

- Play! When you don't understand something like fgets or strcat, write a small program and test various inputs. It is educational and fun.
- Some programs below are associated with a question. **Answer the questions** using C comments below your code in the program file unless the instructions say otherwise.

For each numbered problem below you will write a small program. Name each program lab10.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab10.1.c

Programs

1. Create the functions myStrcpy and myStrcat. In your program, create some sample strings (what is a string in C?) and demonstrate the C function strcpy, followed by myStrcpy, then the function strcat followed by myStrcat. Remember that all C string handling functions are written with a sentinel loop.
2. The C string function strcmp compares two strings. However, it does not return true when they are equal; its behavior is more complex. Because type char elements can be treated as integers, strcmp traverses both arrays being compared, subtracting the elements (one char from another) as it goes (and it goes until it hits a what?).

As long as the result of subtraction is zero, it knows the letters are the same and it keeps going. If two letters are different, it stops and returns a result with the same sign¹ as the result of the subtraction.

In your program, call strcmp on "apple" and "doughnut", then on "doughnut" and "apple" to see which way the sign goes. Then write myStrcmp and demonstrate that it returns the same sign as strcmp for those cases and: apple and applet, bat and bat, bats and bat.

3. Create a four row, eight column array of char and read four words in from the user: biscuit, anagram, amalgamation, and porcine. **After all four** words have been stored, print them out as strings. Annotate your script by hand and briefly explain what prints out.
4. Copy the previous program and reverse the loop, i.e. if you had looped from 0 to 3, now loop from 3 to 0. **After all four** words have been stored, print them out as strings. Annotate your script by hand and briefly explain what prints out.
5. Create two data files, with a different number (< 20) of integers in each. Declare two arrays in main(), size 20. Write a function that will read all of the integers (use the feof() function from class) from a file into an array. Call the same function twice, once for each data file. The size of the data files must not appear anywhere in your program.
6. Write a program to read in a single char from the user. Then use fgets(), as shown in class, to be sure your program isn't confused by extra input. Use a switch statement to have your program print a (real) word starting with the letter entered by the user, but any non-alpha character should print an error message. Lower and upper case letters should be treated the same (hint: see 9.6 of H&K).

Your main should not be longer than ten lines.

¹or even the same number, depending on the implementation

7. Write a program that will print **all** the words entered on the command line by the user, not including the name of the program.
- 8.
- 9.
10. Write a program to read in two doubles and an int from the command line. Use a function from the beginning of Appendix B in H&K to convert the string representation of the numbers to int or double. Have your program print the highest of the three. Your program should not perform more than two numeric comparisons to find the max.

You should have a total of 10 programs named lab10.01.c to lab10.10.c. Make a single script file (see lab00 for the scripting instructions) where you cat, compile, and run each one in its final form (if it didn't compile, don't run it in the script - mark the place in the printed script file with a colored marker so it stands out).

On the first page of every printed copy for this course, your name, section, and TA's name must appear.

Submit all program and script files on MyCourses before midnight Thursday of next week, and give the paper version to your TA at the beginning of your Friday lab (or in lecture Friday if you have a Wednesday lab). Note: cat, compile, and run each program in order! Do *not* cat all programs, then compile, etc.