```
    if ((fp = fopen(filename, "r")) != NULL) {
      while (fgetc(fp) != EOF)
        if (fgetc(fp) == '.')
          n++;
      fclose(fp);
    }

    return n;
}
```

13. Write the following function:

```
int line_length(const char *filename, int n);
```

The function should return the length of line `n` in the text file whose name is `filename` (assuming that the first line in the file is line 1). If the line doesn't exist, the function should return 0.

Section 22.5

W 14. (a) Write your own version of the `fgets` function. Make it behave as much like the real `fgets` function as possible; in particular, make sure that it has the proper return value. To avoid conflicts with the standard library, don't name your function `fgets`.

(b) Write your own version of `fputs`, following the same rules as in part (a).

Section 22.7

W 15. Write calls of `fseek` that perform the following file-positioning operations on a binary file whose data is arranged in 64-byte "records." Use `fp` as the file pointer in each case.

(a) Move to the beginning of record `n`. (Assume that the first record in the file is record 0.)
(b) Move to the beginning of the last record in the file.
(c) Move forward one record.
(d) Move backward two records.

Section 22.8

16. Assume that `str` is a string that contains a "sales rank" immediately preceded by the # symbol (other characters may precede the # and/or follow the sales rank). A sales rank is a series of decimal digits possibly containing commas, such as the following examples:

```
989
24,675
1,162,620
```

Write a call of `sscanf` that extracts the sales rank (but not the # symbol) and stores it in a string variable named `sales_rank`.

Programming Projects

1. Extend the `canopen.c` program of Section 22.2 so that the user may put any number of file names on the command line:

```
canopen foo bar baz
```

The program should print a separate `can be opened` or `can't be opened` message for each file. Have the program terminate with status `EXIT_FAILURE` if one or more of the files can't be opened.

W 2. Write a program that converts all letters in a file to upper case. (Characters other than letters shouldn't be changed.) The program should obtain the file name from the command line and write its output to `stdout`.