

ECE 71/191T – Data Structures and Algorithms

Dr. Gregory R. Kriehn, Fresno State
C++ Homework Assignment: Chapter 9

Code Due By: Midnight on Sat, Feb 04

Write-up Due By: Class on Tue, Feb 07

HOMEWORK #17 – Football!

It's the Super Bowl – time to keep track of some statistics for your favorite team!

Specifications:

Write a program that declares a **struct** called **footballPlayerType** to store the data of a football player (player's last name, player's position, number of touchdowns, number of catches, number of passing yards, number of receiving yards, and the number of rushing yards). Declare an array of structures to store the data of 10 football players.

Your program must contain functions to input data from a file and output data to a file. The input and output files are called "**PackersData.txt**" and "**PackersOutput.txt**", respectively. Add functions to print menu options, print all of the team data, print a single player's data, search the array to find the index of a specific player, update the data of a player based on the number of touchdowns, catches, etc. Before the program terminates, give the user the option to save the data. Assuming **N** is a constant variable expression, use the following function prototypes:

```
// Display the menu
void showMenu();

// Get team data from the input file
void getData(ifstream &inputFile,
             footballPlayerType list[N]);

// Save current team data to output file
void saveData(ofstream &outputFile,
              footballPlayerType list[N]);

// Print entire team data to the screen
void printData(footballPlayerType list[N]);

// Print a single player's data to the screen
void printPlayerData(footballPlayerType list[N],
                    int playerNum);

// Search the data for a player
int searchData(footballPlayerType list[N], string n);
```

```

// Update the number of touchdowns for a player
void updateTouchDowns(footBallPlayerType list[N],
    int tDowns, int playerNum);

// Update the number of catches for a player
void updateCatches(footBallPlayerType list[N], int catches,
    int playerNum);

// Update the number of passing yards for a player
void updatePassingYards(footBallPlayerType list[N],
    int passYards, int playerNum);

// Update the number of receiving yards for a player
void updateReceivingYards(footballPlayerType list[N],
    int recYards, int playerNum);

// Update the number of rushing yards for a player
void updateRushingYards(footBallPlayerType list[N],
    int rushYards, int playerNum);

```

A copy of the input data file can be found on the website.

Select one of the following options.

- (1) Print the Team's Data
- (2) Print a Player's Data
- (3) Update a Player's Touchdowns
- (4) Update a Player's Catches
- (5) Update a Player's Passing Yards
- (6) Update a Player's Receiving Yards
- (7) Update a Player's Rushing Yards
- (99) Quit the Program

Input Selection: 1

Name	Position	Touchdowns	Catches	Passing	Receiving	Rushing
Bill	QB	70	0	8754	0	577
Jackson	RC	55	87	59	5490	574
...						

Select one of the following...

Input Selection: 2

Enter player's name: Bill

Name = Bill
Position = QB
Touchdowns = 70
Number of Catches = 0
Passing Yards = 8754
Receiving Yards = 0
Rushing Yards = 577

Select one of the following...

Input Selection: 3

Enter player name: Bill

Enter number of touchdowns added: 7

Select one of the following...

Input Selection: 2

Enter player's name: Bill

Name: Bill
Position: QB
Touchdowns: 77
...

Select one of the following...

Input Selection: 4

Enter player name: Billy

Enter number of catches to be added: 3

Invalid player number.

Select one of the following...

Input Selection: 99

Would you like to save data (Y/N): Y

```
~> more PackersOutput.txt  
Bill QB 77 0 8754 0 57  
Jackson RC 55 87 50 5490 574  
...
```

Once you verify the operation of your program, submit your source code to the Grader Program.