CS 410: Homework 2

Due: Thursday, Nov. 18, 2021 at 11:59pm **100 pts**

Instructions:

YOU WILL SUBMIT **MULTIPLE FILES** FOR THIS PROGRAMMING PROJECT.

PLEASE FOLLOW THE INSTRUCTIONS FOR SUBMISSION INCLUDING FOLDER STRUCTURE AND FILE / FOLDER NAMES (THEY ARE CASE SENSITIVE) ELSE THE GRADING SCRIPT MIGHT NOT LOCATE YOUR HW2 SUBMISSION.

Create a subdirectory named "hw2" in your cs410 directory. Use that subdirectory for all your file submissions on this assignment. At the end of the homework assignment, as a minimum, these files should be found in your hw2 directory:

- **hw2_functions.cpp** (function definitions)
- **hw2_functions.h** (function declarations)
- **hw2.cpp** (main)
- a "typescript" file demonstrating program compilation, execution and testing

You may include other source/header files if you please but you must submit a typescript demonstrating program compilation, execution and testing.

Background: After some years of work, Krusty finally decides to get his University degree. But now, the Beacons (those generous Professors at UMASS Boston) have made it so that he can't graduate unless he completes a number of courses in some required order. And to do that, he has to take courses within the same year or higher years, and he can only take the courses one at a time. Once a course is taken from a particular year, he cannot take any courses from previous years (First digit in the course code represents the year). Also, once he takes a course, he cannot take another course with a lower course number, not even in the years ahead (Second digit in the course code represents the course number). Finally, once he takes a version of a course, he cannot take another version of any other course with a lower version number (Third digit in the course code represents the course version). There's just one problem.....Krusty in his bid to increase the number of courses he can take and make it easier for him to graduate, requested the school authorities to be a bit flexible on the criteria for taking courses. His wish was granted with a tweak: He can take a course with a lower course number than a previously taken course only if the new course number is not lower than any previously taken course version, and the new course version is not lower than any previously taken course number[1].

1 As an example: course1 is 2.2.0, course2 is 1.1.0. These can NOT satisfy Krusty's graduation criteria since course2 is from a previous year. As another example: course1 is 2.1.0 and course2 is 2.4.0. These two CAN satisfy Krusty's graduation criteria since course2 has a higher course number than course1, course2 is not from a previous year compared to course1, and the version of course2 is not lower than that of course1. So, you see, you must compare the courses in all 3 instances. Both the course number and course version of a previous course cannot be greater than either the course number or course version of a current course, and at least one of the course number or the course version must be strictly greater than at least one of the previous course number / course version. So taking 3 courses in the order 4.2.1, 4.4.3, 4.3.6 is valid and satisfies Krusty's graduation criteria.

Specifications: Your program is to simulate this situation. Krusty will be given the total number of courses he will take from the start. You are to help Krusty decide if his course selection will lead to graduation. Krusty needs a minimum of 8 "eligible"[1] courses to graduate. Your program will take as first input, the total number of courses Krusty will take, followed by a list of numbers, each representing the index of courses that Krusty plans to take (In the order they appear in the "course_catalog.txt" file). If there is a combination of the selected courses that can lead to graduation, then print the statement: "**Congratulations!!! Krusty has earned enough credits to graduate**", followed by a list of the course codes and course names in the order in which he can take them (Can be ascending or descending). If there is no such combination for him to graduate, then print the statement: "**Krusty hasn't received enough credits to graduate yet**".

Details:

- You are expected to use a struct for a course, and that includes its name and the details of the course code.
- You are expected to use arrays in this assignment as appropriate.
- Graduation rule: Krusty can graduate if he has taken 8 (eight) "eligible"[1] courses.
- If Krusty takes a course more than once, then only one of them counts towards graduation.
- The data to use (courses and their codes) can be found in the file "course_catalog.txt". There are 14 courses in all. Your code should read in the file and use the contents as necessary. A sample of the file contents is below:

COURSE	**CODE(*year.num.ver*)**
Introduction to computing	1.1.0
Applied Discrete Mathematics	2.2.0
Intermediate Computing with Data Structures	2.1.0
Programming in C	2.4.0
Advanced Data Structures and Algorithms	3.1.0

Special Observation: You will find your coding easier if you were to sort Krusty's courses by the criteria given in the **background** section. Think carefully how this has to be done.

Examples:

Inputs: 10, [11, 9, 5, 2, 7, 6, 0, 8, 1, 4]
Output:

```
Welcome to the UMB Graduation status check app!

Enter the number of available courses...
Enter which course you are taking...

4.4.6 (Introduction to Internetworking added)
4.4.3 (Cybersecurity in the Internet of Things added)
3.4.1 (Computer Architecture added)
2.1.0 (Intermediate Computing with Data Structures added)
4.2.1 (Competitive Programming added)
4.2.0 (Intro to Software Engineering added)
1.1.0 (Introduction to computing added)
4.3.6 (Database Application Development added)
2.2.0 (Applied Discrete Maths added)
3.1.0 (Advanced Data Structures and Algorithms added)


Congratulations!!! Krusty has earned enough credits to graduate:
4.4.6 (from Introduction to Internetworking)
4.3.6 (from Database Application Development)
4.4.3 (from Cybersecurity in the Internet of Things)
4.2.1 (from Competitive Programming)
4.2.0 (from Intro to Software Engineering)
2.2.0 (from Applied Discrete Maths)
2.1.0 (from Intermediate Computing with Data Structures)
1.1.0 (from Introduction to computing)

Thanks for using the UMB Graduation status check app!
```

Inputs:11, [11, 9, 5, 10, 2, 7, 6, 3, 8, 1, 4]
Output:

```
Welcome to the UMB Graduation status check app!

Enter the number of available courses...
Enter which course you are taking...

4.4.6 (Introduction to Internetworking added)
4.4.3 (Cybersecurity in the Internet of Things added)
3.4.1 (Computer Architecture added)
4.4.4 (Introduction to Operating Systems added)
2.1.0 (Intermediate Computing with Data Structures added)
4.2.1 (Competitive Programming added)
4.2.0 (Intro to Software Engineering added)
2.4.0 (Programming in C added)
4.3.6 (Database Application Development added)
2.2.0 (Applied Discrete Maths added)
3.1.0 (Advanced Data Structures and Algorithms added)


Krusty hasn't earned enough credits to graduate yet!

Thanks for using the UMB Graduation status check app!
```

And, as always, let your TA or instructor know if you need any help.