DUE DATE: **By the deadline specified on Canvas**

CSE-443/543: High Performance Computing

Exercise #2

Max Points: 20

Objective: The objectives of this exercise are to refresh concepts related to:

- Work with files specified as a command-line argument
- Read columnar data from a file
- Review working with a `std::vector`
- Review working with a `std::unordered_map`

Fill in answers to all of the questions. For some of the questions you can simply copy-paste appropriate text from the Terminal window into this document. You may discuss the questions with your neighbor, TA, or your instructor.

Name:

Part #1: Deduping user entries

Estimated time: 30 minutes

Background: In this exercise you will be required to work with `std::vector` and `std::unordered_map`. Ensure you review the commonly used methods for these two containers from Canvas →`Files`→`Handouts`→`CommonMethodsAndCommands.pdf`

Problem: You are supplied with a data file (as command-line argument) with each line having 3-columns (all strings) in the format: `UserID IP port` (separated by whitespaces). You are expected to complete a method called `dedupe` that returns a list of lines with no more than the most recent 2 lines for each user. Here is a long form of that statement – if there are more than 3 for an user, then retain last 2; if there is 3 retain last 1; if less than 3 then retain all. For example, given the following list of values, the program should return the list as shown below, by clearing out duplicate entries for the user `bill`.

Input data
bill 128.199.152.105 60844
smbuser 119.45.159.254 35124
bill 185.228.113.216 54928
vagrant 221.6.32.34 52420
bill 176.111.173.8 54816
smbuser 159.75.129.200 45058
admin 107.175.83.11 47498
vagrant 20.67.27.149 40406
bill 35.184.62.5 43960
bill 2.236.108.242 58246
vagrant 211.253.8.225 59634
bill 154.8.151.45 24253

List returned by dedupe
smbuser 119.45.159.254 35124
smbuser 159.75.129.200 45058
admin 107.175.83.11 47498
bill 2.236.108.242 58246
vagrant 211.253.8.225 59634
bill 154.8.151.45 24253

DUE DATE: **By the deadline specified on Canvas**

Exercise

1. Log into OSC's OnDemand portal via https://ondemand.osc.edu/. Login with your OSC id and password that you setup.
2. Startup a VS-Code server and connect to VS-Code
3. Next, create a new VS-Code project in the following manner:
   a. Start a new terminal in VS-Code
   b. In the VS-Code terminal use the following commands:

```
$ # First change to your workspace directory
$ cd ~/cse443
$ # Use ls to check if workspace.code-workspace file is in pwd

$ # Next copy the basic template for a C++ project
$ cp -r /fs/ess/PMIU0184/cse443/templates/basic exercise3
$ # Copy the starter code for this exercise
$ cp /fs/ess/PMIU0184/cse443/exercises/exercise3/* exercise3
$ # Now add the newly created exercise3 directory to VS-Code
```

4. Now add the newly created exercise3 directory to VS-Code Briefly study the starter code in `main.cpp`. It is straightforward.
5. Briefly review the data in `list1.txt`.
6. Suitably implement the dedupe method. Refer to the methods for these two containers from Canvas →`Files`→`Handouts`→`CommonMethodsAndCommands.pdf`. If you take the approach of erasing values from a vector you can use the following approach:

```
vec.erase(std::begin(vec), std::begin(vec) + 2);
```

7. You can test the program by setting command-line argument as demonstrated in the `OSC OnDemand & VS-Code demonstrations` page on Canvas. See video titled `Using command-line arguments in VS-Code`.

Part #2: Submit to Canvas via CODE plug-in

Estimated time: 5 minutes

In this part of the exercise, you will be submitting the necessary files via the Canvas CODE plug-in.

1. Download your source file from VS-Code to your local computer. **Ensure your file name does not have special characters**.
2. Upload the files using the "`Upload via CODE`" tab on Canvas.

Ensure you actually **submit** the URL generated by CODE plug-in in the final step as shown in the adjacent figure