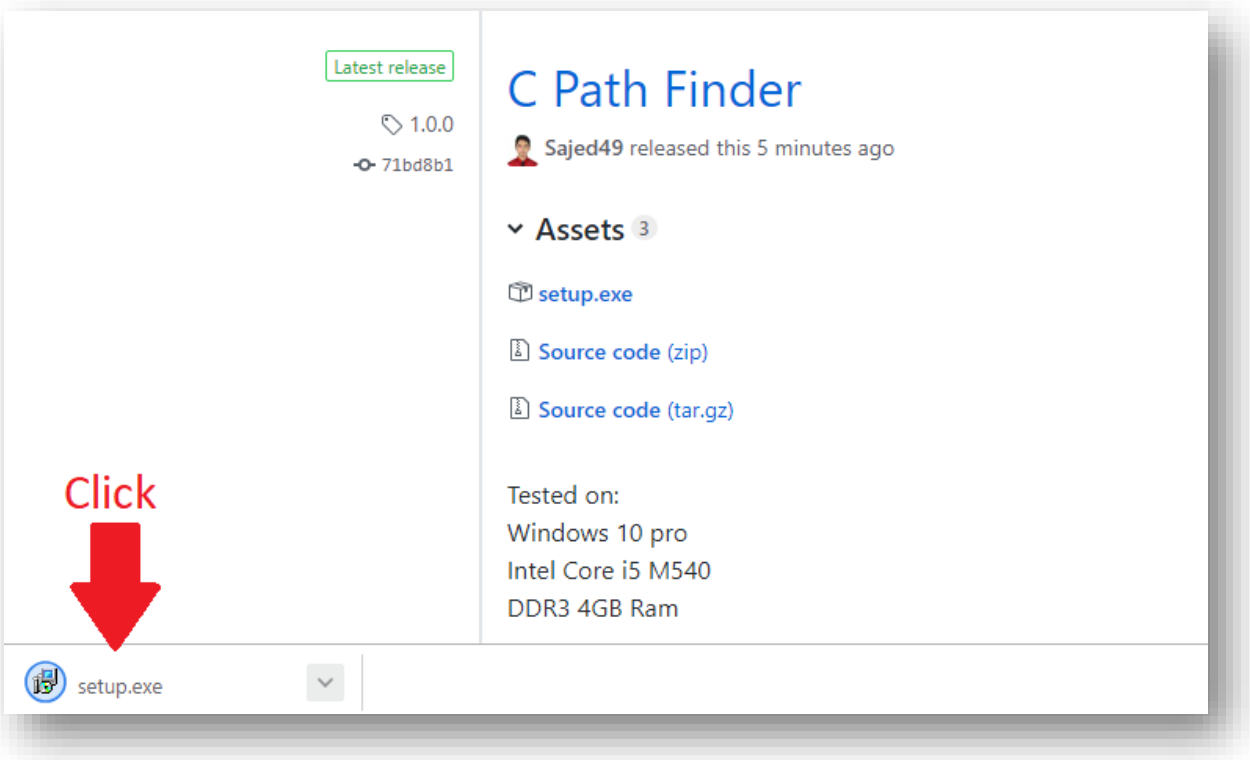

Figure 2: Executable File

Installing the application

In the following, step by step installation process are shown with figure. The user can customize the installation path as provided in our requirement. Moreover, User can choose whether to create a desktop shortcut icon.

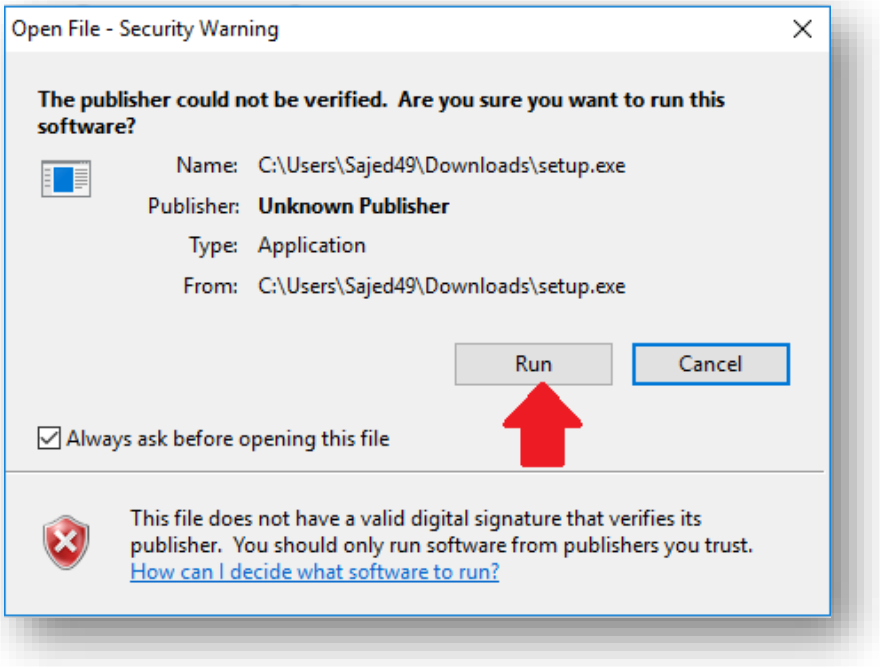

Figure 3: Installation permission 1

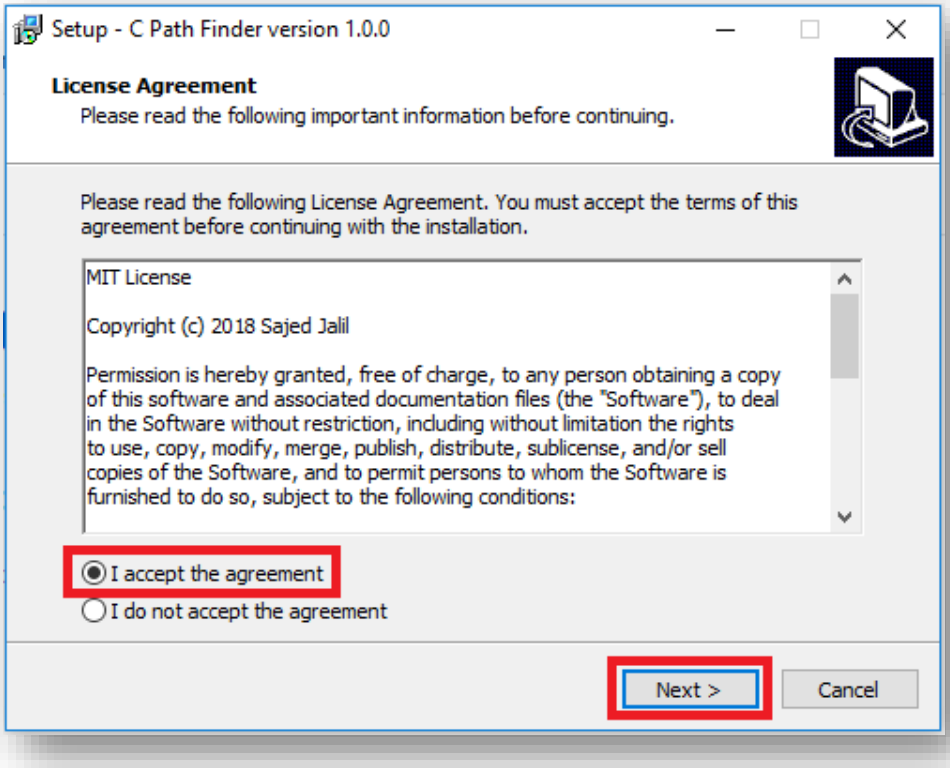

Figure 4: License Agreement

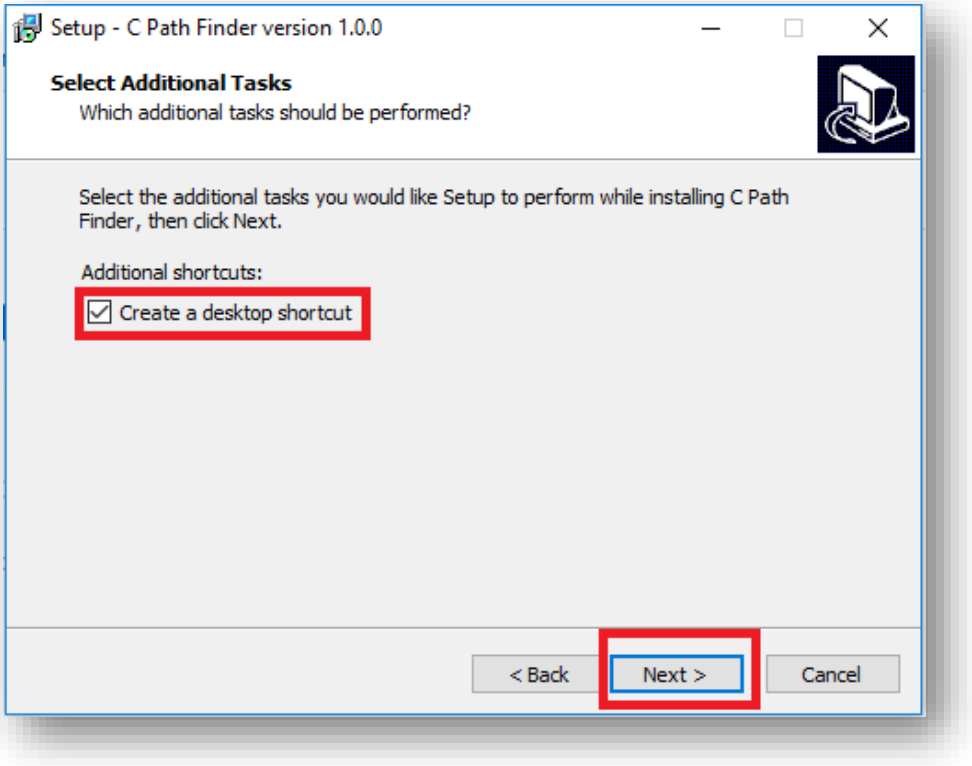

Figure 5: Desktop Shortcut

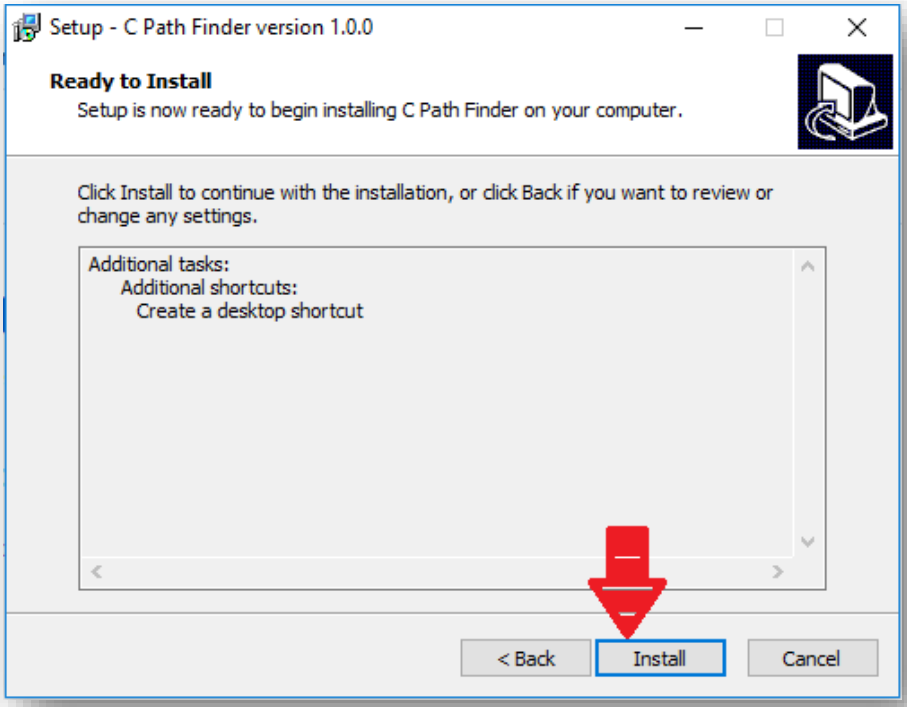

Figure 6: Install button

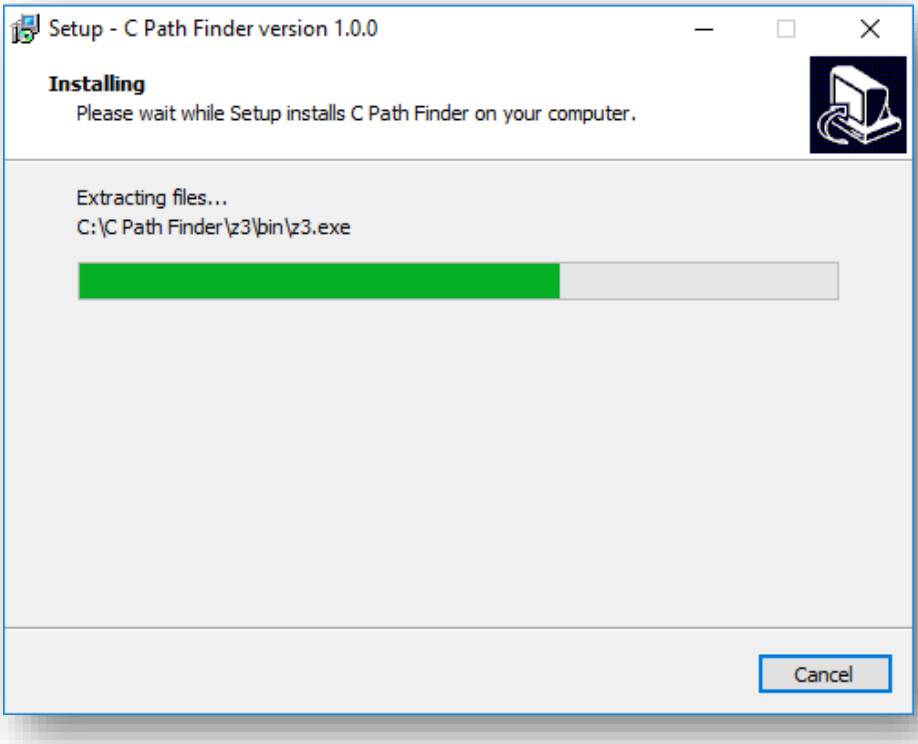

Figure 7: Installation on progress

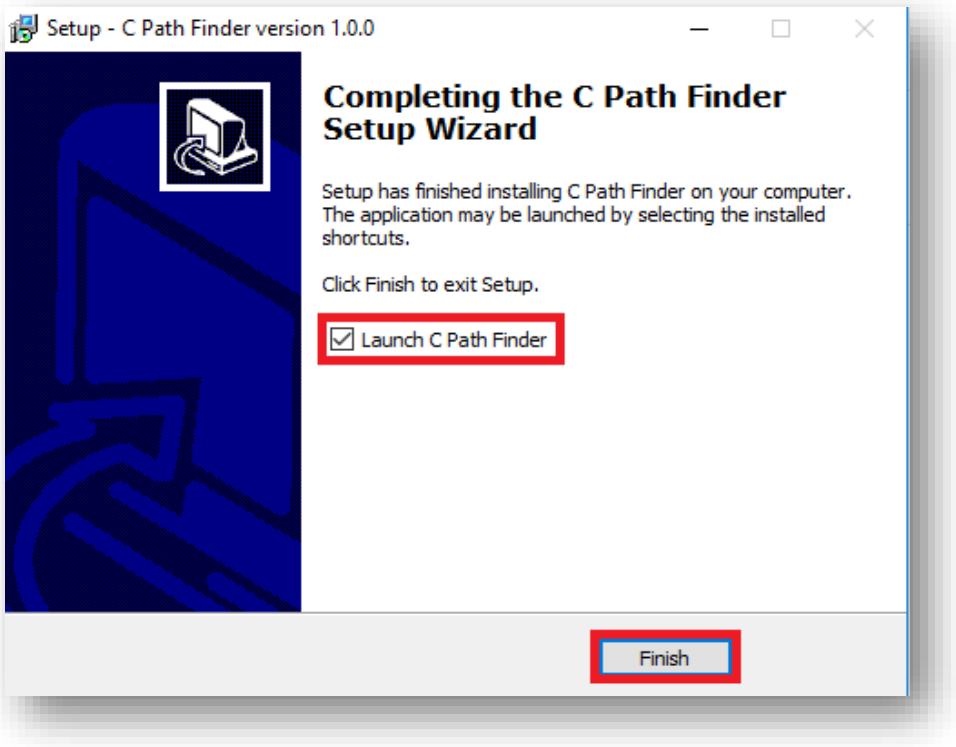

Figure 8: Installation completed

Running the Application

The application can be launched from desktop shortcut or the installation directory.

Figure 9: Launching from shortcut icon

From the **File** option we can open system explorer. From the **Settings** we can change the result generation directory. **Tutorial** menu opens the user manual for the application. To import the test files, we simple have to click the **select** button.

After selection a system explorer view will pop up. Form there we have to select the input directory where C source codes are located. In our example, the folder is located in C drive named as **test**.

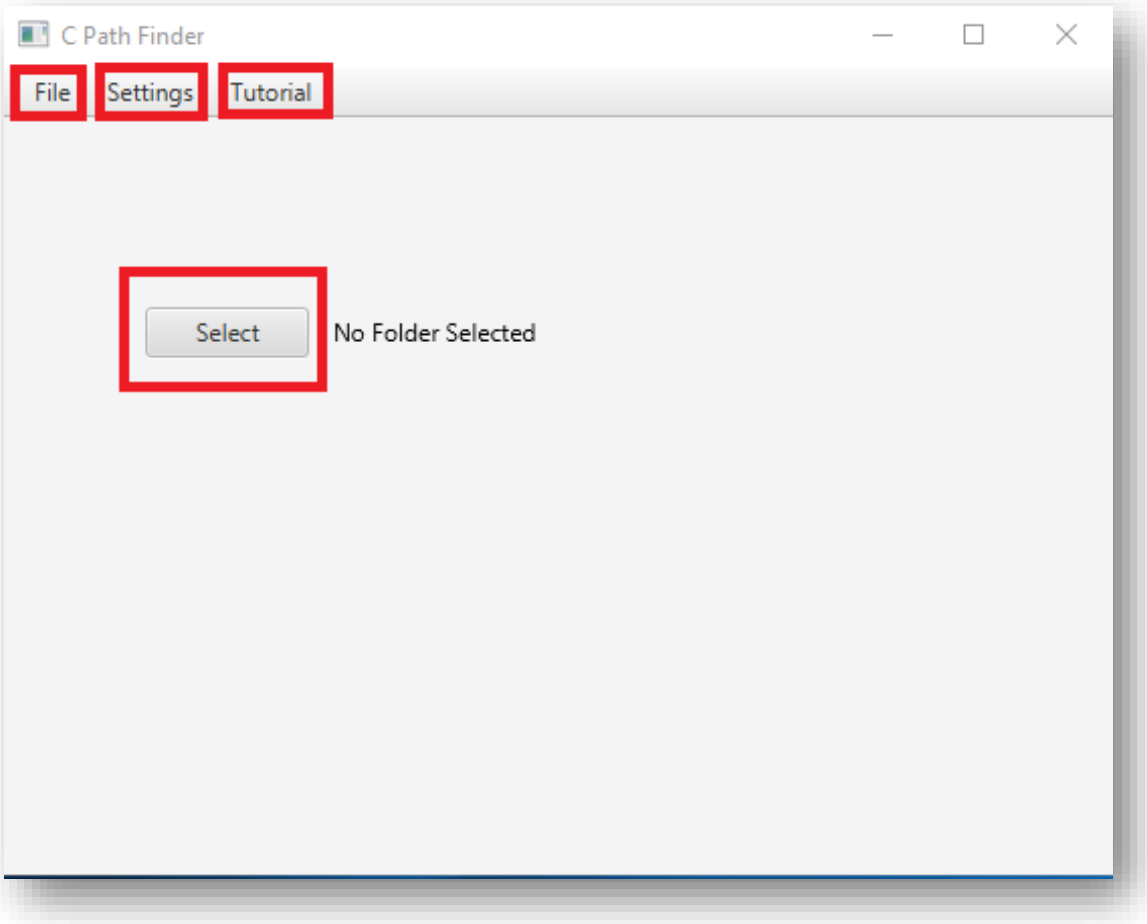

Figure 10: Application basics

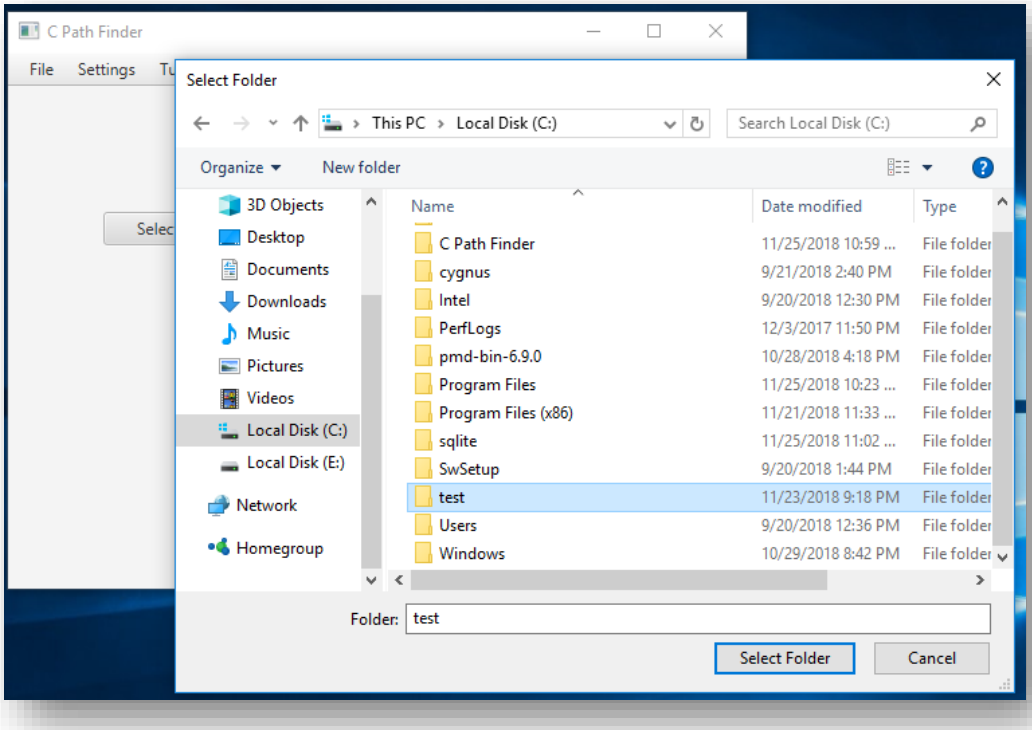

Figure 11: Source File Selection

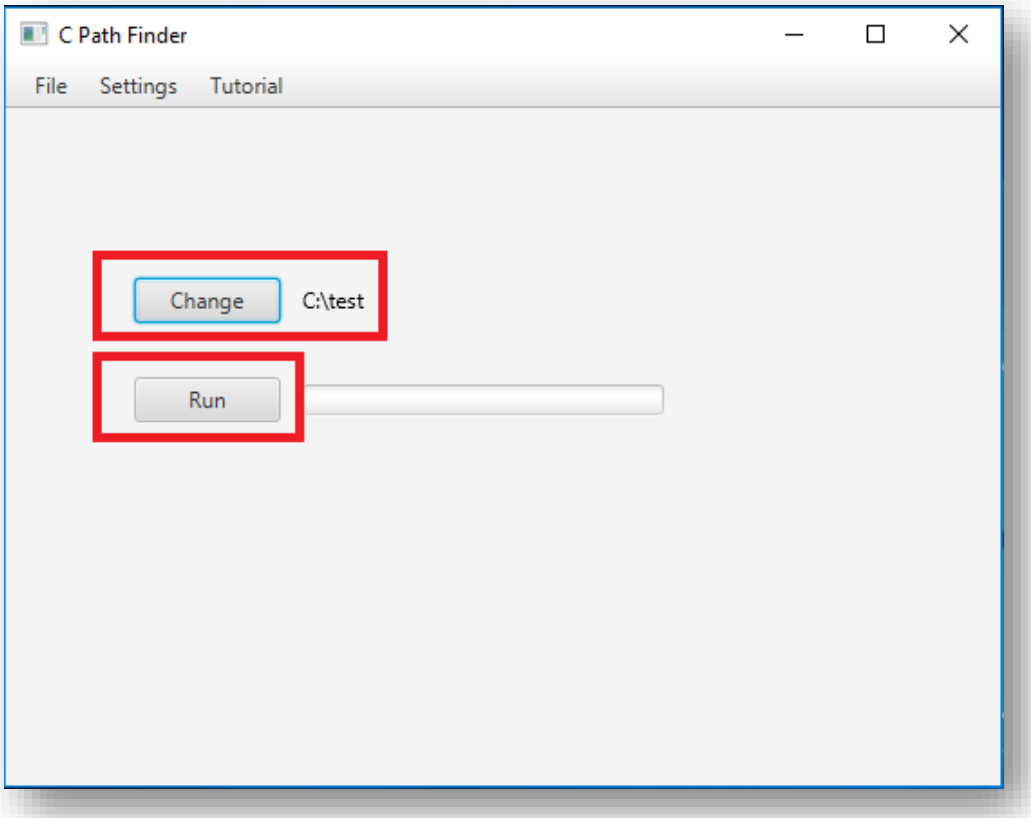

Figure 12: Running Source Files

After selection, we can change the test files or run the test files. When the processing is completed, the progress bar will be filled and a new button named open result will be shown. WE can re-run the source files. In that case, run will use **memorized execution** technique. **Open Result** button will pop up the result directory containing testcases in txt file format.

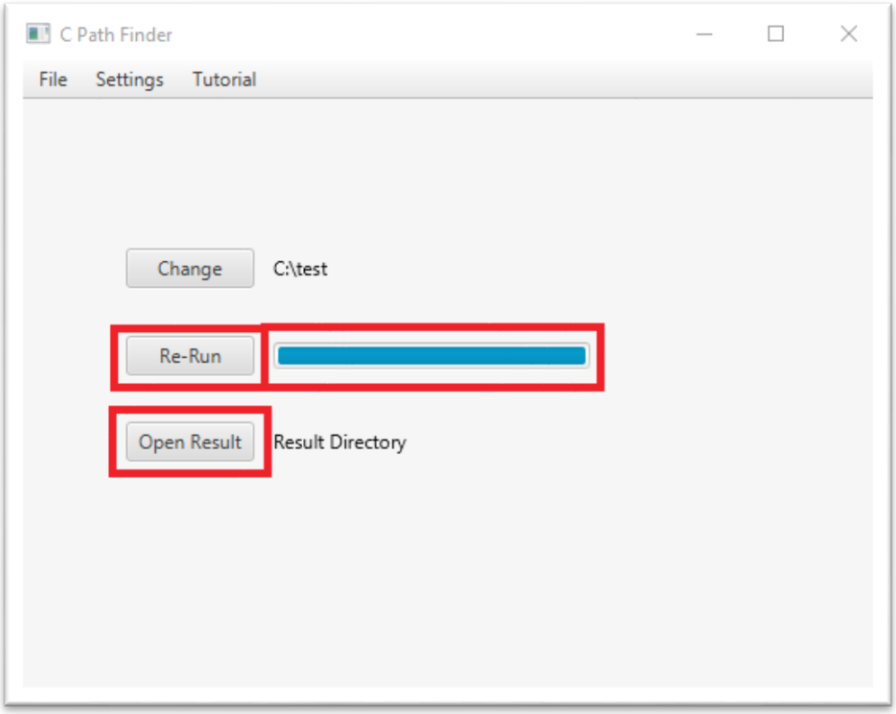

Figure 13: Progress Bar and Result

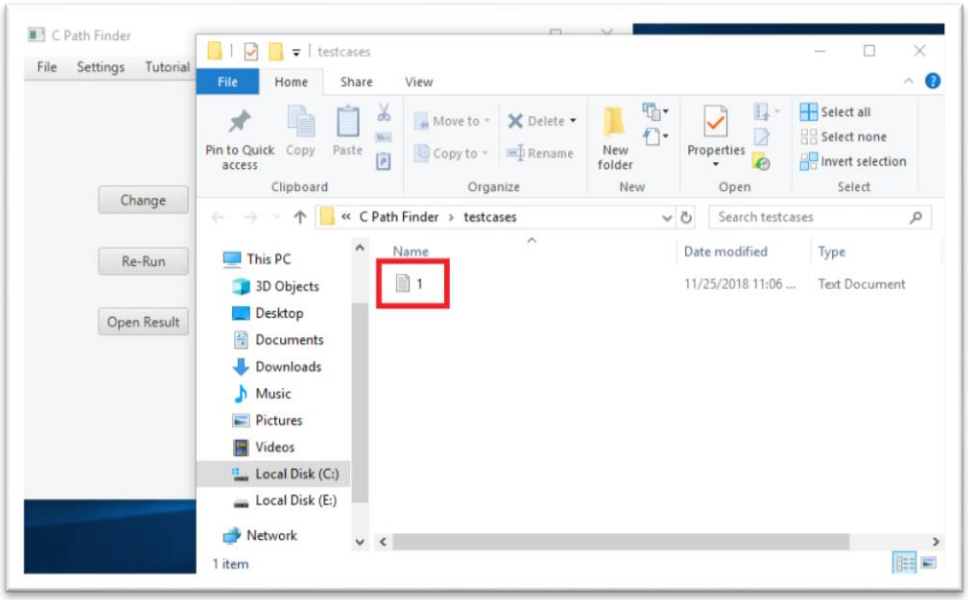

Figure 14: Result Output Directory

This figure below shows the sample test code for the file name 1.c . The source code had three possible paths. WE can see in the next result figure that our program could completely detect three paths and successfully generated test case that traverse these three paths.

```c
#include <stdio.h>

int main(void)
{
    int number1, number2;
    printf("Enter two integers: ");
    scanf("%d %d", &number1, &number2);

    if (number1 >= number2)
    {
      if (number1 == number2) printf("Result: %d = %d",number1,number2);
      else printf("Result: %d > %d", number1, number2);
    }
    Else printf("Result: %d < %d",number1, number2);
    return 0;
}
```

Figure 15: Input Test Code

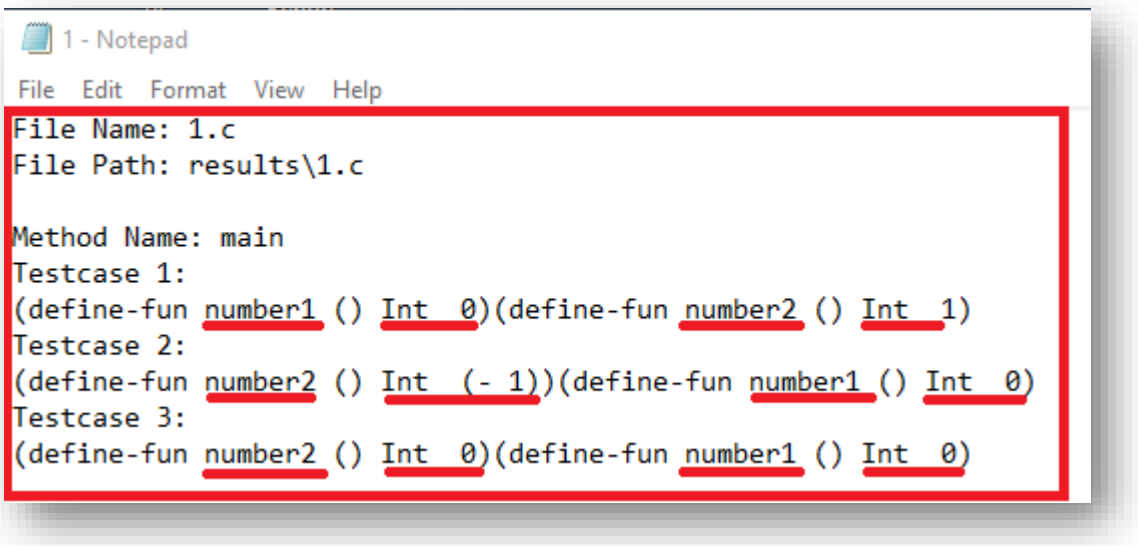

Figure 16: Generated Test Cases

Uninstalling the Application

Besides interactive installation feature, we have also added uninstallation feature in our application. The following shows the steps of uninstallation process.

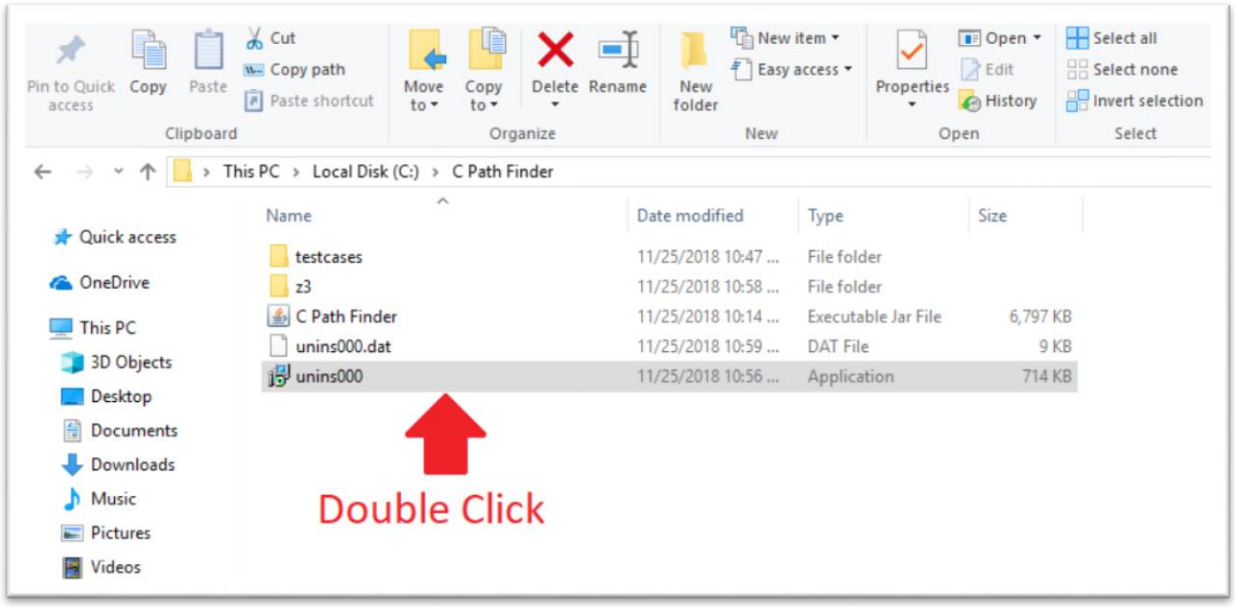

Figure 17: Uninstallation Steps

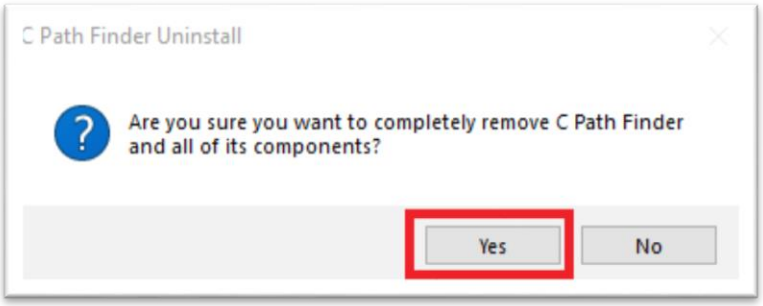

Figure 18: Confirm Uninstallation

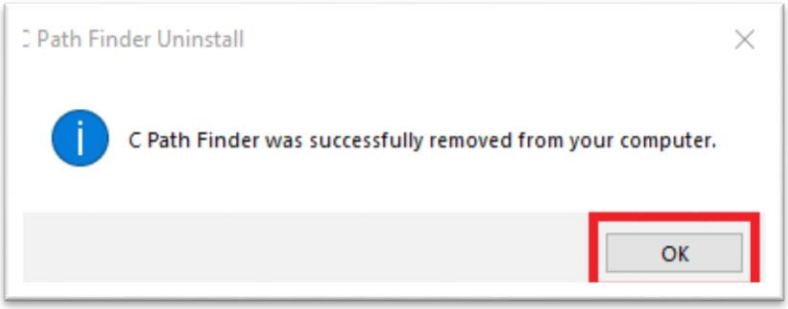

Figure 19: Uninstallation Completed

Troubleshoot

1. For any test generation related problems, please uninstall the application and install again.
2. Logs can be viewed for deeper insight.
3. For further problems, please raise an issue in GitHub repository or mail us at bsse0714@iit.du.ac.bd