

Project 1: Nurse Scheduling Problem

Milestone Due: September 19 by Midnight

Complete Project Due: October 3 by Midnight

Instructions

For this project you will be designing and implementing a system, in either C or C++, to help describe and determine a schedule for a set of nurses at an emergency clinic. The clinic is open 24 hours a day, five days a week, and nurses work in eight hour shifts. Using bit vectors and set operations, your system should provide information about the available nurses and their potential schedules, as well as determine an actual schedule for a week.

Additionally, sample input files can instead be found on the CSE machines at

`/home/jeh0289/public/csce2110/fa18/proj1/` You can `cd` into that directory and copy input the files from there.

For the project you may, if you choose, work in small groups of up to three students. You will need to sign-up for a group on Canvas by no later than midnight (9/7). This can be accessed in the Groups tab in the People section. Any student who does not join a group will be assumed to be working on their own and will be assigned a single member group. A group cannot be changed once created, unless it is completely dissolved. All members of a group will receive the same grade, and all members are expected to contribute equally to the programming and report. Only one submission per group will be required, but a group will not be penalized for having multiple members submit the completed project.

Also, as a reminder, all of the code for this assignment must be written by you. You may not share code or download solutions off the internet, as doing so will be considered cheating.

Requirements

This assignment has two parts: a design portion and an implementation portion.

Design Document

For the design portion, you must generate documentation, in PDF format, describing your system and design process. The purpose of this is for you to explain not just what your system is doing, and how it is doing it, but why. You will need to justify your design decisions in a concise, informative manner. Justifications such as “I did this because it was easy” are not sufficient, as you should actually explain why a particular data structure or algorithm was more efficient, effective, or optimal. Additionally, commented code, while sometimes helpful in small examples, is not a sufficient explanation in and of itself. Your explanations and justifications are expected to be presented in prose and in paragraph format, i.e. not bulleted lists. Further, part of the evaluation of your design document is the apparent amount of thought and effort that went into its creation.

This document should be divided into four main parts, each with an appropriate header.

In the first part, you should describe your design process. Did you work out the algorithm on paper or a whiteboard before hand? Did you draw UML diagrams of the system? Did you create a small prototype? Did you simply start coding away and then recode once or twice with newfound understanding? In a few paragraphs, describe in detail how you went about designing the system, and be sure to provide sufficient justification of your methodology.

For the second part, you should describe the data structures you used in your system. What, if any, objects or structs did you create to store data? How did you organize and manage them? What types of formal data structures did you make use of (trees, graphs, arrays, hashes, etc)? In a few paragraphs, describe in detail how you stored the various data elements in your system, and be sure to provide sufficient justification of your methodology.

For the third part, you should describe functionality of your system. How is data moved and transformed? How is it read in? How is it output? What are the various major functions you constructed and how do they work? In a few paragraphs, describe in detail how your system works, and be sure to provide sufficient justification of your methodology. You might also consider including diagrams to more easily visualize how all of the pieces fit together.

Finally, you will need to briefly describe what share of the work was performed by each group member. Be sure to be specific about tasks.

Implementation

Your program must provide the following functionality and adhere to the following constraints:

- Allow the user to choose the file containing the list of nurse schedules
 - For a nurse's schedule, the first line will contain the name of the nurse
 - The next four lines will contain the day and times they can work
 - The times will be in 24 hour, military-style format
- Your program should read in each these schedules and create five bit vectors for each nurse, one for each day of the week, and store their availability information with each hour of the day being a slot in the bit vector
- Using appropriate set operations you implement, your program should output the following
 - A yes or no answer as to whether there is simple, complete coverage of all 120 hours during the week, i.e. can at least one nurse work during each hour of the week
 - A 5x24 bit matrix listing which hours during the week are more than one nurse available to work, i.e. 0 if less than two are available to work, and 1 if two or more are available to work
 - A 5x24 bit matrix listing which hours during the week are no nurses available to work, i.e. 0 if at least one can work, and 1 if none can work
 - A 5x24 bit matrix listing which hours during the week exactly one nurse can work, i.e. 0 if zero or more than two can work, and 1 if exactly one can work
- Your program should also output a 5x24 matrix that contains the initials of each nurse on the days they will work, such that the schedule has complete coverage and the total number of used nurses is minimized
- Each 5x24 bit matrix should have an appropriate vertical and horizontal labels and be easy to read and understand
- You may not use any of the set containers or set operations/algorithms provided by C or C++. You must write your own versions of them.
 - Examples: bitsets, sets, multisets, set_union, set_intersection, set_difference, etc...
- Your code must be well commented.
- You must provide a short README file which includes your name and explains how to compile and run your program.
- Additionally, you may write a makefile if you want your code to compile with additional flags.

Milestone Submission

Your program must be able to read in the file as well as provide the yes or no answer concerning simple, complete coverage. You must submit a .zip file containing the following:

1. All files necessary to compile and run your program
2. A README file explaining how to compile and run your program

Complete Project Submission

Your program must provide all requested functionality. You must submit a .zip file containing the following:

1. All files necessary to compile and run your program
2. A README file explaining how to compile and run your program
3. Your design document in PDF format

Rubric

The entire assignment is worth 100 points. The breakdown of those points is as follows.

- 40 points: Design documentation
- 20 points: Code satisfies milestone requirements
- 30 points: Code satisfies complete requirements
- 10 points: Professional coding style
 - 5 points: Adequate comments
 - 3 points: Modularity
 - 2 points: Readability
- If your code fails to compile on the CSE machines you may not receive credit for the programming portion of the assignment. I recommend not making changes to your code without checking for compilation before you submit.

Bonus

For 10 bonus points, your program should also output a 5x24 matrix that contains the initials of each nurse on the days they will work, such that the schedule has complete coverage and the total number of used shifts is minimized.