

Lab 6: grep, awk, pipe, and others**Due date:** 10/26/2021

In this lab assignment we will be practicing with **grep** and **pipng** along some other commands. Some of the features used in this assignment are explained in the textbook and slides but there might be a few things that you will need to learn from other sources. You can always use a search engine to get more information on commands or ways to do certain things.

To begin, create a directory named **lab6** in your home directory, then cd to it. Copy files **/files/labs/lab6/etc.passwd** and **/files/labs/lab6/judgeHSPC05.txt** to this directory.

Write a **single shell script** named **lab6scr_2344_t#** (where # is your team number) which uses **grep** to get the following information from **judgeHSPC05.txt**:

1. List the lines and their corresponding numbers where the strings "Problem 12" and "Problem 13" appear in the file?
2. How many lines contain at least one occurrence of the string "problem" (case does not matter)?
3. How many lines start with a capital letter or a period (.)?
4. How many lines start with a period (.) followed by an asterisk (*)?
5. How many lines end with a period (.) or a question mark (?)?
6. How many lines contain a dollar sign (\$) followed by a two-digit number?¹
7. How many lines contain at least one six-letter word (case does not matter)?¹
8. How many lines start with anything but a letter?

Now add to **lab6scr_2344_t#** the commands that will get the information specified by questions 9 through 13 from the file **etc.passwd**. Note that this file is organized as follows:

account:password:UserID:GroupID:user name:home directory:shell

Note that you may need to use a variety of commands along with piping and redirection to accomplish these tasks.

9. Which are the account names that have the word "daemon" in its record? Display **account** names only (no other piece of information should be shown).
10. Which are the different groups between 0 and 1000 (including 0 and 1000) that are in the file? Display them sorted in ascending order.
11. List the **account** name of users who have **bash** as their default shell in descending order.
12. List **account** name and user id of those users who do not have a **user name** (the corresponding field is empty). The list must sorted by used id.
13. How many users in total are there in groups 44 and 100?

¹: See clarifications about these two questions at the end of this handout please.

Each answer to the above questions in the script must be preceded by a line like the one shown below:

echo == Answer to question X == (where X is the question number).

Add a heading to your script file like the one shown below.

```
# =====
# Script name: lab6scr_2344_t#
# By: Joe Doe and Jane Dane
# Date: 4/3/19
# Purpose: Get information from judgeHSPC05.txt and etc.passwd
# =====
```

Once your script has been thoroughly tested run it again and redirect its output to a file named `mysolution`. Copy file `/files/labs/lab6/solutiongd` (this is the output generated by my solution) to your current directory and then use command **diff** to determine if your solution is correct or not. If your solution is correct, submit it. If your solution is not correct, fix it and test it until you get it right and then submit it.

When done, submit lab6scr_2344_t# through Blackboard using the “Assignments” tool. Do Not email it.

Questions 6 and 7 clarifications:

6. How many lines contain a dollar sign (\$) followed by a two-digit number?

This means a \$ sign followed strictly by 2 digits (which may be followed by any other characters except digits). For example "\$10 ", "\$10.0", "\$10;", and "\$10/week" should be counted while "\$100", "\$.10", and "\$1.0" should not be counted. I used the double quotes just to show you the string that I am considering in the analysis. In the first example a blank space follows the 2 digits; in the next three you have a decimal point, a semicolon, and a slash. To solve this problem you have to use a certain option so use `man` to see which one will work for you :-)

7. How many lines contain at least one six-letter word (case does not matter)?

The command to answer this question is similar the one above; the difference is the definition of the pattern (in this case a word that consists precisely of 6 letter regardless of the case). The words are "surrounded" by non-letter characters.

These are a few examples:

```
Test output:          ->  " output:"
Problem 12 (30 points) ->  " points)"
  leading spaces.     ->  " spaces."
```