

## Laboratory: Files

- Exercises
  - Exercise 0: Preparation
  - Exercise 1: Summing Values
  - Exercise 2: File Length
  - Exercise 3: Missing Files
  - Exercise 4: Reading All Lines
  - Exercise 5: Displaying Files
  - Exercise 6: Finding Divisors
  - Exercise 7: Reusing Output Files
  - Exercise 8: Counting Characters

### Exercises

#### Exercise 0: Preparation

- a. Scan through this lab to determine what kinds of tasks you'll need to complete.
- b. You should also scan through the reading on files in Scheme.

#### Exercise 1: Summing Values

The file `/home/rebelsky/Web/Courses/CS151/2001S/Examples/numbers` contains five hundred and twenty-eight natural numbers. What is their sum?

Note: That file was copied from a similar file produced by Mr. Stone.

#### Exercise 2: File Length

Using `sum-of-file` as a pattern, write a Scheme procedure `file-size` that takes as argument a string that names a file and returns the number of characters in that file (that is, the number of times that `read-char` can be called to read a character from the file without returning the end-of-file object).

#### Exercise 3: Missing Files

Find out what happens if `sum-of-file` or `file-size` is given a string that does not name any existing file.

## Exercise 4: Reading All Lines

Write a procedure, `read-lines`, that takes the name of a file as a parameter and returns a list of all the lines of the file (with each line represented as a string).

## Exercise 5: Displaying Files

Write a procedure, `display-file`, that takes the name of a file as a parameter and displays the contents of the file with each line preceded by its line number. You may find it useful to call upon the previous procedure to do the reading.

## Exercise 6: Finding Divisors

Use the `store-divisors` procedure to draw up a list of the divisors of 120, storing them in a file named `divisors-of-120`. Examine the file afterwards and confirm that the answer is correct. (Don't give this procedure an extremely large number as argument -- it's too slow. There are more efficient ways to find divisors!)

## Exercise 7: Reusing Output Files

The Scheme standard says that if you try to open an output port to a file that already exists, “the effect is unspecified,” i.e., anything might happen. Hence, designers of a particular implementation of Scheme are free to do what they choose.

- a. Find out through experimentation what DrScheme does in this situation.
- b. Find out using DrScheme's Help Desk what DrScheme allows programmers to do in this situation. (The help desk document is entitled *Opening file ports*.)

## Exercise 8: Counting Characters

Write a Scheme procedure that takes as arguments two file names (an input file and an output file), counts the number of occurrences of each vowel in the input file, and writes the result to the output file.

If you have time or inclination, extend your procedure to deal with all 127 ASCII characters.