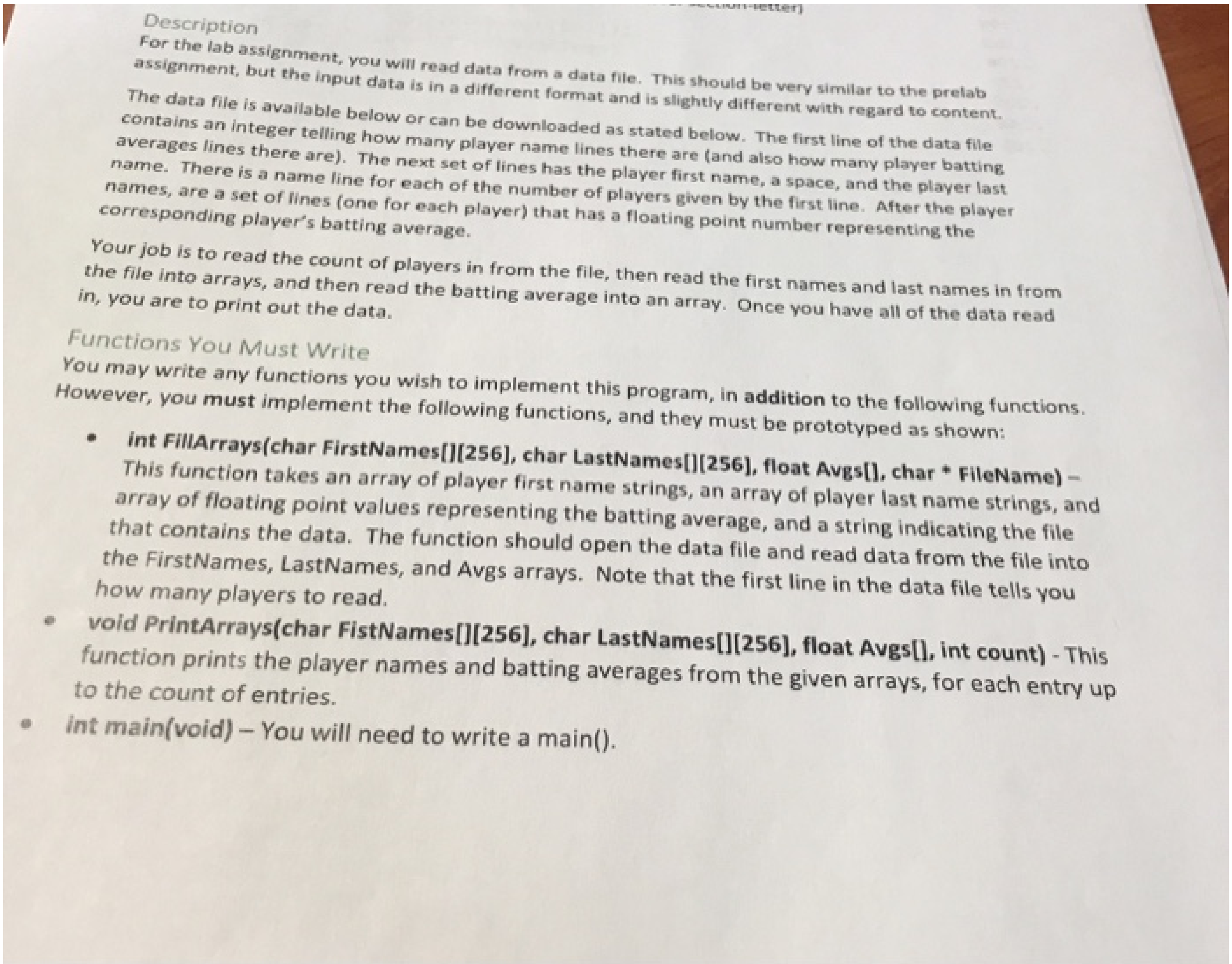


ction-letter)

Description

For the lab assignment, you will read data from a data file. This should be very similar to the prelab assignment, but the input data is in a different format and is slightly different with regard to content.

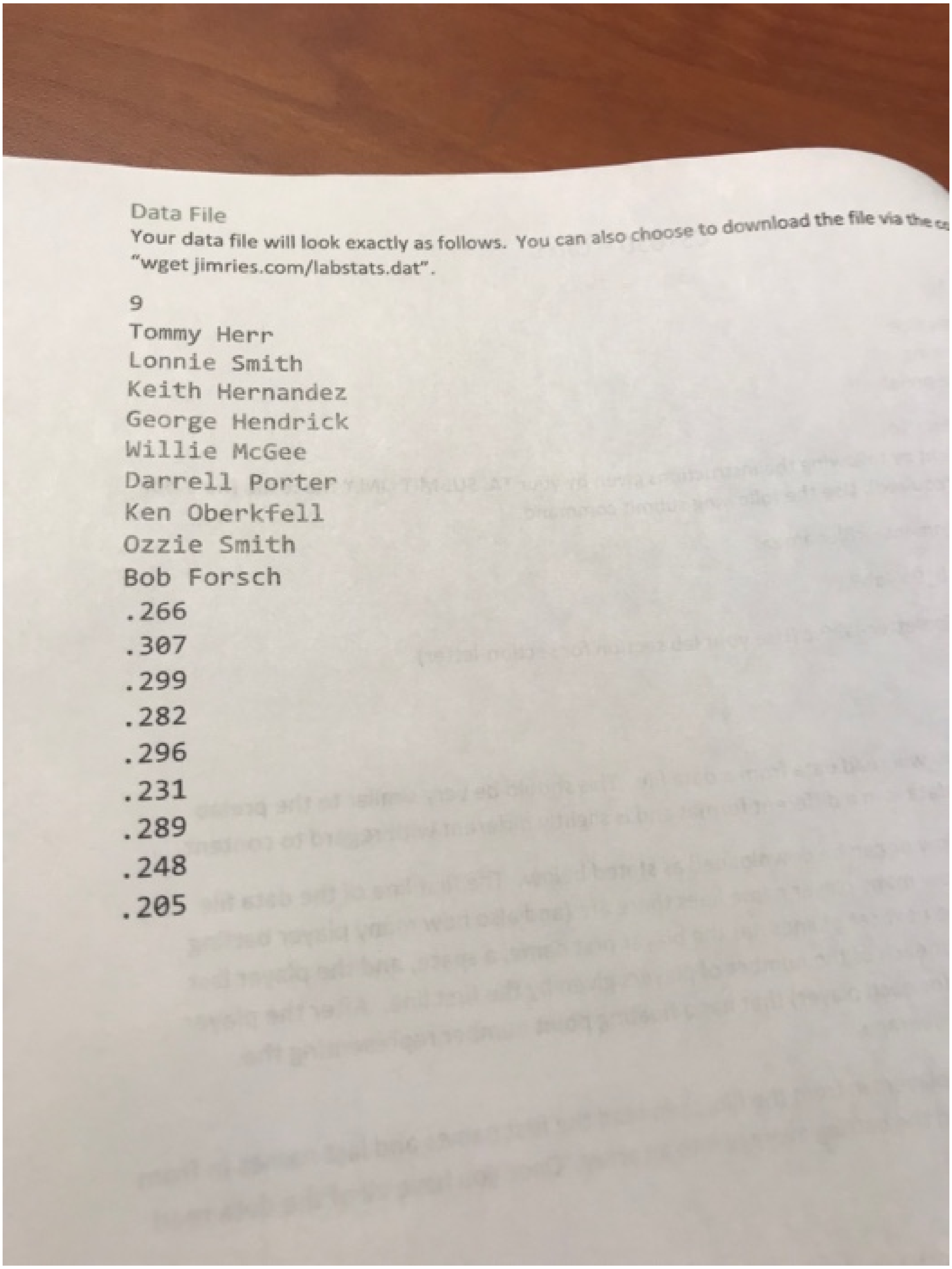
The data file is available below or can be downloaded as stated below. The first line of the data file contains an integer telling how many player name lines there are (and also how many player batting averages lines there are). The next set of lines has the player first name, a space, and the player last name. There is a name line for each of the number of players given by the first line. After the player names, are a set of lines (one for each player) that has a floating point number representing the corresponding player's batting average.

Your job is to read the count of players in from the file, then read the first names and last names in from the file into arrays, and then read the batting average into an array. Once you have all of the data read in, you are to print out the data.

Functions You Must Write

You may write any functions you wish to implement this program, in **addition** to the following functions. However, you **must** implement the following functions, and they must be prototyped as shown:

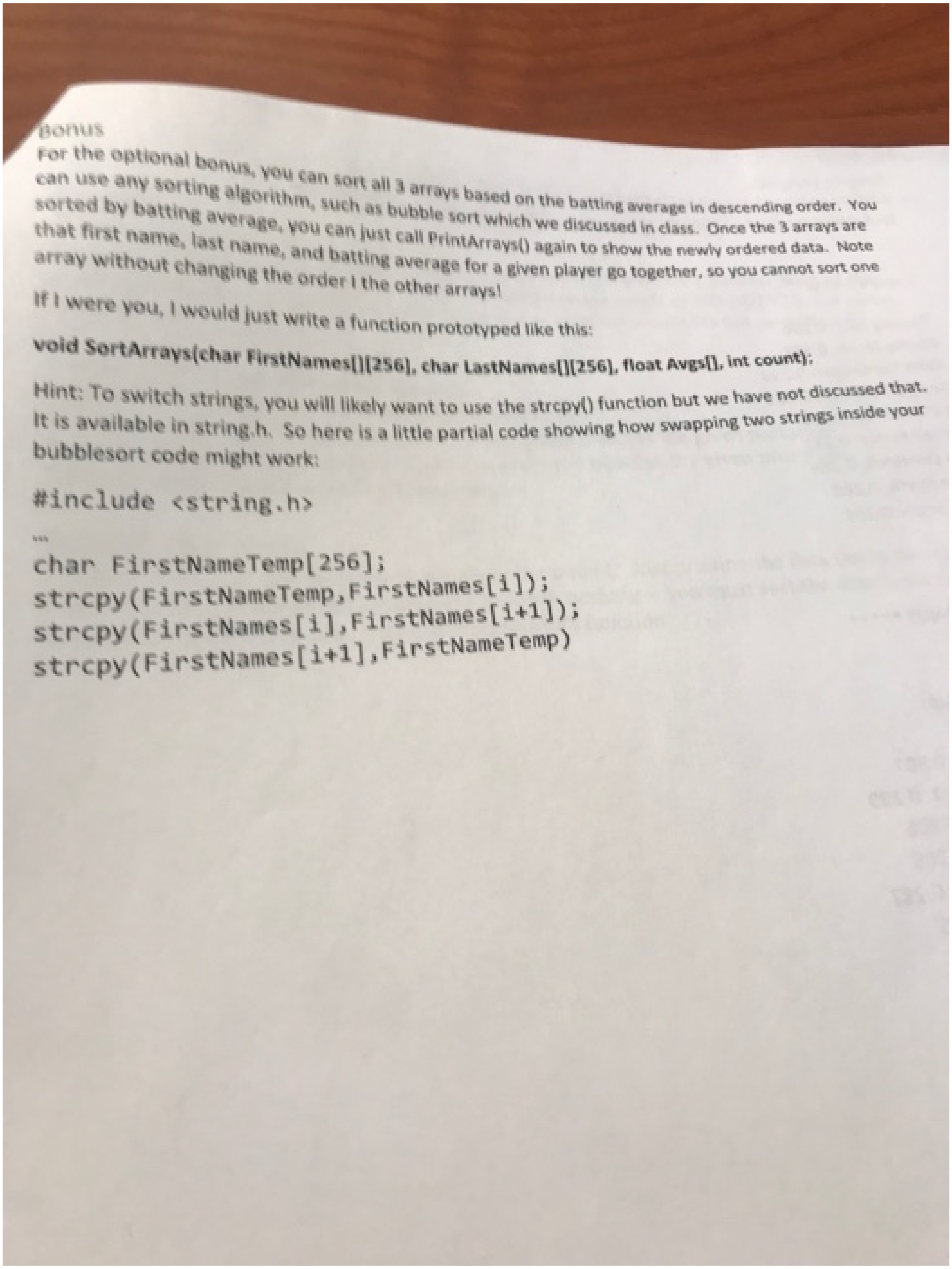
- **int FillArrays(char FirstNames[][256], char LastNames[][256], float Avgs[], char * FileName)** – This function takes an array of player first name strings, an array of player last name strings, and array of floating point values representing the batting average, and a string indicating the file that contains the data. The function should open the data file and read data from the file into the FirstNames, LastNames, and Avgs arrays. Note that the first line in the data file tells you how many players to read.
- **void PrintArrays(char FistNames[][256], char LastNames[][256], float Avgs[], int count)** - This function prints the player names and batting averages from the given arrays, for each entry up to the count of entries.
- **int main(void)** – You will need to write a main().



Data File

Your data file will look exactly as follows. You can also choose to download the file via the
"wget jimries.com/labstats.dat".

```
9
Tommy Herr
Lonnie Smith
Keith Hernandez
George Hendrick
Willie McGee
Darrell Porter
Ken Oberkfell
Ozzie Smith
Bob Forsch
.266
.307
.299
.282
.296
.231
.289
.248
.205
```



Bonus

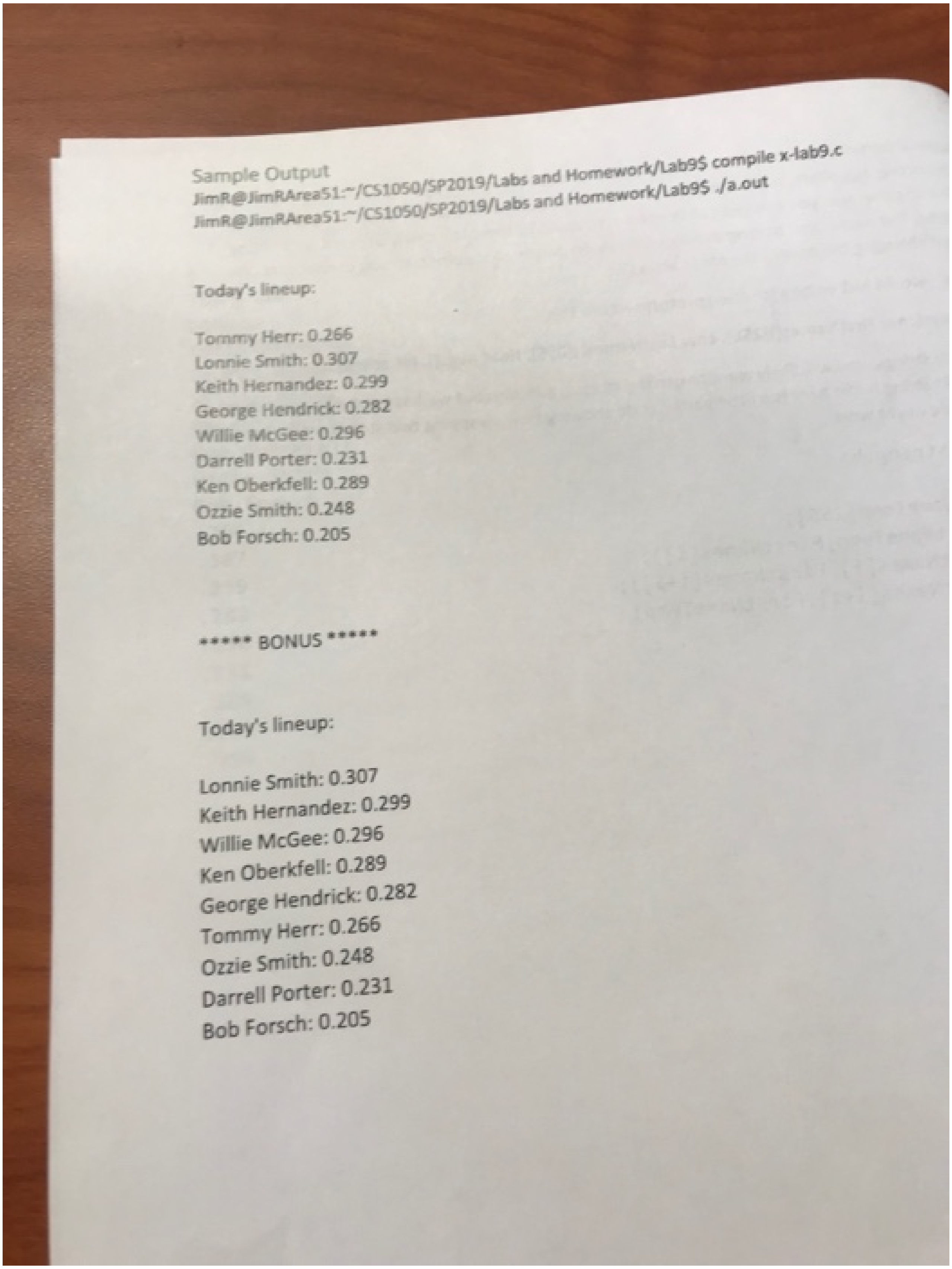
For the optional bonus, you can sort all 3 arrays based on the batting average in descending order. You can use any sorting algorithm, such as bubble sort which we discussed in class. Once the 3 arrays are sorted by batting average, you can just call PrintArrays() again to show the newly ordered data. Note that first name, last name, and batting average for a given player go together, so you cannot sort one array without changing the order I the other arrays!

If I were you, I would just write a function prototyped like this:

void SortArrays(char FirstNames[][256], char LastNames[][256], float Avgs[], int count);

Hint: To switch strings, you will likely want to use the strcpy() function but we have not discussed that. It is available in string.h. So here is a little partial code showing how swapping two strings inside your bubblesort code might work:

```
#include <string.h>
…
char FirstNameTemp[256];
strcpy(FirstNameTemp,FirstNames[i]);
strcpy(FirstNames[i],FirstNames[i+1]);
strcpy(FirstNames[i+1],FirstNameTemp)
```



Sample Output

```
JimR@JimRArea51:~/CS1050/SP2019/Labs and Homework/Lab9$ compile x-lab9.c
JimR@JimRArea51:~/CS1050/SP2019/Labs and Homework/Lab9$ ./a.out

Today's lineup:

Tommy Herr: 0.266
Lonnie Smith: 0.307
Keith Hernandez: 0.299
George Hendrick: 0.282
Willie McGee: 0.296
Darrell Porter: 0.231
Ken Oberkfell: 0.289
Ozzie Smith: 0.248
Bob Forsch: 0.205


***** BONUS *****


Today's lineup:

Lonnie Smith: 0.307
Keith Hernandez: 0.299
Willie McGee: 0.296
Ken Oberkfell: 0.289
George Hendrick: 0.282
Tommy Herr: 0.266
Ozzie Smith: 0.248
Darrell Porter: 0.231
Bob Forsch: 0.205
```