

Offline Assignment-01

TASK A: File Length Checker

Write a bash script named `file_length` that interacts with the user to check the number of lines in multiple files and categorizes each file as **empty**, **small**, or **large**.

Subtasks

1. **Query Count:** Prompt the user to enter the total number of files to check.
2. **Iterations:** Use a for loop to iterate exactly `num_queries` times.
3. **Remaining Queries:** At the start of each iteration, display how many queries remain.
4. **Filename Input:** Prompt for a filename on each iteration:
5. **Line Count:** Count the number of lines in the specified file.
 - You may assume that the file exists. Use `wc -l`.
 - Extracting the number only can be tricky. But there are multiple ways.
6. **Output:** Print the line count and based on that print its category.
 - **Empty:** `num_lines == 0`
 - **Small:** `1 <= num_lines < 10`
 - **Large:** `num_lines >= 10`

Deliverable

A single executable script named `file_length` that reproduces the sample output shown below. Lines beginning with `>` asks for user input (prompts), while the remaining lines are program output.

```
$ ./file_length
> Number of queries: 3
You have 3 queries remaining
> Filename: dummy.sh
Number of lines: 0
dummy.sh is empty!
You have 2 queries remaining
> Filename: if_checker
Number of lines: 23
if_checker is large!
You have 1 queries remaining
> Filename: loop_checker
Number of lines: 5
loop_checker is small!
```

TASK B: Shell-Script Audit

Write a program `audit_scripts` that discovers shell scripts and counts occurrences of a user-supplied keyword.

Subtasks

1. **Directory Prompt:** Prompt the user for a directory path.

2. **Find Scripts:** Use `find` to locate **all** files ending in `.sh` under the given directory. Store them into an array. Here's a sample code snippet for creating such a list and accessing the first item.

```
files=$(find . -type f)
echo ${files[0]}
```

3. **Keyword Prompt:** Prompt the user for a keyword that will be searched for.
4. **Count Matches:** For each `.sh` file returned by `find`, run a case-insensitive search to count the number of matching lines using `grep`.
5. **Report Results:** Print one line per script file in the format:
`./path/to/script.sh : X occurrences`

Deliverable

A single executable script named `audit_scripts` that reproduces the sample output shown below. Lines beginning with `>` asks for user input (prompts), while the remaining lines are program output.

```
$ ./audit_scripts
> Enter a directory to scan for shell scripts: ./projects
> Enter a keyword to count in found scripts: TODO
./projects/install.sh : 3 occurrences
./projects/tests/run_tests.sh : 0 occurrences
./projects/utils/helpers.sh : 1 occurrences
```

Submission Instructions

Create two bash scripts named `file_length` and `audit_scripts`, and place them both in a single directory named with your student ID. Compress this directory into a ZIP archive and upload it to eLMS.