**Documentation for AWK in Ubuntu**

**Brayan Rosario**

**CIS-245**

**LINUX ADMINISTRATION**

**Part 2**

**1 print the first and last names of those who contributed more than $110 in the second month.**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: singles quote to insert the rest of your code and complete your syntax.**

**$4>110: the dollar sign with the number four in the front and the single greater than next to the 110 means. Match the last thing on the string or Any name in the file that contains more than 110 show be display on the screen.**

**{print $1}: and this command with the brackets on and the dollar sign with the number 1 in the front will print the names just in the first spot of the file followed of the file name of your preference.**

**Awk -F: ‘$4>110 {print $1}’ lab3october.txt**

Text

Description automatically generated

**2 print the names and phone numbers of those who contributed less than $75 in the last month.**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: the quotes to insert the syntax of your commands.**

**$5<75: the dollar sign with the number 5 in the front will match the end of the string and the single less than with the 75 next to it and will match the 75 less than we are looking for those who contributed less than 75 in the second month.**

**{print $1, $2}: and the print option with dollar sign with the number 1 and the number 2 in the front separate with the comma will print the names and the phone number of those who contributed less than 75 in the second month followed in the file as where they are located.**

**Awk -F: ‘$5<75 {print $1, $2}’ lab3october.txt**

****

**3 print the names of those who contributed between $75 and $150 in the first month.**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: the quotes to insert the syntax of your commands.**

**‘$3<75, $3>150 : as we want to print the values between 75 and 150 in the first month, I use the dollar sign with the number 3 in the front to located the spot where the first spot is which is the first month and one value with the single greater than and another one with the single less than to just display the both amounts separated with a comma.**

**{print $1}: and for the end of the syntax we are going to use the print option and the dollar sign with the number 1 in the front to just print the first line of the file.**

**Text

Description automatically generated**

**4 print the name of those who contributed less than $700 over the three-month period.**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: the quotes to insert the syntax of your commands.**

**{sum=$3 + $4 + $5}: the sum command in Linux is used to add things together in a file. The dollar sign with the 3 different numbers followed with the plus sign will do the account over the three months in our file.**

**Sum>700: the sum and the single less than, will match and add things together in the file and will find the less than 700 according to the question.**

**{print $1}: and for the end of the syntax we are going to use the print option and the dollar sign with the number 1 in the front to just print the first line of the file that will take the name of those who contributed less than 700.**

****

**5 Print the names and addresses of those with an average monthly contribution greater than $100.**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: the quotes to insert the syntax of your commands.**

**{average=($3+$4+$5)/3}: as we want the average monthly contribution, average stand for the result you get by adding two or more amounts and as in our file we have some amounts we assign the dollar sing with the numbers in the front to indicate the spots in the file where we want amount to be printed.**

**Average>100: and as we want the amounts greater than 100 we use the average followed of the singles greater than and also add the 100 to specify we only want the amount greater than 100.**

**{print average}: as we want just the average to be display on the screen we select print and then average insert of those brackets to displays the amounts that we are looking for followed of your file name.**

**Text

Description automatically generated**

**6 print the first name of those not in the 916 are code.**

**Awk: Programming language to manipulating data.**

**-F”[ :]”: this option is followed by a regular expression enclosed in brackets. If a space, colon, or tab is encountered and AWK will use that character as a field separator. The expression is using double quotes so that the shell will not pounce on the metacharacters for its own.**

**‘’: the quotes to insert the syntax of your commands.**

**$3 !~/916/: as we want those names not in 916 area code we select the dollar sign with the number 3 in the front to match the end of those in the line to be display on the screen, followed !~/916/ this will match the ones who no are in the 916 inserted with slashes to include the number we want to looking for, for the names who are not in the area code.**

**{print $1}: and for the end of the syntax we are going to use the print option and the dollar sign with the number 1 in the front to just print the first line of the file.**

**Awk -F”[ :]” ‘$3 !~/916/{print $1}’ lab3october.txt**

**Text

Description automatically generated**

**7 Print each record preceded by the number of the records.**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: the quotes to insert the syntax of your commands.**

**{print NR, $0}: as we want to print the record NR is use to get the records of our actual file and as we only want the record I select the dollar sign and the number 0 to just display the record number next to the names.**

**Text

Description automatically generated**

**8 Print the name and the total contribution of each person.**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: the quotes to insert the syntax of your commands.**

**{sum=$3+$4+$5}: as we want the total contribution of each person of our record we use sum which stand for adding thing together and the dollar sign with the numbers in the front to indicate where is the amount in our file and what we want to be display in our screen.**

**{print $1, sum}: and for the end of the syntax we are going to use the print option and the dollar sign with the number 1 in the front to just print the first line of the file where the names are and the sum to print the amount of each person.**

**Text

Description automatically generated**

**9 Add $10 to Chet’s second contribution**

**Awk: Programming language to manipulating data.**

**-F: is a way to change the delimiter.**

**‘’: the quotes to insert the syntax of your commands.**

**$1 ~/Chet/: we use the dollar sign and the number 1 in the front to indicate the chet name in our file and the tilde to match any pattern in the file and the name chet inside on the slashes.**

**{c=$4+10}: as we want to add 10 to chet we use the (c) which stand for count and the dollar sign with the number 4 to indicate the amount of the second month and the plus + sign and the 10 to add 10 to the amount of the second amount.**

**{print c}: as we want jus to see the 10 added to the chet amount in the second month we just take where the c with the dollar sign and the number 10 is with the plus sign and print it using the print option.**

**Text

Description automatically generated**

**10 change Nancy McNeil’s name to Doris Shutt.**

**I couldn’t get this one done.**