CS1010 Laboratory 01

Logistics, Set-up, Exercise 0

Zhang Puyu

Group BD04

August 29, 2024

Plan of the Day

Quick Demographic Statistics

- There should be 11 students in our lab group.
- How many of you have learnt programming before?
- How many of you have learnt C or C++ before?

About Me...

- Name: Zhang Puyu
- Pronounced as Zhāng Púyù
- Currently a Year 3 student at NUS
- Major: Mathematics (unlike most other lab tutors)
- 2nd Major: Computer Science
- My E-mail: e0969139@u.nus.edu
- My GitHub: https://github.com/Z-Puyu.

Introduce Yourself! (While Taking Attendance)

- What is your name, year of study and major?
- One reason why you want to learn programming (other than to fulfil degree requirement).

House Rules

- Read the lab exercise questions (better if you attempt them) beforehand.
- There will be **NO** solution for any exercise.
- **Detailed feedback** will be given for exercises (if you submit them).
- **Ask questions** (no matter how silly they might seem) whenever you are in doubts.
- **Seek help** (either from me or from your friends) if you are stuck at an assignment question.
- You should use **the computers in the lab** in **Ubuntu**.
- Expect to be the **Chosen One** — either voluntarily or by the prowess of probability ;) — to present your ideas or solutions during the labs.

Class Telegram Group

After-Class Contact

- The slides will be uploaded to **Google Drive**.
- If you have any question/doubt, make sure to clarify with me before you leave.
- We use **Telegram** as the preferred channel of communication.
- Pose your questions directly in the Telegram group so that everyone can participate in the discussion.
- In the Telegram group, I will not reply to your questions immediately so that you have some time to think your questions over and discuss with the others first.
- Use this anonymous form to tell me any feedback/complaint regarding my teaching that you think I should know.

Regarding Exercises and Assignments

- You **SHOULD** complete all lab exercises although they are ungraded (the benefit of doing so is obvious).
- Hint: assignments count for 5% but practical exams count for 45%, which has two implications:
  1. The marginal benefit of plagiarism is so small that it should not be worth trying.
  2. Using AI to help earn 5% imposes *huge risk* of losing 45%.
- **No late submission will be accepted** even with valid reason. This is because you are allowed to miss **up to** 10% **of quizzes and exercises** without penalty.

My Tips to You as a Survivor

- **Solve every programming problem** you encounter throughout the course.
- Try to write out the solution step by step **in English** (or whatever language you prefer to speak) first before translating them into code.
- When you come across a bug or get stuck, try to **trace your code and debug on yourself** first.
- **Ask Google instead of ChatGPT** to train yourself in asking better and more targeted questions.
- **Use Piazza** wisely both to pose your questions and to spy other people for hints on programming problems.
- **Make your code work first** before trying to optimise it.
- **Find a study partner** to make use of collective intelligence.

Access PE Nodes

- Use the command `ssh <username>@<hostname>`.
- Preferably, use the node `pe113` (to distribute server load).
- If warned of "connecting to an unknown host", answer `yes` in the terminal.
- Input your password. Note that the characters will be invisible while doing so.
- Check the Troubleshooting Guide at https://nus-cs1010.github.io/2425-s1/guides/environments.html#troubleshooting
- Read https://nus-cs1010.github.io/2425-s1/guides/environments.html if you are interested in other stuff you can do with the PE Nodes.

Import `.vimrc` and Plugins

- Run `cp ~cs1010/.vimrc ~` in your home directory.
- Run `ls -a` and you should see a file named `.vimrc`.
- In your home directory, run `mkdir -p ~/.vim` to create a new directory called ".vim".
- Run `cp -r ~cs1010/.vim/* ~/.vim` to copy the default Vim settings from the CS1010 official directory.
- You may run `vim ~/.vimrc` to open the `.vimrc` configuration file to take a look.

(Very) Useful Configurations

- Display line numbers: `set number`.
- Display more accurate colours: `set termguicolors`.
- Find the line `color molokai`. You can change the colour scheme used for code highlighting by changing `molokai` to the name of your preferred colour scheme.
- We have pre-installed three colour schemes: `molokai`, `gruvbox` and `onedark`.
- Configure tab: `set tabstop=4` and `set shiftwidth=4`.
- Go back to your home directory and run `mkdir -p ~/.backup`. This creates the backup folder for Vim to regularly create automatic saves for your files.
- More Vim tips can be found at https://nus-cs1010.github.io/2425-s1/guides/vim-operations.html.

How to Deal With Swap Files

Sometimes when you try to open a file with Vim, you will encounter a *recovery file*. This is usually because the previous editing session was disrupted (e.g. you closed the terminal accidentally or lost Internet connection). In this case, you generally

- Press E to **edit anyway** if the current file is newer; or
- Press R to **recover** an autosaved swap file if the swap file is newer.

Set up `.gitconfig`

https://nus-cs1010.github.io/2425-s1/guides/github.html

In your **home directory on the PE host**, create a new file named `.gitconfig` and add the following to it:

```
[user]
name = Your name
email = Your e-mail
[github]
user = Your GitHub ID
```

Save the file and run `git config --get github.user` in your terminal. You should see your GitHub username.

Set up SSH Keys

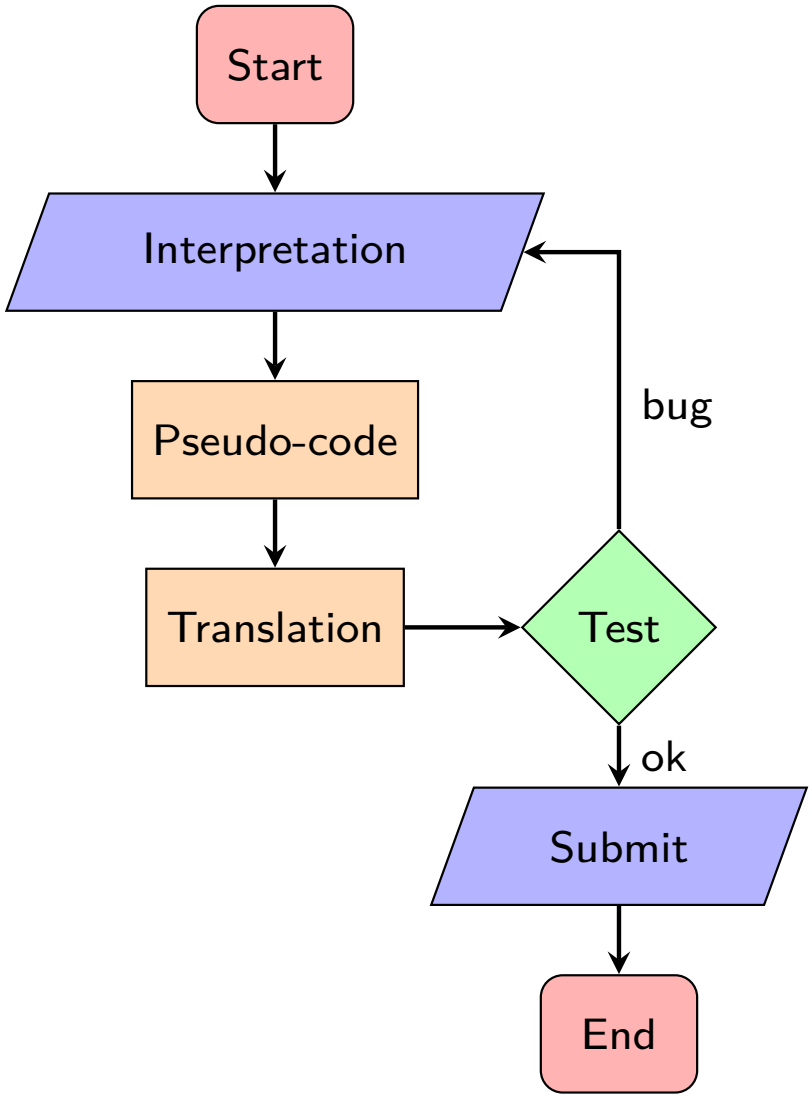
https://nus-cs1010.github.io/2425-s1/guides/github.html#2-setting-up-password-less-login

In your home directory, run the command `ssh-keygen -t ed25519 -C "email@example.com"` by replacing `email@example.com` with the actual e-mail linked to your GitHub account. Then, continue pressing Enter whenever you are prompted with some message.

Log in to your GitHub and click on your avatar picture on the top-right. Go to "Settings" → "SSH and GPG keys" → "New SSH key" and set up a new key.

On the terminal, run `cat ~/.ssh/id_ed25519 pub` and copy the SSH key printed in the terminal over to the GitHub page.

General Guidelines for Programming Problems



- Read the question and take note of the **inputs**, **outputs** and **constraints**.
- **Plan** your solution in *pseudo-code* (basically, try to figure out what is going on using human language).
- **Code** your program (basically, translate your pseudo-code written in human language to how machines talk).
- **Test** and **debug**.

Skeleton Code

- Accept the exercise via GitHub Classroom.
- Run `~cs1010/get ex00` to download the exercise files to your PE node.
- Go to the exercise directory and run `ls -a`. You will see these files and directories:

`Makefile`
`compile_flags.txt`
`.clang-tidy`
`.gitignore`
`test.sh`

You **DO NOT** need to edit nor view these!

`echo.c`
`quadratic.c`
`divide.c`
`ones.c`
`cuboid.c`
`bmi.c`
(generally, `file.c`)

These are the *skeleton codes* for the questions, which you **MUST** edit!

`inputs`
`outputs`

These directories contain *test cases* and their *expected outputs*.
You probably need to view them when debugging.

Important Commands

- Compile: `make program` (remember to exclude the ".c" file extension).
- Test: `./test.sh program` (remember to exclude the ".c" file extension).
- All-in-one: `make` — compiles **all** programs and tests **all** programs for **all** test cases and checks your *coding style*.
- View test case: `cat inputs/x.n.in`, where `x` is your program's name and `n` is the index of the test case.
- View expected output: `cat outputs/x.n.in`, where `x` is your program's name and `n` is the index of the output.
- View your program's output: `./x < inputs/x.n.in`, where `x` is your program's name and `n` is the index of the test case.
- Submit: `~cs1010/submit ex00` (change the exercise label accordingly).