CSC 362 Programming Assignment #2
Due Date: Monday, October 10

In this assignment, you are to use multiple functions and multiple files. If a function is to compute and return a *single value*, it can do so using a return statement where you would call it using an assignment statement like `x = compute(a, b, c);`. Otherwise, if the function is required to compute or input *multiple values*, you *must pass* these values through the parameter list. You will have to decide which parameters to pass normally versus as addresses. The primary purpose of this assignment is to test your understanding of parameter passing; doing so incorrectly or improperly will result in a loss of points on your grade. Note: only pass parameters as addresses for those parameters that will (or may) *change* in the function; all other parameters should be passed by copy. Remember strings are already passed as addresses.

The program's purpose is to predict football games. It will do so by inputting from a disk file various ratings about two football teams. The file will contain a number of games where each row of the file will contain the information for a single game (two teams' worth). Input each row using a single fscanf instruction, pass the relevant data on to a compute function and then pass the prediction information on to an output function. This repeats until the program reaches the end of the file. Also accumulate some statistics for the end of the program. All information is returned to main to pass on to the next function. This is all explained in detail below.

Each row of the input file consists of 11 items, each separated by a space. These items are in order: HT (home team name, a string), HTO HTD HTS HTH HTC (home team's offensive strength, defensive strength, special teams strength, strength at home, home field advantage, all int values), VT (visiting team name, a string), VTO VTD VTS VTR (visiting team's offensive strength, defensive strength, special teams strength, and road strength, all ints). The int values are on a scale of 1 (worst) to 10 (best). Use a separate input function that uses fscanf to input all 11 values. As these variables are needed in main, you will have to properly pass parameters between main and input.

Given the 9 input int values, compute 5 "preference" values (here referred to as preference1, preference2, etc). Do this in a single compute function which receives the 9 int values, compute the 5 preferences and return them. The 5 preferences should all be doubles. Use a single compute function although if desired, this function can call other simpler functions, one per preference. The preference formulas are given on the next page. Items like OFFENSIVE_FACTOR are constants as defined on the next page. Make sure you declare all of the constants in a header file.

Main should then call a prediction function which will receive the 5 preference values and return its prediction (a double) which indicates how much the winning team will win by. A positive number for the prediction means a home team win and a negative number means a visiting team win. The prediction function must be a separate function from your compute function. Upon receiving the prediction, main should then call an output function to output the two teams and the prediction of who will win and by how much. This output should be a positive int value (use a cast, and to convert, you can use the math.h function abs). If the predicted value is 0 (a tie), then predict the home team will win by 1 point. Next, main will call an update function to maintain running totals on the number of games predicted and the number of home teams predicted to win. After processing all of the input (outside the main loop of the program), call a separate summary function to output the number of games predicted and the percentage of the home teams predicted to win.

Your program will consist of the following functions:

- main: declare variables, open the input file, iterate through the file (until EOF), and call the various functions listed below. The following functions are called from within the loop except for summary, which is called after. Outside of the loop, also close the file. You may of course name your functions whatever you like.
    - **input** the 11 data items from the file
    - **compute** the 5 preferences using the following formulas

preference1 = HTO * *OFFENSE_FACTOR* - VTD
preference2 = HTD + 2 – VTO * *OFFENSIVE_FACTOR*
preference3 = HTS * *SPECIAL_TEAMS_FACTOR* – VTS
preference4 = HTH + HTC ***** *HOME_FIELD_ADVANTAGE* - VTS
preference5 = HTO * HTD * HTH * *OVERALL_FACTOR* – VTO * VTD * VTR

Italicized items are constants (declared using #define in your header file) as defined below:

- `OFFENSIVE_FACTOR = 1.22`
- `SPECIAL_TEAMS_FACTOR = 1.13`
- `HOME_FIELD_ADVANTAGE = 1.1`
- `OVERALL_FACTOR = 1.2`

- **prediction**: given the five preference values, compute a prediction of who wins using the following function (remember a negative value means a visiting team win)

  sum = preference1 * *OFFENSIVE_WORTH* + preference2 * *DEFENSIVE_WORTH* + preference3 * *SPECIAL_WORTH* + preference4 * *HOME_WORTH* + preference5 * *OVERALL_WORTH*

  Italicized items are constants (declared using #define in your header file) as defined below:

  - `OFFENSIVE_WORTH = 0.27`
  - `DEFENSIVE_WORTH = 0.31`
  - `SPECIAL_WORTH = 0.18`
  - `HOME_WORTH = 0.09`
  - `OVERALL_WORTH = 0.15`

- **output** the winning team name and the losing team name and the amount, such as `Predicted Bengals over Steelers by 16.`
- **update**: modify running totals: number of games and number of home team wins predicted

- **summary** outputs the number of games predicted and the percentage of those games in which the home team is predicted to win. NOTE: This must be output as a percentage such as 71.33%.

Divide your functions into multiple files as follows:

1. A file containing your main function and an include statement for your header file
2. A header file that contains all function prototypes, all include statements for C libraries, and all constants used in your program. Prototypes should be commented as your .h file is visible but your .c files may not be
3. A file containing all of your computation functions (compute, prediction and update and the 5 preference computations if you separate them into other functions)
4. A file containing your I/O functions (input, output and summary)

Run your program on the two data files on the website. Example output for first file is given below. Hand in your source code and the output of running your program on the *second* of the two data files.

```
Predicted Cincinnati over Denver by 8
Predicted NY_Giants over Washington by 9
Predicted Indianapolis over San_Diego by 9
Predicted Pittsburgh over Philadelphia by 20
Predicted Dallas over Chicago by 29
Predicted New_Orleans over Atlanta by 1
Predicted Arizona over Buffalo by 20
Predicted Carolina over Minnesota by 12
Predicted LA_Rams over Tampa_Bay by 6
Predicted Kansas_City over NY_Jets by 9
Predicted Miami over Cleveland by 1

Out of 11 games predicted, 72.73% favor the home team
```