

Programming Assignment #3

Using 2-Dimensional Arrays

Two dimensional arrays can be thought of as tables of data. A familiar example of a table of data is the spreadsheet, such as Microsoft Excel, where you arrange data into rows and columns. To declare a 2-Dimensional array in C, we do the following:

```
const int NUM_ROWS = 8;  
const int NUM_COLUMNS = 5;  
int table[NUM_ROWS][NUM_COLUMNS];
```

The above, as you can probably guess, creates an array called `table`, that can hold/represent a table with 8 rows and 5 columns. Of course, by changing the named constants `NUM_ROWS` and/or `NUM_COLUMNS`, you can change the size of the table that is being created and represented. However, unlike a spreadsheet, 2-dimensional arrays in C must always start at index 0 for both the row index and the column index. By convention, the first index is used to index the row number, and the second index the column number. This convention is somewhat arbitrary, they could be reversed, but in most of mathematics and programming you will see people using this ordering when working with 2-dimensional tables, so it is a good idea to just memorize this ordering.

So, a visual representation of the 2-dimensional `table` variable, declared above, would be:

<i>table</i>	[0]	[1]	[2]	[3]	[4]
[0]					
[1]					
[2]					
[3]					
[4]		25			
[5]					
[6]					
[7]					

To assign a value into a 2-dimensional table, we do the same as for a 1-dimensional array, except we specify 2 indexes, the row, column. So, to assign the value 25 to the cell shown in the table, which is in the row with index 4 and the column with index 1, we would do:

```
table[4][1] = 25;
```

In this program, you will use and manipulate data in a 2-dimensional array.

Objectives

Become familiar with using 2-dimensional array to solve problems. Also, learn how to correctly pass a 2-dimensional array to a function.

Instructions

1. Download the PlayBall.cpp source file from the class web site. Also get the playerstats.dat file, which contains some data you will use in the program.
2. The PlayBall.cpp source file already contains code to open up and read the player statistics from the playerstats.dat data file. The input() function performs this task. Two arrays are created in the input function, a 1-dimensional array of the player names, and a 2-dimensional array of the player stats. Each row of the 2-dimensional stats array holds statistics for a particular baseball player. For example, the name of the player at index 5 of the players[] array might be “Jose Consenco”. The statistics for “Jose Consenco” will be found in the 2-dimensional stats array in the row with index 5. The columns of the stats table hold the following information:

<i>stats</i>	<i>[0] Hits</i>	<i>[1] Walks</i>	<i>[2] Outs</i>
[0]			
[1]			
[2]			
...			
e.g. 1 row for each player on the team			

Therefore, if you wanted to find the number of hits that “Jose Consenco” has, and he is the player at index 5, you would look in row 5, column 0 like this:

```
joseConsencoHits = stats[5][0];
```

3. Write a function that discovers which player has the most number of hits on the team. The function should be called `maxHits()` and should take the stats 2-dimensional array and the number of players on the team as parameters. In this function, you need to search through the values in the "Hits" column (column 0) and find the maximum. Your function should return the row number of the player with the maximum number of hits on the team. Use this row number to display the players name from the players array who has the most hits.
4. Write a function called `battingAverage()` that will determine the batting percentage of a player. You should pass in the row number of the player you want to calculate the batting percentage for, as well as the stats 2-dimensional array. To calculate the batting percentage, you need to know the number of Hits the player had (from column 0), as well as the total number of bats. You must calculate the total number of at bats by adding the number of hits for the player, plus his walks and outs. Therefore $atbats = hits + walks + outs$. Of course, you need to get these pieces of information from the correct locations of the stats table. You calculate the batting average by dividing the number of hits by the total number of at bats. Make sure you cast this calculation as a float because the batting percentage will be a number between 0 and 1. Return this float as the result of calling the `battingAverage()` function.
5. Once you have the `battingAverage()` function working, write a function called `calculateTeamBattingAverages`. You should pass this function 3 pieces of information, the stats 2-dimensional array, the players name array, and the number of players on the team. Loop through all of the players, and use the `battingAverage()` function to calculate each player's batting average. Display the name of the player and his batting average on the console. This function does not return a value, it simply calculates the batting averages for all of the team members and displays them all nicely on the console.
6. For your final task, create a function that will display all of the statistics (Hits, Walks, and Outs) for all of the players. Call the function `displayStats()`, and pass in the stats 2-dimensional array, the players array and the number of players as parameters. Display the contents of the stats table by writing a nested loop. You will need a loop within a loop to display all of the rows and all of the columns in each of the rows of the stats table. Display the players name before each row when you print out the player stats on the console. Try and format your output so that the table looks nice and presentable on the console output (look at the `iomanip` stream functions for formatting your output).

Program 3 Finished

You have now completed Program 3. As with previous assignments, the final step you should perform when you complete an assignment is to upload your finished assignment to your eCollege account. Go ahead at this point and upload your PlayBall.cpp source file to eCollege. Once this is done you have successfully completed the programming assignment #3.