MyLs

Deadline: October 3rd, 2014

1 Instructions

Implement a program named `myls` that matches the following documentation:

```
Usage: myls [OPTION]... [FILE]...

List information about the FILEs (the current directory
by default). Sort the output alphabetically by default.

Options:
  -R       Recurse into subdirectories.
  -l       Use a long listing format.
```

- The program must terminate with code 0 if the operation completes successfully, 1 otherwise.
- Use only one column for the output.
- Entries whose name start with a period (`.`) must be omitted, except if explicitly given on the command line.
- For the long listing, use the following format: file mode, number of links, owner user ID, group ID, number of bytes in the file, datetime of last modification in format YYYY-MM-DD HH:MM:SS, entry name, optionally followed by "`->`" and the link target for symbolic links.
- You may use any standard C function (either from ISO C 1999/2011 or POSIX).
- You may not use `system` or any other mechanism that invokes an external program.

You may implement the following for a higher grade:

- multi-column output if the output is a terminal;
- output sorting: by default alphanumeric on the entry name; no sorting if `-f` is specified; by modification time if `-t` is specified, by size if `-S` is specified. Tip: use `qsort(3)`;
- any extra feature you deem particularly useful.

2 Grading

- 4 points if `myls` works without arguments or with only one file/directory argument.
- +1 point if `-R` is properly implemented.
- +1 point if `-l` is properly implemented.
- +1 point if `myls` also supports more than 1 file/directory argument.
- +0.5 point if multi-column output is properly implemented.
- +1.5 point if output sorting is properly implemented (including `-f`, `-t` and `-S`);
- +1 point if any additional feature is implemented and duly documented in an enclosed `README` file, and your examinator agrees it is indeed useful.

---

3 Copyright and licensing