Name: Kevin Tran | **PID:** A16809193 | **(page 1)**

`find «path»:` Recursively traverse the given path and list all files in that directory and subdirectories
- **`-name "pattern"`** option to only match files with a specific pattern (like **`find . -name "*.java"`**)
- (**`find`** has many more options! You can look them up online or with **`man`**)

`wc «file»`: Print the number of lines, words, and characters in a file or files

`grep «string» «files»:` Search a file or files for the given string, print matching lines
- **`-r`** option that can a directory path and search it recursively (instead of just one or more listed files)

`«command» > «file»` Save the output of the command in the given file. Overwrites the file!

`*` (asterisk, star) Used to create **patterns**, which expands to all matching paths.
Examples: `lib/*.jar`, `*.txt`

`echo «args»` Print the arguments to the terminal

`rev` Prints the text of each line reversed (from a file arg or standard input)

`cut -d'«delimeter»' -f «n»`
Split each line by **`delimiter`** and print the **`n`**th element

`tail -n` Take the last **`n`** lines of input (file arg or stdin)
`head -n` Take the first **`n`** lines of input (file arg or stdin)

`sort` Print lines in sorted (lexicographic) order
`uniq` Remove adjacent identical lines from input

https://github.com/bluesky-social/social-app

Bluesky, also known as **Bluesky Social**, is a microblogging social platform and a public benefit corporation. The service is focused on microblogging, and has been called "Twitter-like". Bluesky Social was made open source under the MIT license in May 2023.[27]

```
social-app % ls
Dockerfile                  bskyweb                     patches
Gemfile                     docs                        plugins
LICENSE                     eas.json                    scripts
Makefile                    google-services.json.example src
... a few more files ...
```

What's a command for counting the lines/characters of all the TypeScript (`.ts`) files in `src`?

find then pipe "|" to wc

- name *.ts

wc $(find src -name *.ts)

wc src/*.ts <-- does not find .ts files recursively

wc src/*.ts src/*/*.ts <-- doesn't find 2 directories deep

What about all of the `.ts` and `.tsx` files in `src`?

find . -name '*.ts' | xargs wc

wc $(find src -name '*.ts' '*.tsx')

what about .js files in src/?

find src -name '*.js'

note the quotes! don't want bash to expand *

xargs wc < typescripts.txt

`xargs «command»` perform «command» after reading standard input to get all the command-line arguments
`«command» | «command»` **"pipe"** – Take the output of the first command and use it as the input to the second command.

Any other ways to write the example above with **`|`** and **`xargs`**?

find . -name '*.ts' | xargs wc <-- "one liner"find . name '*.ts' >

What about getting all the file extensions in use across the whole project?

start with find (on ALL files)
isolate the file extension part
see which unique extensions are present

cut -d'.' -f 1
gets the first element splitting on dot

cut -d'/' -f 2
gets 2nd element splitting on /

find . > all-files.txt
find . -type f | rev | cut -d'.' -f 1 | rev | sort | uniq

What about counting the number of lines/characters *by file extension*? (e.g. how many lines of .ts code, how many lines of .tsx code)

Another useful resource (if we have time) – there's a dictionary installed in a plain text file on most operating systems. Anything interesting we can do with it?

```
bash-3.2$ wc /usr/share/dict/words
  235976  235976 2493885 /usr/share/dict/words
bash-3.2$ head -n 10 /usr/share/dict/words
A
a
aa
aal
aalii
aam
Aani
aardvark
aardwolf
Aaron
```

```
bash-3.2$ tail -n 10 /usr/share/dict/words
zymotoxic
zymurgy
Zyrenian
Zyrian
Zyryan
zythem
Zythia
zythum
Zyzomys
Zyzzogeton
```