1. Both a C structure (a struct) and an array may store multiple data values, but otherwise structs and arrays are very different types. Explain how a struct object named S differs from an array object named A in terms of these aspects:

   a. (4 pts) The types of data that can be stored in S and A.

   b. (4 pts) The ways that data stored in S and A are accessed.

   c. (7 pts) Ways that the names, S and A, can be used in a C program. Hints: One of these names can be reassigned to a different object, but the other cannot. And one is equivalent to a pointer but the other one is not.

2. (7 pts) Define a C structure named struct book that has three data members: title - book title, a character string (may be as long as 50 characters), year - the year the book was published, an integer, and pages - the total number of pages in the book, another integer.
Both problems 3 and 4 refer to the struct book you defined for problem 2.

3. Write C statements to accomplish the following steps in order:
   a. (3 pts) Define an object of struct book named cs16text.
   b. (5 pts) Let char s[] = "Engineering Problem Solving with C"; copy this string into the title field of cs16text. Properly use the library function strcpy, or copy the characters one-by-one using a loop.
   c. (4 pts) Set cs16text's year field to 2013 and the pages field to 460. You must use two separate statements to do these things.

4. (10 pts) Write a function named printBook that takes a pointer to a struct book as its only parameter, neatly prints the fields of the structure like the following example, and does not return anything. Here is how the function would work at the ch prompt (using the object created in problem 3):
   ch> printBook( &cs16text );   /* notice address of the object is passed */
   Engineering Problem Solving with C
   published 2013
   460 pages