Dylan Buchheim

Group N

Lab 502 / Lecture 004

Questions:

I had no questions.

Acknowledgement:

I was the only one who contributed to my code.

//Main Pseudocode:

Data: FNAME[], character array to store the name of the file the user wants to open.

\*fp, the file pointer.

Plan: //P1: get user input for the file name.

//P2: set fp = to the opened file named FNAME.

//P3: check if the entered file is NULL.

//P3.1: if the file is null print a failure and exit the program.

//P4: Call the FindGreatestString function with fp as the parameter.

//P5: Close fp.

//Function Definition

Function Name: FindGreatestString

Input: \*fp, the file pointer.

Output: N.A.

side effect: printing the full name of the person who has the greatest last name to the screen.

Data: firstName[], the character array that stores the current first name.

lastName[], the character array that stores the current last name.

greatestLastName[], the character array that stores the greatest last name.

greatestFirstName[], the character array that stores the greatest first name.

\*s1 , \*s2, the character pointers which are used to reference the two last names we are comparing.

Plan: //P1: set \*s1 to greatestLastName and \*s2 to lastName.

//P2: start and infinite loop, for(;;).

//P2.1: read the next name from fp and set it to firstName[] and lastName[].

//P2.2: if StringComp(s1,s2) == -1.

//P2.3.1: use strcpy to copy the current name to greatestFirstName[] and greatestLastName[].

//P2.3: if we are at the end of fp break from the loop.

//P3: print the greatestFirstName[] and greatestLastName[].

//Function Definition

Function Name: StringComp

Input: \*s1, \*s2 the two character pointers that reference the strings we are comparing.

Output: 0, if the two strings are the same.

-1, if s1 is less than s2.

+1, if s1 is greater than s2.

Data: comparisonValue, the integer which stores either 0, -1, +1 depending on the comparison.

Plan: //P1: if \*s1 == \*s2 and \*s1 != the null character.

//P1.1: call StringComp((s1 + 1),(s2 + 1)).

//P2: else if \*s1 < \*s2 or \*s1 == the null character.

//P2.1: set the comparison value to -1.

//P3: else if \*s1 > \*s2 or \*s2 == the null character.

//P3.1: set the comparison value to 1.

//P4: return the comparison value.