Test 1

Information:
- Thursday 2/20, 11:20-12:20 (I will lecture from 11:00-12:20).
- In class, closed book, closed notes, clean desk
- 15% of your final grade
- 60 minutes to complete it:
- Bring your ID card!
- NO: calculators or cell phones.
- NO: headphones/earbuds.

Test format:
100 points total:
- 12 (or more) multiple choice questions (~48 points total)
- 2 questions: write a function to do X (~26 points each)

Probably 4 total pages (2 pages front+back)
I'll have extra paper if you need it for writing the functions.

Content:
These lectures:
- Unit 1: Functions, Arrays, And Structures
- Linux (slides 2-13)
- Unit 2: Searching, Sorting, and Analysis

See the lecture slides for the corresponding book sections.

Sample questions:

Multiple choice:

1. See squarecap questions
2. What is the output given some code?
3. What is the error in the given code?
4. What is the command to do X in Linux?
5. What will the contents of the array be after 1 pass of the bubble (or selection) sort
6. What are the first two values that the target value X would be compared to during a linear (or binary) search.
7. What is the worst case run time efficiency big O function for the Bubble Sort (or selection sort, or linear search, or binary search)?
8. Of *these* efficiency functions, which is the **slowest** growing? $O(\log n)$ or $O(n)$ or $O(n^2)$ (or some other group of efficiency functions)
9. Also know definitions of terms and methods of parameter passing, scope rules, etc.

Coding questions:

1. Define a struct for a soccer player that stores their name, jersey number, and total points scored.
2. Using the struct in #1, write a function that takes an array of soccer players and its size as arguments and returns the average number of points scored by the players.
3. Using the struct in #1, write a function that takes an array of soccer players and its size as arguments and returns the index of the player who scored the most points.
4. Using the struct in #1, write a function that sorts an array of soccer players by name.
5. Using the struct in #1, write a function that takes an (unsorted) array of soccer players and its size and a number as arguments, and returns the name of the soccer player with that number. It should not do any extra unnecessary work.