Assignment 2: Marriage Pact 💖

Due ***February 1 at 11:59pm***

Table of Contents

Overview

Happy assignment 2! This is meant to be a very short and sweet bit of practice to get you started working with the STL's containers and pointers.

Download the starter code here.

Part 1: Get all applicants

You've been waiting for days to get your Marriage Pact initials this year, and they've finally arrived in your inbox! This year, they're implementing a new rule: your match MUST share your own initials to be eligible. However, even after talking about it for hours with your friends, you have no idea who your match could be! There are thousands of students on campus, and you can't just go through the whole roster by hand to draft up a list of your potential soulmates. Fortunately enough for you, you're in CS106L, and you remember that C++ has a pretty quick method of going through collected, similar information – containers!

We've included a .txt file of all of the (fictional) students who signed up for The Marriage Pact this year. Each line includes the first and last name of the student. You will write:

- **get_applicants**: From the .txt file, parse all of the names into a set. As you add names to your set, print them out to the console as well, with each name on a different line. In your implementation, you're free to choose between an ordered and unordered set as you wish!

Additionally, please answer the following short answer question in **short_answer.txt**:

- It is your choice to use either an ordered or unordered set. In a few sentences, what are some of the tradeoffs between the two? Additionally, please give an example (that has not been shown in lecture) of a valid hash function that could be used to hash student names for an unordered set.

NOTE: All names appearing in this assignment are fictitious. Any resemblance to real persons, living or dead, is purely coincidental.

Part 2: Find matches

Great detective work! Now that you've narrowed down your list of potential soulmates, it's time to put it to the test. After a long day of acapella and consulting club meetings, you return to your dorm to learn from your roommate that there is a mixer for Marriage Pact matches at Main Quad that night! Your best chance of finding true love is imminent — if only you can get out of your Ultimate Frisbee practice. Quickly, you decide to

interview everyone who shares your initials at the mixer, and you get to work coding up a function that will compile the order for you automatically.

For this section, you will write the function:

- **find_matches:** From the set, take all names that share your initials and place pointers to them in a new std::queue.
    - If you're having trouble figuring out how to iterate through a set, it could be helpful to look back over Thursday's lecture on iterators and pointers.
- From here, please print in the main function your "one true match." This can be determined as you see fit; choose some method of acquiring one student from the queue, ideally something with a bit more thought than a single pop(), but it doesn't have to be particularly complicated! Consider random values or other methods of selection.
    - If your initials have no matches in the dataset, print "NO STUDENT FOUND." Better luck next year :(

Afterwards, answer the following questions in **short_answer.txt**:

- Try changing the backing container of the std::queue from its default value to either a list or a vector. Implementation wise, what changes? Why would the C++ developers leave this as an option? Explain container adaptors in this manner in 3-5 sentences.
- Note that we are saving pointers to names in the queue, not names themselves. Why might this be desired in this problem? What happens if the original set where the names are stored goes out of scope and the pointers are referenced?

Feedback Survey

Please fill out this anonymous feedback form to help us improve these short assignments in the future!

Submitting Instructions

When you have completed this assignment, upload all of the files to Paperless under the correct assignment heading.
Your deliverables should be:

- **main.cpp**
- **short_answer.txt** with all of your responses.

You may resubmit as many times as you'd like before the deadline. Please let us know if you have any questions as you work on the assignment!