Week 1 Trainings Homework

Prabhvir Babra and Pratyay Pandey

October 26, 2022

§1 Introduction

This handout is split into three general parts, the Command Line, Git, and C. However, there is no neat way to segregate these topics bits of each topic throughout the handout.

To complete the assignment, there are a mandatory number of points you must earn by completing problems on the handout.

If you are struggling with a problem, we urge you to stick with it. Often, the best way to learn is to find the answer after struggling. For example, if you are facing an error, Google it and browse forum posts because these sort of debugging skills are necessary as a programmer.

However, if you are really stuck or need clarification on a problem, you can ask either of us (Prabhvir or Pratyay) for help on Slack.

§2 Initial Setup

Before you can get started with the homework, you need to download programs that you may not have on your system. If you are struggling with the installation, ask Prabhvir or Pratyay for help sooner rather than later.

§2.1 Terminal

You will need a Terminal to work with the Command Prompt. If you are using a MacOS, you can simply launch the Terminal app. If you are using Windows, talk to Prabhvir to get WSL set up.

§2.2 Git

Git is the Version Control System that we use. In order to download it, follow the instructions on the Git download page at `https://git-scm.com/downloads`.

§2.3 C

C is the programming language we worked with on Tuesday and Thursday. The easiest way to work with C on your system is to download Visual Studio Code. The download instructions are straightforward and should be easy to follow.

§3 Command Line

To get started, clone the following Github repository (https://github.com/Prababab/valk-training-repository.git). Keep this repository in a place you will remember because this repository will be used in future handouts.

For this section, use only you terminal.

Change your current working directory to the Github repository that you just cloned. From there, change directory into hw-1.

§3.1 Problem 1

You are currently in the Github repository that you just cloned. What are the list of all non-hidden files?

§3.2 Problem 2

There are some sneaky hidden files in this repository that you couldn't see before. What are these hidden files in this repository?

§3.3 Problem 3

Notice a .txt file. Within it is a secret password. What is the password?

§3.4 Problem 4

Notice the folder, dir. Change directory to it. There is another file, garbage.txt. In this file, there is a lot of garbage. However, there is also a password in it. In the same line, there is the string "abeijpd." What is the secret password?

§3.5 Problem 5

The name garbage.txt is frankly quite disrespectful to the file; a file doesn't deserve to be called garbage! Rename the file to something more tasteful and write the command you used.

§3.6 Problem 6

Within your current directory, you might notice two other folders, util and lib. Cd (change directory) into util. What files are executable?

§3.7 Problem 7

Unfortunately, greeting.sh is not executable when it should be. Can you make the file executable? What command did you use?

§3.8 Problem 8

Run the file greeting.sh. What is the last line of output?

§3.9 Problem 9

Can you place the output of greeting.sh in a new file named greeting_output.txt?

§3.10 Problem 10

You have been exploring this repository, but why don't you make your mark? Create a file named hello.txt in the current working directory with any text you desire. What is the command you used?

§3.11 Problem 11

Go back to //hw-1. Can you make a new directory called my-work? What command did you use?

§3.12 Problem 12

The hello.txt file you made in Problem 10 is so appealing that you want to copy it to the my-work directory. Write one command that copies you copy hello.txt file from Problem 10 into the my-work directory?

§3.13 Problem 13

Change your directory to lib. Notice two directories, temp1 and temp2. In temp1, can you remove the file, file1.txt? What is the command you use?

§3.14 Problem 14

There are still 20 more files in temp1.txt, and it's exhausting to remove all of them one by one. What is the one command needed to remove all the files in temp1 at once?

§3.15 Problem 15

temp1 is unnecessary now. Remove the temp1 directory. What is the command you used to do this?

§3.16 Problem 16

Try removing temp2 without removing any of its files. What is the command that you need to remove it?

§4 C Code

This section will be more open ended than the previous section. In hw-1, go to the directory **c-code**. This is where your C code will go.

This task will mainly consist of two types of problems: debugging problems and original code writing questions.

For Debugging Problems, you will be given code that has build/runtime errors and a description of what the code should do. Your job will be to fix the errors.

In order to Debug Error read the build errors in addition to using print statements to see parts of the program at different times. There are more formal debugging programs such as gdb. However, print statements will be sufficient.

There will be additional questions accompanying the Debugging questions.

With Original Code Problems, you will be given a task to solve.

§4.1 Problem 1

Your job is to make a calculator that supports the following operations: addition, subtraction, multiplication, division, and exponentiation.

You will recieve an integer **n** denoting the number of operations you will read. You will then receive n operations.

Each operation consists of an operation symbol ('+', '-', '*', '/', and '**') followed by an integer. You will create an interactive calculator where you start off with the number 0 and each operation works to modify that number.

To clarify the problem statement, here are some sample test cases:

```
3
+ 2
- 5
* 4
/ 8
** 2
```

Answer: 2.25

Explanation:

- You are given 3 operations. Starting with '+ 2', add 2 to the initial number 0. This yields 2.
- The next operation is '- 5', which denotes subtracting 5 from the current number, 2. Thus, this yields -3.
- The third operation '* 4', denotes multiplying the current number, -3, by 4, which yields -12.
- The fourth operation, '/ 8', denotes dividing the current number by 8, which yields -1.5.

- Finally, the fifth operation, '** 2', denotes to take the current number to the power of 2, which yields a final answer of 2.25.

Write your final code to the file, **calculator.c**.

§4.2 Problem 2

Go to **traverse_array.c**. This program receives an integer **n** followed by n integers that denote an array.

Then the program takes in two integers, a left index, **l** and a right index, **r** (both are 0-indexed ie index 0 accesses the first element).

The goal of this program is, starting from index r, print each element and move left until it prints the element at index l.

This is an example of a testcase that will work.

```
5
1 2 3 4 5
1 3
```

Answer: 4 3 2
Explanation: Element 3 of the array is 4, element 2 of the array is 3, and element 1 of the array is 2.

§4.2.1 Problem 2.1

There are two exploitable issues that the program can face given the right testcase. What are they?

§4.2.2 Problem 2.2

Write two testcases that will cause each of these errors.

§4.2.3 Problem 2.3

What is a possible solutions to fix each of these bugs? Make sure to implement these fixes to the program file, and try to change as little of the program as possible.

§4.3 Problem 3

Create the file **airlinetickets.c**. In this problem, you will be working with airline tickets.

For this task, you are given the integer, **n**, the number of airline tickets you will receive.

Each Airline Ticket has the following information:

```
airline_ticket: {
    customer_name: string;
    customer_id: int;
    boarding_group: integer;
    airline_name: string;
    gate: integer;
```

```
7               destination: string;
8           }
```

You will receive the input given in the order above.

Your job is to create an array of airline tickets and make functions that can solve the following sub-tasks.

§4.3.1 Problem 3.1

Sort the airline tickets by customer name. A string is lexicographical smaller than another string if at the first character at which the two string differ, the character from the first string is alphabetically before the character of the second string.

§4.3.2 Problem 3.2

Given a gate number and airline, print out all the names of the customers that have that specific gate number and airline.

§4.3.3 Problem 3.3

Given a customer_id, find the name of the customer in the airline_tickets. This would be straightforward, except your solution must run in a time faster than O(n).

§5 Git

For this Git section, in a separate folder, complete modules from this website: https://hackmd.io/@sparker/git-workshop-happy-path?print-pdf#/.

§6 Conclusion

Overall, this handout is meant to help you build on certain parts of C programming, Git, and the command line. Further reading on C is highly encouraged through Kerningham's C.

We also recommend reading more on the command line and learning more on Git. A good, free read would be www.git-scm.com, which also talks about Git internals. While not necessary to know, it's highly encouraged to understand the design of Git and how it works under the hood.

Make sure to follow up with Pratyay to get specific feedback on what problems would best help you.