# Tutorial 07

## Structures; Break, Continue and Goto

1. How are structures like arrays?
2. How are structures different from arrays?
3. How do you computer the size of an array?
4. How do you compute the size of a structure.?
5. What does the keyword typedef do?

```
struct grade {
  char grade;
  float score;
  int percentage;
};
```

Figure 1;

6. Write the declaration of a grade variable as defined in figure 1.
7. Write a function that assigns values to grade, score, and percentage.
8. Write statement that will print out the values of the grade variable whose values were assigned in question 20.
9. Define an array, called grades, of grade elements as defined in figure 1.
10. Write a loop that will read values from the keyboard to assign to each of the elements of the array, rades, defined in 17.
11. Define a function whose declaration is float grade_score(char grade_letter, struct grade grades[]); that will return the score associate with the grade_letter.
12. Define a function whose declaration is char grade_letter(float score); that will return the grade whose score greater than the score associated with, but less than the next highest score.
13. What do break and continue do? How do they differ?

# Arrays: Sorting and Median

14. What is an accumulator?
15. How do you initialize an array?
16. Write an array that is initialized to three ints: 1, 2, and 3.
17. Write an array that is initialized to three strings: “hi”, “there”, and “world”.
18. Why do you need to pass an address to “swap”?
19. What is computational complexity?
20. What is the median of a vector of numbers?
21. How do you calculate the median?
22. How long does it take to find an item in an unsorted array?
23. How long does it take to find an item in a sorted array?
24. How many times do you need to search an array so that it takes less time to sort the array and do the searches than it would to search the unsorted array if you use a bubble sort?
25. How many times do you need to search an array so that it takes less time to sort the array and do the searches than it would to search the unsorted array if you use a merge sort?
26. What is the difference between a list data structure and an array?
27. What are the common actions that one can perform on a list?
28. What data is needed in addition to an array to implement a list?
29. Write a one line function to implement an addition to a list?
30. Why can't you delete an item from a list in a single line like you can add?

# Arrays of Sructures

31. Why does one usually write output functions first?
32. How do you access a field of a structure? Give an example.
33. How do you access a field of a structure that is part of an array? Give an example?
34. Why can't we copy a string (i.e. char *) the same way we copy and int?
35. Write a definition of an array of structs.

36. How would you print the array of structs defined in the previous question if you had a function that printed a single struct?
37. When searching an array of records, under which condition does the computer do the fewest comparisons? Under which condition does it do the most?
38. What are boundary conditions?
39. What is a failure condition?
40. How do you delete a record from an array of structures?