**CS 2060 Programming with C - Fall 2017**

**Assignment #11**

Due Date: **Dec 4, 2017** at 9:25am (MW class), **Dec 5, 2017** at 9:25am (TR class)

Purpose: Understand arrays of structures and linked lists.

Effort: Individual

Points: **100**

Deliverables: Upload the .c source code file to Blackboard by due date.

Please hand in a hardcopy version of your code at beginning of class.

**NOTE: Please hand in a hand drawn version of your data structures.**

**Assignment Description**

For this assignment, assume 4 friends are going to take part in 3 different running races. Write a simple program that **manually** initializes an **array of structures** and then answers questions based on that data structure. The program will also create a **linked list** based on the structure used to represent a racer.

**Specifications**

1. Create a C project called **Assignment11 (please use this exact name)**
2. Follow "CS2060 Programming Assignments Policy"
   1. Pseudocode is **NOT** needed for this assignment
3. 1st write code that creates 3 new data types using structures to represent:
   1. **Racer**
      1. Name
      2. List of results (array of 3 **integer** race results)
      3. Overall (you’ll write a function to compute this integer value)
   2. **Race**
      1. Name
      2. Number of racers
      3. List of participating racers (array of 4 **racers**)
   3. **RacerNode**
      1. Name
      2. List of results
      3. Overall
      4. Link to next RacerNode
4. 2nd write code to perform the following tasks:
   1. Create **4 racers:**
      1. Sue with results 25th, 7th, 4th
      2. Stacy with results 81st, 3rd, 9th
      3. Betty with results 71st, 82nd, 45th
      4. Joe with result 29th, 4th, 2nd
   2. Create **3 races** each with the 4 racers as participants:
      1. Pikes Peak Ascent
      2. Garden of the Gods 10-mile
      3. Boulder Boulder
   3. Create **an array of 3 races**
   4. Compute **overall** value for each racer (make this a function)
      1. overall = results of 3 races added together
   5. Print the race array
      1. Print each race name, the names of each racer in that race, and result for each racer
   6. Find which friend had the highest finish at the Pikes Peak Ascent
   7. Find which friend had the worse finish at the Pikes Peak Ascent
   8. Find which friend had the best finish out of the 3 races
   9. Find which friend had the worse finish out of the 3 races
   10. Write a function to add a node to a linked list
       1. Base the insertion criteria on the racer’s overall value, from lowest to highest
       2. Place racers into the linked list in this order (Sue, Stacy, Betty, Joe)
   11. Print the linked list
5. **Turn in hand drawn picture of your data structures – please staple your hardcopy.**

**Output**

Your output will look like the following.

**Output - Example #1**

Pikes Peak Ascent

Racer Finish

----------------------

Sue 25

Stacy 81

Betty 71

Joe 29

Garden of Gods 10-Mile

Racer Finish

----------------------

Sue 7

Stacy 3

Betty 82

Joe 4

Boulder Boulder

Racer Finish

----------------------

Sue 4

Stacy 9

Betty 45

Joe 2

Best finisher at Pikes Peak Ascent => Sue finishing 25

Worse finisher at Pikes Peak Ascent => Stacy finishing 81

Best finish for the 3 races occurred at Boulder Boulder => Joe finishing 2

Worst finish for the 3 races occurred at Garden of Gods 10-Mile => Betty finishing 82

The list in order from lowest to highest overall

Joe {29 4 2} --> Sue {25 7 4} --> Stacy {81 3 9} --> Betty {71 82 45} --> NULL