Hacettepe University
Computer Science
and Engineering Department

Name and Surname	**:** ATAKAN AYYILDIZ
Identity Number	**:** 21526681
Course	**:** BBM203 PROGRAMMING LAB
Subject	**:** Linked Lists
Project	: ASSIGNMENT 3
Experiment	: Developing a simple search engine.
Data Due	**: 16.12.2018 (23:59:59)**
Main Program	**:assignment3.cpp**

PROBLEMS:

1.I have to create a single linked list that stores player datas then this linked list contains another different linked list that stores away team datas.

2.C++ does not have split function. So i have to write my split function.

3.To create sorted linked list so there has to be a function to control where the datas place.

4.On the other hand if name of the next line is already in linked list we iterate over down node and insert it to the next down node.

5.Also I have to sure that linked list do not enter NULL. Otherwise we can not go back.

6.After passing datas to the linked list we ensure that passing linked list to functions without passing NULL

SOLUTIONS

1.First of all we have to read input file line by line. Therefore i used getline function 2 times it is not efficent but it is guaranteed.Then datas pass to the insert function.

2.C++ does not specific split function like strtok in C so I have to use Container, stringstream and vector<string>. Using them was simple, they do not have any disadvantages.

3.Linked list has to be sorted. So insert function control linked list is empty or new player name ASCII number smaller than first node or greather.

4.Otherwise new datas playername is already in linked list and we have to insert goal datas to down node with insertdown function. Therefore we have to control where the goal data inserted.

5.I used call by value method.It seems quite simple but that uses too much memory more than call by reference.

FUNCTIONS

First two functions are inserting datas to linked list. (Insert and Insertdown functions).

1)Insert function:

Parameters:

Playerdata struct and vector string (they store footballer name –team name -away team name datas) and goal minute-match id.

This function returns linked list its type Playerdata.

Function controls 4 conditions.

First condition is that linked list is null if it is create new Playerdata and operate.

Second condition is if we want to add a player that starts with “A” and first footballer in the linked list start with “B” we have to add leftward new playerdata.

Third condition is that iterate until new datas player name alphabetically sorted.

Fourth condition is if footballer scores again go Insertdown function.

2)Insertdown Function:

Parameters:

Teamdata struct and vector string and int goal minute-match id.

Function returns a linked list its type Teamdata.

Function controls 2 condition.
First condition is if first down match id is greater than new match id.
Second condition is that iterate until match id is find the appropriate place.

3) MostScoredHalf Function:

Parameters:

2 int counters.First counter stores first half goal number.Second counter holds second half goal number.

Ofstream & output file name .

4)GoalScorer Function:

Parameters:

Linked list type of Playerdata.

Ofstream & output file name .

Function iterate each double list of a single linked list until reach Null then double linked list go back initial point.

5) HatTrick Function:

Parameters:

Linked list type of Playerdata.

Ofstream & output file name .

It controls second and third down nodes are NULL if so go next iter. Otherwise control 1st and 3rd match ids are equal if so print that and go next iter.

6)Teams Function:

Parameters:

Linked list type of Playerdata.

Ofstream & output file name .

In this function i create a vector <string> that holds each teams only one times.And iter type of Playerdata.

Iter is iterate until NULL.

And used variable says team name is already used or not. Each node of double linked list is controlling.Then cotrolling node is not NULL. Then teams printing.

7)DisplayPlayers Function:

Parameters:

Linked list type of Playerdata.
Ofstream & output file name .

That prints playername of each node.

8)FootballerMatches Function:

Parameters:

Linked list type of Playerdata.
String player name.
Ofstream & output file name .

That compares intended player name is same as linked list nodes player name. If so show every match of the player.

9) AscendingOrder Function:

Parameters:

Linked list type of Playerdata.
String player name.
Ofstream & output file name .

Function iterate until NULL.
After control names it iterate over doubly linked list untill NULL then it prints.

10) DescendingOrder Function:

Parameters:

Linked list type of Playerdata.

String player name.

Ofstream & output file name .

This function works same as ascendingOrder but this function go down until NULL then go up.

MAIN FUNCTION:

First create a player type of Playerdata.It holds NULL.

Then files are reading then parsing after that passing function with call by value method.

DATA STRUCTURES:

I used to diffrent linked list.

First doubly linked list calllig as Teamdata

It holds string away team name, int minute-match id and *down-*up type of Teamdata.

Second single linked list calling as Playerdata

It holds string player name and team name.

*next type of Playerdata. *down type of Teamdata.(Holds doubly linked list).