COSC1415 Sorting and Searching Arrays of Strings

Write a program that works with 1 dimensional arrays of strings. You may assume a maximum array size of 50. Write the following functions:

- A function to accept values for an array ***from an input file***. The input file stream will be passed as a parameter along with an array to hold strings. Read values from the input file and store into an array until end of file or until 50 values have been read. (You do NOT want to read and store more than 50 values.) The function should return the number of valid entries in the array.
- A function to print the valid entries of an array ***to an output file***. This function needs three parameters: the array, the count of valid entries, and the output stream name. Make your output neat and easy to read. Number each element of the array.
- A function to sort the elements of the array in alphabetical order. Use the selection sort explained in class.
- A function which performs a binary search on an array. The parameters to this function are: the array, the # of valid entries in the array, the search value. This function returns the position of the element if the search value is found or -1 if the search value is not found. Be sure to note in the precondition comments that this function assumes a sorted array!

Your main program will work with two different arrays and will call the above functions to perform the tasks. ***Design the pseudocode for your main function first!! Decide what each function does, what parameters it needs and what the function returns, if anything.***

Your main needs to perform the following tasks **in this order**, using the functions described above: input the first array, input the second array. (Use the provided input files words1.txt and words2.txt.) Print the values in the first array. Do the same for the second array. Sort each array and then print the sorted version of each array. Before you print each array, print a title that explains what you are printing, for example: "First list, unsorted:" or "Second list, sorted:" .

Main will now read a third input file that has a new list of words to be used as search values. (Use the file searchvalues.txt.) Set up a priming read loop in main that reads a value and then calls the search function to determine if the search value is in each of the arrays. Main will then print the search value and the results. If it was found, print its position in the sorted array. (Be sure to match this position with the number of the word when you printed the entire list.) Continue this process for each search value until the end of this file. You do not need to put these search values in an array; you can handle them one at a time. Example output for this part of the program:

> The word global was found in array 1 at position 16 and was not found in array 2.
> The word fantasy was found in array 1 at position 9 and was found in array 2 at position 14.

Print all your output to one output file.

Do not use any global variables! Have your functions do the detail work and main be the "air traffic controller."

Remember to put this statement at the top of your program: **#include <string>**

Hint: Design your main and all your function prototypes first. Write and debug the program one function or two functions at a time. There is no rule that says you have to write every function before you test anything. In fact, you will create correct code faster if you test one function at a time.