Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2021
Homework # 5
Due date Dec 15th 2021

In this homework, you will write an abstract base class **`BoardGame2D`**. This class will have at least the following functions some of which might be pure virtual functions.

- **`playUser`** takes a string as a parameter and plays the game accordingly. The parameter depends on the game, for the game of Peg Solitaire the string could be "2B UP"
- Another overload of **`playUser`** is a *final* function. It does not take any parameters, it takes a string from the user for the next move in a loop and plays the game until it is over. It also prints the board between the moves.
- **`playAuto`** plays the game by the computer for one move.
- **`playAutoAll`** is a *final* function. It plays the game until it is over. This one calls **`playAuto`** for all the moves. It also prints the board between the moves after some pause.
- **`print`** and **`operator<<`** prints the game on the screen starting from the top left corner of the terminal. See ANSI escape sequence to move your cursor anywhere on the screen https://en.wikipedia.org/wiki/ANSI_escape_code#In_C.
- **`endGame`** returns true if the game is ended.
- **`boardScore`** returns an **`int`** score value for the current board. It returns a positive integer that indicates the goodness of the current board. Smaller the returned value, better the board. If the game is finished for the current board, it returns return 0, which is the best case.
- **`initialize`** initializes the board. For some games the initial board is the same, for other games the initial board is random.
- static **`playVector`** function takes a vector of **`BoardGame2D *`** objects. It plays all the games in the vector until they end.

Derive the following concrete classes from the base class

- **`PegSolitaire`** class will play the type 2 game of https://en.wikipedia.org/wiki/Peg_solitaire
- **`EightPuzzle`** class will play the https://en.wikipedia.org/wiki/15_puzzle game for a 3x3 board.
- **`Klotski`** class will play the Klotski puzzle on a 4x5 board https://en.wikipedia.org/wiki/Klotski.

Write your driver function that

- makes 2 objects of each class, puts them in a vector and calls **`playVector`**.
- tests each function of your classes one by one. Show how your score function performs.

Write a report about your class design decisions such as your move strings, your score function, etc. You should submit a PDF file as your report.

Notes:

- Use namespaces, separate header and source files. Include makefile to compile your submission.

- Do not use any functions from the standard C library (like `printf`)
- Do not use anything that we did not learn in the lectures.
- Do not forget to indent your code and provide comments.
- Check the validity of the user input.
- **Submit your test results as TXT files.**
- You should submit your work to the Teams page.