CSC161 2010F *Imperative Problem Solving*

Assignment 8: A Sorting Utility

Assigned: Friday, 12 November 2010
Due: 11:00 p.m., Wednesday, 17 November 2010

Summary: In this assignment, you will develop a sorting utility similar to the standard Unix `sort` utility.

Purposes: To give you more experience with strings, function pointers, and more.

Expected Time: Two to three hours.

Collaboration: I encourage you to work in groups of two or three students. However, you may work on your own or in groups of up to size four. You may discuss the assignment with anyone you wish, provided you clearly document such discussions.

Submitting: Email me a tarball of your important files (your `.c` files, your `.h` files, your `Makefile`, a typescript of a sample session, and anything else you deem appropriate).

Warning: So that this assignment is a learning experience for everyone, I may spend class time publicly critiquing your work.

Sorting

Write a Linux utility, `srt`, that sorts lines of text.

If `srt` is called with no command-line arguments, it should read its input from standard input, sort the lines alphabetically, and print the output to standard output.

If `srt` has `-n` as a command-line argument, it should treat each line as a number.

If `srt` has `-r` as a command-line argument, it should "reverse sort", from largest to smallest.

If `srt` has `-u` as a command-line argument, it should "unsort" the values, putting them in random order.

If `srt` has `-w`*`num`* as a command-line argument, it should support lines of up to *num* characters.

If `srt` has `-l`*`num`* as a command-line argument, it should support up to *num* lines.

If `srt` is called with `-k`*`num`* as a command-line argument, it should break up the input at the separator character (a space, by default) and sort using the *num*th column.

If `srt` is called with `-t`*`char`* as a command-line argument, it should use *char* as the separator. (This option is pointless when used without `-k`.)

Anything else on the command line should be treated as a file name. No more than one file name should be permitted. If no file name appears, you should read from standard input.

If `-w` is not specified, you need only handle lines of up to 256 characters.

If `-l` is not specified, you need to handle only up to 1024 lines.

Examples

```
srt
```

Sort lines read from standard input.

```
srt students
```

Alphabetically sort the lines in the file `students`.

```
srt students -r
```

Alphabetically sort the lines in the file `students` in reverse order.

```
srt -n grades
```

Sort the lines of grades numerically.

```
srt -w511 courses
```

Sort the lines of courses, allowing up to 511 characters per entry.

```
srt -t, -k2 gradebook -n -r
```

Given a grade book of the form `last,first,num` by grade from largest to smallest.

```
srt -l10000 -k1 students
```

Given a list of no more than 10,000 students of the form `last first middle`, sort alphabetically by first name.

```
srt -n students grades
```

An erroneous command because two file names are given.

```
srt -u students
```

Sort the lines of students in “random” order.