**Lab 4 - Arrays, Strings, and Pointer**

Part 1: Strings and Arrays

# Part A

* + For this part of the lab you will implement your own version of the library function strstr(), with the following function signature:   
    *int myStrStr (char haystack[], char needle[], char buffer[]);*
    - Your function will take 3 strings, a 'haystack' string, a 'needle' string, and a 'buffer' string. You will search the 'haystack' string for a sequence matching the 'needle' string, and copy the found result from the 'haystack' string into the 'buffer' (do not copy the 'needle' string). You should return a 1 if the matching sequence in the 'haystack' is found and a 0 if the 'needle' is not found. (You are required to implement your own logic for searching matches instead of using c library function).
  + Test your function with the following strings.
    - haystack="apple", needle="app"
    - haystack="orange", needle="ge"
    - haystack="blueberry", needle="ueber"
    - haystack="strawberry", needle="strawberry"
    - haystack="grapefruit", needle="terrible"
  + Within your main, after your function call, neatly print the 'haystack', 'needle', 'buffer' from the myStrStr function.

Part 2 -- Tournament

In this part of the lab you are going to implement your own tournament program. You can name your teams whatever you like.

# Part A

* + You will need to create a struct called Team that contains a string buffer for the team name. After you've defined 8 teams, you will place pointers to all 8 into an array called league[], defined the following way: Team \* league[8]. This must be an array of pointers to teams, not an array of Teams.
  + Write a function called game() that takes pointers to two teams, then randomly and numerically determines a winner using the rand() function (you do not have to store the score in the struct). Your game() function should
    - take pointers to two teams as parameters
    - randomly determine a winner using rand() (don't forget to seed with srand() )
    - print out the teams names, scores, and the winner of the game
    - Return a pointer to the winner.
  + Make sure this works correctly before moving on to the next part.

# Part B

* + Once you have your league, create a function called tournament() that takes an array of 8 pointers to Team structs, then returns a pointer to a single winning team from the array. Use your game function and a loop for each round (there should be 3 rounds) to determine the winner. Because this is an elimination style tournament, each team should lose only once, while the winner goes on to the next round. The easiest way to accomplish this is to use a different array of pointers for all 3 rounds.
  + You should print the round number and the winners of each round to the console.
  + Once you have determined that the tournament() function works correctly, then go on to part C.

# Part C

* + Add a handicap (your choice of how to implement) to the Team struct, and assign a value to each team by hand or randomly (again, up to you). Alter your game function to use the handicap to weight the tournament in favor of some teams, and run your results again.

Part 3 - Submission

Create a tar archive with the command “tar -czvf lab4.tar.gz .”, and then email your archive to bu580u2017@gmail.com and cc your TA dmu1@binghamton.edu before the submission deadline. Make sure you do not include the executable in your archive (make clean before creating the archive). Late assignments will not be accepted under any circumstances. Plan to turn in your assignments early.

Demo your lab before the demo deadline (after the submission deadline) by downloading your submission from class Gmail and extracting your archive with the command “tar -xvf lab4.tar.gz”. Then compile (with your makefile), and run your code, show your source to your TA, and answer any questions your TA may have.

Grading Guidelines

## Part 1:

* + myStrStr function returns the correct value for all cases: 2 points

## Part 2:

* + Part A: 2 points
  + Part B: 3 points
  + Part C: 2 points

## Style Guidelines

* + Follows Style Guidelines: 1 point

Submission Deadline: 11:59pm 10/26/2017 Eastern Time

Demo Deadline: 2:00pm 11/3/2017 Eastern Time