

Instructions

Complete the problems assigned below and turn in your answers by the due time above. To receive full credit, you must show both the relevant command(s) and output. If a question asks about what a command does, or what the output would be, give both the command and the output. Demonstrate that you actually performed the command, and didn't just guess.

Homework Problems

1. Implement the phonebook programs **add**, **rem**, and **lu** from Chapter 7, with the following modifications:
 - a. Use a TAB separator between the name and phone number fields, not spaces.
 - b. Do not use the TAB character itself - rather, use the appropriate **echo** escapes.
 - c. Modify **lu** to ignore case during lookups.
 - d. do not use the **/tmp/phonebook** file for the **rem** program - instead, use **./phonebook.tmp**.
 - e. Use variables for all filenames in each program.
 - f. Do not allow duplicate entries to be added. [hint: no extra code or commands is required]
2. Answer the questions in Exercise 2, page 147. Also, what happens when you pass a space as an argument?
3. What happens if you only give 1 argument to the **add** program?
4. For problem 1, above, modification (d), why did I have you not use the file **/tmp/phonebook**?
5. Modify the program from problem 6, homework 3 to use two numbers passed on the command line, instead of from two files. If wrote your program appropriately, this will only require two simple changes at the top of your file.
6. Do Exercise 6, page 148.
7. Do Exercise 7, page 148.
8. Write a program that does the following:
 - a. Line 1 of the output states how many arguments were passed.
 - b. Line 2 of the output states how many actual words existed on the command line (ignoring the command itself).
 - c. Outputs the first nine command line arguments, one per line. Each argument should be output such that it is surrounded by single quotes, so that you can exactly see what the argument was. Also precede each single quoted argument with an appropriate label.
9. Write two additional programs that output a single line. The output line announces the program's name and the arguments it received (single quote each for clarity). The first program is to run the second program; the second program runs the program you created from problem 8. Be sure all arguments pass correctly.
10. Write three programs called **trm**, **treco**, and **tempty**. The **trm** program is like **rm**, but instead of removing files, it places them in a directory named **Trash** (similar to the Recycle Bin in Windows or the Trash in Mac OS). The **treco** program recovers the files named on the command line from the **Trash** and places them in the current directory. Finally, the **tempty** program permanently deletes items in the **Trash**. You do not have to worry about duplicate files names or errors messages - these programs should be simple and straight forward, using what has been explored thus far in the course. We will add to these programs and make them more robust in future assignments.

Optional Problems - I would recommend that you try the Exercises on pages 147-148 that were not required above. You do not need to turn these in.