Data analysis/ manipulation (Awk)

Id Employee Name Job Title **Base Pay** Overtime Pay Other Pay Total Pay TotalPayBenefits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 NATHANIEL | | GM | 167411 | 0 | 400184 | 567595 | 567595 |
| 2 GARY | | CAPTAIN | 155966 | 245131 | 137811 | 538909 | 538909 |
| 3 ALBERT | | CAPTAIN | 212739 | 106088 | 16452 | 335279 | 335279 |
| 4 | CHRISTOPHER | MECHANIC | 77916 | 56120 | 198306 | 332343 | 32343 |
| 5 PATRICK | | DEPUTY CHIEF | 134401 | 9737 | 182234 | 326373 | 326373 |
| 6 DAVID | | ASST DEPUTY | 118602 | 8601 | 189082 | 316285 | 316285 |
| 7 ALSON | | BATTALION CHIEF | 92492 | 89062 | 134426 | 315981 | 315981 |
| 8 | DAVID | DEPUTY DIRECTOR | 256576 | 0 | 51322 | 307899 | 307899 |
| 10 JOANNE | | CHIEF | 285262 | 0 | 17115 | 302377 | 302377 |
| 12 PATRICIA | | CAPTAIN | 99722 | 87082 | 110804 | 297608 | 297608 |
| 13 EDWARD | | EXECUTIVE | 294580 | 0 | 0 | 294580 | 294S80 |

1. Print EmployeeName and Total Pay who has BasePay greater than 10000
2. Read data file 'data.csv' from command line and extract rows which have BasePay > 10000
3. Print only EmployeeName and TotalPay

Ans: cat data.csv | awk '{ if ($4 < 10000) print $4 }'

1. What is the aggregate TotalPay of employees whose jobtitle is 'CAPTAIN'
2. Read data file 'data.csv' from command line and extract rows which have 'CAPTAIN' in the column 'jobtitle'
3. Extract TotalPay and calculate sum. Print the result on terminal.

Ans: cat data.csv | grep CAPTAIN | awk '{ sum+=$7 } END { print sum }'

1. Print JobTitle and Overtimepay who has Overtime pay is between 7000 and 10000
2. Read data file 'data.csv' from command line and extract jobtitle and overtimepay for column value range between 7000-10000
3. Print the result on terminal.

Ans: cat data.csv | awk '{ if ($5 > 7000 && $5 < 10000) print $3 " " $5 }'

1. Print average BasePay
2. Read data file 'data.csv' from command line and extract BasePay values and calculate its average
3. Print the result on terminal.

Ans: cat data.csv | awk '{sum+=$4} END {print sum/NR}'

# Move files from one folder to the respective folders.

E.g current folder have files abc.txt, def.txt, ghi.txt, jkl.txt

You have to move these files to the folder like abc.txt => abc/, def.txt =>def/ ... Expected outcome -

abc/abc.txt def/def.txt ghi/ghi.txt jkl/jkl.txt

1. Create files in current directory or any temporary directory - abc.txt, def.txt, ghi.txt, jkl.txt
2. Print list of files to move.
3. Segregate basename and extension of a file.
4. Create folder using basename.
5. Move file to newly created folder.
6. Iterate above steps for all files.

Ans:

touch abc.txt def.txt ghi.txt jkl.txt

ls

mkdir abc def ghi jkl

mv abc.txt abc/

mv def.txt def/

mv ghi.txt ghi/

mv jkl.txt jkl/

# Append current date to all log files name which has extension .log.1 from a folder

E.g original file - access.log.1

New updated file name - access-20102019.log

1. Create files with name abc.log.1, def.log.1 , ghi.log.1, jkl.log.1, mno.log.1
2. Print list of files to rename.
3. Segregate basename and extension of a file
4. Print Date Command to show in ddmmyy
5. Append Date to the log file name
6. Iterate above steps for all files which has extension .log.1

Ans:

touch abc.log.1 def.log.1 ghi.log.1 jkl.log.1 mno.log.1

ls

abc.log.1 def.log.1 ghi.log.1 jkl.log.1 mno.log.1

mv abc.log.1 abc-14032022.log

mv def.log.1 def-14032022.log

mv ghi.log.1 ghi-14032022.log

mv jkl.log.1 jkl-14032022.log

mv mno.log.1 mno-14032022.log

ls

abc-14032022.log ghi-14032022.log mno-14032022.log

def-14032022.log jkl-14032022.log