Colby O’Donnell

Lab 4 – Awk

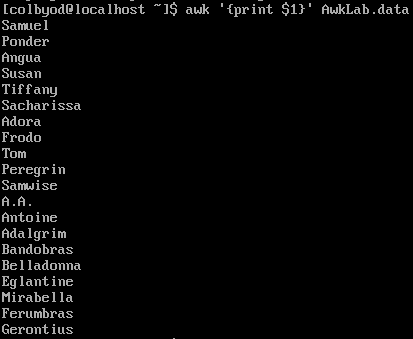
Linux Admin

November 2, 2023

This is Awk-ward

1. Print all the First Names.

Using awk we print the first field (**$1**) that ends at the first delimiter of “:”



2. Print phone numbers for Tom and Frodo after their names

Using the **-F** option, we set the delimiter to “:” then search for any line beginning with either “Tom” or “Frodo” and print the first field **$1** as well as the second field **$2** which are the names and phone numbers respectively in the file **AwkLab.data**



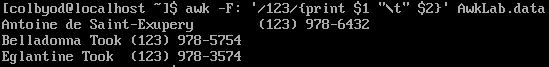
3. Print Peregrin’s full name and phone number area code only.

Again using **-F** we set the delimiter to “:” then search for any line that begins with “Peregrin” and print the first field **$1**, then insert a tab (**“\t”**) into the printed string then print the second field **$2.** I couldn’t figure out how to only print the area code without printing the entire field.



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

Again we set the delimiter to “:” using **-F** then search for any line containing the string “123”. We then print the first and second fields with a tab in between them containing the string “123”



5. Print all Last names beginning with either a T or D (careful of middle names!)

Can’t figure this one out.

6. Print all first names containing four or less characters.

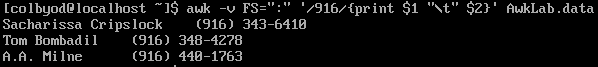
Here we’re using an **if statement** and search if the length first field **$1** is less than or equal to (**<=**)

four (**4**). **If** this statement is true, we print the first field. In this case, the delimiter is the default of a **space**.



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

Here we use the **FS=** option to set the delimiter to “:” which operates the same as the **-F** option we’ve used previously. We then search for any lines containing the string “916” and print the first **$1** and second **$2** fields separated by an inserted tab **“\t”** to separate the fields appropriately.



8. Print Sacharissa’s campaign contributions following her name. Each value should be printed with a leading dollar sign; e.g., $250 $100 $175. 1

Here we changed the location of the **FS=** option to the end of the command but still change the delimiter to “:” just the same. First we search for the string “Sacharissa” and print the fields containing the data in regards to her contributions. Before fields **$3 $4** and **$5** we insert a tab **“\t”** as well as a “**$**” to indicate the data as currency.



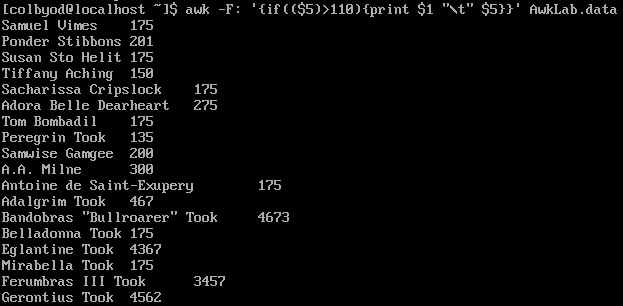
9. Print last names followed by a comma and the phone number. Be careful of the last names’s format.

Can’t figure this out.

10. Print the first and last names of those who contributed more than $110 in the last month. Make sure to include their last month contribution amount after the name

**awk -F: ‘{if(($5)>110{print $1 “\t” $5}}’ AwkLab.data**

First I changed the delimiter to “**:**” and then employ an **if** statement to factor whether or not the 5th field **$5** is greater than 110. If that is true, it will print the first field **$1** which is the first and last names, then the fifth field **$**5, which is the last month, from the file **AwkLab.data**.



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

**awk -F: ‘{if($3<150){print $1 “\t” $2 “\t” $3}}’ AwkLab.data**

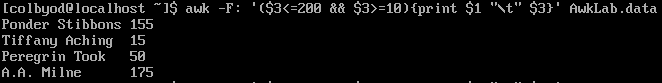
First we change the delimiter to “**:**” and then employ an if statement to test if the third field **$3** is less than 150. If this is true, it will then print the first field **$1**, the name, the second field **$2**, the phone number, and then the third field **$3**, which is the first month’s contribution. All printed fields are separated by a tab space to increase readability.



12. Print the first names and contribution of those who contributed between $10 and $200 in the first month.

**awk -F: ‘($3<=200 && $3>=10){print $1 “\t” $3}’ AwkLab.data**

After changing the delimiter to **:** we test to see if the third field **$3**, the first month, is less than 200 **AND (&&)** greater than 10. It is to then print out the first field **$1**, the name, and the third field **$3**, the first month with a tab space **“\t”** in between from the file **AwkLab.data**.



13. Print the first name, last names and total contributions of those who contributed less than $700 over the three-month period.

**awk -F: ‘($3+$4+$5<700){print $1 “\t” $3+$4+$5}’ AwkLab.data**

First we change the delimiter to **:** and then test to see if the sum of all contributions, fields **$3 $4 $5**, is less than 700. If this is true, it is to print the first field **$1** and then the sum of field 3, 4, and 5 **$3+$4+$5** with a tab space in between.



14. Print the first names and first letter of the last name, and average contribution of those who had an average contribution of more then $300

**awk -F: ‘($3+$4+$5<300){print $1}’ AwkLab.data**

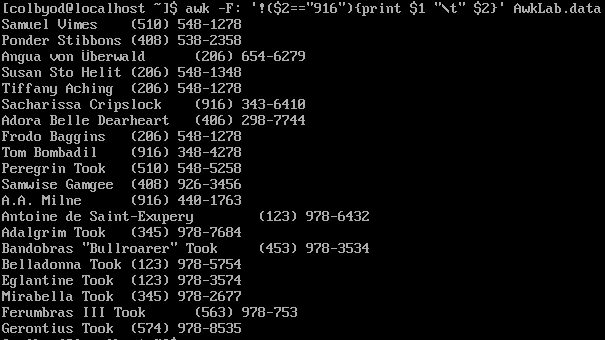
**-F:** changes the delimiter to **:** then we add the three contribution fields **$3+$4+$5** together. Whatever line matches to less than 300, it will print the first field **$1** from the file **AwkLab.data**



15. Print the last name and area code of those not in the 916 area code.

**awk -F: ‘!($2==”916”){print $1 “\t” $2}’ AwkLab.data**

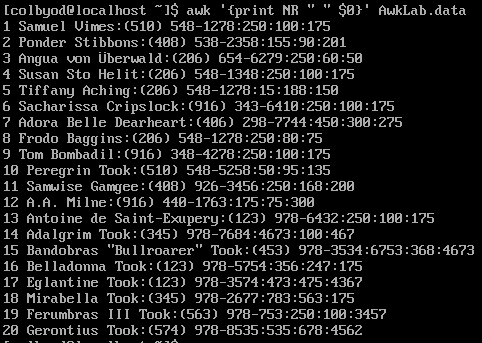
After changing the delimiter to **:** we test to see if the second field **$2** does not equal **916**. I believe instead of **!($2==”916”)** I could also write it as **($2!=”916”)** but maybe not. If this test is true, print the name **$1** and phone number **$2** from the file **AwkLab.data**



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

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

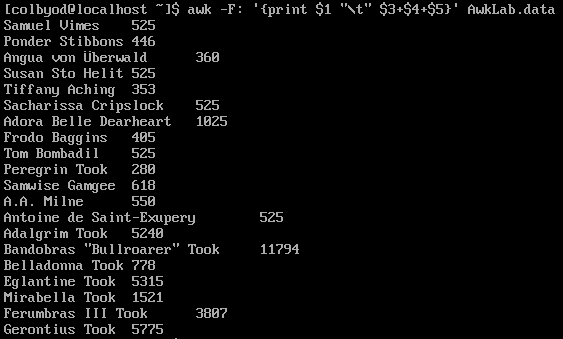
**NR** prints a number chronologically before each record indicated by **$0** which puts it before the first found field of each record.



17. Print the name and total contribution of each person.

**Awk -F: ‘{print $1 “\t” $3+$4+$5}’ AwkLab.data**

**-F:** changes the delimiter to **:** then print the first field **$1** as well as the sum of field 3, 4, and 5 **$3+$4+$5** which are the fields holding the contribution data from the file **AwkLab.data**



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

**Awk -F: ‘/^\Tiffany Aching\>/{$3=$3+10;print $1 “\t” $3}’ AwkLab.data**

We change the delimiter to **:** then anchor the search to the beginning of the line using **^** and search for the string matching **Tiffany Aching**. Using **>/** we search and initiate a change to the third field **$3** as well as adding **10** to it **$3=$3+10**. We then print the name and replaced total of the first month **$1 “\t” $3** with a tab space in between.



19. Change Samwise Gamgee’s name to Sean Astin

**awk -F: ‘$1==”Samwise Gamgee”{$1=”Sean Astin”;print $1}’ AwkLab.data**

Again, changing the delimiter to **:** then searching if the first field **$1** matches a string of **Samwise Gamgee**. If that is true, it will replace the matching string with **Sean Astin** in the first field. It is to then print the first field **$1** from the file **AwkLab.data**



20. Write an awk script to do the following (MUST be an awk script not just a bash script or commands on the commandline)

(a) Prints first name of the all the Tooks followed by their total campaign contributions .

(b) Print the full names and contributions of anyone who contributed between $10 and $200 in the last contribution

(c) Prints the full names and average contribution of those who contributed less than $300 on average

Surprise! I don’t know how to do this one either...