HACETTEPE UNIVERSITY DEPARTMENT OF COMPUTER ENGINEERING
BBM203

Prepared by Burak YILMAZ
Number : 21627868
E-Mail : burak040898@gmail.com
Subject : Assignment 3

Comments

I compiled my project on dev g++ -std=c++11 main.cpp as you mentioned on piazza because otherwise some libraries make trouble and I created a makefile.

MAIN GOAL

In this experiment, our aim is design two linked list structures for storing the footballer's and their goals of matches (can be different match or same match)

PROBLEM

In this assignment, we 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.. The point is the information can be mixed.

SOLUTION

I built two different linked list one is a single linked list the other one is double linked list but actually the second linked list is a part of the first linked list. I store the name of the footballer and his team and one pointer of the second linked list. So, I can access the footballers by using the main linked list and using the pointer of the other linked list I can access each footballer scores and matches. After built the linked list I just read operations.txt and did some calculations about them considering the rules that given in the pdf.

ALGORITHM

Firstly, I read input.txt file and create the linked lists that I mentioned above. So, in linked lists I can access each player besides I can access the goals and matches of each player. While creating linked list I compare each footballer name and built my linked list by considering the ascending names of the footballers. For each player, I create another linked list for their goals and while I was creating that linked list I consider the ascending order of match Id's. So, for the future calculations It made my job easier. After the calculations, I read the second file line by line and get the names

from the lines and calling the related function to do more calculations like printing the given players matches ...

FUNCTIONS/METHODS/STRUCTS

```
typedef struct Away_Team{
        string against_team;
    int score_time{};
    int match_id{};

Away_Team *awayTeam_prev{};
    Away_Team *awayTeam_pointer{};
}Away_Team;

typedef struct Home_Team{
        string name;
    string home_team;
    Away_Team *awayTeam_pointer{};
    Home_Team *next_player_pointer{};
}Home_Team;
```

I used these structs for storing the data.

Home_Team* file_reader(char*,Home_Team*);

-This function for reading input file line by line and calling an auxilary function to create the footballers and returns the linked list of these footballers.

Home_Team* create_player(const string &,Home_Team*);

-This function for creating each player.It splits each line with delimiter comma and assign the values to corresponding feature of the variable.

Home_Team* add_player(Home_Team*,Home_Team*);

-This function is the most important function because ıt basically creates the linked list correctly. It considers the alphabetic order(ascending) of the footballer's name and for each player ıt finds the correct location and locate them to the correct order at the same time ıt creates the other linked list for each player. If the current player already added to main linked list ıt determines that and If it encounters with the same player name again, It knows that it is another match or goal of the player so it adds only the away match linked list

of each player by considering the match id of the given line. Finally, it uptades the linked list and that linked list is type of double linked list.

void operations(Home_Team*,ostream&);

-This function for the first 5 operations. It finds the top scorer, most scored half, the names of the footballers who scored hat-trick, list of the teams, and all the footballers who scored on that league and writes the results to the output file.

void operations2(char*,Home_Team*,ostream&);

-This function for the rest of the calculations starting from sixth to eighth. For the given footballers in each line ıt calls the corresponding auxiliary function to perform the operations and after the calculations it writes the results to the output file.

void match_finder(Home_Team*, const string &,ostream&)

-This functions finds the all matches of the given footballer name and writes the all results to the output file.

void ascending_order_printing_matches(Home_Team*, const string &,ostream&);

-This function writes the given footballers matches to the output file by considering the order of ascending match id's.

void descending_order_printing_matches(Home_Team*, const string &,ostream&);

-This function writes the given footballers matches to the output file by considering the order of descending match id's and it calls the an auxiliary function (print recursively function) to write the results like that way.

void print_recursively(Away_Team*, const string &,int,ostream&);

-This function writes the matches of given player recursively because in linked list I stored the matches by considering the order of ascending match id's. So, it should start writing from the last element of that linked list.