Structures

```c
struct sname {
    T_1 d_1;
    T_2 d_2;
    ...
    T_k d_k;
};
```

defines a structure with name = "sname" and data components $T_1$, $d_1$, $T_2$, $d_2$, ..., $T_k$, $d_k$ of type $T_1$, $T_2$, ..., $T_k$ respectively. Here $T_i$ can be the fundamental data types (int, float, char etc) or even other structures that have been defined before.

Structures can be used exactly like the fundamental data types:

- The assignment statement `s1=s2` copies the contents of `s2` into `s1`.
- Structures can be passed as arguments to functions. By default they are passed by value. To pass by reference just add the ampersand after the structure name in the function declaration.
- The return type of a function can be a structure.

Practice Problems:

Write a program that creates an array of the structure student (that was discussed in class). Also write functions to

1. sort this array in alphabetical order (of student names)
2. sort it in decreasing order of scores.
3. calculate the mean score and the standard deviation.
4. search the array for the score of a student. (return -1 if the student is not in the list and the student’s score otherwise)