

---

## Homework 1

System Skill (Term II/2022)

*built on 2023/01/18 at 22:58:50*

**due:** *Thursday, February 2nd @ 11:59pm*

**Be sure to read this problem set thoroughly, especially the sections related to collaboration and the hand-in procedure.**

### Collaboration

We interpret collaboration very liberally. You may work with other students. However, each student **must** write up and hand in his or her assignment separately. Let us repeat: You need to write your own code. You must not look at or copy someone else's code. You need to write up answers to written problems individually. The fact that you can recreate the solution from memory will be taken as proof that you actually understood it, and you may actually be interviewed about your answers.

*Be sure to indicate who you have worked with (refer to the hand-in instructions).*

### Hand-in Instructions

To submit this assignment, please follow the steps below:

1. Make sure your scripts run and work correctly on a Linux machine. We will run the grading script on the server.
2. Zip up all the scripts and name it `a1.zip`

---

```
> zip a1.zip lines.sh quota.sh extrm.sh flatten.sh invite.sh happy_countries.sh git.sh
```

---

3. Find out the MD5 hash of your zip file. You will need to submit this code on Canvas. We use it for keeping track of your submission time. You may resubmit your work but the MD5 hash has to match.

---

```
> md5sum a1.zip
```

---

4. Submit the zip file in our syskill server at `/handin/a1/uXXXXXX` where XXXXXX is your student ID and put the MD5 on your Canvas submission.

### Task 1: Tenth Line (10 points)

Given a directory `someDir`, print the tenth line of all files in this directory. You do not need to print any files in its subdirectory. Save this script into `lines.sh`

### Task 2: Disk Quota (10 points)

Assume that you were a system admin. Your team was assigned with gathering information of disk space in use. In particular, you are charged with writing a shell script called `quota.sh` to determine how much a particular directory has consumed its disk quota. To make this information easy to understand, we classify a directory as follows:

- **Small** if the directory uses less than 128 kB disk space
- **Medium** if the directory uses less than 1 MB disk space
- **Large** if the directory uses equal to or more than 1 MB disk space

**Expected output** Given a directory, the script should output the classification of the directory's size as Low, Medium, or High. For those directory that is classified as Low, append the path to the directory you tested to `~/ListOfSmallDirs.txt`.

---

```
> ./quota.sh /usr/bin
High
[and then append /usr/bin to ~/ListOfBigDirs.txt]
> ./quota.sh /tmp
Low
```

---

### Task 3: File Extension Remover (10 points)

Given a specific extension and a directory, write a shell script called `extrm.sh` to remove the extension from the name of any file with that specific extension. For example, assume that the extension `jpg` is given as input to the script along with a path to a directory, and the given folder contains two files that have that extension, named `image1.jpg` and `image2.jpg`. The script in `extrm.sh` should rename these two files these two files to `image1` and `image2`, respectively.

---

```
> ls images
a.jpg b.jpg c.txt
> ./extrm.sh jpg ./images
> ls images
a b c.txt
```

---

### Task 4: Flatten then Zip (10 points)

We have had headache every term from students' submissions that do not conform to the assignments' file requirements, especially the unwanted directories in the submitted compressed files (especially when you submit the entire workspace on your IDEs). Therefore, you are assigned to help us out by writing a bash script called `flatten.sh` that 1) flatten all files in the current directory and all other files

in all subdirectories inside this current directory into one single level and 2) compress **only .c files** into a zipped file. In case of files having the same filename, you are required to keep only one of them.

For example, the current directory contains the following files and directories:

---

```
| - a1
|   | - p1.c
|   | - p2.c
|   | - README.txt
| - p2.py
| - p3.c
| - README
```

---

Here is the expected output of your script.

---

```
> ./flatten.sh ./handin.zip
```

---

will create a file called handin.zip with p1.c p2.c p3.c all together in one single directory when unzipped.

## Task 5: CSV Extractor (10 points)

An organization is organizing an virtual event with a raffle and is identifying a subset of the attendees that born in a certain month. Therefore, the event organizer needs to list only eligible members. The members' information is given in a CSV file, and the record format in this data file is as follow:

firstName,lastName,YYYYMMDD

However, the year can either be in a Gregorian calendar, where the current year is 2022, or a Buddhist calendar, where the current year is 2565.

You are asked to help them by writing a shell script `invite.sh` that accept a location of a CSV file that contains records of the original format, performs this filtering, and outputs the information. Please note that you can check the type of calendar using a normal human lifespan.

### Expected output

---

```
> cat guest.csv
James,Smith,19990123
Maria,Garcia,25580512
Jane,Hernandez,20031010
Peter,Wong,20000421
> ./invite.sh May guest.csv
Maria,Garcia,25580512
```

---

## Task 6: Happy Countries (10 points)

Write a bash script called `happy_countries.sh` that lists out the names of the countries reported by 2022 World Happiness Report ranked by the happiness.

Your script will retrieve the information directly from the following wikipedia page, [https://en.wikipedia.org/wiki/World\\_Happiness\\_Report](https://en.wikipedia.org/wiki/World_Happiness_Report). However, parsing the data directly from HTML

is a headache. Luckily, you can request the page in an alternative format (raw wiki format) by using ([https://en.wikipedia.org/wiki/World\\_Happiness\\_Report?action=raw](https://en.wikipedia.org/wiki/World_Happiness_Report?action=raw)). Notice the suffix `?action=raw`.

To keep things simple, you are to list only top 5 countries. For this task, you may only use the following tools: `curl`, `sed`, `awk`, `grep`, `tr`, `cut`, `sort`, `head`, `tail`.

Below is expected output of your script.

---

```
Finland
Norway
Denmark
Iceland
Switzerland
Netherlands
Canada
New Zealand
Sweden
Australia
```

---

## Task 7: Git (10 points)

Sometimes it is handy to have a script that just print out a shorter form of git log. This question will ask you to create a script to create a brief version of the git commit history. Assuming that you are in a folder that is also a git repository. Please write a script called `git.sh` to print out a list of the all commits, with each line containing the following format [commit author]: commit date. For example, if I print out a log of a git repo with two commits on 7th Feb and 14th Feb, both by me, then the script should print

```
[Rachata Ausavarungnirun]: 7 February 2021
[Rachata Ausavarungnirun]: 14 February 2021
```

Note that you can just use the authors' name and date given by the command `git log`. No need to discover all the logs of all branches or anything, just call the `git log` command inside your script and process it.

**Notes:** All operations must be performed through command lines, unless the instruction specifies otherwise. A step may require multiple commands.