**Exercise 7-1.** Write a program that converts upper case to lower or lower case to upper, depending on the name it is invoked with, as found in argv[0].

**Exercise 7-2.** Write a program that will print arbitrary input in a sensible way. As a minimum, it should print non-graphic characters in octal or hexadecimal according to local custom, and break long text lines.

**Exercise 7-3.** Revise minprintf to handle more of the other facilities of printf.

**Exercise 7-4.** Write a private version of scanf analogous to minprintf from the previous section.

**Exercise 5-5.** Rewrite the postfix calculator of [Chapter 4](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.3815\tcpl\chapter4.html) to use scanf and/or sscanf to do the input and number conversion.

**Exercise 7-6.** Write a program to compare two files, printing the first line where they differ.

**Exercise 7-7.** Modify the pattern finding program of [Chapter 5](file:///C:\Users\Valentin\AppData\Local\Temp\Rar$EXa1580.3815\tcpl\chapter5.html) to take its input from a set of named files or, if no files are named as arguments, from the standard input. Should the file name be printed when a matching line is found?

**Exercise 7-8.** Write a program to print a set of files, starting each new one on a new page, with a title and a running page count for each file.

**Exercise 7-9.** Functions like isupper can be implemented to save space or to save time. Explore both possibilities.