**Exercise 5-1.** As written, getint treats a + or - not followed by a digit as a valid representation of zero. Fix it to push such a character back on the input.

**Exercise 5-2.** Write getfloat, the floating-point analog of getint. What type does getfloat return as its function value?

**Exercise 5-3.** Write a pointer version of the function strcat that we showed in [Chapter 2](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter2.html): strcat(s,t) copies the string t to the end of s.

**Exercise 5-4.** Write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise.

**Exercise 5-5.** Write versions of the library functions strncpy, strncat, and strncmp, which operate on at most the first n characters of their argument strings. For example, strncpy(s,t,n) copies at most n characters of t to s. Full descriptions are in [Appendix B](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\appb.html).

**Exercise 5-6.** Rewrite appropriate programs from earlier chapters and exercises with pointers instead of array indexing. Good possibilities include getline ([Chapters 1](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter1.html) and [4](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter4.html)), atoi, itoa, and their variants ([Chapters 2](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter2.html), [3](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter3.html), and [4](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter4.html)), reverse ([Chapter 3](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter3.html)), and strindex and getop ([Chapter 4](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter4.html)).

**Exercise 5-7.** Rewrite readlines to store lines in an array supplied by main, rather than calling alloc to maintain storage. How much faster is the program?

**Exercise 5-8.** There is no error checking in day\_of\_year or month\_day. Remedy this defect.

**Exercise 5-9.** Rewrite the routines day\_of\_year and month\_day with pointers instead of indexing.

**Exercise 5-10.** Write the program expr, which evaluates a reverse Polish expression from the command line, where each operator or operand is a separate argument. For example,

expr 2 3 4 + \*

evaluates 2 \* (3+4).

**Exercise 5-11.** Modify the program entab and detab (written as exercises in [Chapter 1](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter1.html)) to accept a list of tab stops as arguments. Use the default tab settings if there are no arguments.

**Exercise 5-12.** Extend entab and detab to accept the shorthand

entab *-m +n*

to mean tab stops every *n* columns, starting at column *m*. Choose convenient (for the user) default behavior.

**Exercise 5-13.** Write the program tail, which prints the last *n* lines of its input. By default, *n* is set to 10, let us say, but it can be changed by an optional argument so that

tail *-n*

prints the last *n* lines. The program should behave rationally no matter how unreasonable the input or the value of *n*. Write the program so it makes the best use of available storage; lines should be stored as in the sorting program of [Section 5.6](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.23260\tcpl\chapter5.html#s5.6), not in a two-dimensional array of fixed size.

**Exercise 5-14.** Modify the sort program to handle a -r flag, which indicates sorting in reverse (decreasing) order. Be sure that -r works with -n.

**Exercise 5-15.** Add the option -f to fold upper and lower case together, so that case distinctions are not made during sorting; for example, a and A compare equal.

**Exercise 5-16.** Add the -d (``directory order'') option, which makes comparisons only on letters, numbers and blanks. Make sure it works in conjunction with -f.

**Exercise 5-17.** Add a field-searching capability, so sorting may bee done on fields within lines, each field sorted according to an independent set of options. (The index for this book was sorted with -df for the index category and -n for the page numbers.)

**Exercise 5-18.** Make dcl recover from input errors.

**Exercise 5-19.** Modify undcl so that it does not add redundant parentheses to declarations.

**Exercise 5-20.** Expand dcl to handle declarations with function argument types, qualifiers like const, and so on.