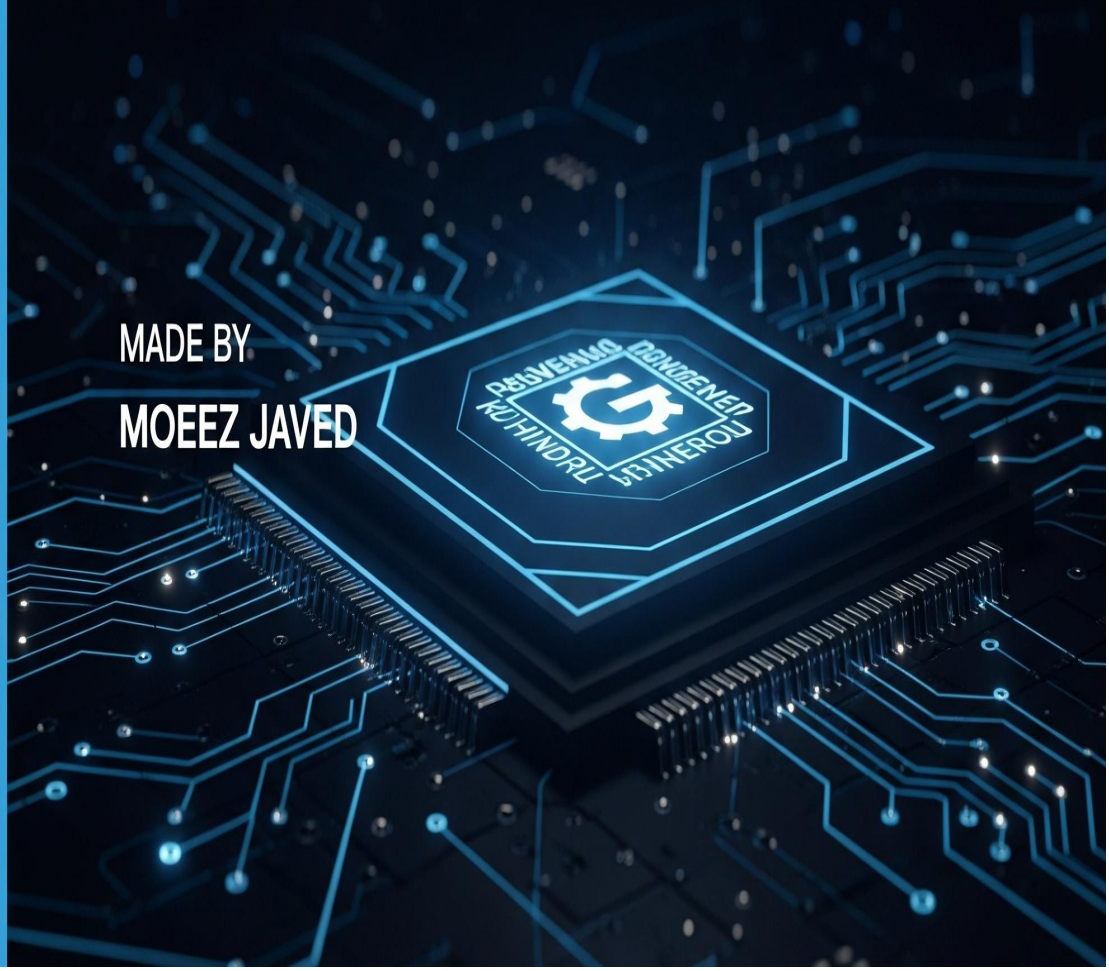
GHIDRA

REVERSE ENGINEERING TOOL

GHIDRA MANUAL



MADE BY

MOEEZ JAVED

Ghidra Manual(Step-by-Step)

Scope & ethics — Use Ghidra only on software and firmware you own or have explicit permission to analyze. Reverse engineering can be restricted by law or license. This manual is for defensible, educational use in a cybersecurity course.

1) What is Ghidra & why it matters

Ghidra is a free, open-source reverse-engineering (RE) suite from the NSA. It supports Windows, Linux, macOS; PE/ELF/Mach-O formats; x86/x64/ARM/MIPS/RISC-V and many others. It provides:

- Disassembly and a high-level **Decompiler** to read C-like pseudocode.
- Interactive program database (symbols, types, comments) that you can save/share.
- Static analysis (function discovery, string/constant finding, cross-refs, call graphs).
- Binary patching and diffing/version tracking.
- **Scripting** and **headless** (CLI) automation for repeatable labs.

Why students should learn it - Understand how compiled code actually executes. - Perform safe, *static* analysis of suspicious binaries and CTF challenges. - Audit closed-source software components in a SOC/DFIR workflow. - Document findings with reproducible projects and scripts.

Learning outcomes - Import a binary → run Auto-Analysis → navigate code/data. - Recover function prototypes, rename symbols, apply types. - Trace cross-references to understand behavior. - Use the Decompiler effectively and annotate clearly. - Automate tasks with scripts and the **analyzeHeadless** CLI. - (Optional) Patch small changes and export.

2) Prerequisites

- A lab machine (Windows, Linux, or macOS) with **Java 17+** installed.
- Sample binaries you are allowed to analyze (course pack or your own test builds).
- Basic C/C++ reading skills help, but aren't required.

3) Installation (step-by-step)

A) Install in Kali Linux (beginner-friendly guide)

Kali Linux does not ship with Ghidra pre-installed, but the setup is simple.

1. **Update your system**

   sudo apt update ***&&*** *sudo apt upgrade -y*

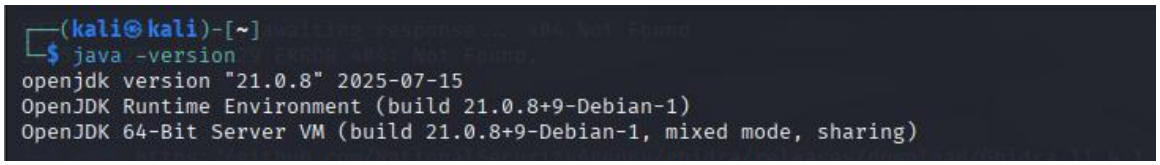
   This ensures your packages are up to date.

2. **Install Java 17 (required runtime)**

   sudo apt install openjdk-21-jdk -y

   Confirm installation:

   java -version



   You should see something like openjdk version "17.x".

3. *Download Ghidra Go to the official site: https://ghidra-sre.org*

- Download the latest release (a .zip file).

In Kali, you can also use wget to download directly:

wget https://github.com/NationalSecurityAgency/ghidra/releases/download/Ghidra_11.4.1_build/ghidra_11.4.1_PUBLIC_20250731.zip

```
┌──(kali㉿kali)-[~]
└─$ wget https://github.com/NationalSecurityAgency/ghidra/releases/download/Ghidra_11.4.1_build/ghidra_11.4.1_PUBLIC_20250731.zip

--2025-08-25 00:31:56--  https://github.com/NationalSecurityAgency/ghidra/releases/download/Ghidra_11.4.1_build/ghidra_11.4.1_PUBLIC_20250731.zip
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://release-assets.githubusercontent.com/github-production-release-asset/173228436/512b7e23-91d7-4586-95d9-445c6bd0f546?sp=r&sv=2018-11-09&sr=b&spr=http
s&se=2025-08-25T05%3A19%3A58Z&rscd=attachment%3B+filename%3Dghidra_11.4.1_PUBLIC_20250731.zip&rsct=application%2Foctet-stream&skoid=96c2d410-5711-43a1-aedd-ab1947aa7a
b0&sktid=398a6654-997b-47e9-b12b-9515b896b4de&skt=2025-08-25T04%3A19%3A18Z&ske=2025-08-25T05%3A19%3A58Z&sks=b&skv=2018-11-09&sig=jxp%2BPM%2BcjMZH4BgEBOteQ55qLuWAyRZ6a
WyyZgG72RY%3D&jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmVsZWFzZS1hc3NldHMuZ2l0aHVidXNlcmNvbnRlbnQuY29tIiwia2V5Ijoia2V5MSIsImV4cCI
6MTc1NjA5NjYxOCwibmJmIjoxNzU2MDk2MzE4LCJwYXRoIjoicmVsZWFzZWFzc2V0cHJvZHVjdGlvbi5ibG9iLmNvcmUud2luZG93cy5uZXQifQ.JkpQuf_WxOc_hzGLROETnmnJKOzT96SFjuw974-7A9c&response-c
ontent-disposition=attachment%3B%20filename%3Dghidra_11.4.1_PUBLIC_20250731.zip&response-content-type=application%2Foctet-stream [following]
--2025-08-25 00:31:57--  https://release-assets.githubusercontent.com/github-production-release-asset/173228436/512b7e23-91d7-4586-95d9-445c6bd0f546?sp=r&sv=2018-11-0
9&sr=b&spr=https&se=2025-08-25T05%3A19%3A58Z&rscd=attachment%3B+filename%3Dghidra_11.4.1_PUBLIC_20250731.zip&rsct=application%2Foctet-stream&skoid=96c2d410-5711-43a1-
aedd-ab1947aa7ab0&sktid=398a6654-997b-47e9-b12b-9515b896b4de&skt=2025-08-25T04%3A19%3A18Z&ske=2025-08-25T05%3A19%3A58Z&sks=b&skv=2018-11-09&sig=jxp%2BPM%2BcjMZH4BgEBO
teQ55qLuWAyRZ6aWyyZgG72RY%3D&jwt=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmVsZWFzZS1hc3NldHMuZ2l0aHVidXNlcmNvbnRlbnQuY29tIiwia2V5Ijoi
a2V5MSIsImV4cCI6MTc1NjA5NjYxOCwibmJmIjoxNzU2MDk2MzE4LCJwYXRoIjoicmVsZWFzZWFzc2V0cHJvZHVjdGlvbi5ibG9iLmNvcmUud2luZG93cy5uZXQifQ.JkpQuf_WxOc_hzGLROETnmnJKOzT96SFjuw974-
7A9c&response-content-disposition=attachment%3B%20filename%3Dghidra_11.4.1_PUBLIC_20250731.zip&response-content-type=application%2Foctet-stream
Resolving release-assets.githubusercontent.com (release-assets.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.133, ...
Connecting to release-assets.githubusercontent.com (release-assets.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 456576486 (435M) [application/octet-stream]
Saving to: 'ghidra_11.4.1_PUBLIC_20250731.zip'

ghidra_11.4.1_PUBLIC_20250731.zip          94%[===================================================================>      ] 409.39M  2.97MB/s    eta 9s
ghidra_11.4.1_PUBLIC_20250731.zip         100%[===================================================================>] 435.42M  3.12MB/s    in 2m 24s

2025-08-25 00:34:34 (3.03 MB/s) - 'ghidra_11.4.1_PUBLIC_20250731.zip' saved [456576486/456576486]
```

4. *Extract the archive*

unzip ghidra_11.4.1_PUBLIC_20250731.zip -d ~/tools/

```
┌──(kali㉿kali)-[~]
└─$ unzip ghidra_11.4.1_PUBLIC_20250731.zip -d ~/tools/
cd ~/tools/ghidra_11.4.1_PUBLIC/
./ghidraRun
Archive:  ghidra_11.4.1_PUBLIC_20250731.zip
   creating: /home/kali/tools/ghidra_11.4.1_PUBLIC/
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/ghidraRun.bat
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/ghidraRun
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/bom.json
   creating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/build-docker-image.sh
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/Dockerfile.dockerignore
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/Dockerfile
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/README.html
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/README.md
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/entrypoint.sh
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/LICENSE
   creating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Modified_Nuvola_Icons_-_LGPL_2.1.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Tango_Icons_-_Public_Domain.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/BSD-3-JUNG.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/zlib_License.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Nuvola_Icons_-_LGPL_2.1.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Oxygen_Icons_-_LGPL_3.0.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/LGPL_2.1.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Python_Software_Foundation_License.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/INRIA_License.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/BSD-3-ORACLE.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/GPL_2_With_Classpath_Exception.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Public_Domain.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Jython_License.txt
```

This creates a folder like ~/tools/ghidra_X.Y.Z_PUBLIC.

5. *Run Ghidra*

cd ~/tools/ghidra_11.4.1_PUBLIC/

```
┌──(kali㉿kali)-[~]
└─$ unzip ghidra_11.4.1_PUBLIC_20250731.zip -d ~/tools/
cd ~/tools/ghidra_11.4.1_PUBLIC/
./ghidraRun
Archive:  ghidra_11.4.1_PUBLIC_20250731.zip
   creating: /home/kali/tools/ghidra_11.4.1_PUBLIC/
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/ghidraRun.bat
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/ghidraRun
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/bom.json
   creating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/build-docker-image.sh
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/Dockerfile.dockerignore
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/Dockerfile
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/README.html
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/README.md
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/docker/entrypoint.sh
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/LICENSE
   creating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Modified_Nuvola_Icons_-_LGPL_2.1.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Tango_Icons_-_Public_Domain.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/BSD-3-JUNG.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/zlib_License.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Nuvola_Icons_-_LGPL_2.1.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Oxygen_Icons_-_LGPL_3.0.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/LGPL_2.1.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Python_Software_Foundation_License.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/INRIA_License.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/BSD-3-ORACLE.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/GPL_2_With_Classpath_Exception.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Public_Domain.txt
  inflating: /home/kali/tools/ghidra_11.4.1_PUBLIC/licenses/Jython_License.txt
```

./ghidraRun

```
┌──(kali㉿kali)-[~/tools/ghidra_11.4.1_PUBLIC]
└─$ ./ghidraRun
```

If Java is installed properly, Ghidra will start and show the Project Manager window.

6. ***(Optional) Create a shortcut/alias Add this line to your ~/.bashrc:***

   alias ghidra='~/tools/ghidra_X.Y.Z_PUBLIC/ghidraRun'

   Then reload:

   source ~/.bashrc

   Now you can launch Ghidra anywhere by typing:

   ghidra

Setup complete! You're ready to create your first project.

B) Installation on other platforms (summary)

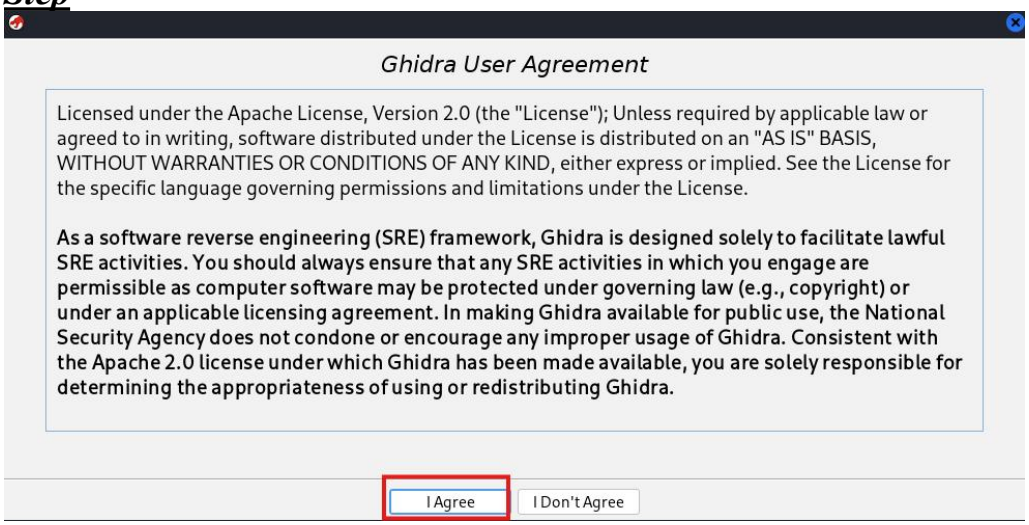
Windows 1. Download the latest Ghidra zip. 2. Extract to C:\Tools\ghidra. 3. Run ghidraRun.bat.

macOS/Linux (generic) 1. Extract the zip. 2. Launch with ./ghidraRun.

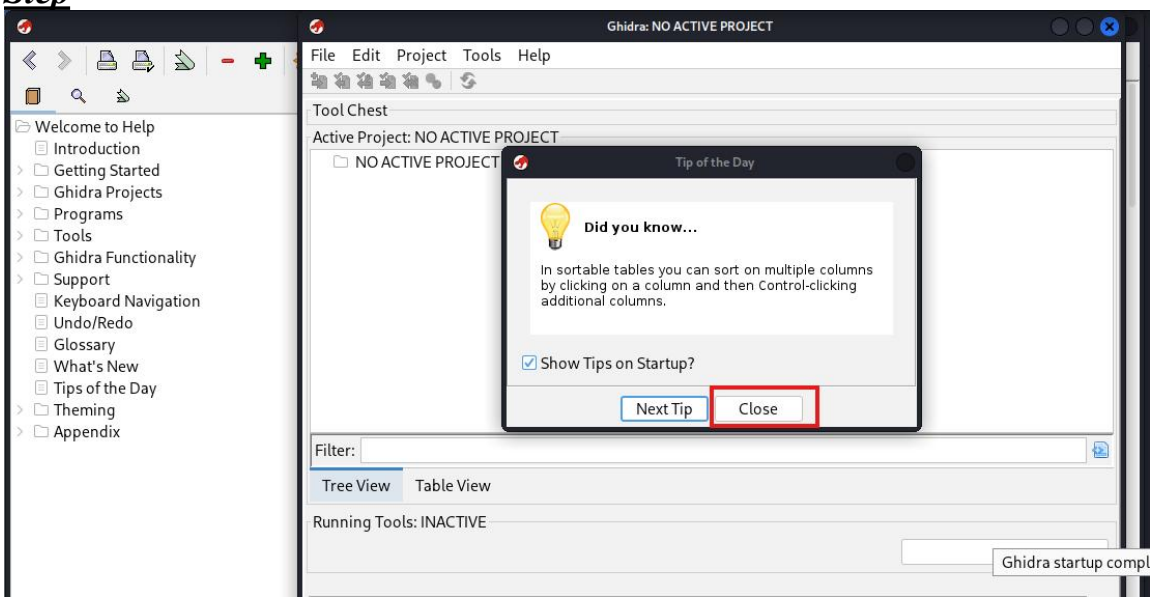
4) First project (GUI quick start)

Goal: Import a binary and run analysis to produce readable functions and pseudocode.

Step

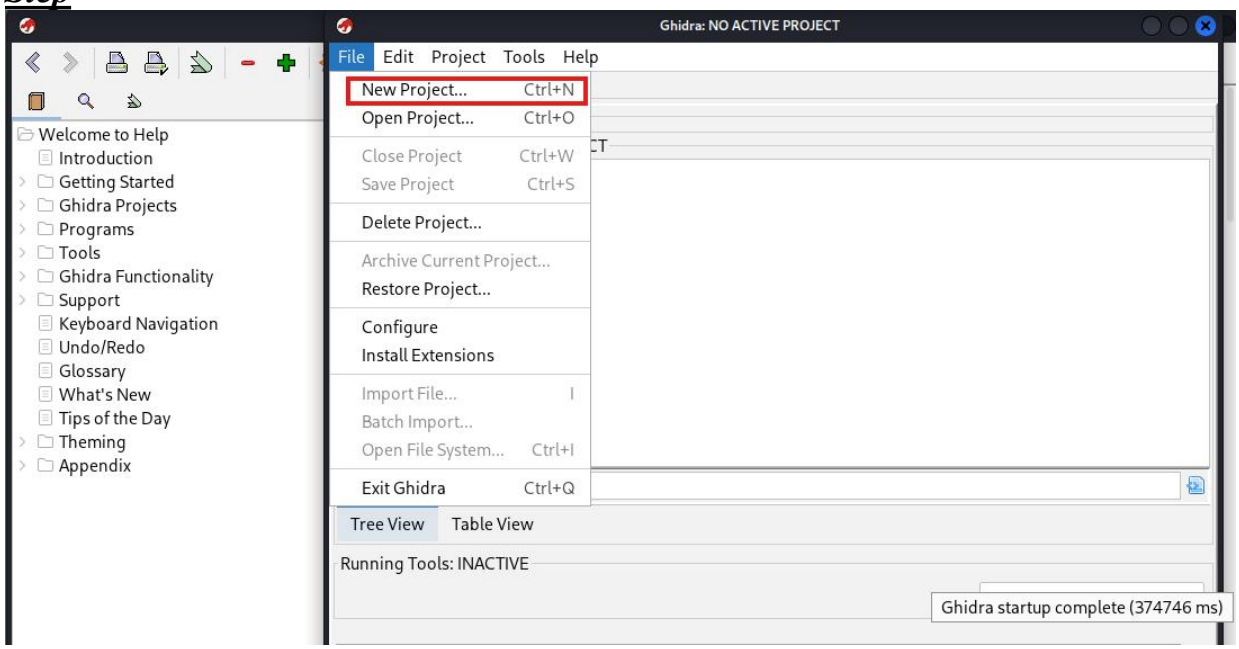


Step



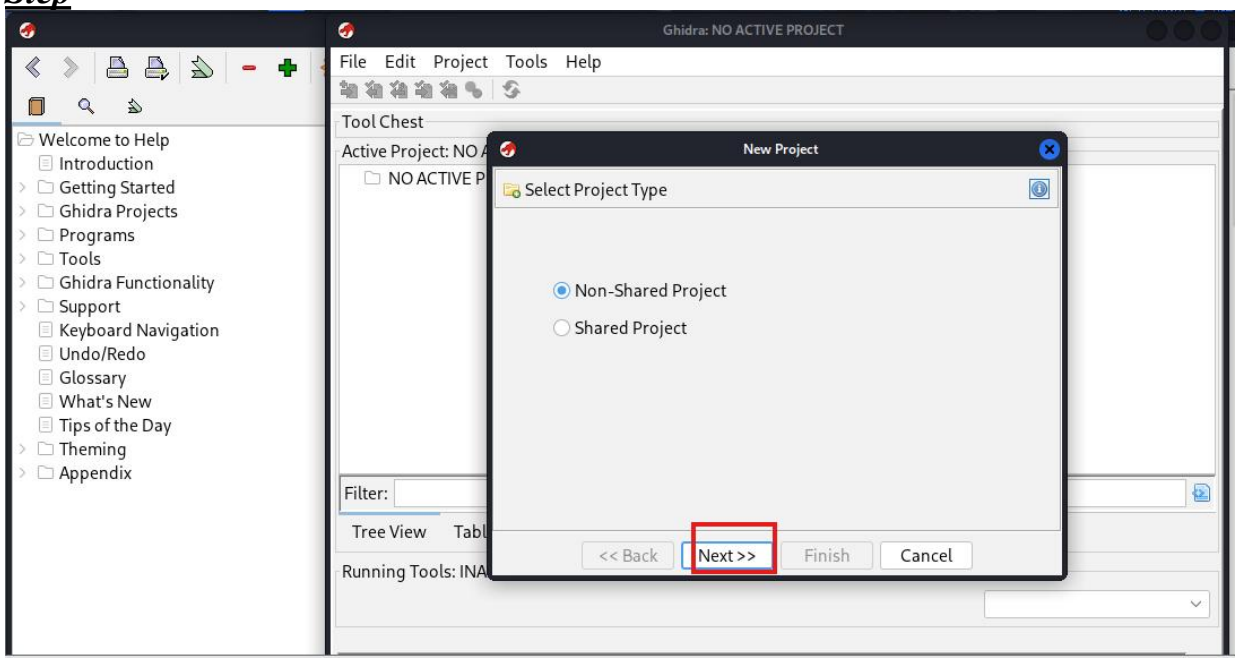
1. ***Open Ghidra → File → New Project.***

Step



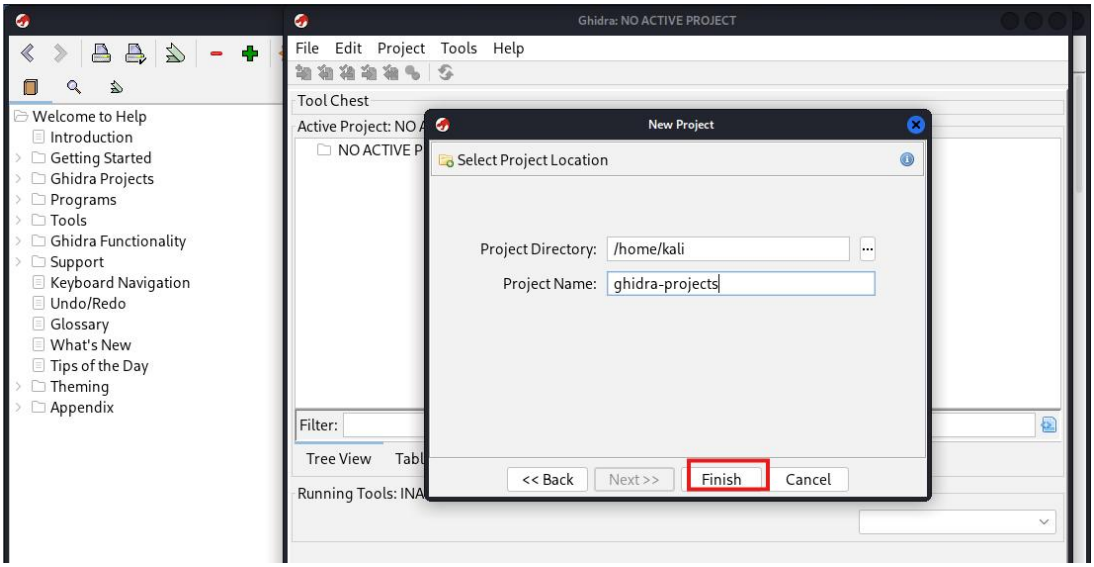
2. ***Choose Non-Shared Project → Next.***

Step



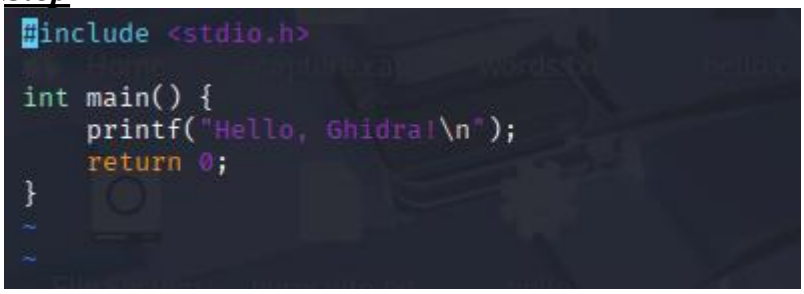
3. ***Choose a Project Directory (e.g., ~/ghidra-projects) and a Project Name (e.g., IntroLab).***

Step



4. ***Make a File***

Step



```
#include <stdio.h>

int main() {
    printf("Hello, Ghidra!\n");
    return 0;
}
~
~
```

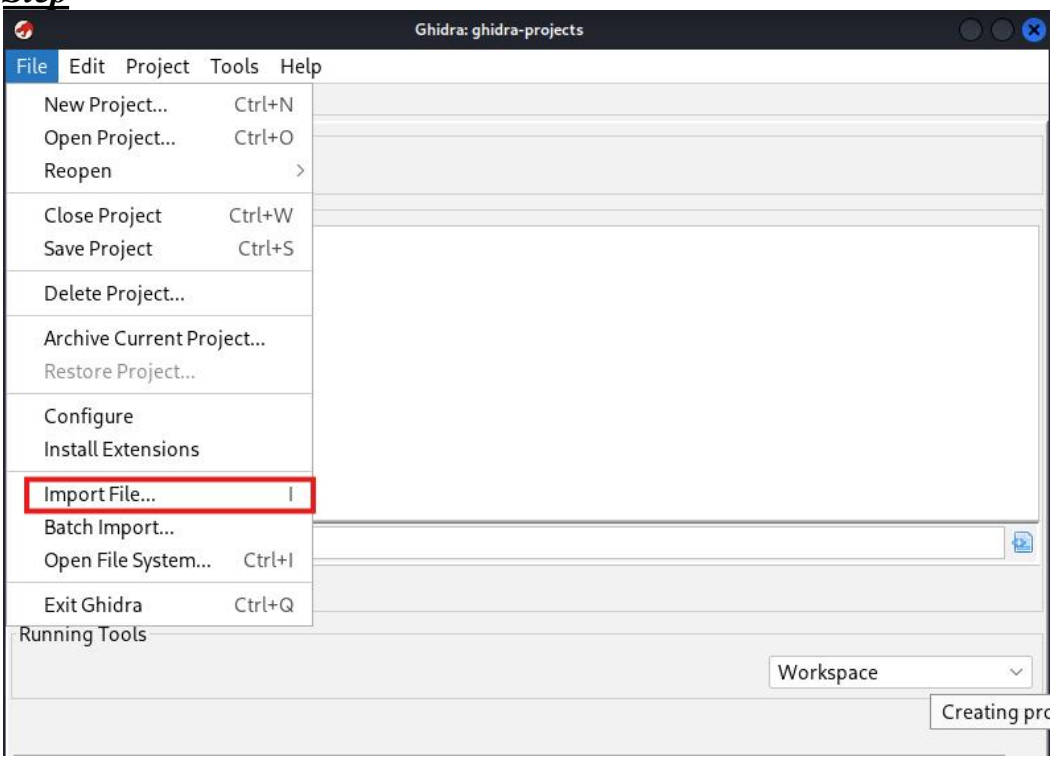
cd Desktop
gcc hello.c -o hello
./hello

Step

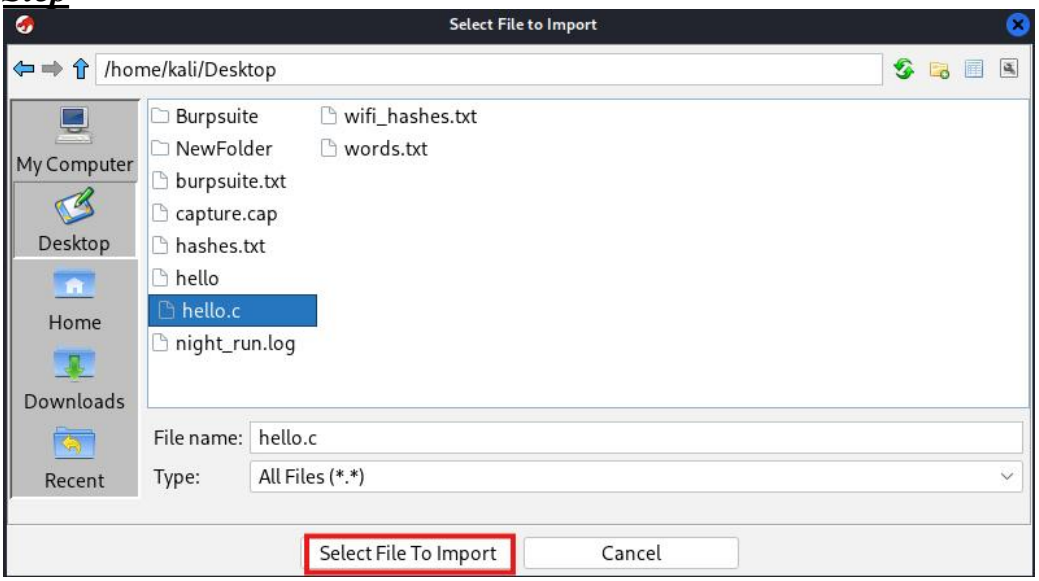
```
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ gcc hello.c -o hello


┌──(kali㉿kali)-[~/Desktop]
└─$ ./hello

Hello, Ghidra!

┌──(kali㉿kali)-[~/Desktop]
└─$
```

5. ***File → Import File… → pick your sample binary (e.g., hello or sample.exe).***

Step

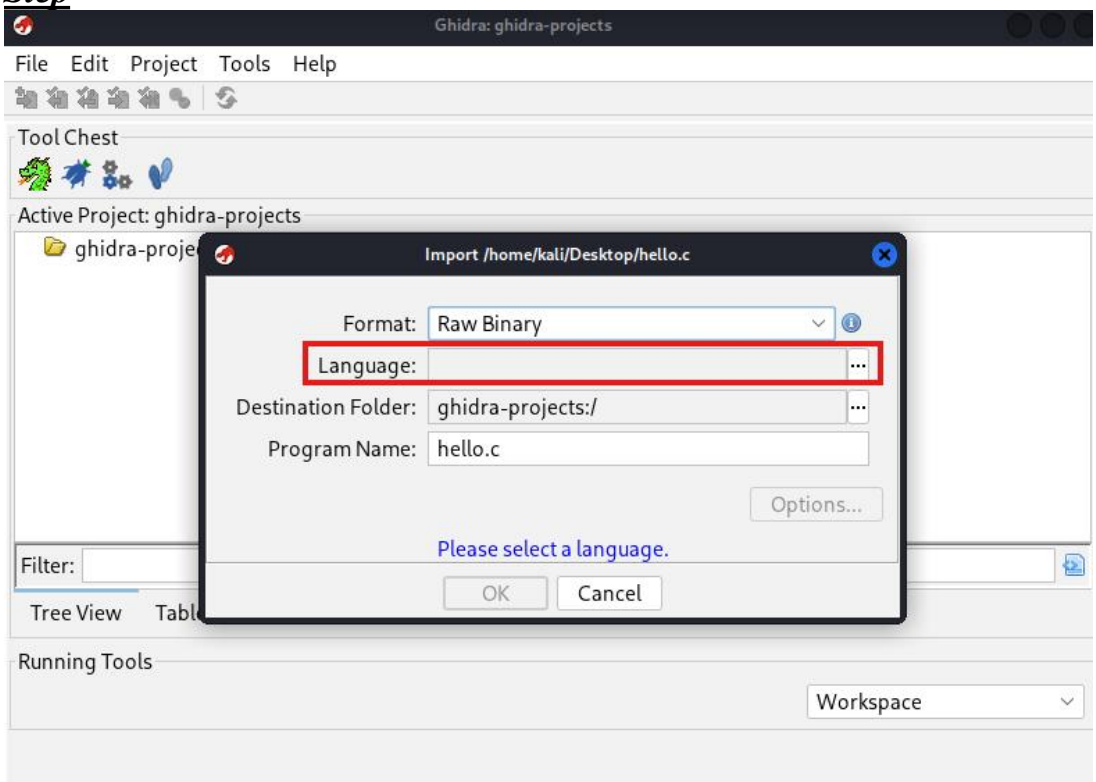


Step

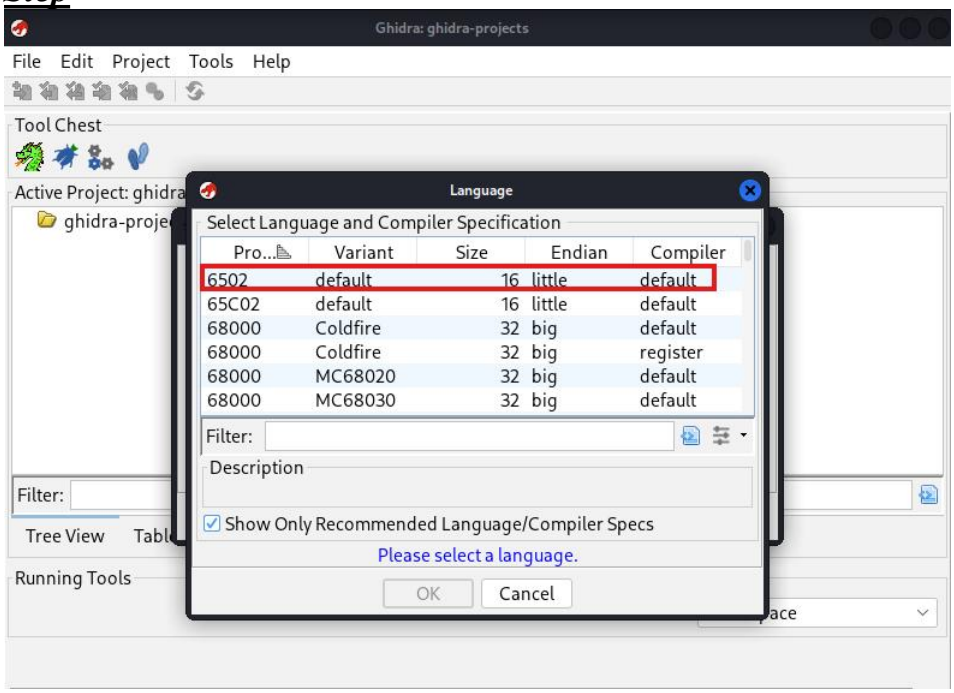


6. ***In the Import dialog, confirm the Format and Language (Ghidra will usually detect these). Click OK.***

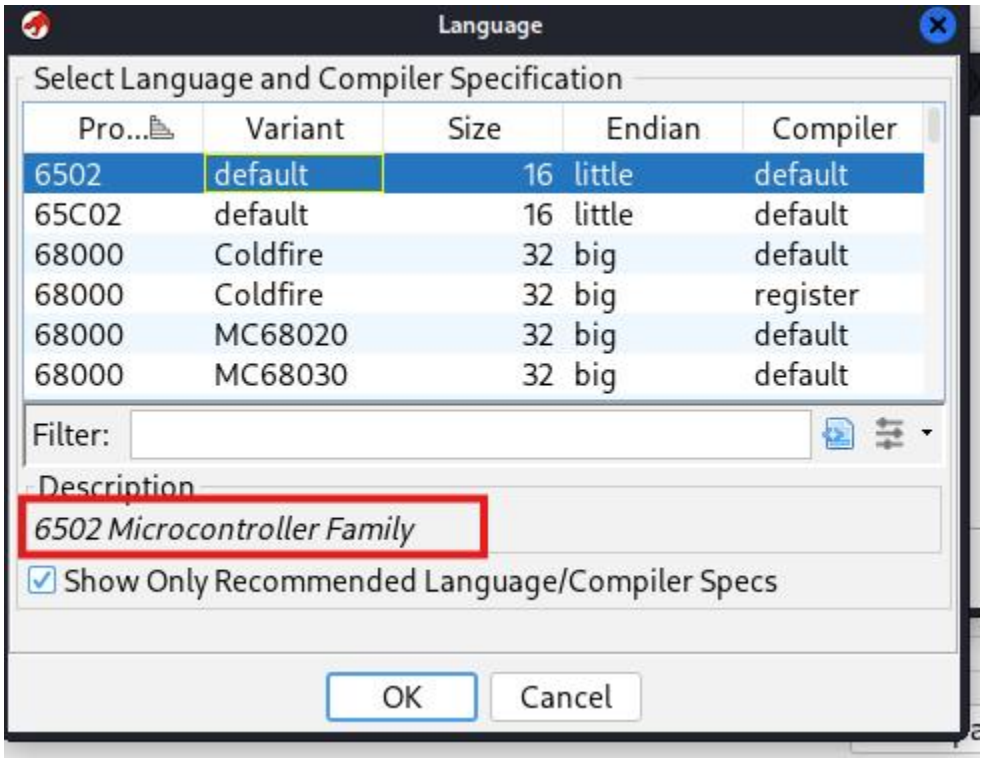
Step



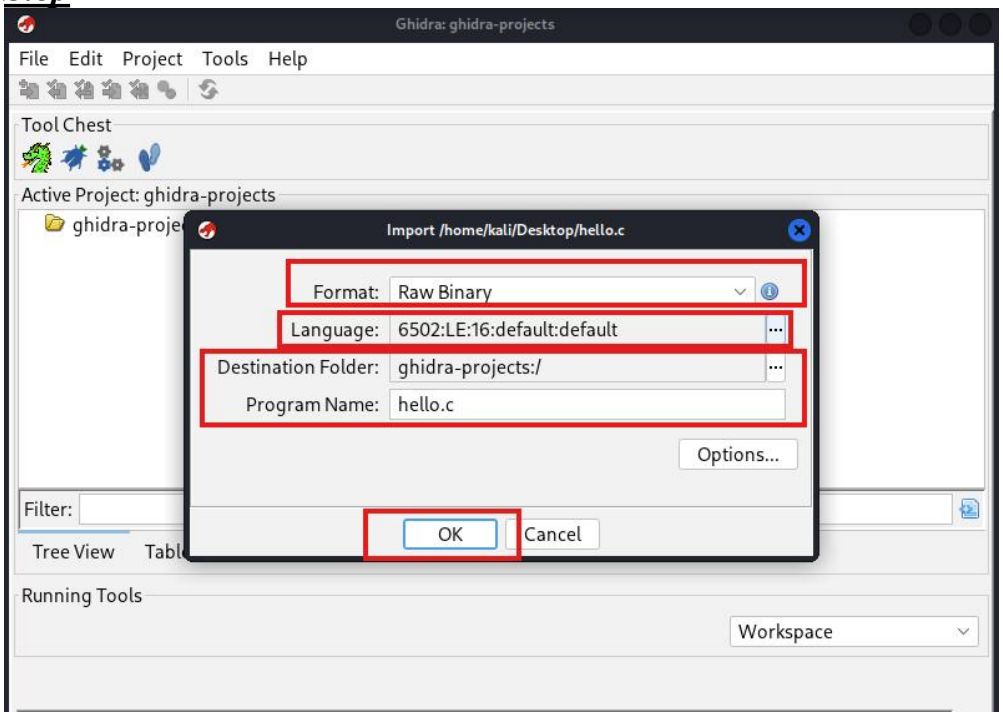
Step



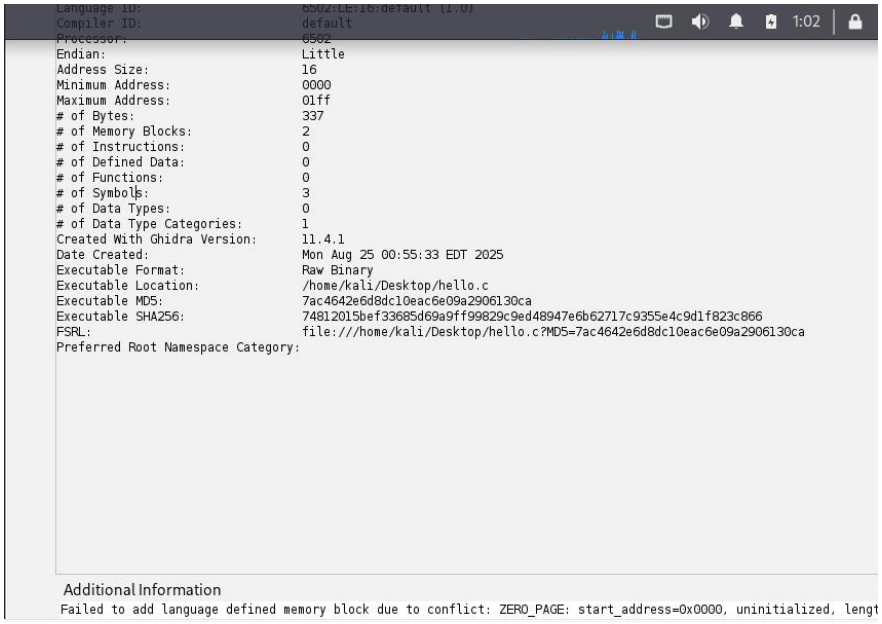
Step


Language
Select Language and Compiler Specification
Pro... Variant Size Endian Compiler
6502 default 16 little default
65C02 default 16 little default
68000 Coldfire 32 big default
68000 Coldfire 32 big register
68000 MC68020 32 big default
68000 MC68030 32 big default
Filter:
Description
6502 Microcontroller Family
Show Only Recommended Language/Compiler Specs
OK
Cancel

Step

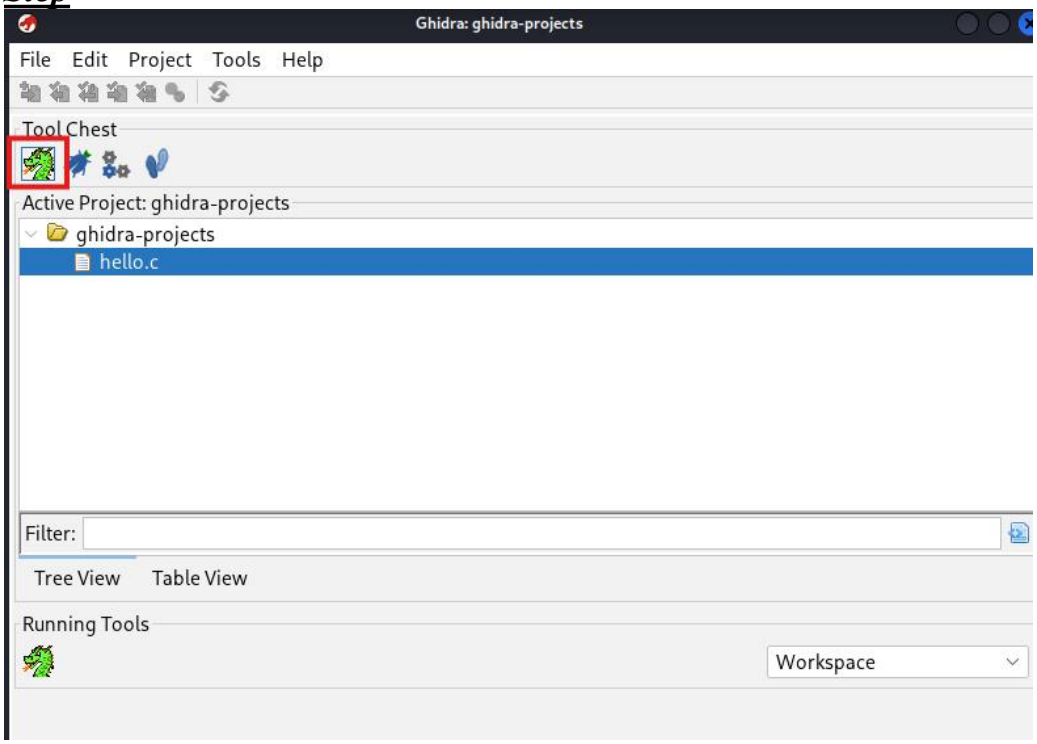

Ghidra: ghidra-projects
File Edit Project Tools Help
Tool Chest
Active Project: ghidra-projects
ghidra-proje
Import /home/kali/Desktop/hello.c
Format: Raw Binary
Language: 6502:LE:16:default:default
Destination Folder: ghidra-projects:/
Program Name: hello.c
Options...
OK
Cancel
Filter:
Tree View
Running Tools
Workspace

Step

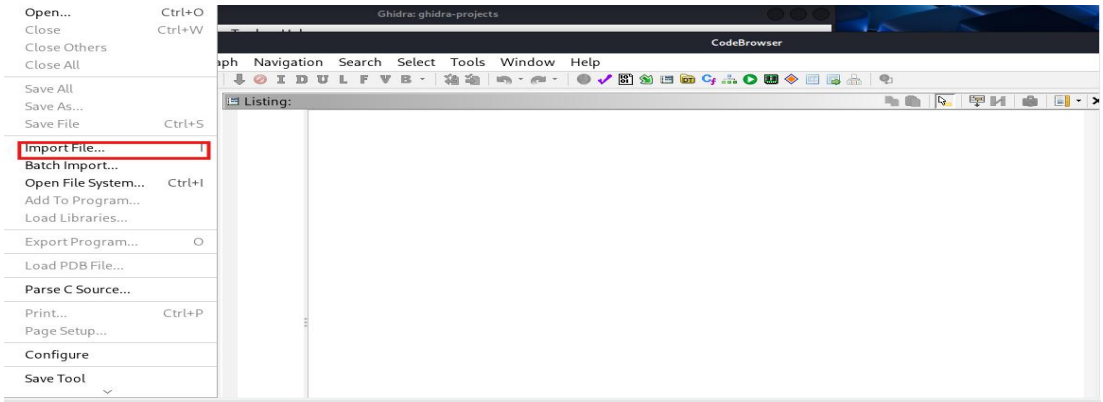


7. ***In the Project window, double-click the newly imported program to open it in the CodeBrowser.***

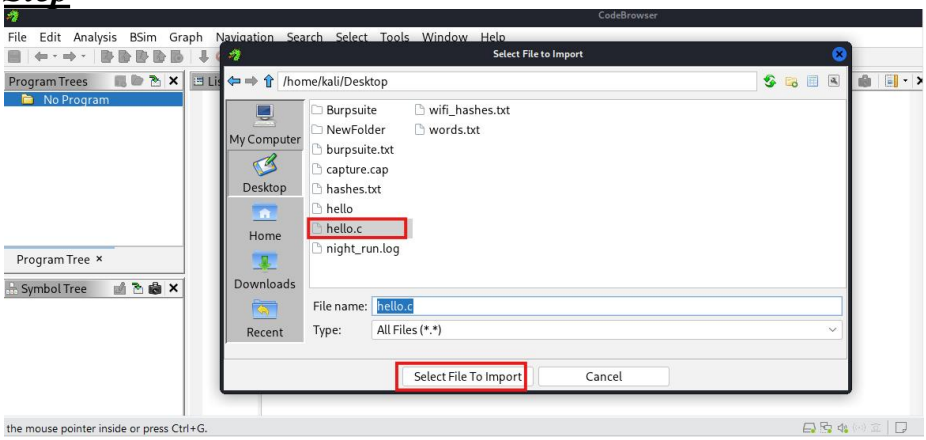
<u>Step</u>



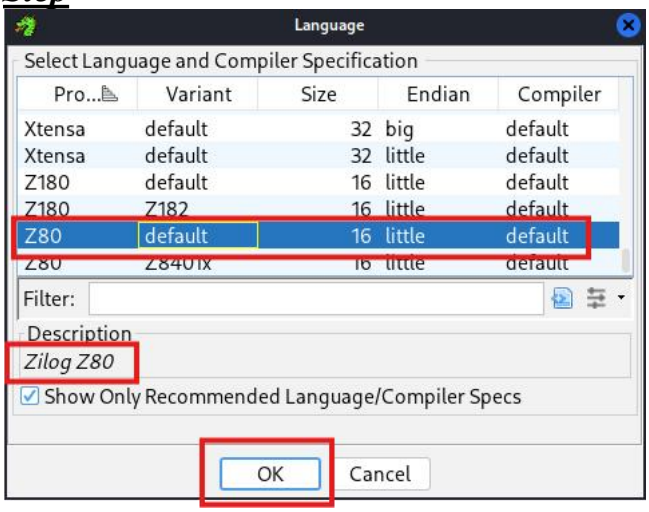
<u>Step</u>



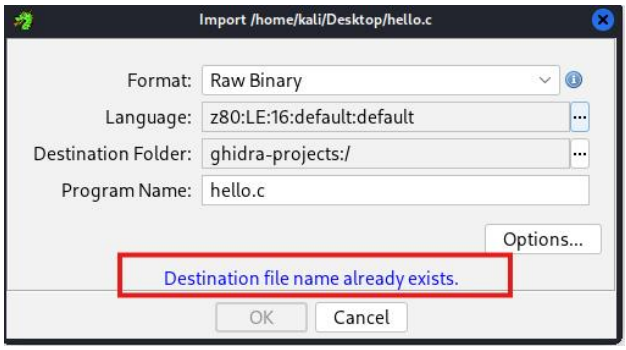
Step



Step



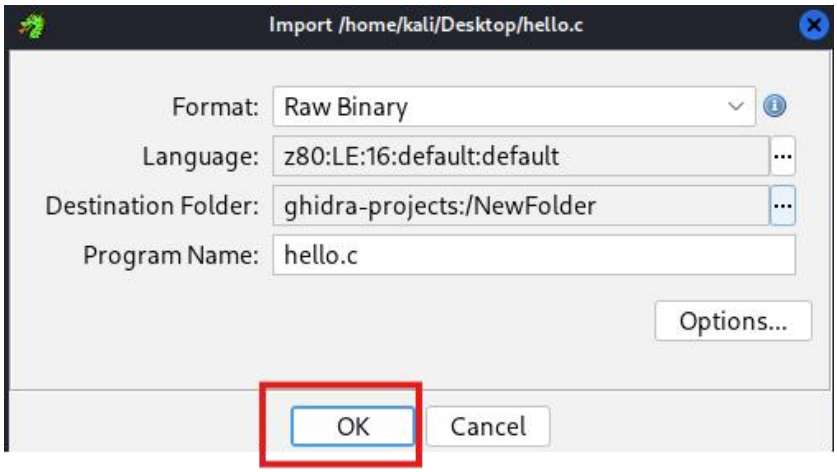
Step


Import /home/kali/Desktop/hello.c
Format:
Raw Binary
Language:
z80:LE:16:default:default
Destination Folder:
ghidra-projects:/
Program Name:
hello.c
Options...
Destination file name already exists.
OK
Cancel

Step

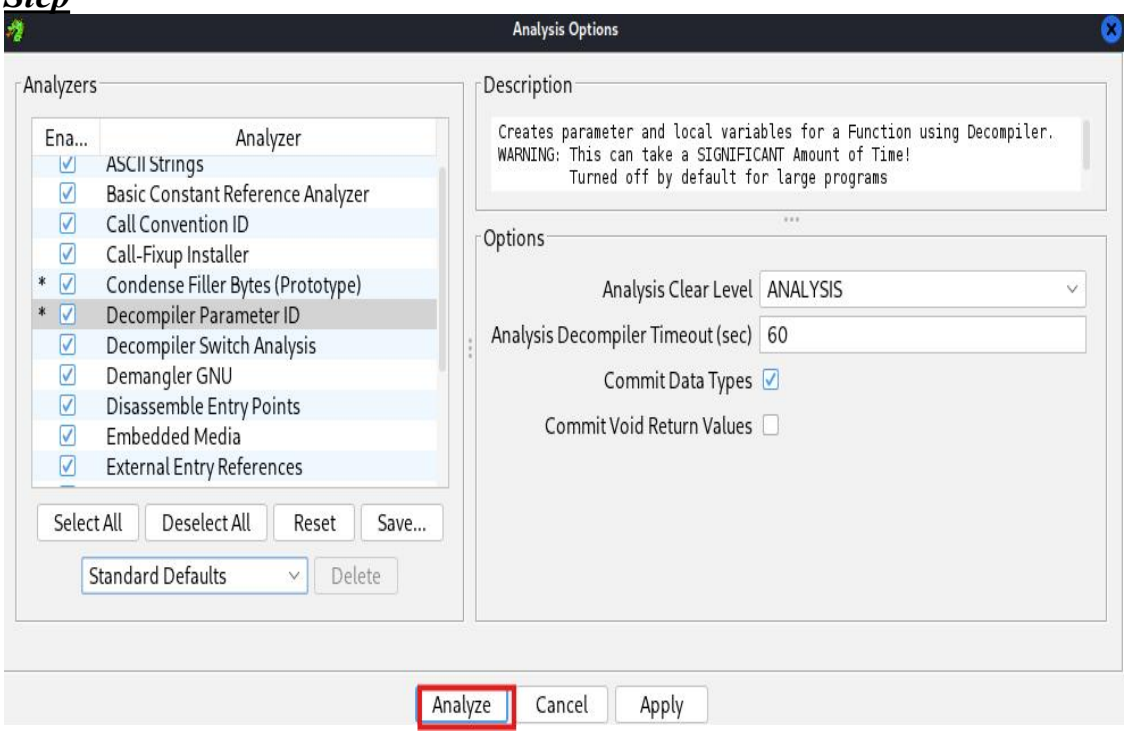

Choose a project folder
ghidra-projects
codebrowser
hello.c
Filter:
Folder Path: /
Folder Name:
/
OK
Cancel

Step

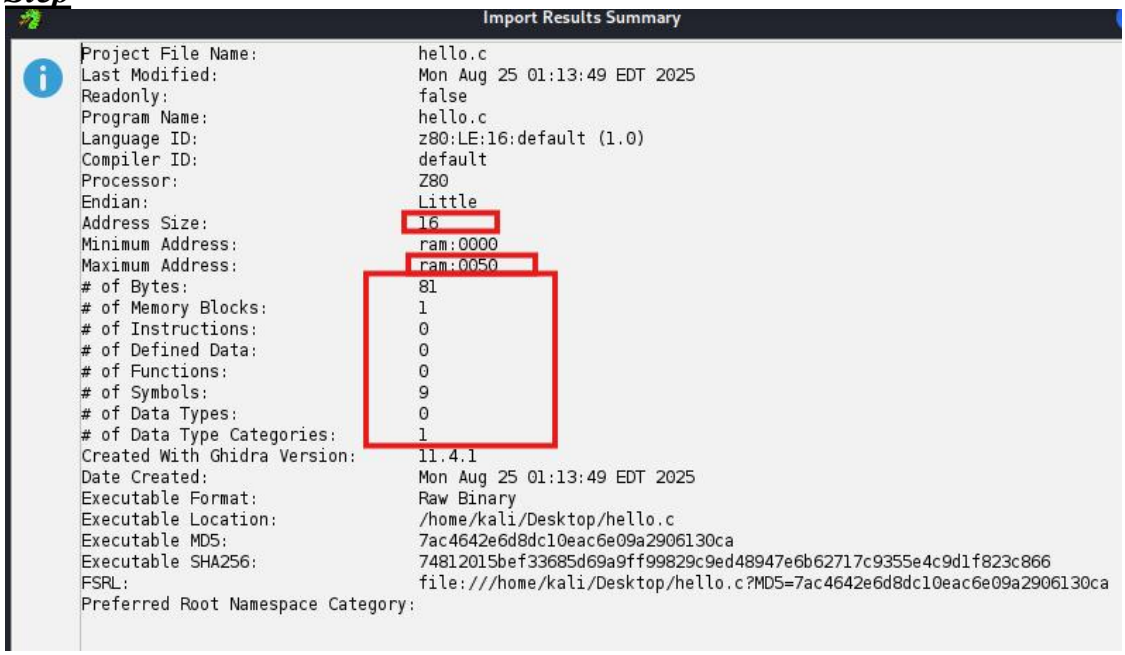


8. ***When prompted for Auto-Analysis, click Yes (accept defaults for your first run). You can re-run analysis later via Analysis → Auto Analyze…***

Step



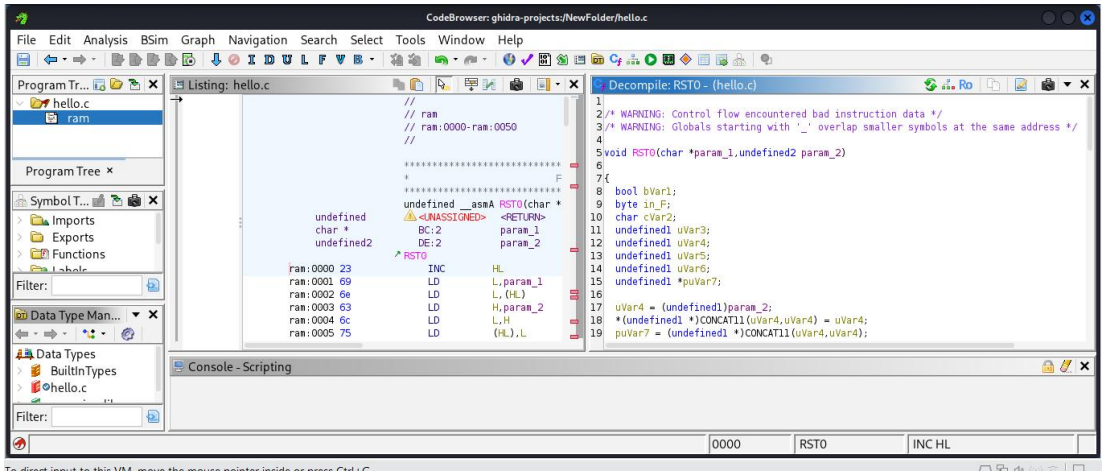
Step



You should now see the **Listing** (disassembly) and **Decompiler** windows.

5) Explanation of sections:

- **Program Tree (left top):** Displays loaded modules and memory sections. Here, hello.c → ram shows the memory segment being analyzed.
- **Symbol Tree (left middle):** Contains functions, imports, and labels discovered in the binary.
- **Data Type Manager (left bottom):** Holds built-in and user-defined types; here, hello.c types are visible.
- **Listing window (center):** Shows raw assembly/disassembly. You can see instructions like LD, INC, and data bytes (23, 69, 6e, 63, etc.).
- **Decompiler window (right):** Translates the assembly into a C-like pseudocode view (RSTOP(char *param1, undefined2 param_2) with variables such as bVar1, cVar2, etc.).
- **Console (bottom):** Used for logs and script outputs.



6) The interface you'll use the most

- **Program Tree**: Segments/sections (e.g., .text, .data, .rdata).
- **Symbol Tree**: Functions, labels, classes, namespaces.
- **Listing**: Mixed code/data view. Right-click here for most RE actions.
- **Decompiler**: High-level pseudocode of the selected function.
- **Console**: Logs, errors, script print output.
- **Data Type Manager**: Structures, typedefs, enums.
- **Bookmarks**: Save jump points during analysis.

If a window is missing: **Window → Reset Window Layout** or open from **Window** → the component you want (e.g., *Decompiler*).

7) Core workflow (step-by-step "commands" via menus)

*sudo apt update **&&** sudo apt upgrade -y*

sudo apt install openjdk-21-jdk -y

wget https://github.com/NationalSecurityAgency/ghidra/releases/download/Ghidra_11.4.1_build/ghidra_11.4.1_PUBLIC_20250731.zip

unzip ghidra_11.4.1_PUBLIC_20250731.zip -d ~/tools/

cd ~/tools/ghidra_11.4.1_PUBLIC/

```
./ghidraRun
```

You're ready to reverse. Start with small, legal samples, annotate everything, and build habits that make your work reproducible and explainable.