Due date: 04.11.2014 – Tuesday – 23:00

BLG 233E - Data Structures and Laboratory, Fall 2014
Assignment 3 - Multi Linked Lists
Console Solitaire

Problem Definition:

Remember Windows Standard Solitaire game. In order to remember the game or learn the rules better you can play the game from this link: http://www.solitr.com/ You can read the aim and definition of the game below.

Aim: In this game the aim is sorting a card deck of playing cards into four categories as Spades, Hearts, Diamonds, and Clubs in the increasing order (A,2,3,4,5,6,7,8,9,10,J,Q,K). So, you have four "Foundation Lists". You **should** pile cards to the Foundation Lists in the increasing order starting from "A".

On the board: In the beginning of the game, you see 7 lists. Let's call these lists as "Board Lists". The first list includes one card, the second list includes two cards, the third list includes three cards, and so on. Each list shows only the identity of the card on the frontmost, the other cards are downturned. There is also a list on the top which includes rest of the cards. Let's call this card list as "Top List". You can see all the cards in the Top List.

Playing the game: In order to achieve the goal you have to reveal the downturned cards and learn the identity of the cards. For this purpose, you can sort cards on the **board** by the following **rule**; all the cards have to be in the decreasing order, all black cards follow red cards, and all red cards follow black cards. If you can move all upturned cards from a board list, it will reveal the next downturned card if the list is not empty. You can select cards from Top List or from Board Lists and move these cards to Board or Foundation Lists according to rules. If a Board List is empty you can fill it only by the biggest card which is "K". You can understand the rules better by practicing the Solitaire game.

Implementation Details:

You will have several lists; Top List, Board Lists, and Foundation Lists. Board Lists and Foundation Lists have to be multi-linked lists. You are going to read the lists from a text file called "**solitaire.txt**" (Do not change the name of this text file). This text file includes top list, and board lists respectively which are separated by stars (******) (i.e. the first list is Top List, the second list is 1.Board List, the third list is 2.Board List, and so on.). In the text file the cards are symbolized by four tokens. The first token symbolizes the color of the card, the second token symbolizes suit of the card, the third token symbolizes card number and the last token symbolizes whether the card is downturned or upturned.

Symbol of colors:
R = Red
B = Black
Symbol of suits:
S = Spades
H = Hearts

D = Diamonds
C = Clubs

Symbol of numbers:
A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Symbol of upturned or downturned:
Up = Upturned
Down = Downturned
Sample from "solitaire.txt":
R H 8 Down
B S 9 Up

You are going to read "solitaire.txt" and initialize the lists. First of all, you are going to print out Top List, then Board Lists by indicating lists as "1. list", "2.list", "3.list" and so on, and then you are going to print out Foundation Lists which are indicated by "Spades", "Hearts", "Diamonds", and "Clubs" **(20 points)**. Remember Foundation Lists are empty initially. In your program, **you will always update and list the current situation of the Top List, Board Lists and Foundation Lists, and then list a menu which asks the next operation in the game**. The menu will be like below:

Choose an operation:

1. Select from Top List to Foundation Lists
2. Select from Top List to Board Lists
3. Move on the Board Lists
4. Move from Board List to Foundation List

Please enter your choice (1, 2, 3, or 4):

- If the user selects **1**, the program will print out: **(20 points)**

Select a card from Top List:

The user will insert identity of a card in "Symbol_of_colors Symbol_of_suits Symbol_of_numbers" format. And the program will move this card to the related Foundation List if it is valid to move this card to a Foundation List (remember, Foundation Lists hold cards in the increasing order in the category of suits). If the movement is successful the program will print out "Movement is successful!", otherwise it will print out "Wrong movement!". For example, if the user enter "B S A" which means "Black Spades Ace", the program will move this card to "Spades Foundation List" since it is a valid operation (since Ace is a card with the smallest value it can be moved to an empty Foundation List") and print out that the movement is successful. For example, after this operation if the user selects operation 1 and wants to move "B S 3" card to Foundation Lists from Top List, the program will print out that the movement is wrong (because "B S 2" have to be inserted after "B S A").

- If the user selects **2**, the program will print out: **(20 points)**

Select a card from Top List:

Select the number of the destination Board List:

The user will insert identity of a card in "Symbol_of_colors Symbol_of_suits Symbol_of_numbers" format to select a card from Top List, and then insert the number of a

Board List to select a destination Board List. Then the program will move this card to the selected Board List if it is valid to move this card to this Board List (remember the rule for sorting cards on the board.). If the movement is successful the program will print out "Movement is successful!", otherwise it will print out "Wrong movement!".

- If the user selects **3**, the program will print out: **(20 points)**

Select the number of the source Board List:
Select the number of the destination Board List:
Select a card from the selected source Board List to determine starting point of movement:

The user will insert a number to select a source Board List, and a number to select a destination Board List, and then insert identity of a card from the selected Source Board List in "Symbol_of_colors Symbol_of_suits Symbol_of_numbers" format to determine starting point of the movement. Then the program will move the cards starting from the selected card from Source Board List to the Destination Board List if it is valid to move these cards to this Board List (remember the rule for sorting cards on the board.). If the movement is successful the program will print out "Movement is successful!", otherwise it will print out "Wrong movement!".

- If the user selects **4**, the program will print out: **(20 points)**

Select the number of the source Board list:

The user will insert a number to select a source Board List. Then the program will move the upturned card which is frontmost of the selected Board List if it is valid to move this card to a Foundation List (remember, Foundation Lists hold cards in the increasing order in the category of suits). If the movement is successful the program will print out "Movement is successful!", otherwise it will print out "Wrong movement!".

Please examine the given "**SAMPLE RUN.pdf**" file. The program has to run as shown in the "SAMPLE RUN". The characters printed by red are just samples which are inserted by a user.

Submission Details:

Your program **should be compiled and run on Linux** environment using **g++**. Include all necessary header files to your code. Do not use precompiled header files, and header files and functions which are specific to Windows environment. Keep the input file name as written (**solitaire.txt**), and read it from your code. Add comments to your code to explain what you did. Make sure to include the following heading in the beginning of your code:

```
// Student Name:
// Student ID :
```

Please just zip all .cpp and .h files for uploading your project. All projects should be submitted through Ninova, other projects will not be graded. You can discuss your questions through Ninova or you can contact with Meryem Uzun-Per (uzunper@itu.edu.tr).

Policy: You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual project. Plagiarism and any other forms of cheating will have serious consequences, including failing the course.