# Search the files that satisfy certain requirements

Write a bash script that find all the files in a directory and its subdirectories that are owned by a user but can be read by any users. The script takes two arguments. The first argument is the pathname of the directory and the second argument is a user id.

Note that your script needs to traverse the directory and check the files under its subdirectories, sub-subdirectories, etc. During the traversal, for each file (assuming file name saved in variable `filename`), your script need 1) to use command `ls -l ${filename}` to get the information of the file, 2) parse the line generated by the `ls` command using `grep` or `expr` and determine whether the file satisfies the above requirements or not, and 3) if the file satisfies the requirements, print out the following information of the file:

- file name
- permissions (a group of 9 characters consists of r, w, or x, do not include the character for file type at the beginning of the line)
- time of creation or last modification.

For the format of the information printed out by ls -l, refer to these pages: https://cr.yp.to/ftp/list/binls.html, https://linuxize.com/post/how-to-list-files-in-linux-using-the-ls-command/ . Check the owner field and the read permissions to determine whether a file satisfies the requirements. A file that can be read by any users is the one with three "r" permissions.

When you extract the time of creation or last modification, your code should be flexible to handle two time formats: month+day+hour+minute for files modified/created within the last six months, and month+day+year for other files.

To use `grep` to process the line printed out by `ls -l,` you can use a pipe to connect `ls` command and `grep` command. For example the following commands extracts all the numbers.

```
$ ls -l /bin/bash | grep -o '[0-9]*'
1
1113504
6
2019
```

This exercise is for you to practice the use of regular expressions. DO NOT use commands find and cut in your script. To extract the desired information from a string, consider to use sub-string, or `grep -o`.

Escape ("\") parentheses and braces if you use BRE.

**Testing**: to test the script, run it with "root" and "/usr/share/docutils/writers/" as arguments, you should see the information of all 36 files. Randomly select a few of these files, and manually check whether the information printed out by your script matches the corresponding information printed out by `ls -l`.