Ian Camire

00325798

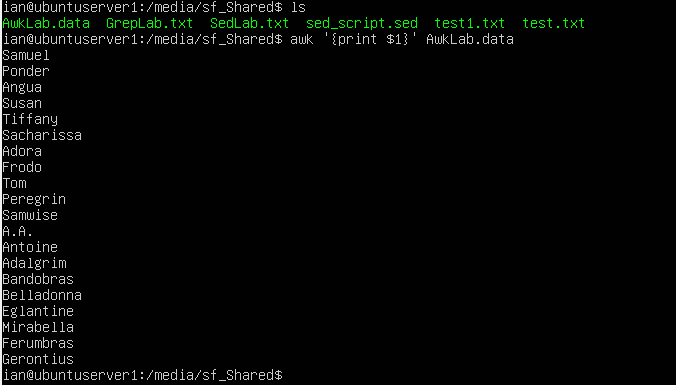
CIS 245 B1A

AWK

1. Print All first names

**Awk ‘{print $1}’ AwkLab.data**

Print the first field of each file line ($1 indicates the first field)



1. Print the phone numbers of Frodo and Tom after their names

**Awk -F: ‘/^Tom/ || /^Frodo/ {print $1, $2)’ AwkLab.data**

Search through AwkLab.data for lines beginning with Tom (^indicates lines starting with) or (|| indicates the OR operator) Frodo and print the first and second fields (-F: sets the field separator to the : character).



1. Print Peregrins full name and phone number, area code only

**Awk ‘/Peregrin/ {print $1,$2}’ AwkLab.data**

Search through AwkLab.data for the pattern (// sets the pattern) Peregrin and print the first 2 fields {$1, $2}

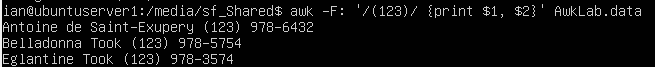


1. Print all phone numbers (full number) in the 123 area code along with their names.

**Awk -F: ‘/(123)/ {print $1, $2}’ AwkLab.data**

Search through AwkLab.data for any string of (123) (within the // is the pattern to search), separate the lines by : (-F: sets the field separator to :) and print the first 2 fields

YouTube. (2020, April 4). *01-Introduction to AWK*. YouTube. https://www.youtube.com/watch?v=I-uWvNvtJcY&list=PLY-V\_O-O7h4fzqbPT0kpQMl8XlPvSse9H



1. Print all last names beginning with either a T or D

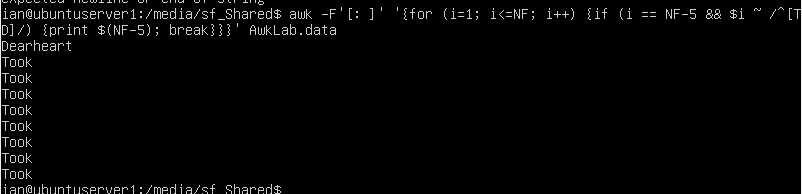
**Awk -F’[: ]’ ‘{for (i-1; i<=NF; i++) {if (i == NF-5 && $i ~ /^[TD]/) {print $(NF-5); break}}} AwkLab.data**

-F’[: ]’ sets the field separator to ‘ ‘ and ‘:’

for (i-1; i<=NF; i++) begins the for loop

{if (i == NF-5 && $i ~ /^[TD]/) if the last field (NF) -5 (^) begins with a T or D save it in i

{print $(NF-5); break}}} print the lines that fit the parameters and break (end the loop)



1. Print All first names containing 4 or less characters

**Awk ‘(length($1) < 5) {print $1}’ AwkLab.data**

Search through AwkLab.Data and print the first field if the length of that field is less than 5 characters (length is a built in expression, if it contains less than 5 characters then print $1)

yaelyael                      1, JohanJohan                      4, GuruGuru                      5, & BernhardBernhard                      12.3k44 gold badges6060 silver badges7070 bronze badges. (1959, January 1). *AWK + print line only if the first field start with string as linux1*. Unix & Linux Stack Exchange. https://unix.stackexchange.com/questions/72734/awk-print-line-only-if-the-first-field-start-with-string-as-linux1



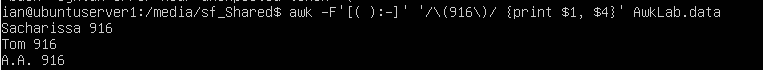
1. Print the first names and area codes of all those in the 916 area code

**Awk -F’[( ):-]’ ‘/\(916\)/ {print $1, $4}’ AwkLab.data**

-F sets the FS to ( ) : -

/\(916\)/ specifies the pattern to look for, 916 in parenthesis

{print $1,$4} this specifies to print the $1 and $4 if the pattern is found



1. Print Sacharissa’s campaign contributions following her name. Each value needs a leading dollar sign.

**Awk -F’[( ) :-]’ ‘/Sacharissa/ {print $1, “$” $8, “$” $9, “$” $10}’ AwkLab.data**

-F sets the field separator to ‘ ‘ or : then I used the /Saracharissa/ to search for that name. Finally I had the output printed as her name in the first field ($1) then her contributions preceded by a “$” so the output would be formatted.



1. Print Last names followed by a comma and the phone number

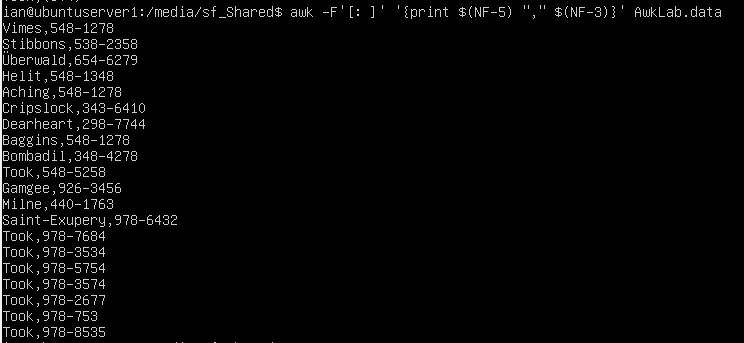
**Awk -F’[: ]’ ‘{print $(NF-5) “,” $(NF-3)}’ AwkLab.data**

-F’[: ]’ sets the field separator to ‘ ‘ and ‘:’

‘{print $(NF-5) NF is the last field on the line, -5 back is the last name field

“,” prints a comma

$(NF-3)}’ NF is the last field on the line -3 back is the phone numbers

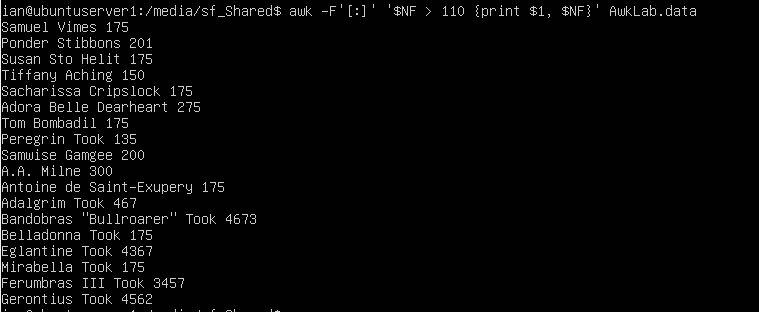


1. Print the first and last names of those who contributed over $110 in the last month.

**Awk -F’[:]’ ‘$NF > 110 {print $1, $NF}’ AwkLab.data**

-F sets the field separator to :

$NF is the last field on the line, any NF that is greater than 110 print the first and last field.



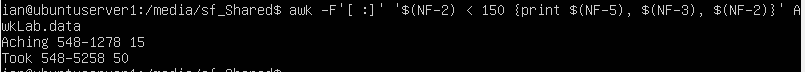
1. Print the last names, phone numbers, and first month contribution of those who contributed less than 150$ in their first month

**Awk -F’[ :]’ ‘$(NF-2) < 150 {print $(NF-5), $(NF-3), $(NF-2)}’ AwkLab.data**

-F sets the field separator to [ :] (blank space as well as :)

‘$(NF-2) < 150 search the file for the last field in a line -2 and if its less then 150 then

{print $(NF-5), $(NF-3), $(NF-2)} print the last field -5 (last name) print the last field -3 (phone number) and print the last field -2 (first contribution)



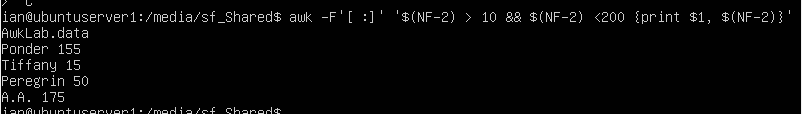
1. Print the first names and their contributed between 10 -200 in the first month

**Awk -F’[ :]’ ‘$(NF-2) > 10 && $(NF-2) < 200 {print $1, $(NF-2}’ AwkLab.data**

Awk -F’[ :]’ sets the field separator to [ :] (blank space or :)

‘$(NF-2) > 10 && $(NF-2) < 200 takes the last field -2 (first contribution) that is above 10 and less than 200

{print $1, $(NF-2}’ print the first field (first name) and the last field -2 (first contribution) that fit the pattern



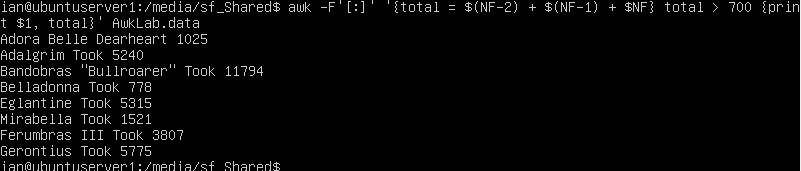
1. Print the first name, last name, and total contributions of those who contributed more than $700 in the last 3 months

**Awk -F’[:]’ ‘{total = $(NF-2)+$(NF-1)+$NF} total >700{print $1, total}**

Awk -F’[:]’ set the field separator to :

‘{total = $(NF-2)+$(NF-1)+$NF} creates the variable total and sets it to the last 3 fields added together (the contributions)

total >700{print $1, total} if total is less than 700 do nothing, if its above, print the name and their total contributions.



1. Print the first name and first letter of the last name, and the average contribution of those where the average contribution is more than 300.

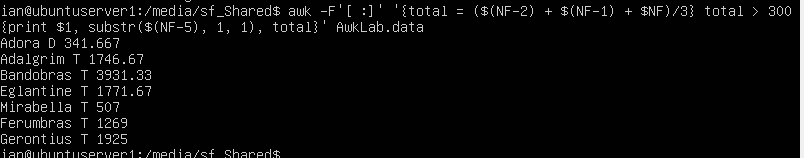
**Awk -F’[ :]’ ‘{total=($(NF-2)+$(NF-1)+$NF)/3} total >300 {print $1, substr($(NF-5), 1, 1), total} AwkLab.data**

Awk -F’[ :]’ Set the field separator to [ :] (blank space and :)

‘{total=($(NF-2)+$(NF-1)+$NF)/3} add the last 3 fields together (NF is last field) and divide by 3 and put that into the total variable

} total >300 if the total average is over 300

{print $1, substr($(NF-5), 1, 1), total} print the $1 field (first name), substr($(NF-5), 1, 1) prints the first letter of the last name field, then prints total.



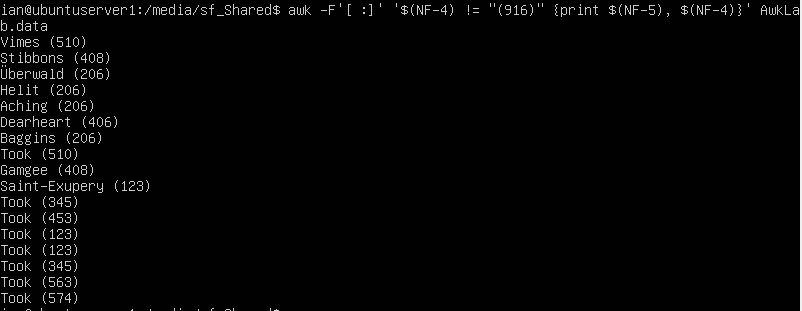
1. Print the last name and the area code of those NOT in the 916 area code

**Awk -F’[ :]’ ‘$(NF-4) != “(916)” {print $(NF-5), $(NF-4)}’ AwkLab.data**

Awk -F’[ :]’ Set the field separator to [ :] (blank space and :)

‘$(NF-4) != “(916)” look in the last field – 4 (zipp code) for anything that does not equil (!=) (916)

{print $(NF-5), $(NF-4)} print the last field – 5 (Last Name), and the last field-4 that does not match the (916)

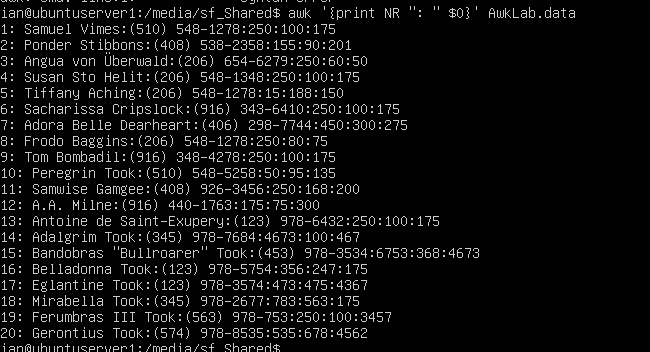


1. Print each record preceded by the number of the record.

**Awk ‘{print NR “: “ $0}’ AwkLab.data**

‘{print NR “: “ $0}’ Print the NR (record number) followed by a “:” the the $0 (entire record)

Stackoverflow: What are NR and FNR comment by Tom Fenech



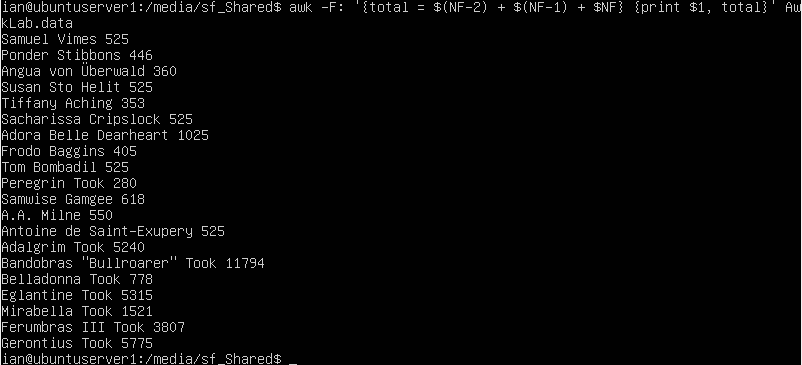
1. Print the names and total contributions of each person

**Awk -F: ‘{total = $(NF-2) + $(NF-1) + $NF} {print $1, total}’ AwkLab.data**

Awk -F: Set the field separator to [ :] (blank space and :)

‘{total = $(NF-2) + $(NF-1) + $NF} add the last three fields together and store it in total

{print $1, total}’ Print the first field (Name) and the contents of the total variable



1. Add $10 to Tiffany Aching’s first contribution and print her full name and full contribution.

**Awk -F’[ :]’ ‘$1 == “Tiffany” {$(NF-2) += 10; print $1, $(NF-2)}’ AwkLab.data**

Awk -F’[ :]’ Set the field separator to [ :] (blank space and :)

‘$1 == “Tiffany” {$(NF-2) += 10; In the first field find the pattern “Tiffany” then add 10 to the total of $(NF-2) (first contribution)

print $1, $(NF-2)}’ Print the first field (name) followed by the new $(NF-2)



1. Change Samwise Gamgee’s name to Sean Astin

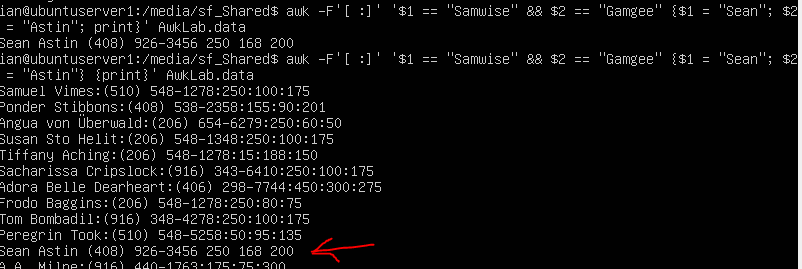
**Awk -F’[ :]’ ‘$1 == “Samwise” && $2 == “Astin” {$1 = “Sean”; $2 = “Astin”} {print}’ AwkLab.data**

Awk -F’[ :]’ Set the field separator to [ :] (blank space and :)

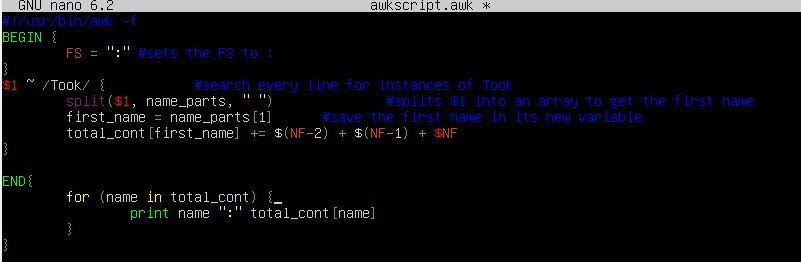
‘$1 == “Samwise” && $2 == “Astin” search the dataset for a pattern match in both fields (&& is and)

{$1 = “Sean”; $2 = “Astin”} replace $1 with Sean and $2 with Astin

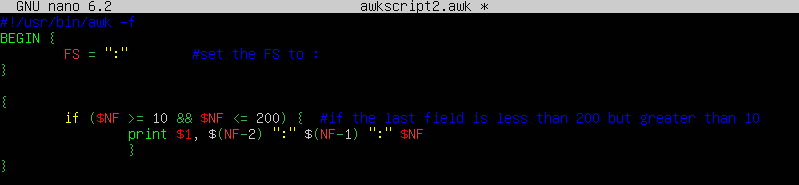
{print} print the entire dataset with the modified line



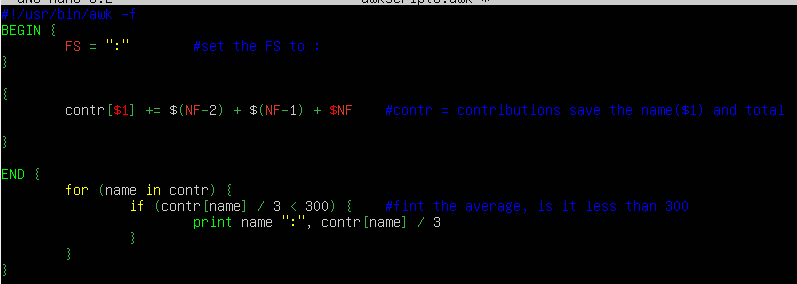
1. Write an awk script to do the following:
2. Prints the first name of all the Tooks followed by their total campaign contributions.



1. Print the full names and contributions of anyone who contributed between $10 and $200 in their last contribution



1. Prints the full names and the average contribution of those who contributed less than $300 on average



Call all scripts from the awk script awkscript1.awk AwkLab.data

