CS270 - System Software
Lab Assignment #3
Fall 2016

The purpose of this assignment is to give you practice writing shell scripts.

Many systems today provide the users with a *trashcan* for deleting files. Unwanted files are not actually deleted, but rather are moved to a designated subdirectory. They still exist as regular files until the user does an "empty trash" operation, at which time the files truly are deleted.

You are to write a shell script, called `rf` (for *round file*), that implements the *trashcan* function. When `rf` is invoked without any options, the specified files should be moved to a subdirectory called `roundfile`. If `roundfile` does not exist, it should be created in the user's home directory, and a message stating that it has been created should be printed. If a directory is specified, then the directory (along with all of its contents) should be moved. If a file specified in the `rf` command has the same name as a file that already exists in `roundfile` (i.e., a file with the same name has already been `rf`'ed), then one of two things should happen:

- If the newly deleted file's contents are the same as the previously deleted file (as would be the case if the same file were deleted twice), then the new file should replace the old file - that is, the file's date should reflect the most recent deletion.
- If the newly deleted file's contents are different than the previously deleted file, then both files should be kept. The newly deleted file's name should be changed to include a "version" number. For example if we try to delete a file named `file.txt`, but a file already exists with that name, the newly deleted file should have the name `file.txt.1` . If subsequently another file named `file.txt` is deleted, its name should be `file.txt.2`, etc.

The `rf` command should support at least the following options:

- `-e` - *empty* the trash. All files in the `roundfile` directory should be (really!) deleted.
- `-f` - *flush*. Like `-e`, except that the `roundfile` directory itself should also be deleted.
- `-i` - *interactive*. The program should ask the user before `rf`'ing any file (similar to `rm -i`).
- `-l` - *list*. The files in `roundfile` should be listed, similar to issuing the command `ls -l roundfile`. Directories should be listed, but their contents don't need to be.
- `-r` - *retrieve*. Copy the specified file from the `roundfile` directory to the current directory. If a file with the same name as the retrieved file already exists in the current directory, then your script should ask the user if the retrieved file should replace the existing one. If the answer is "no," then nothing should happen.

You can write your shell script using any of the standard UNIX shell languages, but it should execute properly if invoked from `bash`. It should also work correctly if issued from any subdirectory and the permissions on the file are appropriate for the operation.