1.Software Using Documentation

1.1. Software Usage

The program works by taking two input files.

- First file gives us informations of footballers splitted by commas.

<footballerName>,<teamName>,<awayTeam>,<minuteofGoal>,<matchID>

- Second file includes some footballer's name and we do some operations taking this footballer's name.

2.Software Design Notes

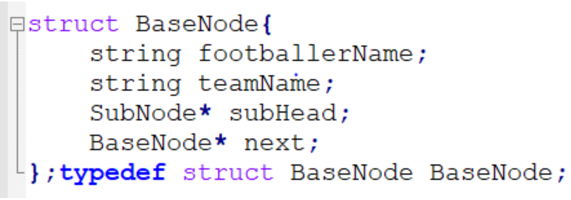
2.1. Description of the Program

2.1.1. Problem

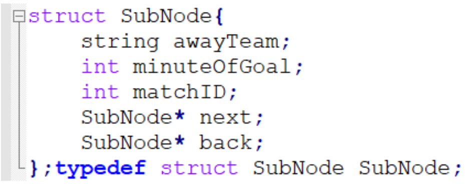
In this experiment, we are expected to write a program that creates a linked list structure. There is a base linked list which includes the name of the footballer, the team name of footballer, a pointer to point next footballer and a pointer which points to another linked list. Second linked list includes away team name, minute of goal, match ID, a pointer to point next member of the list and a pointer to point previous member of the list. After creating this structure, program prints to file requested information given in the assignment paper.

2.1.2. Solution

Since the program reads the information needed to create the linked list structure from the first file, we send the informations to the function<insertToBaseNode()> by separating each line according to the comma. There are two structure in the program.



```
struct BaseNode{
    string footballerName;
    string teamName;
    SubNode* subHead;
    BaseNode* next;
};typedef struct BaseNode BaseNode;
```



```
struct SubNode{
    string awayTeam;
    int minuteOfGoal;
    int matchID;
    SubNode* next;
    SubNode* back;
};typedef struct SubNode SubNode;
```

In the insertToBaseNode() function we create a new BaseNode and add it to list according to alphabetical order. After this we call the function<insertToSubNode> with required informations. In this function we create a new SubNode and add it to list according to match ID order. After creating the structure, the program moves on the list and performs the operations according to the desired output and print them to a file.