14/8/2023

1. Write a bash script that accepts a few arguments(all numbers) and performs the following functions. Prints the string Error if the number of arguments supplied is not equal to 2. If the number of arguments is equal to two, print their sum.
2. Write a script to print the users(one on each line) who are logged in successfully. Extract the information from the file named myauth.log located in the current working directory. The output should contain usernames only and should be unique. Hint: Use uniq command to get all distinct lines of the output. Contents of myauth.log is given below:  
   Jan 20 20:11:34 IITMBSC systemd-logind[897]: Session 27 logged out. Waiting for processes to exit.

Jan 20 20:11:34 IITMBSC systemd-logind[897]: Removed session 27.

Jan 20 20:17:01 IITMBSC CRON[70999]: pam\_unix(cron:session): session opened for user root by (uid=0)

Jan 20 20:17:01 IITMBSC CRON[70999]: pam\_unix(cron:session): session closed for user root

1. Given a file words.txt containing a string in each line in the format FIRST\_second. Every string is a combination of two words joined with an underscore(\_), the first word FIRST consists of all uppercase letters and the second word second consists of all lowercase letters. Write a bash command/script using sed to convert all the string to SECOND\_first.

After conversion

- The first and the second words should be swapped.

- The uppercase word should be converted to lowercase word and vice versa.

The file `words.txt` is located in the current working directory.

1. Given a file input.txt containing a word on each line, print all the words(one in each line) that occur between the words "FROM" and "TO"(but excluding these words). The match should be case sensitive for the given words and the words in the file are not unique, they can repeat. For e.g. for Input file

$ cat input.txt

This

is

TO

some

4word

FROM from FROM

THE

1. Write an awk script to find unintentionally repeated (duplicate) words in the file 'myfile.txt'. For example, sometimes a file can contain sentences like "The the building is beautiful". Print the repeated words on the order of occurence at one per line. [awk '<code> ' myfile.txt]
2. EmployeeDetails.csv file contains the Employee ID, Employee Name, Leaves taken this year and Gender, of all the employees working in a company XYZ, born between the years 1997 and 2000 (including both). Total employees in the company is less than 1000.

The employee ID is of the format: DepartmentYearOfBirthCode Where:

- Department is the department to which the employee belongs to (Department A to G)

- YearOfBirth is the birth year of the employee (Eg. 2000)

- Code is a three digit number unique to each employee.

For e.g. B1997122 is employee id of an employee working in department B, born in the year 1997 having unique code as 122. The email ID of an employee is in the format EmployeeID@xyz.com, where EmployeeID is the employee id of the employee.

For example email id of Ram having employee id as A1998001 is A1998001@xyz.com. Email ids are case sensitive.

Write an awk script that takes input as file EmployeeDetails.csv and calculate and prints the average number of leaves taken by the employees born in each year from 1997 to 2000(both 1997 and 2000 included). The average for each year should be printed on a newline starting from the year 1997 to 2000 in the same sequence i.e. your script should print 4 lines of output always one for each year 1997, 1998, 1999 and 2000. If there are no employees born in some year, print 0 for that years average leaves. Print only the integer part of the average(i.e. if the average is 7.3333 print 7). Use int() function to get the integer part of any float number.

Sample

Suppose your awk script is named as yourScript.awk. For the below sample file the average number of leaves taken by employees born in years 1997 to 2000 is printed in 4 lines in the same sequence.

$ cat EmployeeDetails.csv

A1998001,Ram Kumar,10,Male

A1998002,Mohammed Iqbal,5,Male

A1998003,Priya Lal,7,Female

A1999001,Sunita Sharma,25,Female

A2000001,Rose Mary Thomas,13,Female

B1999001,Sri Lakshmi Jai,5,Female

​

$ awk -f yourScript.awk EmployeeDetails.csv

0

7

15

13

15/08/2023

1. Write a bash script which takes one argument as the name of a file and prints Yes if the file has read permission only for the owner and no other permissions for owner or other users, else do not print anything. The file given in the argument will be present in the current working directory.
2. Print the previous login time of the user guest in the format MMM DD HH:MM:SS. Where MMM, DD, HH, MM and SS corresponds to Month (E.g. Nov), Date, Hours, Minutes and Seconds respectively. Extract the information from the logs available in the file myauth.log in the current directory. Sample log file below.

Jan 21 19:42:14 IITMBSC PackageKit: uid 1000 obtained auth for org.freedesktop.packagekit.system-sources-refresh

Jan 21 19:42:17 IITMBSC PackageKit: uid 1000 is trying to obtain org.freedesktop.packagekit.system-sources-refresh auth (only\_trusted:0)

Jan 21 19:42:17 IITMBSC PackageKit: uid 1000 obtained auth for org.freedesktop.packagekit.system-sources-refresh

Jan 21 19:42:20 IITMBSC PackageKit: uid 1000 is trying to obtain org.freedesktop.packagekit.system-sources-refresh auth (only\_trusted:0)

1. Without using the wc or awk commands(instead use sed as Bash command), write a bash script that accepts any number of arguments. Out of these some would be options(hyphen plus a character like -l or -c) and the last argument will be a file path(use ${@: -1} to access the last argument, there is a space before -1). Only four options are accepted by your script -l, -w, -n and -s.

Assume that file path given will always be for a valid file and we will refer it as file in the next lines. For options,

If no option is supplied to your script do nothing.

If -l option is supplied, print the number of lines in the file.

If -w option is supplied, print the number of words in the file. Assume that any string between spaces is a word. i.e. if using awk count the number of fields in each line to get the word count.

If -n option is supplied, print the number of lines having only digits(no alphabets or spaces) in the file.

option -s also accepts an argument say str. In this case print the number of lines containing the string str.

The above options can be supplied together or more than once. Print the required count for each appearance of the option on a new line. For e.g.

if -l and -w are both supplied together in the sequence print count of lines and count of words each on separate lines.

If -l, -n and -l options are supplied in the sequence then print number of lines, number of lines containing only digits and finally again number of lines in the file each on separate line.

Note: Your bash script should not even contain any variable or comment that contians the string wc or awk.

Hints:

Use while getopts style code.

Use sed to find the count of lines as specified in the conditions aboce.

Sample

Suppose your bash script is named as myCount.sh. In the below sample the argument to -s option is "say" so this should count all the lines containing the string "say". For the public test case all the commands given in the below sample are executed one by one on the input file.

$ cat somefile.txt

This is a sample file

this is not end justsay start

that contains say

some number

say like 10

or

20

or

233

444

or say 3444

and now it ends.

$ bash myCount.sh -l somefile.txt

12

$ bash myCount.sh -w somefile.txt

32

$ bash myCount.sh -n somefile.txt

3

$ bash myCount.sh -s say somefile.txt

4

$ bash myCount.sh -l -n somefile.txt

12

3

$ bash myCount.sh -l -s say -l -n somefile.txt

12

4

12

3

$ bash myCount.sh

$ bash myCount.sh somefile.txt

1. Write a sed command to print the count of lines that starts with a digit in the file input.txt. Assume that there is at least one line in the file input.txt that starts with a digit. Do not use the commands wc or awk , or even these keywords in comments or anywhere in your answer.
2. Write an awk script that reads a value n from the stdin within awk script, then prints the sum of odd numbers and sum of even numbers each on a separate line respectively, from the set of natural numbers from 1 to n (ends inclusive).
3. You have a csv file named groceries.csv that contains a list of grocery items and their unit cost. The two fields are separated by comma(,). This file will be given as input to your Awk script.

Write an Awk script that takes two arguments(command line) named item and n, where item is the item name and n is the number of units, then prints the total cost of purchasing n units of the item item. The script prints only a number. i.e. you need to find the item cost of the item given in argument while parsing the input file.

Note: You can directly use these variables with the given name in your Awk script. Assume that the item given in the argument will always be present in the csv file.

Sample(suppose your script is named as yourScript.awk)

Here the cost of 3 Tomatoes needs to be calculated. Cost of one Tomato is 40 as seen from the csv file. So total cost = 3\*40

$ cat groceries.csv

1,Tomato,40

2,Brinjal,35

3,Banana,60

$ awk -f yourScript.awk groceries.csv Tomato 3

120

16/08/2023

1. Write a bash script that reads a value from the standard input stream and prints PNUM if the value is a postive number or 0; prints NNUM if it is a negative number; else print STRING.
2. Write a Bash command to print the number of failed login attempts which are recorded in the file myauth.log located in the current working directory.

Contents of myauth.log is given below

Jan 20 20:11:34 IITMBSC systemd-logind[897]: Session 27 logged out. Waiting for processes to exit.

Jan 20 20:11:34 IITMBSC systemd-logind[897]: Removed session 27.

Jan 20 20:17:01 IITMBSC CRON[70999]: pam\_unix(cron:session): session opened for user root by (uid=0)

Jan 20 20:17:01 IITMBSC CRON[70999]: pam\_unix(cron:session): session closed for user root

Jan 20 20:21:10 IITMBSC su: (to root) student on pts/4

Jan 20 20:21:10 IITMBSC su: pam\_unix(su:session): session opened for user root by student(uid=0)

Jan 20 20:21:21 IITMBSC su: pam\_unix(su:session): session closed for user root

1. In the lines that start with a digit, if there is a words "delta"(case sensitive) replace it with the word "gamma". Replace only the first occurrence of the word "delta" in the desired lines. The filename where the contents present are input.txt.
2. Given a file that contains current years board exam scores of students in all the schools in an area. Each line in the file contains four comma-separated fields: school code, roll number of the student, name of the student, and marks in the below format.

School\_code,Student\_Roll\_no,Name,Marks

All the fields could be alphanumeric values except the last field Marks which is a number less than 500, as the maximum marks of the exam is out of 500.

Write an AWK script to print the roll numbers of the toppers of each school. For example if there is marks details of students of 11 schools of an area in the file, then your output should contain 11 roll numbers in any order, one for each school.

Test case description: Input is the contents of the input file. Your script does not need to read any input. Output is the expected output from your script. Your script may print expected roll numbers in any order, evaluation script sorts the input before printing.

1. Write a script using AWK to print the file with the maximum number of lines. Assume only one file that have the maximum number of line among the given files.
2. EmployeeDetails.csv file contains the Employee ID, Employee Name, Leaves taken this year and Gender, of all the employees working in a company XYZ, born between the years 1997 and 2000 (including both). Total employees in the company is less than 1000.

The employee ID is of the format: DepartmentYearOfBirthCode Where:

- Department is the department to which the employee belongs to (Department A to G)

- YearOfBirth is the birth year of the employee (Eg. 2000)>

- Code is a three digit number unique to each employee.

For e.g. B1997122 is employee id of an employee working in department B, born in the year 1997 having unique code as 122. The email ID of an employee is in the format EmployeeID@xyz.com, where EmployeeID is the employee id of the employee.

For example email id of Ram having employee id as A1998001 is A1998001@xyz.com. Email ids are case sensitive.

Write an awk script takes the file EmployeeDetails.csv as input and prints the name of the employee(s) with lowest number of leaves taken this year. If there are more than one employees with the lowest number of leaves, print the name of each employee on a new line.

Sample

Suppose your awk script is named as yourScript.awk. For the below sample file the lowest number of leaves taken by any employee is 5. And there are two employees who have taken only 5 leaves, print both employee names on a separate line.

$ cat EmployeeDetails.csv

A1998001,Ram Kumar,10,Male

A1998002,Mohammed Iqbal,5,Male

A1998003,Priya Lal,7,Female

A1999001,Sunita Sharma,25,Female

A2000001,Rose Mary Thomas,13,Female

B1999001,Sri Lakshmi Jai,5,Female

$ awk -f yourScript.awk EmployeeDetails.csv

Mohammed Iqbal

Sri Lakshmi Jai

17/08/2023

1. Write a bash script that takes any number of inputs(all numbers) and prints the maximum and minimum value from all the inputs in the format Maximum: max | Minimum: min, where max is the maximum value and min is the minimum value.
2. Mine the logs given in the file myauth.log present in the current working directory to print all the usernames to which user student switched to using su command.

Note: switching back to the previous user should not be accounted.

Hint: Basically you have to grep all the lines where 'su' command is run successfully and fetch the username to which the user student switched to.

Contents of myauth.log is given below

Jan 21 20:32:17 IITMBSC polkitd(authority=local): Unregistered Authentication Agent for unix-session:c2 (system bus name :1.525, object path /org/freedesktop/PolicyKit1/AuthenticationAgent, locale en\_US.UTF-8) (disconnected from bus)

Jan 21 20:32:32 IITMBSC login[119373]: pam\_unix(login:session): session opened for user guest by LOGIN(uid=0)

Jan 21 20:32:32 IITMBSC systemd-logind[966]: New session 31 of user guest.

Jan 21 20:32:32 IITMBSC systemd: pam\_unix(systemd-user:session): session opened for user guest by (uid=0)

Jan 21 20:32:41 IITMBSC login[119373]: pam\_unix(login:session): session closed for user guest

Jan 21 20:32:41 IITMBSC systemd-logind[966]: Session 31 logged out. Waiting for processes to exit.

1. Consider a special programming file functions.sh that contains several functions (A function is a block of code). Write a bash script/command using sed to insert a line "# START FUNCTION" before the starting of a function and a line "# END FUNCTION" at the end of the function.

Starting of a function in this file can be identified as a line that has some string followed by "(", then followed by ")" or some string followed by ")", and this line should end with "{".

Ending of a function can be identified by a line containing only "}" in the whole line.

In this file curly braces "{" and "}" are not used for any other purpose. Do not change the original file just print the output to STDOUT.

1. Write a sed script to Swap the first and second fields in the given input having field separator as a colon :

Replace every occurrence of the character ? if found at the end of a line to !

Note: complete partial tasks for partial marking. Test case description: The input is the input file. And output is the output of running your sed script.

1. Consider a file named marks.csv containing roll number and marks of variable number of subjects of students. The values are comma separated values and in the format

RollNo,Subject1,Subject2,Subject3,So on...

Write an Awk command to print all the roll numbers(RollNo) in the file.

1. A software company has published some best practices for writing the code. One of the best practice mentioned is that if no line in your code should exceed 50 characters in total including all type of characters or spaces.

Given a bash script that intends to print the names of all .c files that contain one or more lines with length more than 50 characters(as specified above).

The awk script within this bash script to check the files as per above condition is missing in the code, complete that

18/08/2023

1. Write a bash script that takes a number as an argument and prints "Yes" if the number is a prime number, else prints "No".
2. Write a script to print the users(one on each line) who are logged in successfully. Extract the information from the file named myauth.log located in the current working directory. The output should contain usernames only and should be unique.

Hint: Use uniq command to get all distinct lines of the output.

Contents of myauth.log is given below

Jan 20 20:11:34 IITMBSC systemd-logind[897]: Session 27 logged out. Waiting for processes to exit.

Jan 20 20:11:34 IITMBSC systemd-logind[897]: Removed session 27.

Jan 20 20:17:01 IITMBSC CRON[70999]: pam\_unix(cron:session): session opened for user root by (uid=0)

Jan 20 20:17:01 IITMBSC CRON[70999]: pam\_unix(cron:session): session closed for user root

Jan 20 20:21:10 IITMBSC su: (to root) student on pts/4

Jan 20 20:21:10 IITMBSC su: pam\_unix(su:session): session opened for user root by student(uid=0)

1. Given some raw programming files, we want them to adhere to the company guidelines. Write a sed script that will run for all ".sh" files in the current directory and print the contents after performing the following actions. You just need to write the sed script, running that for all the files will be taken care of by our driver bash script.

Insert a copyright message at the start of the file(before the first line) as "# Copyright IITM 2022"(Note that there is a space after #).

Insert a copyright message at the end of the file(after the last line) as "# Copyright IITM 2022".

Insert a line "# START FUNCTION" before the starting of a function and a line "# END FUNCTION" at the end of the function. Check GrPA 4 for more details on identifying function boundaries. Use the same logic here.

Change the function "background\_sleep" to "inactive\_sleep". So replace all the occurrences of the word "background\_sleep" in any line with "inactive\_sleep". Assume that these keywords are used only in context of a function and nothing else.

Also, the function "active\_sleep" is deprecated and we do not have an immediate replacement. So insert a line "# TODO:DEPRECATED" before the function "active\_sleep" and in every instance. i.e. before every line containing the word "active\_sleep".

After every 10th line (in line numbers 10, 20, 30,... ) add a line with four hashes such as "####" after applying all the above actions.

Perform all the above actions in the order given from top to bottom.

For example, for the input file

echo Hello

EOF

analysis.sh

script() {

sum=0

for i in $(cat result); do

while read hash name; do

if [ $i == $hash ]; then

inv=$(grep INVESTMENT $name)

inv=${inv//INVESTMENT $/}

sum=$((sum+inv))

fi

done < map

done

echo $sum

}

mkdir data

read fnos

for (( i=0; i<fnos; i++ )); do

read line

echo $line | cut -d ":" -f 2- | tr '#' '\n' > ./data/${line%%:\*}

done

read mnos

for (( i=0; i<mnos; i++ )); do

read line

echo $line >> map

done

read rnos

for (( i=0; i<rnos; i++ )); do

read line

echo $line >> result

done

script

Output should be

# Copyright IITM 2022

echo Hello

EOF

analysis.sh

# START FUNCTION

script() {

sum=0

for i in $(cat result); do

while read hash name; do

if [ $i == $hash ]; then

inv=$(grep INVESTMENT $name)

####

inv=${inv//INVESTMENT $/}

sum=$((sum+inv))

fi

done < map

done

echo $sum

}

# END FUNCTION

mkdir data

####

read fnos

for (( i=0; i<fnos; i++ )); do

read line

echo $line | cut -d ":" -f 2- | tr '#' '\n' > ./data/${line%%:\*}

done

read mnos

for (( i=0; i<mnos; i++ )); do

####

read line

echo $line >> map

done

read rnos

for (( i=0; i<rnos; i++ )); do

read line

echo $line >> result

done

####

script

# Copyright IITM 2022

1. Write a sed script to

Replace every three letter abbreviation of day of week to the respective number as given in the table. The abbreviation can be in any case or mixed case

Abbreviation(case insensitive match) Replace with

sun 1

mon 2

tue 3

wed 4

thu 5

fri 6

Sat 7

Change the first and third occurrences of the character ! to ., on each line.

Note: complete partial tasks for partial marking. Test case description: Input is the input file to sed. Output is the output printed on running your sed script.

1. Write an Awk command to print the first field of the all the lines containing more than 20 characters in the file marks.csv. The field separator in the file is comma (,).

EmployeeDetails.csv file contains the Employee ID, Employee Name, Leaves taken this year and Gender, of all the employees working in a company XYZ, born between the years 1997 and 2000 (including both). Total employees in the company is less than 1000.

The employee ID is of the format: DepartmentYearOfBirthCode Where:

- Department is the department to which the employee belongs to (Department A to G)

- YearOfBirth is the birth year of the employee (Eg. 2000)>

- Code is a three digit number unique to each employee.

For e.g. B1997122 is employee id of an employee working in department B, born in the year 1997 having unique code as 122. The email ID of an employee is in the format EmployeeID@xyz.com, where EmployeeID is the employee id of the employee.

For example email id of Ram having employee id as A1998001 is A1998001@xyz.com. Email ids are case sensitive.

Write an awk script that that takes the file EmployeeDetails.csv as input and prints the email ids of all the female employees of the company in the same sequence as the employee details appear in the file EmployeeDetails.csv.

Sample

Suppose your awk script is named as yourScript.awk

$ cat EmployeeDetails.csv

A1998001,Ram Kumar,10,Male

A1998002,Mohammed Iqbal,5,Male

A1998003,Priya Lal,7,Female

A1999001,Sunita Sharma,25,Female

A2000001,Rose Mary Thomas,3,Female

B1999001,Sri Lakshmi Jai,5,Female

$ awk -f yourScript.awk EmployeeDetails.csv

A1998003@xyz.com

A1999001@xyz.com

A2000001@xyz.com

B1999001@xyz.com