LAB 1

Free and Open-Source Software

Full name: Nguyen Minh Nhut

Student ID: B2205896

- Note: Screenshots need to be clear and good-looking; submissions must be in PDF format.

1. **Using OSS Methods to Produce Better Products**

   Enumerate a few reasons why adopting OSS methods can improve and accelerate a product's success.

   Answer:

   - OSS methods can improve and accelerate product success by:
   - Fostering innovation through diverse perspectives and rapid iteration.
   - Enhancing collaboration with open development and shared goals.
   - Reducing costs with minimal licensing and leveraging existing code.
   - Improving reliability and security through community review and established codebases.
   - Accelerating time-to-market by leveraging components and receiving community support.
   - Increasing market reach through ecosystem development and community adoption.

2. **What OSS Products Do You Use?**

   - Choose 5 pieces of proprietary software you use regularly, and find a FOSS equivalent for each. What license (like GPL, MIT, a mix of licenses, etc.) do the FOSS products fall into?

   - Write a short comparison (about the UI, the documentation, the bugginess, etc.) for each pair of proprietary and FOSS software

   1. Proprietary: Microsoft Word and FOSS: LibreOffice Writer

   License:

   - Microsoft Word: Proprietary
   - LibreOffice Writer: MPL (Mozilla Public License)

   Comparison:

   - UI: Word has a modern, intuitive interface; LibreOffice Writer has a traditional, customizable UI that can mimic Word's ribbon but may feel outdated to someone.

- Documentation: Word offers extensive, professional documentation and tutorials. LibreOffice Writer provides comprehensive community-driven documentation that may lack the polish of Word's resources.
- Bugginess: Word is generally stable. LibreOffice Writer is also stable but may have minor bugs, especially with complex formatting.

2. Proprietary: Microsoft SQL Server and FOSS: PostgreSQL

License:

- Microsoft SQL Server: Proprietary
- PostgreSQL: PostgreSQL License (similar to MIT License)

Comparison:

- UI: SQL Server has a robust GUI (SSMS) for database management. PostgreSQL uses pgAdmin, which is functional but not as polished as SSMS.
- Documentation: SQL Server has extensive official documentation and training resources. PostgreSQL's documentation is thorough but might be less beginner-friendly.
- Bugginess: SQL Server is very stable and optimized for enterprise use. PostgreSQL is also highly stable, with minor bugs.

3. Proprietary: Adobe Dreamweaver and FOSS: Visual Studio Code

License:

- Adobe Dreamweaver: Proprietary
- Visual Studio Code: MIT License

Comparison:

- UI: Dreamweaver offers a design-oriented interface with WYSIWYG editing. VS Code has a minimalist, code-focused UI without WYSIWYG features.
- Documentation: Dreamweaver includes comprehensive official documentation. VS Code offers rich, community-driven documentation covering many extensions.
- Bugginess: Dreamweaver is generally stable but may slow down with large projects. VS Code is highly stable, with occasional bugs.

4. Proprietary: Google Chrome and FOSS: Mozilla Firefox

License:

- Google Chrome: Proprietary (with some components under BSD license)
- Mozilla Firefox: MPL (Mozilla Public License)

Comparison:

- UI: Chrome has a clean, fast interface focused on simplicity. Firefox has a similar design but allows more customization with extensions and themes.
- Documentation: Both have comprehensive documentation; Firefox's is more community-driven, while Chrome's is developer-focused.
- Bugginess: Chrome is stable but can have memory issues. Firefox is also stable but may have performance hiccups with many extensions.

5. Proprietary: Adobe Photoshop and FOSS: GIMP

License:

- Adobe Photoshop: Proprietary
- GIMP: GPL (GNU General Public License)

Comparison:

- UI: Photoshop has a professional, customizable interface for advanced editing. GIMP's interface is less polished with a steeper learning curve but is also customizable.
- Documentation: Photoshop provides extensive official documentation and tutorials. GIMP's community-driven documentation is comprehensive but less detailed.
- Bugginess: Photoshop is generally stable but can have bugs after major updates. GIMP is stable for most tasks but may have occasional bugs with certain plugins or large files.

3. Contribute to OSS projects

Find an **active** OSS project hosted on GitHub that you are interested in, and spend some time figuring out some important information about the project:

I am interested in VScode.

- Who are the project leaders?
    - The project leaders are primarily led by developers at Microsoft.
- When did the project start?
    - The project began in 2015.
- What license does the project fall into?
    - The project falls into the MIT license.
- Answer the following questions using README file
    + What does this project do?

        + VS Code provides a source code editor that supports the core edit-build-debug cycle for developers, with features such as comprehensive code editing, debugging, and rich extensions.
    + Why is this project useful?
        + This project is useful because it combines the simplicity of a code editor with the advanced features developers need to write, debug, and manage their code effectively. It supports a wide range of programming languages and tools through extensions. Additionally, being open-source, it encourages community contributions and collaboration.
    + How do I get started?
        + To get started with the Visual Studio Code, you can download from the Visual Studio Code website for Windows, macOS, or Linux.
    + Where can I get more help, if I need it?
        + You can get more help form several channels:
            + Stack Overflow
            + GitHub
            + Slack
            + Social Media
            + Wiki and Documentation
            + Code of Conduct
- Answer the following questions using the project contribution guidelines
    + How to file a bug report
        + You can file an issue on the GitHub Issues page by providing a clear description of the bug.
    + How to suggest a new feature
        + You can submit a feature request through the same GitHub Issues page or upvote existing requests.
    + How to set up your environment and run tests
        + You can follow the instructions in the "How to Contribute" document to build from source, set up the development environment, and run tests.
    + The types of contributions the project is looking for
        + Types of contributions the project is looking for: Bug fixes, new features, documentation improvements, and code reviews.
    + The roadmap or vision of the project
        + The VScode have a clearly roadmap and it public on their GitHub. Their roadmap typically looks out 12-18 months and they establish topics they want to work on. They continuously tune the plan based on feedback and they will provide more detail in each of their monthly iteration plans. They will develop their next roadmap in around 12 months from now.

4. Selecting a License

The OSSWatch tool attempts to help its users understand their own preferences about free and open-source software licenses. There are 7 choices that you need to make.

- What question do you need to answer when making a choice?

  Answer:

  That has seven question I need to answer:

  - Do you want to limit the results to licenses that the Open Source Initiative describe as being "popular and widely used or with strong communities"?
  - Do you want to include licensing conditions on reuse?
  - How would you like your license to handle the issue of jurisdiction?
  - What is your attitude to the issue of patent grants in relation to your desired licence?
  - What is your attitude to patent retaliation in your desired licence?
  - Do you want your licence to specify enhanced attribution?
  - Do you want your licence to address the 'privacy loophole'?
  - Do you want your licence to include such a 'no promotion' feature?

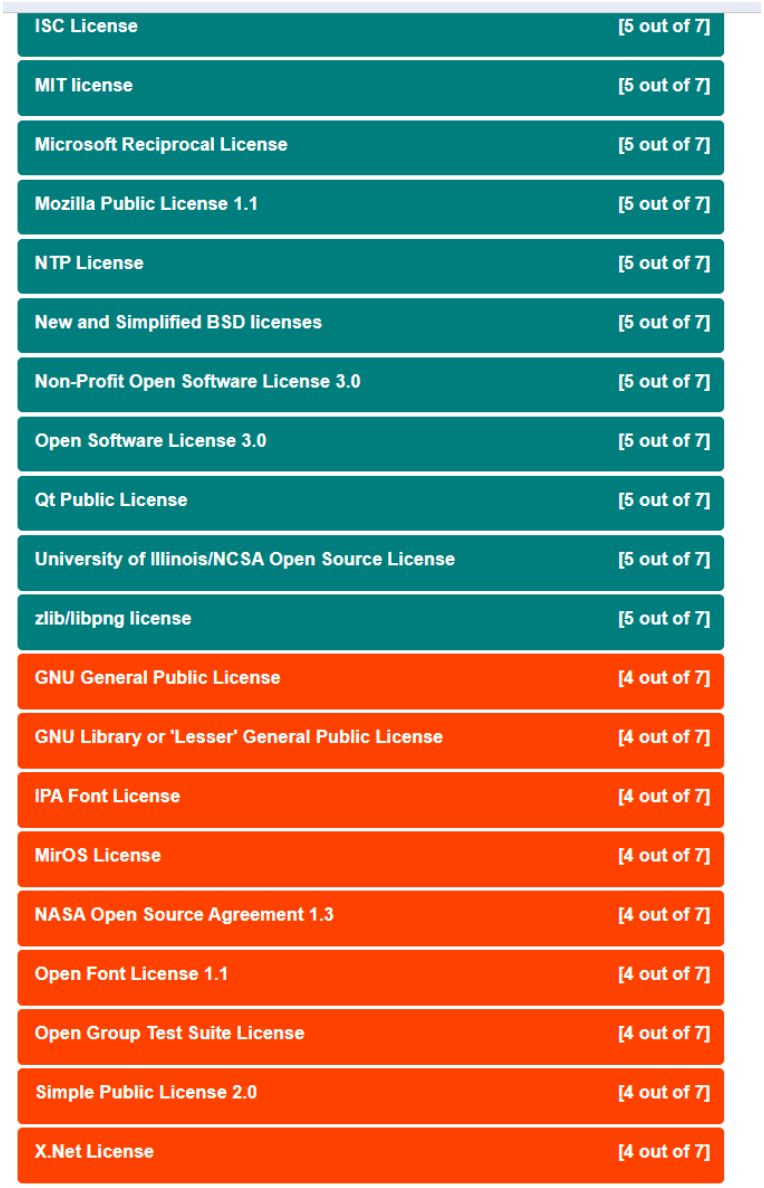
- What license(s) match ALL requirements below (other choices are “I don’t care”)?

  Permissive, exclusively patent grants, specify enhanced attribution

  Answer:

  - Pictures show the licenses match all requirements below:

Academic Free License 3.0	[6 out of 7]
Affero GNU Public License	[6 out of 7]
Apache License 2.0	[6 out of 7]
Artistic License 2.0	[6 out of 7]
Attribution Assurance Licenses	[6 out of 7]
Common Public Attribution License 1.0	[6 out of 7]
Lucent Public License Version 1.02	[6 out of 7]
Microsoft Public License	[6 out of 7]
Reciprocal Public License 1.5	[6 out of 7]
Adaptive Public License	[5 out of 7]
Boost Software License	[5 out of 7]
Common Development and Distribution License	[5 out of 7]
Common Public License 1.0	[5 out of 7]
Eclipse Public License	[5 out of 7]
Educational Community License Version 2.0	[5 out of 7]
Eiffel Forum License v2.0	[5 out of 7]
European Union Public License	[5 out of 7]
Fair License	[5 out of 7]
GNU General Public License v3.0	[5 out of 7]
GNU Library or 'Lesser' General Public License v3.0	[5 out of 7]



Take screenshots to show that you finish this exercise

5. Experimenting with FOSSology (optional)

The FOSSology project offers easy-to-understand options for learning the basics of how to use the tools. The easiest way is to use the project's online testing facility. To do this:

1. Point your browser to https://fossology.osuosl.org/repo/
2. Login with username=fossy and password=fossy.
3. Download the source code of the project in **exercise 3**, then upload it to FOSSology.

4. After the source code is analyzed, In the select action box, click onCopyright/Email/Urlin in the drop-down menu. You see the copyright, license, and URL information highlighted in appropriate colors.

5. Try clicking on some of the other fields in the top ribbon banner to see more useful information.

Take screenshots to show that you finish this exercise

---END---