CSC 180 lab 9: Files

Thurs Nov 8 or Mon Nov 12, 2001

1. The simplest sorting algorithm is the one called ‘‘bubble sort’’. The following C function (available as ~ajr/lab9/bubblesort.c on ECF):

```
void sort(int *a, int size)
{
    int i, j, t;

    for (i = 0; i < size; i++) {
        for (j = 0; j < size - 1; j++) {
            if (a[j] > a[j+1]) {
                /* swap */
                t = a[j];
                a[j] = a[j+1];
                a[j+1] = t;
            }
        }
    }
}
```

takes an int array and the size of that array, and sorts the integers in that array into ascending order. This algorithm will be discussed in lecture shortly, but we’ll use it right now in this lab.

Write a void function which takes two FILE* arguments. It reads a sequence of up to 100 integers from the first file, calls the above function to sort them, then outputs the sorted list to the second, one per line, no additional data. The input file is free-format; i.e. you will want to use fscanf(). It is okay if your program mistakes non-numeric input for end-of-file; i.e. you will probably want to use while (fscanf(...) == 1).

Test your function with the following (which could be in a separate file, or you can put them all into one, whichever’s easiest):

```
#include <stdio.h>

int main()
{
    sortfile(stdin, stdout);
    return 0;
}
```

To use this program, type some numbers, then press control-D. Or run ‘‘./a.out <file’’.

2. Write a main() to prompt for and read names for the input and output files. Then open them both (with fopen(), calling perror() and exiting if the fopen() fails), and call sortfile(infp, outfp).

After calling fgets() to read a file name, remove a possible newline at the end of the resulting input string before passing it to fopen() with the following code:

```
if ((p = strchr(s, '\n')) != NULL)
    *p = '\0';
```

This requires `#include <string.h>` and a previous declaration `char *p;`.

3. Write a different main() which uses command-line arguments so that you can sort the numbers in file1 and output it to file2 by typing

```
./a.out file1 file2
```