ECE220: Computer Systems & Programming **Fall 2020 ZJUI**
Lab 14 **Tuesday 22 December**

Matching Teams with Function Pointers

In today's lab, you must make use of a cyclic, doubly-linked list with a sentinel to add a command to a tool for managing teams in a class. The code for managing and iterating over the list is already written for you, as is a trie for interpreting and executing commands. You need merely implement the "show" command to show individuals (not on teams) and teams that contain people of one sex or the other (F or M). The goal is for you to write some command interpretation code, which is useful for MP12, and some code using function pointers, which will help you understand some of the concepts built into C++. If you have extra time, you may want to take a look at the trie code to see how the interpreter works.

Begin by checking out the **`lab14`** subdirectory in your Subversion repository. The directory contains a copy of this document (**`lab14.pdf`**), a C header file **`lab14.h`**, a C source file **`lab14main.c`** that provides most of the code, a **`Makefile`**, and a C source file, **`lab14.c`**, with which you can start this lab. The directory also contains a trie implementation (**`cmd_trie.h`** and **`cmd_trie.c`**) and a cyclic, doubly-linked list implementation (**`double_list.h`** and **`double_list.c`**) and a list of **`students`** in our class for use with the tool. You can immediately "**`make`**" the executable and play with the tool, but the code as given will generate warnings, and the "show" command will do nothing without your code.

What the Given Code is Doing

The code given to you provides a command interpreter for a tool that manages teams in a class. Initially, all people in the **`students`** file are read in add added to the doubly-linked list **`people`** using the "add" command. Commands are read from **`stdin`** one line at a time and executed using a trie. Commands allow the user to add new people, delete people, list people and teams whose names or members start with a given prefix, pair people into teams, unpair people from teams, and quit. The "show" command (to be written by you) allows the user to show individuals and teams including people of one sex, either F(emale) or M(ale).

The doubly-linked list code implements a cyclic, doubly-linked list with a sentinel and includes functions for initializing a list (to empty), inserting an element, removing an element, executing a function on all elements, and finding the first element in the list.

A trie is a tree with 26 children per node. To use a trie, one starts with a string and follows the corresponding child link for each letter in the string (the trie given is case insensitive). Each trie node contains a function pointer. If the pointer is **`NULL`** when a non-letter is found in the string, the command is not defined. If the function pointer is not **`NULL`**, the function is executed on the rest of the string (the part after the command).

The Task

First, complete the **`cmd_show`** function. Take a look at the "add" command (**`cmd_add`**) for syntax checking and the "list" command (**`cmd_list`**) for the structure of the output and the callback functions. Both are in **`lab14main.c`**. You will need to know how to execute a function all elements of the list **`people`**.

Next, you must implement the two callback functions, **`do_show_individuals`** and **`do_show_teams`**. The first is easier than the second. If you get both working, you should be able to use the in the interpreter. Feel free to make a shorter list of people with which to debug—the file to be used must be named **`students`**, however.