HACETTEPE UNIVERSITY COMPUTER ENGINEERING

BBM203 Programming Lab.

ASSIGNMENT 3

MELTEM TOKGÖZ

21527381

b21527381@cs.hacettepe.edu.tr

Subject : Linked Lists

TAs: Merve Ozdes

Programming Lanuage: C++

Due Date: 16.12.2018

1. Software Usage

Software input is input.txt and operations.txt. Input files reads such that an argument. Software output is write output.txt. (with argument) I wrote with CodeBlocks in C++.

2. Software Design Note

In this experiment, you are supposed to develop a simple search engine that uses linked list.

2.1 What is the problem?

In this assignment I have 2 input file.Firstly, I will have input.txt which has footballer's information on each line about a football league. In each row,there are the name of the footballer, the team name of footballer who scored, the name of the away team, minute of the scored goal and the match ID. Each line is read and parsed one by one, you need to create your linked lists according to this information and add the data you have read from the file to the linked list by the adding method.
Secondly, operations.txt file. In operations.txt, each line corresponds to one item. And each line has two parameters separated by commas.For each line necessary different functions.

2.2 Solution

I used the linked list structure in my homework and did all of the functions by navigating over those linked lists.

I first read the input.txt file in my homework. Then I created two structures to create a linked list. These are footballer and goal structures. Then I separated each line with a comma and added node it to the linked list (list1) that I created for football. In doing so, I took into account the alphabetical order as requested from me. If the same player comes again, I haven't added it to List1 again.I've just added the goal information to my 2nd linked list (list2). Then I started to write functions for the desired in the output. In addition, I have read the operations.txt file after this process and I wrote the necessary functions for it.I summarize all of my functions below.

2.3 Function

1. **int main(int argc, char** argv)**

This function is the main function in which the program is started. It takes two input files and one output file name as an arguments. In this function, the 1st linked list is created and the addNode function is called in order to add the required players. This function, input files are read in line by line and the required functions are called.

2. **void addNode(string name, string team, string a_team, int mnt, int m_id)**
   This function checks whether the footballer to be added is in the linked list. If he is not in the list, he will add the footballer to the connected list and add the goal information to the 2nd linked list created for each distinct footballer. If the player is on the linked list, he will find the footballer and just add the goals information to the second linked list.

3. **bool check(string name)**
   This function checks whether the name to be added is in the linked list. If name exist the list, function return true.

4. **void mostScore(ofstream& outfile)**
   This function takes the output file as a parameter. This function moves on each footballer (1st with linked list). Calls up the totalFirst () and totalSecond () functions for each player.
   - **int totalFirst()**
   - **int totalSecond()**

   These functions moves on the second linked list. The sum of the goals scored in each half is thus found and a return total goal numbers.

   The values returned from these functions are collected for each player. The most goals are found in which half of the match

5. **void bestScore(ofstream& outfile)**
   This function takes the output file as a parameter.Finds the best goal scoring by navigate the goal linked list.
   And the goalScorer() function calls.

6. **void goalScorer(int best, ofstream& outfile)**

This function takes the output file and best score as a parameter. Each footballer the best goal scoring compare best score (given a parameter)by navigate the goal linked list.If equal or grater, write output file.

7. **void hattrick(ofstream& outfile)**
   For each player, he calls the hattrickGoals() function and checks whether he is hattrick.If so, it will print to the output.
8. **bool hattrickGoals()**
   It creates a map like this.
   map<footballer_id, total goals number> my_map
   And checks whether he is hattrick by navigate the map.If so, it return true.

9. **void printTeam(ofstream& outfile)**
   list<string> teamList
   I created a list structure and added the teams to that list (No duplicate team name) and I wrote that list.

10. **void printFootballer(ofstream& outfile)**
    By navigate the footballer linked list.I am writing each footballer name.

11. **void matchesF(string name, ofstream& outfile)**
    I find the names given as parameters from the football player list and I am writing their information on with navigate 2 linked lists(footballer linked-list and goals linked-list).
    - **void printGoal(string name,ofstream& outfile)**

    I use the x function to navigate the second linked list.

12. **void mIDasc(string name,ofstream& outfile)**
    The footballer in the given name finds the from the linked list and calls the matchIDasc() function for writing ascending order his goals.

    - **void matchIDasc(string name,ofstream& outfile)**
      This function keeps the match id in a list structure (no duplicate match id in the list) and I sorted the list in ascending order.I wrote that list.

13. **void mIDdsc(string name,ofstream& outfile)**
    The footballer in the given name finds the from the linked list and calls the matchIDdsc() function for writing descending order his goals.
    - **void matchIDdsc(string name,ofstream& outfile)**
      This function keeps the match id in a list structure (no duplicate match id in the list) and I sorted the list in descending order.I wrote that list.