Hacettepe University

Computer Science and Engineering Department

NAME & SURNAME	Ali Utku ÜNLÜ
ID	21228817
COURSE	BBM203 - Programming Lab
EXPERIMENT	Assignment III
SUBJECT	Linked Lists
DUE DATE	16/12/2018 23:59:59
ADVISORS	Alaettin Uçan, Merve Ozdes, Pelin Canbay, Selma Dilek
E-MAIL	aliutkuunlu@gmail.com

1- Introduction

In this assignment this topic titles expected from us.

- Linked List implementation
- Manipulating Linked Lists
- Read- Write Files

In order to do that we have a simulation like this:

We have an input file and it consists some knowledge about some players and their information.

For example this is a line from the input file;

“Burak Bulut,Denizli,Pamukkale,37,0”

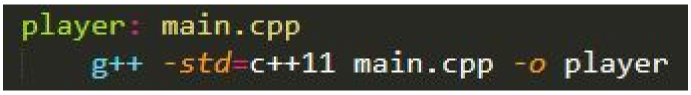
The line tell us this player’s name is “Burak Bulut”
Player’s team name is “Denizli”
Player’s away team name is “Pamukkale”
The minute (37) of player’s goal in the given match id (0)

We will create a linked list which holds information about player’s name, player’s team name and a linked list which holds information about player’s goals (which team the player score against it and minute of goal and match id).

Arguments:

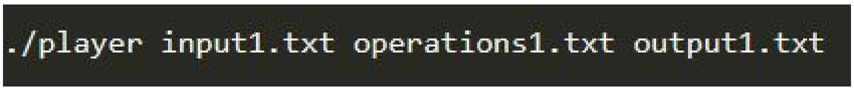
I created a makefile because some of functions that i used in this assignment does not work on dev machine.

The makefile is this.



```
player: main.cpp
    g++ -std=c++11 main.cpp -o player
```

And the command line is:



```
./player input1.txt operations1.txt output1.txt
```

2- Solution

I have 3 classes in this assignment which are Goal, Player and PlayerList

The Goal class consists:

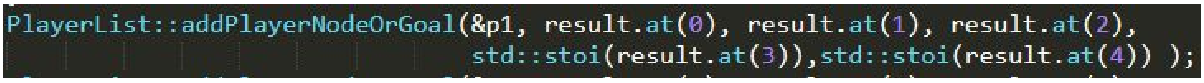
- Next Goal object’s pointer as a Goal * type
- Previous Goal object’s pointer as a Goal * type
- Away team’s name as a string type.
- Minute of goal as an int type
- match id as an int type

The Player class consists:

- Next Player object’s pointer as a Player * type
- Previous Player object’s pointer as a Player * type
- Player’s name as a string type
- Player’s team name as a string type
- Player’s Goals Linked List’s head pointer as a Goal type
- Player’s total goal count as an int type

The PlayerList class has functions for implementing linkedlists and manipulating them.

i read input files line by line than i splitted it. and i send those tokens to addPlayerNodeOrGoal function as a parameters.



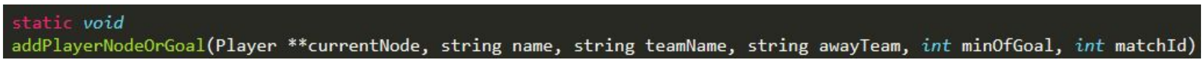
```
PlayerList::addPlayerNodeOrGoal(&p1, result.at(0), result.at(1), result.at(2),
                                std::stoi(result.at(3)),std::stoi(result.at(4)) );
```

3- Functions

a- addPlayerNodeOrGoal function:

This function creates linked list for players and player’s goals linkedlist.

Its parameters are like this:



```
static void
addPlayerNodeOrGoal(Player **currentNode, string name, string teamName, string awayTeam, int minOfGoal, int matchId)
```

It does not return a value just initialize the structure.

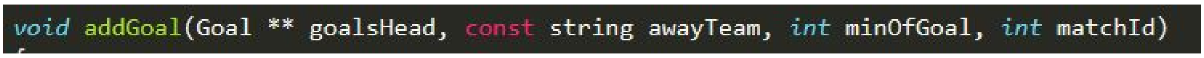
In this function i call some functions they are mostly the attributes’ getters and setters functions. Because they are private.

Other functions are:

a1- addGoal:

This function creates linked list for player’s goals.

Its parameters like this:

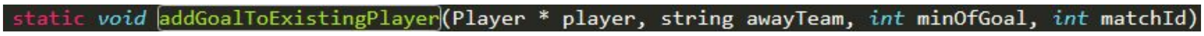


```
void addGoal(Goal ** goalsHead, const string awayTeam, int minOfGoal, int matchId)
```

a2- addGoalToExistingPlayer

This function creates linked list for existing player's goals.

Its parameters like this:

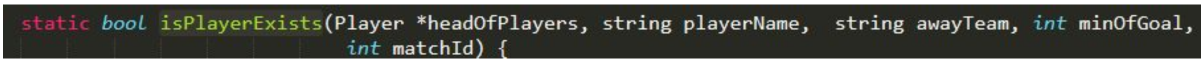


```
static void addGoalToExistingPlayer(Player * player, string awayTeam, int minOfGoal, int matchId)
```

a3- isPlayerExists

This function controls that if given player exists than return true if it is not than return false.

Its parameters like this:

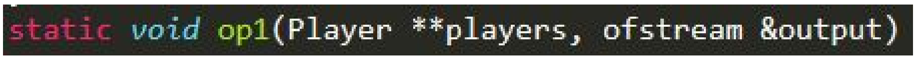


```
static bool isPlayerExists(Player *headOfPlayers, string playerName,  string awayTeam, int minOfGoal,
                           int matchId) {
```

b- op1 function:

This function was created for operation 1 and it calculates the most scored half of all matches

Its parameters like this:

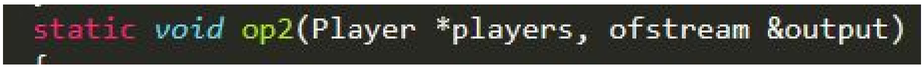


```
static void op1(Player **players, ofstream &output)
```

c- op2 function

This function was created for operation 2 and it calculates top goal scorer players.

Its parameters like this:

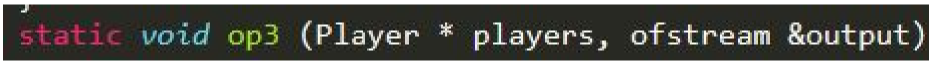


```
static void op2(Player *players, ofstream &output)
```

d- op3 function

This function was created for operation 3 and it calculates the hat-trick players.

Its parameters like this:

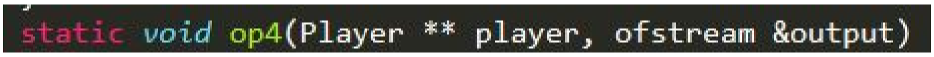


```
static void op3 (Player * players, ofstream &output)
```

e- op4 function

This function was created for operation 4 and it prints the teams in inputfile.

Its parameters like this:

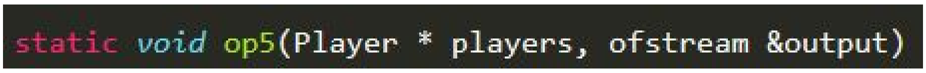


```
static void op4(Player ** player, ofstream &output)
```

f- op5 function

This function was created for operation 5 and it prints the players in inputfile.

Its parameters like this

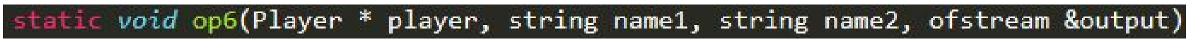


```
static void op5(Player * players, ofstream &output)
```

g- op6 function

This function was created for operation 6 and it prints the matches and goals of given player.

Its parameters like this

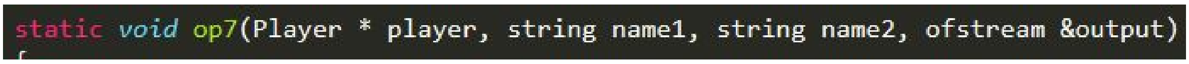


```
static void op6(Player * player, string name1, string name2, ofstream &output)
```

h- op7 function

This function was created for operation 7 and it sorts player’s match by matchID of given player.

Its parameters like this



```
static void op7(Player * player, string name1, string name2, ofstream &output)
{
```