String Lab

LAB 4

SECTION C

Kenneth A. Jacobson

SUBMISSION DATE:

26.9.2017

Problem

Our goal in this lab was to complete the three batches of code and modify them according to the lab manual. We were also asked to find the purpose of some variables within the first batch of code.

Analysis

The ! before the strcmp function is to invert the output of strcmp which returns a 0 if the strings are the same. If you remove the ! the code stops functioning as it was intended. (#1) The flag variable is there to return an error if the string does not trigger either of the first two if-statements. The semantic error was in the final if statement. This error would trigger as long as the flag variable was 1 but adding a ! before flag in the if statement will fix the error.

Design

These programs were given to us more or less complete and the only code we had to design were input and output statements and some small modifications.

Testing

I had a small problem with the last batch of code not returning the values I expected but it turned out to be that I had put the wrong string in the strlen function (#3).

Comments

All good

Source Code

```
//lab3-1.c
//strcmp function - string comparison
//
//NOTE - There is a semantic error in the code below
//      Step 1. Complete the printf and scanf statements below.
//        Compile and run the code.
//        What semantic error do you see in the output?
//      Step 2. Now, fix the error. Compile and run the code to check your
//        result.

#include <stdio.h>
#include <string.h>


int main(void)
{

 char major[100]; //string variable for major

 int flag = 0; //why do we need this variable ?
         //just in case the input does not trigger any of our if statements
 //write one printf to prompt the user
 //to enter a major
 //For example, physics
 //           computer
 /******** YOUR CODE BELOW *********/
 printf("Enter a major: ");

 //write one scanf statement to read
 //the string for the major from
 //the keyboard (no spaces in string)
 /******** YOUR CODE BELOW *********/
 scanf("%s", major);

 //why do we need  ! in front of the strcmp function ?
         //because strcmp returns a 0 if the strings are the same and !0 == true
 //Fix sematic error and then, Try removing all 3 of the !
 //and see what happens with input "physics"?
         //the if statement doesn't trigger and you get a bad result
 if(!strcmp(major, "physics") || !strcmp(major, "chemister") || !strcmp(major, "biology")){

           printf("\nYou are a science major. \n");

           flag = 1;
 }
```

```
        //strcmp returns a value based on the 2 strings. It compares the two strings until it finds an
unmatched value.
        //if there is an unmatched value it returns a value based on the ASCII value of the chars at that
index

 if(!strcmp(major, "computer") || !strcmp(major, "mechanical") || !strcmp(major, "electrical")){

            printf("\nYou are an engineering major. \n");

            flag = 1;

 }

 //what change should be made to the if statement
 //to fix the semantic error ?
 if(!flag)
  printf("\nSorry! I do not recognize your major. \n");
        //add ! before flag to invert the value if flag = 1 then !flag = 0
        //in order to print this the major cannot be found in the above statements

}


//lab3-2.c
//strcpy function - string copy function

#include <stdio.h>
#include <string.h>

int main(void)
{

 char sourceString[100]; //source string variable
 char hold[100]; //to hold destinationString
 char destinationString[100] = "overwrite me"; //destination string variable

 //write one printf to prompt the user
 //to enter a source string
 /******** YOUR CODE BELOW *********/
 printf("Enter a string: ");


 //write one scanf statement to read
 //the source string from
 //the keyboard (no spaces in string)
 /******** YOUR CODE BELOW *********/
        scanf("%s", sourceString);
```

```c
 //write one printf to print source and destination string
 /******** YOUR CODE BELOW *********/
 printf("\nSource: %s\nDestination: %s\n", sourceString, destinationString);

 strcpy(hold, destinationString);
 //added to copy destinationString to hold before copying sourceString into destinationString
 strcpy(destinationString, sourceString);

//print both source and destination string using
// ONE/SINGLE printf statement
 printf("\nSource: %s\nDestination: %s\n", sourceString, destinationString);
 printf("Original Destination: %s\n", hold);
 //added second printf to print hold which holds original value for destinationString
}

//lab3-3.c
//strcat function - string concatenation function

#include <stdio.h>
#include <string.h>

int main(void)
{

  char sourceString[100]; //source string variable
  char destinationString[100] ; //destination string variable
  int sourceStringLength, destinationStringLength, finalStringLength;

//write one printf to prompt the user
//to enter a source string
/******** YOUR CODE BELOW *********/
  printf("Enter a string: ");

//write one scanf statement to read
//the source string from
//the keyboard (no spaces in string)
/******** YOUR CODE BELOW *********/
  scanf("%s", sourceString);

//write one printf to prompt the user
//to enter a destination string
/******** YOUR CODE BELOW *********/
  printf("Enter a second string: ");

//write one scanf statement to read
```

```
//the destination string from
//the keyboard (no spaces in string)
/******** YOUR CODE BELOW *********/
  scanf("%s", destinationString);

//write one printf to print source and destination string
/******** YOUR CODE BELOW *********/
  printf("First String: %s\nSecond String: %s\n", sourceString, destinationString);

  destinationStringLength = strlen(destinationString);
  sourceStringLength = strlen(sourceString);
  strcat(destinationString, sourceString);
  finalStringLength = strlen(destinationString);
 //print both source and destination string using
 // ONE/SINGLE printf statement
  printf("First String: %s\nSecond String: %s\n\n", sourceString, destinationString);
  printf("First String Length: %d\nSecond String Length: %d\nThird String Length: %d\n",
sourceStringLength, destinationStringLength, finalStringLength);
}
```

Screen Shots

1.) Code run without ! before the strcmp function

```
kenneth1@CO2018-19 /cygdrive/u/cpre185/lab4
$ ./a.exe
Enter a major: physics

Sorry! I do not recognize your major.
```

2.) Lab4.2 Final Code

```
kenneth1@CO2018-19 /cygdrive/u/cpre185/lab4
$ ./a.exe
Enter a string: potato

Source: potato
Destination: overwrite me

Source: potato
Destination: potato
Original Destination: overwrite me
```

3.) Lab 4.3 Code with error

```
Kenny@DESKTOP-O9MI33M ~/Lab4
$ ./a.exe
Enter a string: potato
Enter a second string: pancake
First String: potato
Second String: pancake
First String: potato
Second String: pancakepotato

First String Length: 7
Second String Length: 7
Third String Length: 13
```

4.) Lab 4.3 Final Code

```
Kenny@DESKTOP-O9MI33M ~/Lab4
$ ./a.exe
Enter a string: potato
Enter a second string: pancake
First String: potato
Second String: pancake
First String: potato
Second String: pancakepotato

First String Length: 6
Second String Length: 7
Third String Length: 13
```