TCES 201
Introduction to Computer Programming
Homework 6 –Strings
10 Points

This homework tests your understanding of the topics covered so far in C Programming – Strings. Run the *executable* to understand the behavior of the program, make sure that the search_list.txt file is in the same directory as the executable.

In this exercise you will learn how to search an array for the occurrence of a given string. You will use the string comparison function

int strncmp(const char * cs, const char * ct, size_t n)

to find a given string (cs) in an array of strings using what is called linear search. That is, the algorithm starts at the first element of a string array, compares the given string to the one in that slot to see if they are equal. The strncmp function returns a zero if the two strings match, or non-zero otherwise. If the result is non-zero then increment an index counter and repeat. Here is the basic outline for the program.

Program: search_name.c

// open for reading the file search_list.txt (provided)
// setup a two dimensional array of char - names[60][60] this will be big enough to hold up to 60 name even if the name is 59 characters long!
//This is NOT the best way to do this, but you need to work with pointers more to do it a better, more space efficient way.
// set up a char buffer as a fixed array to hold a name to search for.

// write an algorithm that reads (fgets) the file, each line into the names array (names[i]).
// you should have every name in the file in one slot of the array.
// printf each name as it is read from the file and put into the array. printf("%s\n", names[i]) to make sure it is in the array properly.
// now write an algorithm that searches through the array for a specific last name.
//there are several ways you can do this, but we will use a simple function here. You can set up the search_name like this:
// char searchName[] = "Abraham"; looks for a name not in the list, or
// char searchName[] = "Wood"; looks for a name we know is in the list
//
// You can hard code these two cases to test your algorithm.
// You will need to first get the strlen(), string length, of the name to use to control the number of characters used in the comparison, e.g. int n = strlen(name);
// Then the strncmp() function will only use the first n characters of names[i] to see if they are the same lexicographical (alphabetic) order. If they are, the function returns zero so a test for zero tells you if you have found the one you are looking for.

```
/**
Looks for the name in the array of names.  Use the algorithm below.
*/
int find_string ( char  names[][60], const char * name, int file_len) {
   int n = strlen(name);
   int found = 0;
   while (found < file_len) {
      if names[found] is the same as name then break;
      increment found
   }
   if the name was found (hint: found < file_len) then return the found index number
   else return -1 to indicate it was not found
}

int main() {
   /**set up arrays and variables

   call load_array(FILE * fid, char  names[][60]);
   // returns with array names[][] loaded from file as well as the number of names in file, 19
in our case.

  //Get user input for a name to search for.
   call find_string();

   if found print success message on screen
   else print no success message
   **/
}
```

Submission Instructions: Submit the code on Canvas under hw6 Submission link as search_name.c. Formatting, appropriate variable names, readability and commenting are all considered while grading.