CS 279 - Homework 8

Deadline:

Due by 11:59 pm on **SUNDAY**, October 26.

How to submit:

Submit your files using `~ah270/279submit` on nrs-labs, with a homework number of `8`, by the deadline shown above.

Purpose

To practice with the `find` command and with bash arrays (with a touch of `tee` and `wc` as well)

Important notes:

- **Each** bash shell script that you write is expected to include a descriptive opening comment block including your name and the last modified date.
- It is possible that your answers may be collected and posted to the course Moodle site.

The Problems:

Problem 1

In a file `hw8-1.txt`, include:

- your name
- the part you are giving an answer for
- a `find` command for each of the following

1 part a

Write a `find` command that will display the pathnames of all files named `a.out` starting from your home directory.

1 part b

Write a `find` command that will display the pathnames of all files whose names end in `.sh` starting from the current working directory.

1 part c

Write a `find` command that will display the pathnames of all symbolic links starting from your home directory.

1 part d

Write a `find` command that will display the pathnames of all files whose size is less than 100 characters/bytes starting from your home directory.

1 part e

Write a `find` command that will display the pathnames of all directories with the name `submitted` starting from your home directory.

1 part f

Write a single `find` command that will give the pathnames of all directories whose names start with `279`, starting from your home directory AND starting from my `public_html` directory, `~ah270/public_html`

1 part g

Write a `find` command that will display the names of all directories with the name `submitted` that were last modified more than 14 days ago, starting from your home directory.

Then, write another that will display the names of all directories with the name `submitted` that were last modified less than 14 days ago, starting from your home directory.

1 part h

Write a `find` command that will display the pathnames of all directories reachable from the current working directory whose permissions are exactly `755` -- that is, `rwx` for the owner, `r-x` for the group, and `r-x` for the world.

1 part i

Write a `find` command that will display the pathnames of all regular files that are world-readable starting from the current working directory. (That is, whatever their other permissions may or may not be, they are at least world-readable.)

1 part j

Write a `find` command that will display the names of all files, starting from your current working directory, that include `279` somewhere in their name and are newer than the file `~ah270/public_html/f14cs279/279hw08/279hw08.pdf`

1 part k

Write a `find` command piped to an appropriate `wc` command whose result will be the number of regular files starting from your home directory.

1 part l

Write a `find` command that uses an `-exec` to give pathnames preceded by their number of characters/bytes (that is, the result that `wc` gives when it is asked to give JUST the number of characters/bytes) for all files whose size is less than 100 characters/bytes starting from your home directory.

(That is, we are adding an `-exec` to your `find` command from **1 part d**.)

1 part m

Write a single `find` command that uses an `-exec` to give the long-listing of all directories reachable from the current working directory whose permissions are exactly `755` -- that is, `rwx` for the owner, `r-x` for the group, and `r-x` for the world. (Note: we want the long-listing for each directory itself, *not* for its contents.)

(That is, we are adding an `-exec` to your `find` command from **1 part h**.)

1 part n

Write a single `find` command involving piping, an appropriate `tee` command, and an appropriate `wc` command to write to file `world-r.txt` the pathnames of all regular files that are world-readable starting from your home directory but only outputting to the screen the number of such files.

(This is riffing off your `find` command from **1 part i**.)

Submit your resulting `hw8-1.txt` .

Problem 2

Write a bash shell script `stuff-play.sh` that meets the following specifications. You should meet these specifications in order within your resulting script, BUT feel free to echo additional blank lines or "borders" as desired if you'd like your script's output to be more attractive

- First, write a bash statement creating an array `stuff` containing at least 7 but no more than 10 elements of your choice, at least one element containing a blank surrounded by non-blanks. (For example, one element could be "moo oink".)
- Then write a bash statement that will now add a single array element to `stuff` with index `13` whose content is your first and last names, separated by a blank.
- Then write a bash statement that echoes to the screen a descriptive message including the element in `stuff` with index `3` .
- Then write a bash statement that echoes to the screen a descriptive message including the size of `stuff` (the number of elements in `stuff`)

- Then write a bash statement that echoes to the screen a descriptive message including the indices of `stuff`
- Then echo to the screen a descriptive message saying that what follows are the elements of array stuff, one element per line, and finally write a bash loop that will display the elements in `stuff`, one element per line.

Submit your resulting `stuff-play.sh` .

Problem 3

Now adapt your `stuff-play.sh` into `stuff-play2.sh`, making the following changes:

- the script should complain and exit with a non-zero exit status if at least one command-line argument is not given.
- the initial contents of `stuff` should now be the command line arguments, instead of those you previously hard-coded.
  - (hint: don't make this harder than you need to -- it turns out that you can put the appropriate expression inside of `stuff`'s parentheses, but you should remember to quote that expression, also.)
- only echo the value of the element of `stuff[3]` if its length is non-zero -- otherwise, JUST set `stuff[3]` to a value of your choice
  - (hint: its length can be zero if there are fewer than 4 command-line arguments OR if you actually give an empty string as a command-line argument -- the same test works in either case...)

Submit your resulting `stuff-play2.sh` .