

HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING BBM203



Name : Mehmet Taha

Surname : Usta

Number : 21527472

E~mail : b21527472@cs.hacettepe.edu.tr

Subject : Linked Lists

Programming Language : C++

1. Problem Definition

Develop a simple search engine that uses Singly Linked List and doubly linked list

2. Methods and Solution

Firstly, Start with the empty linked list(singly linked list).

Readinputfile() function runs. The readinputfile () function starts to read the input file. It splits each lines and stores data in variables.

linked list is checked after obtaining desired variables. the new player is not created if the same player exists. A new doubly linked

list(MinAndMatch) is created. The player's MinAndMatch (minute, match_id, away_team) structure is added to the end. If the same player does not exist, the new player is created. New player is added from head(singly linked list) to head structure. The input.txt is read and the players are created. then the input.txt file is closed. The

Singly linked list structure is sorted by name using the

single_MergeSort () function. Goals of all players are sorted by match_id with doubly_mergeSort function. Then begins the

read_operations_and_output function. Read_operations_and_output function reads operation.txt as first, then data is stored in variables.

Output file is created. To calculate the the most scored half, all linked list moves in the forward direction. the most scored half and max goal are determined. 0 or 1 in the output file according to the result.

To calculate Goal Scorer, all linked list moves in the forward direction.

If the number of goals of the players is equal to the max_goal result is written to the output file. To calculate Hat-Trick, all linked list moves

in the forward direction. If each player has scored more than three goals in the same match, the result is written to the output file. List of

Teams to calculate all the linked list moves in the forward direction.

Each player's team names are written to the output file. Same team

names are continuously checked to avoid being written to the output

file. To calculate List of Footballers all linked list moves in the forward direction. The names of the players are written to the output file. To calculate the Given Footballers of Matches, all linked list moves in the forward direction. Matching names are written to the output file. Ascending Order According to Match Id All linked list moves in the forward direction to calculate. Matching names are written to the output file. Output is sorted by match ID in ascending order of given footballer. Descending Order According to Match Id All linked list moves in the forward direction to calculate. Matching names are written to the output file. Output is sorted by match ID in descending order of given footballer. Finally deallocates the memory previously allocated with using delete function. Then programs ends.

3. Functions Implemented and not Implemented

ifstream () and ofstream () = Function is used to open a file to perform operations such as reading, writing etc.

Stoi() = Converts the string argument str to an integer (type int).

new()=Allocates the requested memory and returns a pointer to it.

While() = loop in C programming repeatedly executes a target statement as long as a given condition is true.

getline() = Function is used to read a file line by line.

For() = A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

out = Function is used to write the “character, string, float, integer, octal and hexadecimal values” in output file.

delete = Deallocates the memory previously allocated by a call to calloc, malloc, or realloc.

.close() = Function closes the file that is being pointed by file pointer fp

Typedef struct{} = Defines a physically grouped list of variables to be placed under one name in a block of memory, allowing the different variables to be accessed via a single pointer, or the struct declared name which returns the same address

single_append() = Given a reference (pointer to pointer) to the head of a list and an int, appends a new node at the end

doubly_append() = Given a reference (pointer to pointer) to the head of a DLL and an int, appends a new node at the end

single_FrontBackSplit() = Split the nodes of the given list into front and back halves, and return the two lists using the reference parameters. If the length is odd, the extra node should go in the front list. Uses the fast/slow pointer strategy.

single_SortedMerge() = Takes two lists sorted in increasing order, and splices their nodes together to make one big sorted list which is returned.

single_MergeSort() = Sorts the linked list by changing next pointers (not data)

split() = Split a doubly linked list (DLL) into 2 DLLs of half sizes

doubly_merge() = Function to merge two linked lists

doubly_mergeSort() = Function to do merge sort

readinputfile() = Function to reads input.txt file , creates player and append(last node) linked list

read_operations_and_output() = Function to reads operations.txt file, creates output file , write necessary data

Makefile : Makefiles are a simple way to organize code compilation. I have a Makefile (g++ -std=c++11 assignment3.cpp)