Hacettepe University
Computer Science and Engineering Department
Name and Surname : Burak Karademir
Identity Number : 21527123
Course : BBM 203
Experiment : ASSIGNMENT 3
Subject : Linked Lists
Due Date : 25.11.2018 (23:59:59)
Advisors :Alaettin Uçan, Merve Ozdes, Pelin Canbay, Selma Dilek
e-mail : karademirburak@outlook.com
Programming Language : C++

2. Software Using Documentation

2.1. Software Usage

I used netbeans for this assignment.My assignment takes 3 command line arguments .I use dynamic memory allocation for linked list. We have two input files and one output file.My program reads input files and writes output to a text file which name is given with command line arguments.Also input file names are given by command line arguments.

3.Software Design Notes

3.1. Description of the program

3.1.1. Problem

We have two input files.First one includes footballer's goal informations.(footballer's team, away team, goal minute, match id) and other one includes operations about first input. Firstly we need to find the most scored half.Then we need to find the goal scorer. After that we need to find the footballers who scored hat-trick.Then we must find list of teams in input1.Then we need to find list of footballers. After that we have to read operations file and it includes three lines for sixth seventh and eighth items.In seventh item we find matches of given footballers. In seventh and eighth items we find footballers matches and print it ascending and descending order according to match id.

3.2. System Chart

INPUT	PROGRAMS	OUTPUT
input.txt operations.txt (given by command line)	main.cpp	output.txt(given by command line)

Name of the input and output file can change since we take the name of the inputs and output from the command line.

3.3. Main Data Structures

I used linked list , doubly linked list in this assignment.I also used dynamic memory allocation for linked list and structs.

3.4. Algorithm

In the main firstly I open a file for writing outputs which name is taken by command line arguments.(argv[3]). Then I find input1 and operations file's line number with function find_line_number and find_operation_linenum. Then I use dynamic memory allocation for my structs. After that I read my input files with function read_file and read_operation_file. Then I add records about footballers which are in input file with function addFootballer.In addFootballer function firstly I search if the footballer is added before.If it is added before then I add that record to down of that footballer with down pointer. If it is adding firstly than I add the end of the linked list. By the way I use dynamic memory allocation for my linked list. After that in main function I determine the goal minutes and check if the most of the goals scored in first half or second half. If it is first half then I print 0, if it is second half then I print 1 and ıf the goals which are scored in first half and second half are equal then I print -1. After that I call largestElement function for finding most goal scorer footballer. Then I call hattrick_finder function for finding the footballers who scored hat-trick. In hattrick_finder function I use my linked list and search it element by element and in every element I check the down linked list and I check the same footballers goals. If the away team and match id is same I increase a counter which is declared in function. After that function I call print_teamlist in the main. Again it uses my linked list and a struct for determining if the team is already printed. This function prints the teams list. After that in main I call print_footballers function for printing the name of the given footballers. This function use my linked list and before printing the footballer names it checks the same footballers with a struct. After printing the

footballer list I call print_operation1 function from the main. In that function I find the matches of the footballers using linked list.
Footballer names are given in the operation.txt file's first line. After printing the matches of given footballers, I call print_operation2 for finding the given footballer's ascending order according to the match id. In the print_operation2 function I find the matches of the footballers (footballer names are given in operation.txt second line) and I did not make any alignment since the input file is in the ascending order. After that I call print_operation3 function and it prints the given footballer's descending order according to the match id. I did same things with function print_operation2 except I reversed the order of the given input file. I print everything to a output file using ofstream.