**Name: Jesus Liriano**

**ID: 00316389**

**Course: CIS-245**

**Prof: Adrianna Holden-Gouveia**

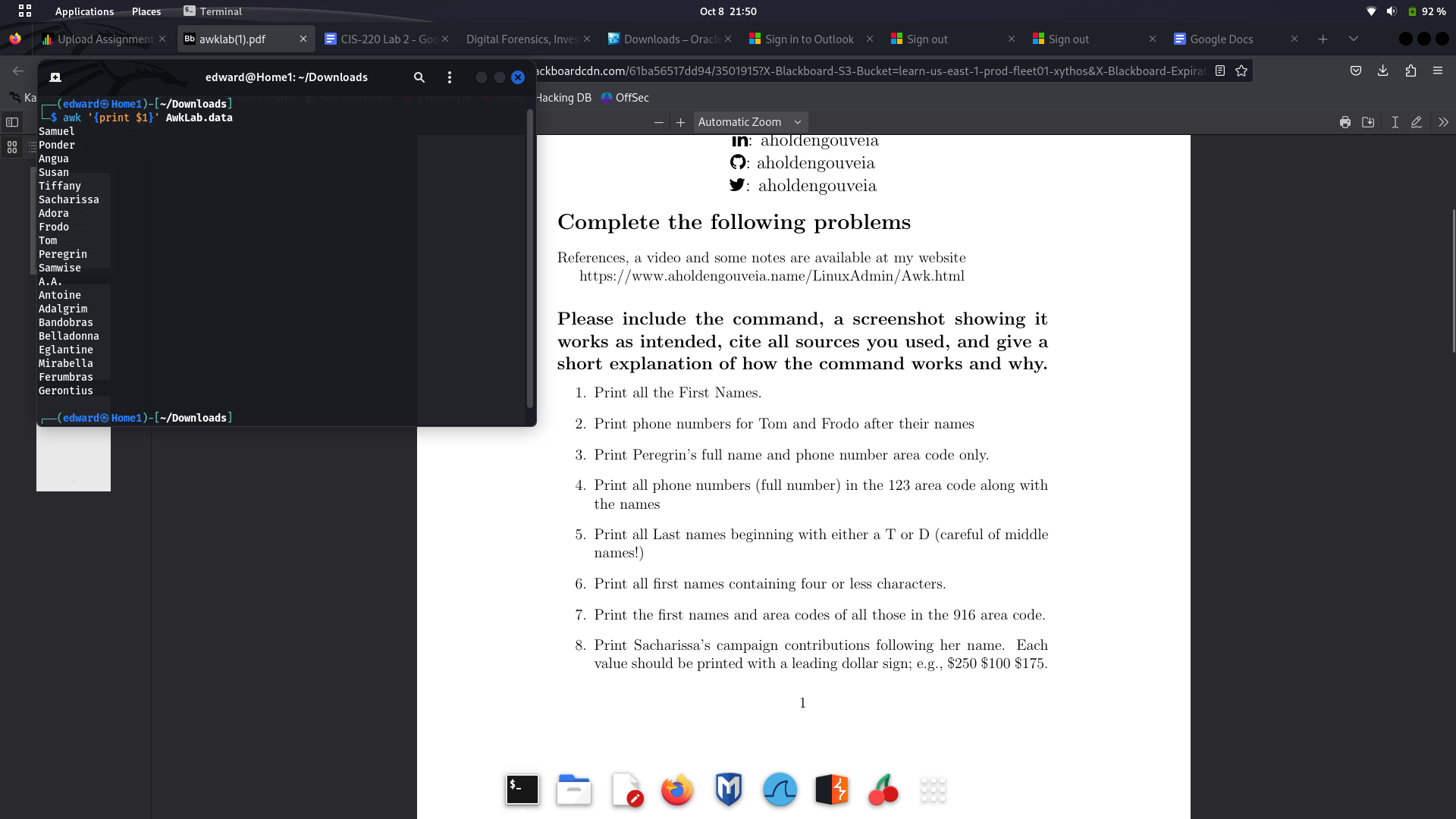
**Date: 10/10/2023**

Please include the command, a screenshot showing it works as intended, cite all sources you used, and give a short explanation of how the command works and why.

1. Print all the First Names.

**Command: awk ‘{print $1}’ AwkLab.data**

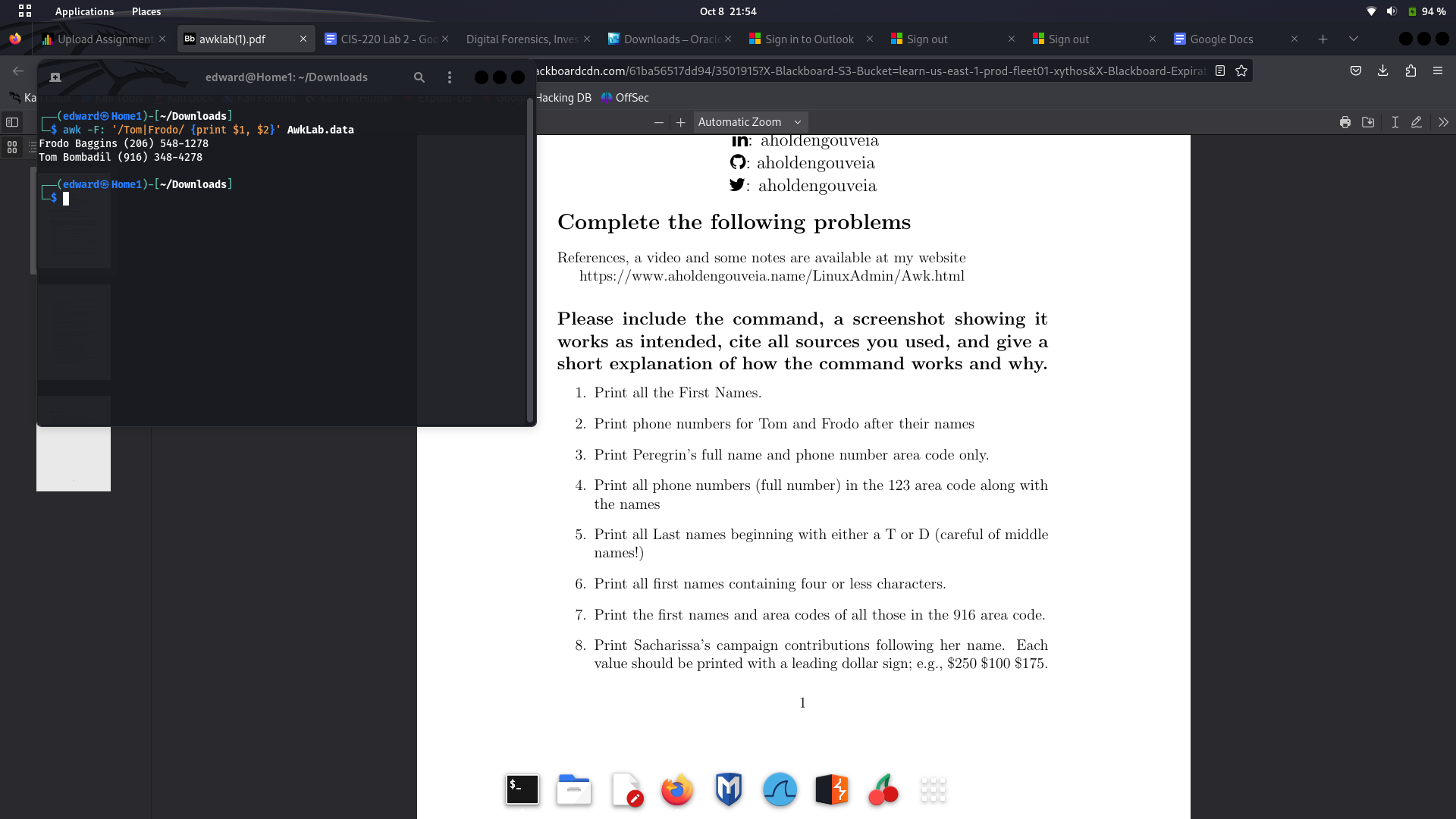
With this command the action we tell the system to use **awk** and “**print $1**” to print the first column in the file



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

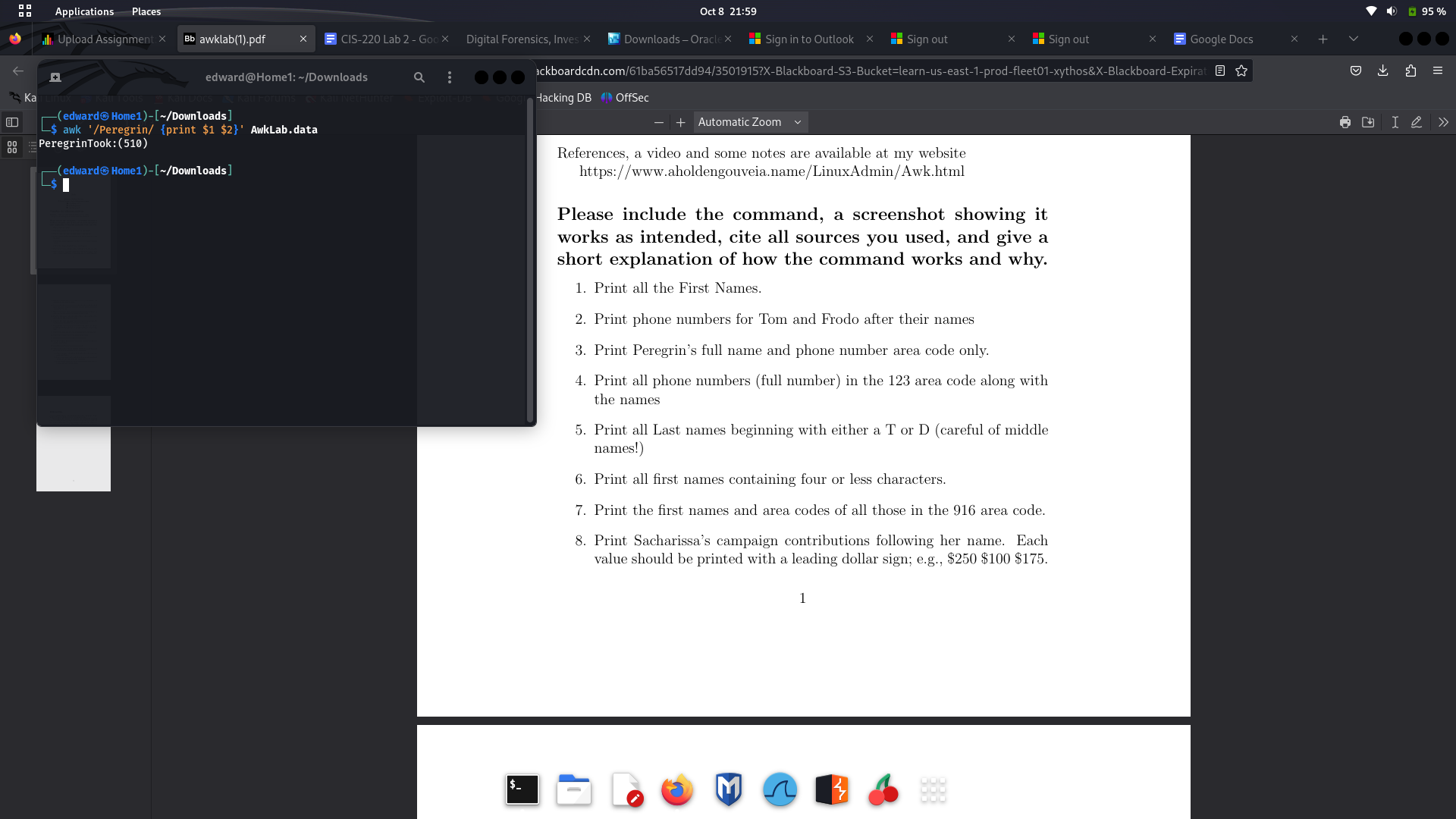
**Command: awk -F: ‘/Tom|Frodo/ {print $1, $2}’ AwkLab.data**

Here we specify with the “**-F:**” option to look for delimiters in that separate our names in the file. We then use the “**/Tom|Frodo/**” to tell awk to look for names specifically and then proceed to “**print $1, $2**” number 2 being our second column which in this case is the phone number



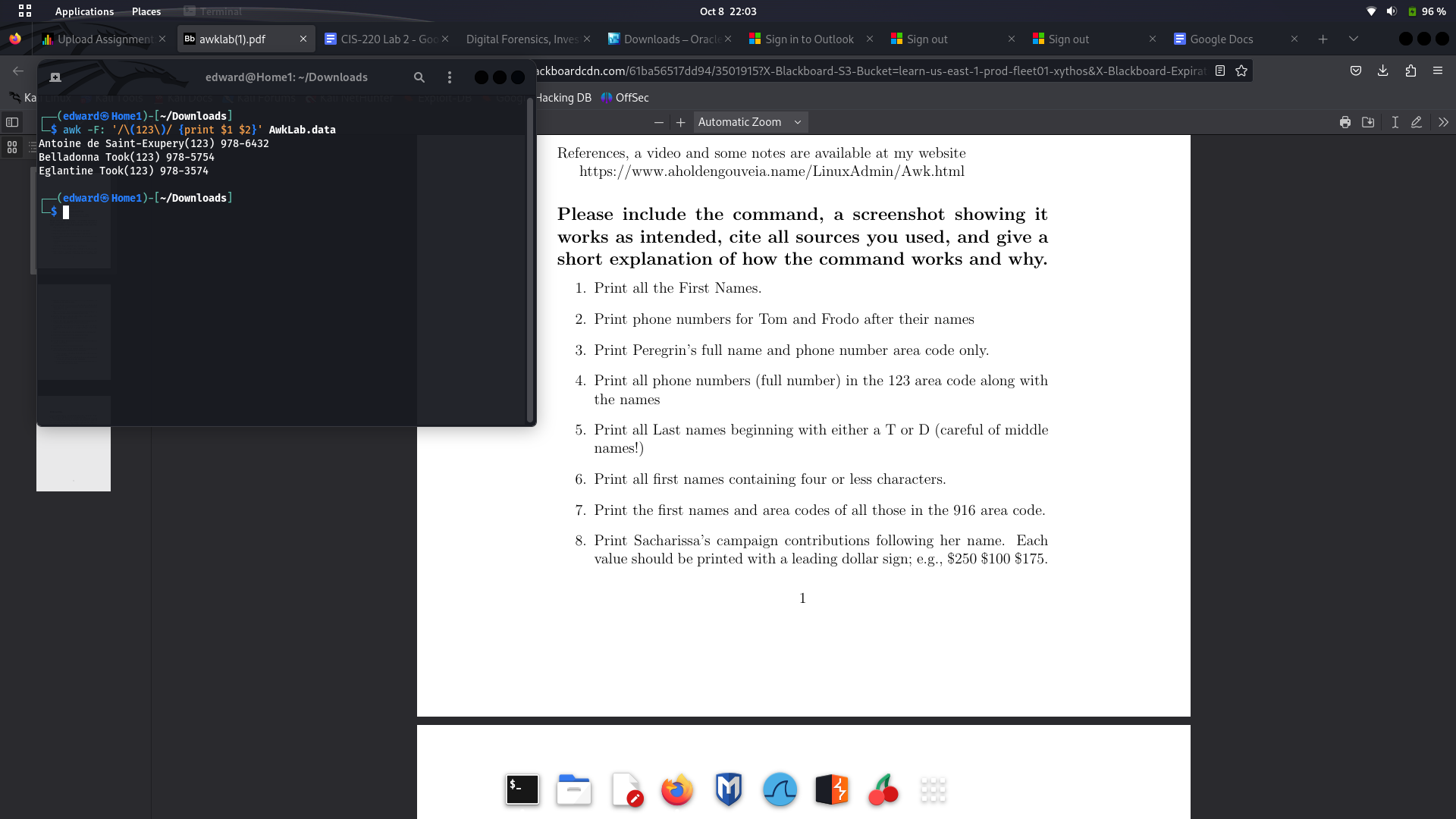
3. Print Peregrin’s full name and phone number area code only.

**Command: awk ‘/Peregrin/ {print $1 $2}’ AwkLab.data**Here the command is very similar to the previous one but in this case since we didn’t use the **-F:** to rule out delimiters our second column returned just the area code and not the full name



4. Print all phone numbers (full number) in the 123 area code along with the names  
**Command: awk -F: '/\(123\)/ {print $1 $2}' AwkLab.data**

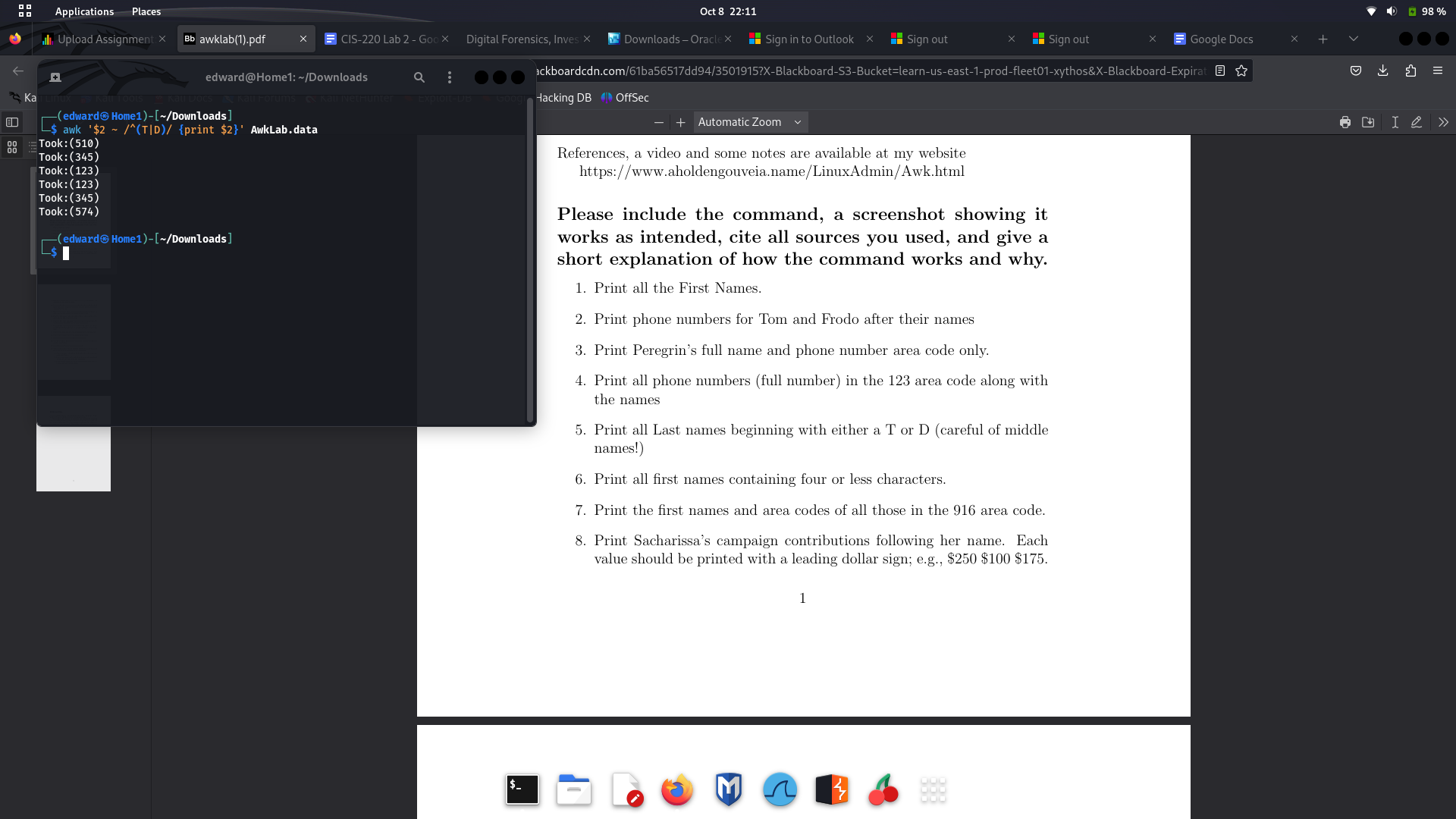
Here we use the **-F:** to print out the full second column and we give this format to awk **/\(123\)/** to specify that we want to include fields with the 123.



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

**Command: awk '$2 ~ /^(T|D)/ {print $2}' AwkLab.data**

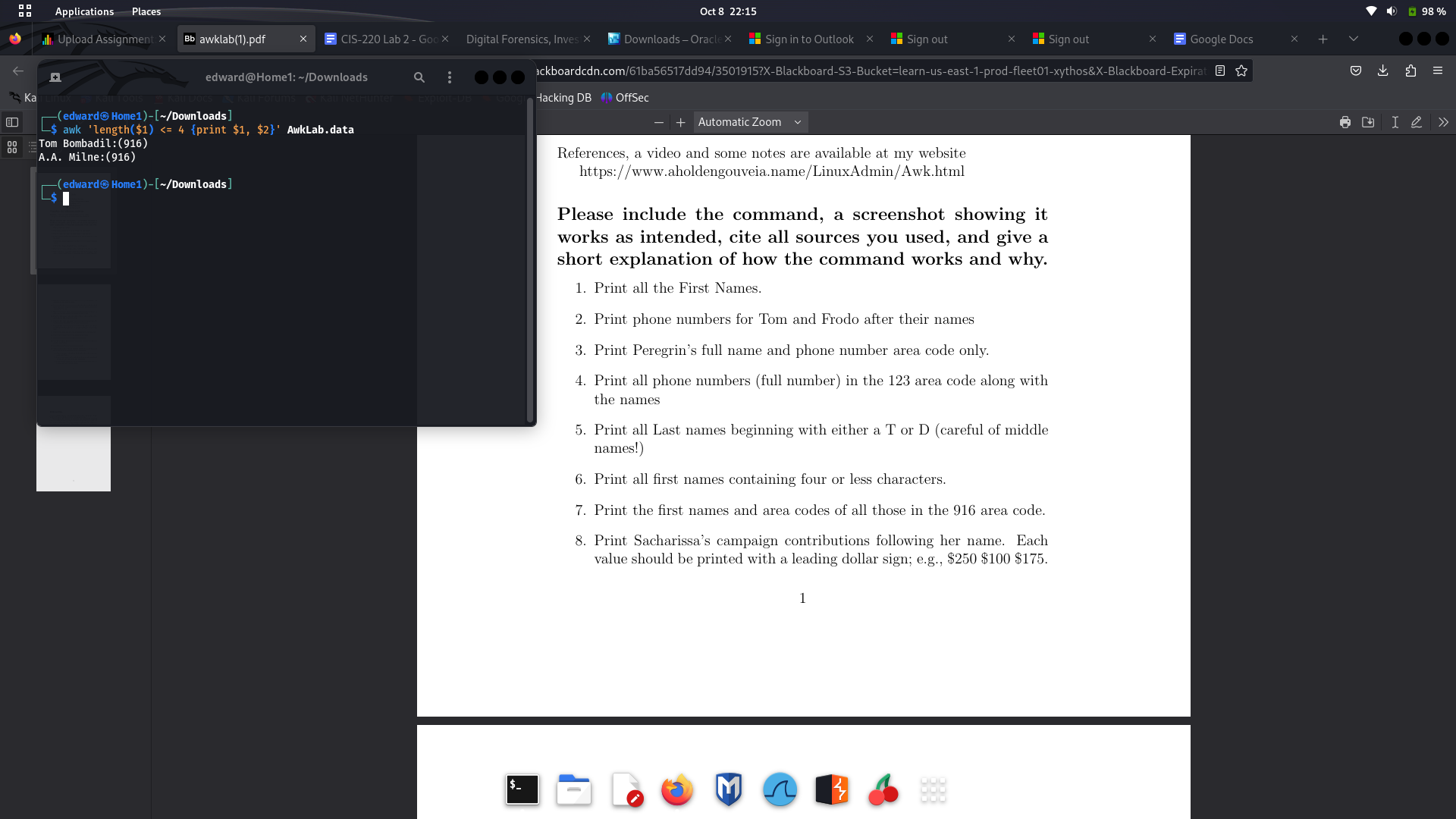
By combining the **“~”** (match) and **“^”** (to take the beginning of the field) we are telling **awk** to take the first letter of the second column and match it so we can then print it out as just the last names



6. Print all first names containing four or less characters

**Command: awk 'length($1) <= 4 {print $1, $2}' AwkLab.data**

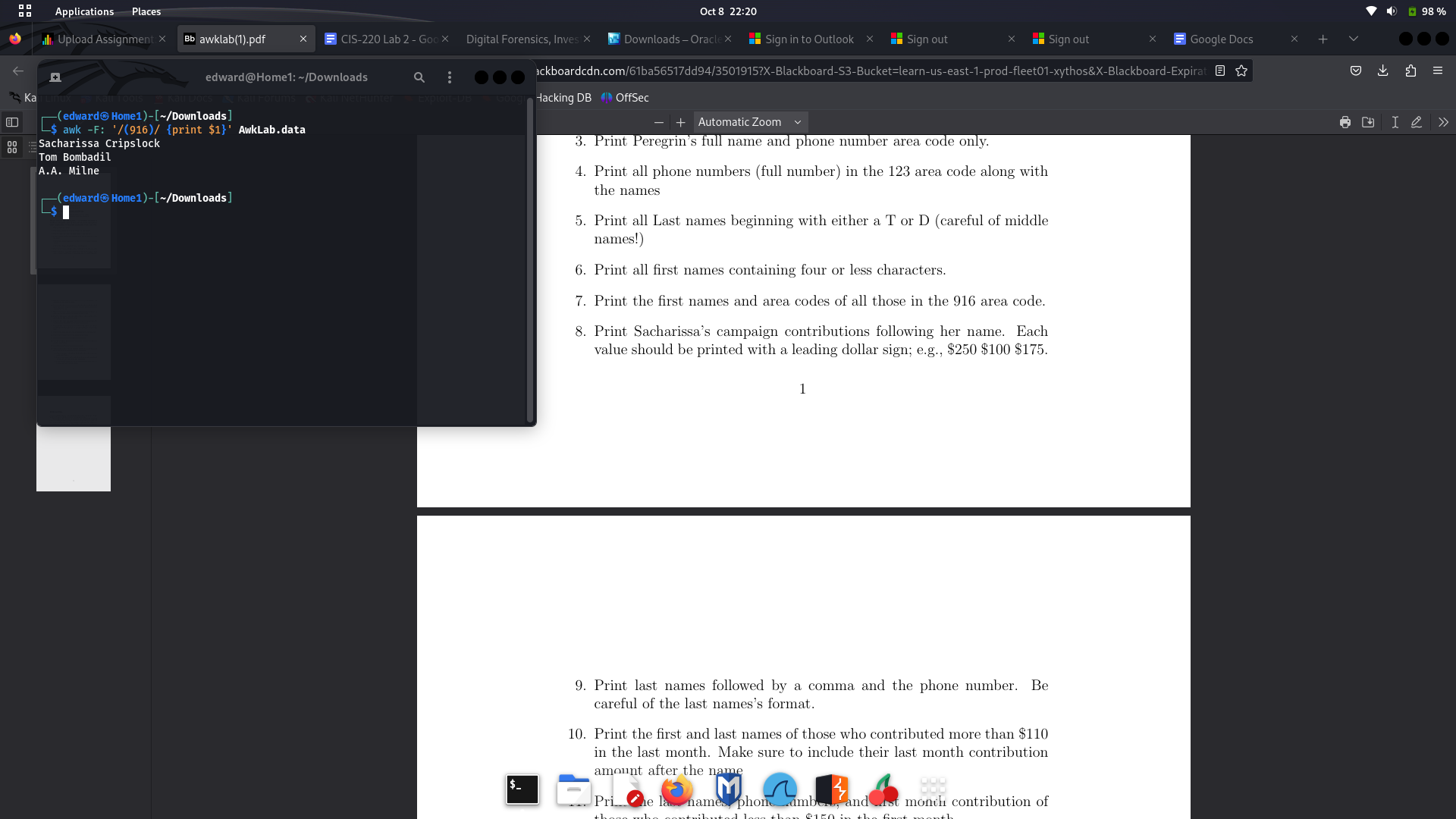
**“length($1) <= 4”** This condition checks if the length of the first name is 4 letters or less. If the condition is met then awk will print out the full names



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

**Command: awk -F: '/(916)/ {print $1}' AwkLab.data**

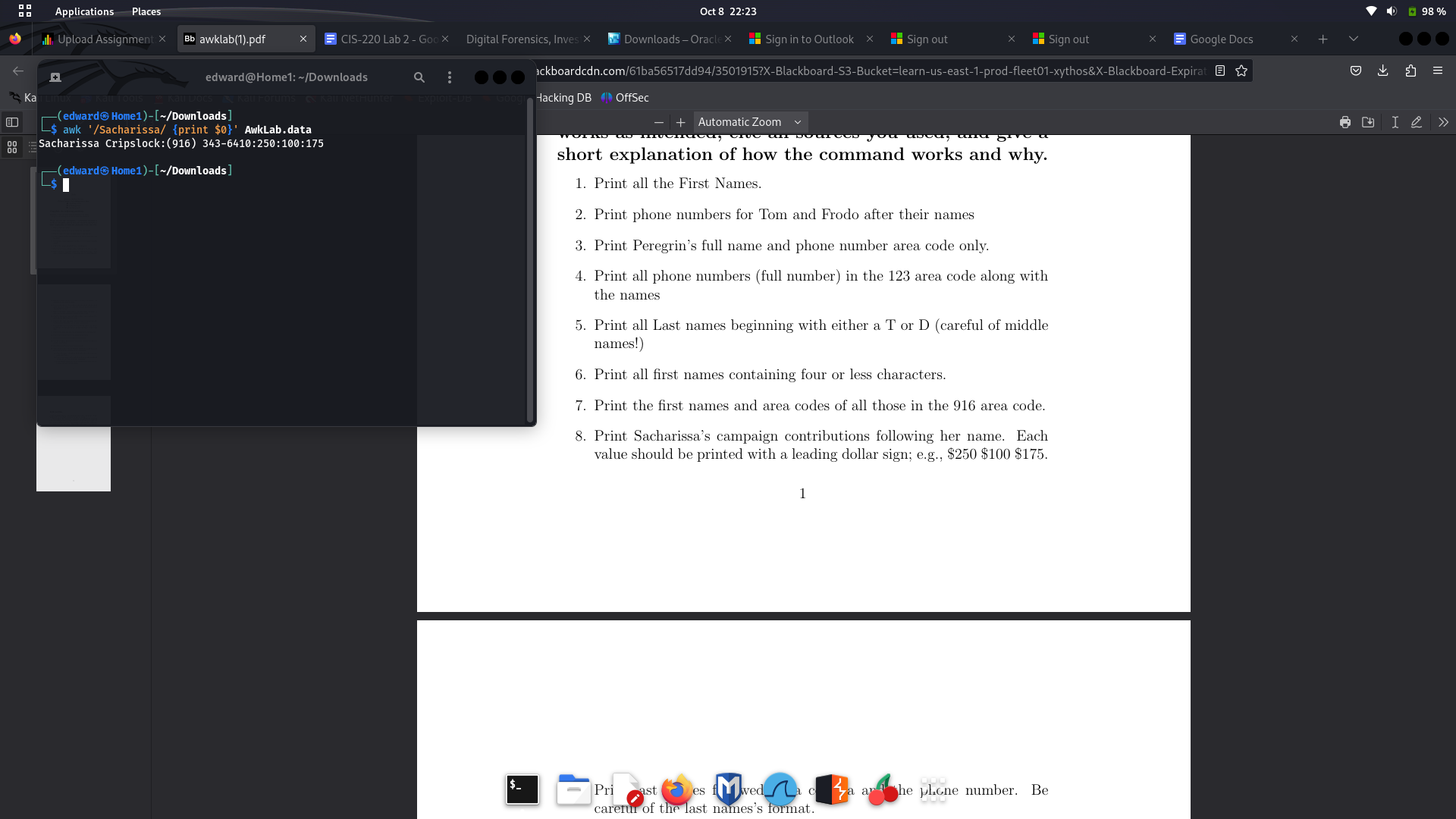
Just like in problem 4 here we tell **awk** to take the names of those in the area code 916 by giving it this format **/(916)/,** if the condition is met then the name gets printed out.



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

**Command: awk '/Sacharissa/ {print $0}' AwkLab.data**

Herewe tell **awk** that we are looking for lines that contain **/Sacharissa/,** if the line contains this then **“print $0”** will print the entire line



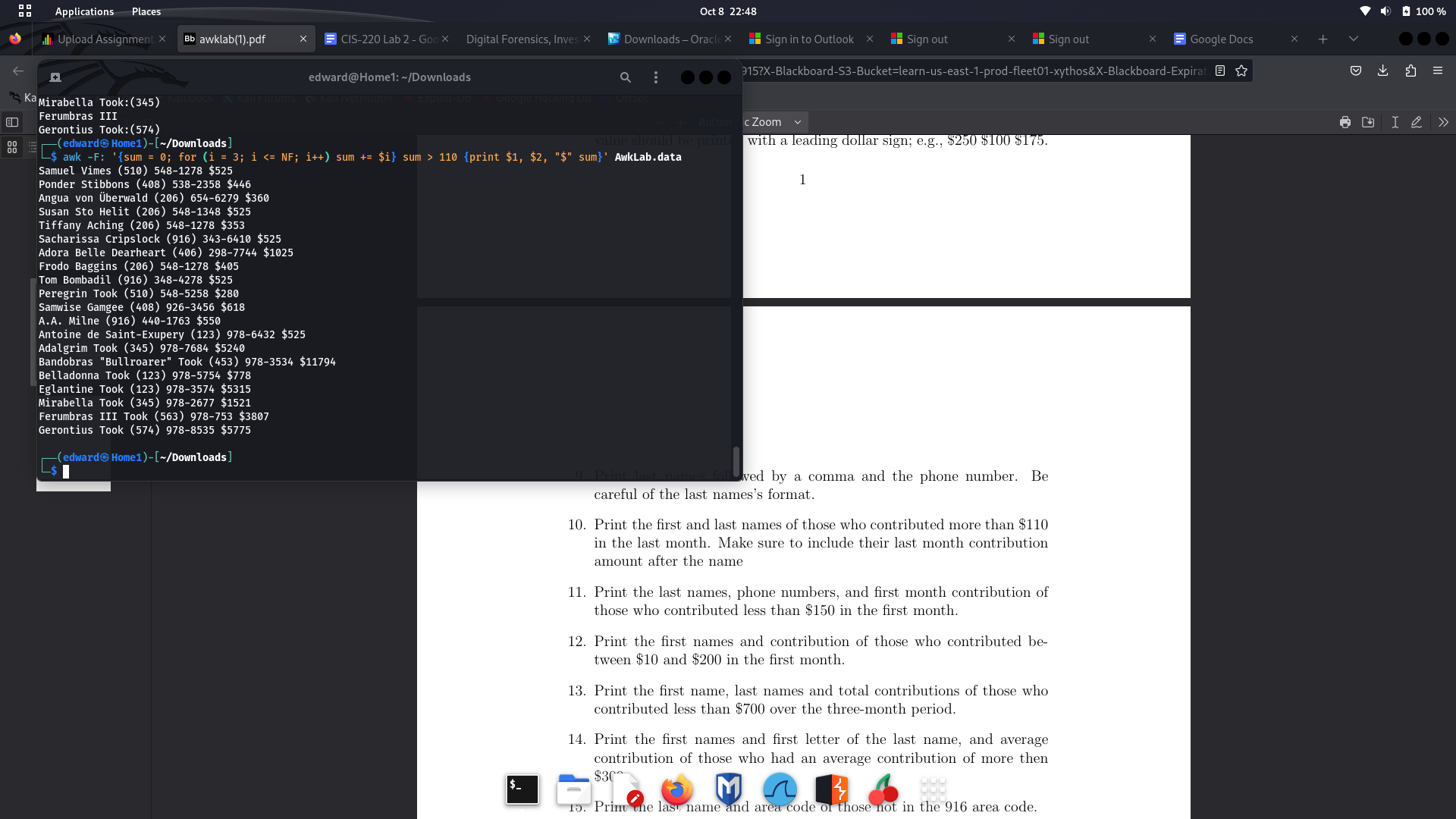
1. Print last names followed by a comma and the phone number. Be careful of the last names’s format.

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

**Command: awk -F: '{sum = 0; for (i = 3; i <= NF; i++) sum += $i} sum > 110 {print $1, $2, "$" sum}' AwkLab.data**

So the first part of this command **'{sum = 0; for (i = 3; i <= NF;** calculates from the third field to the end how much a person contributed and stores it the value **sum**

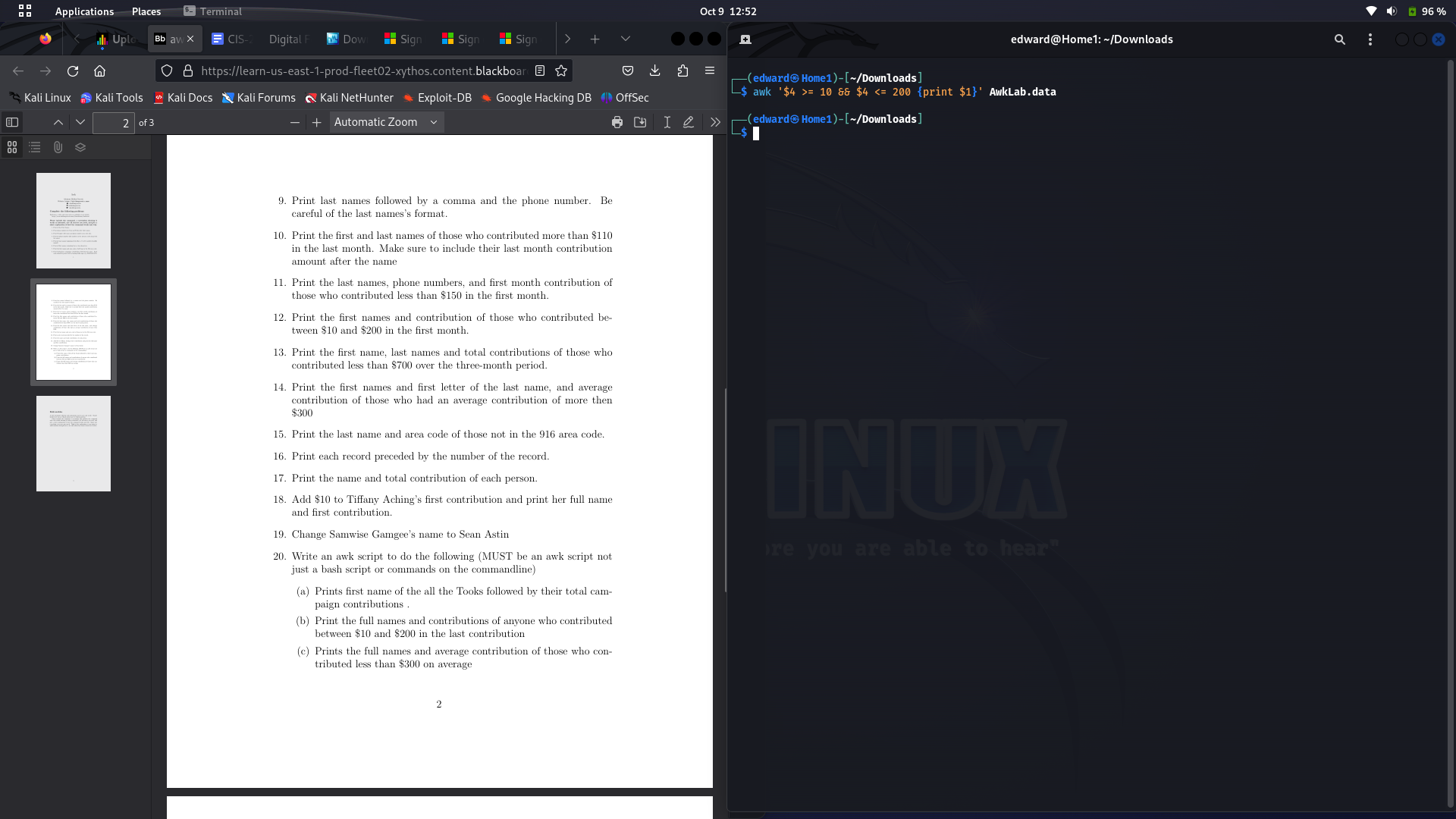
The second part **sum > 110 {print $1, $2, "$" sum}'** will then see if the condition is met or if the person has contributed more than $110 and print out the full name and total contribution followed by the dollar sign **$**



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

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

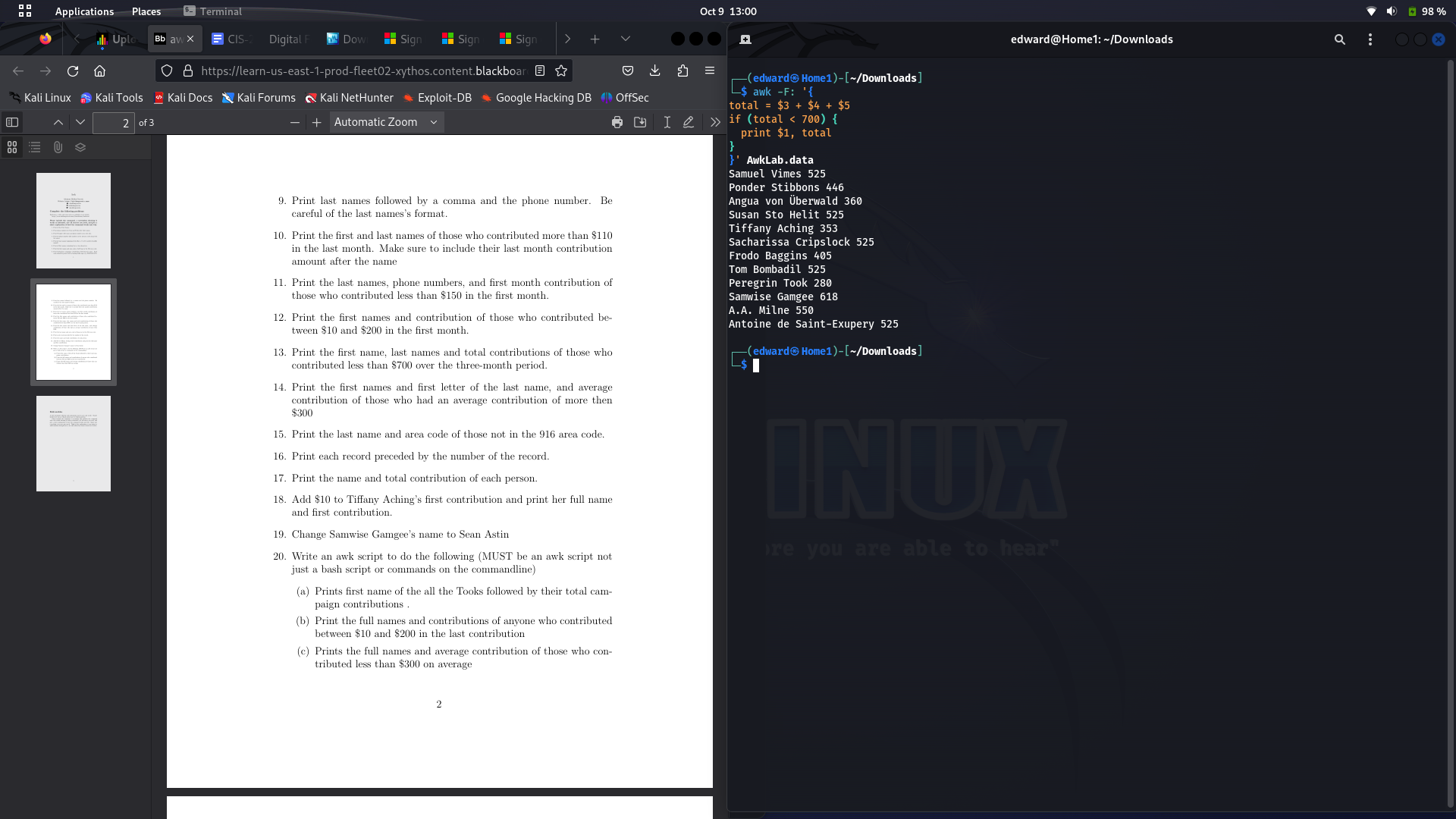
**Command: awk '$4 >= 10 && $4 <= 200 {print $1}' AwkLab.data**Here I did a search to look for contributions between 10 and 200 in the first month but the condition didn’t seem to be met so it returned nothing.



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

**Command: awk -F: ‘{ total = $3 + $4 + $5 if (total < 700) { print $1, total}}’**

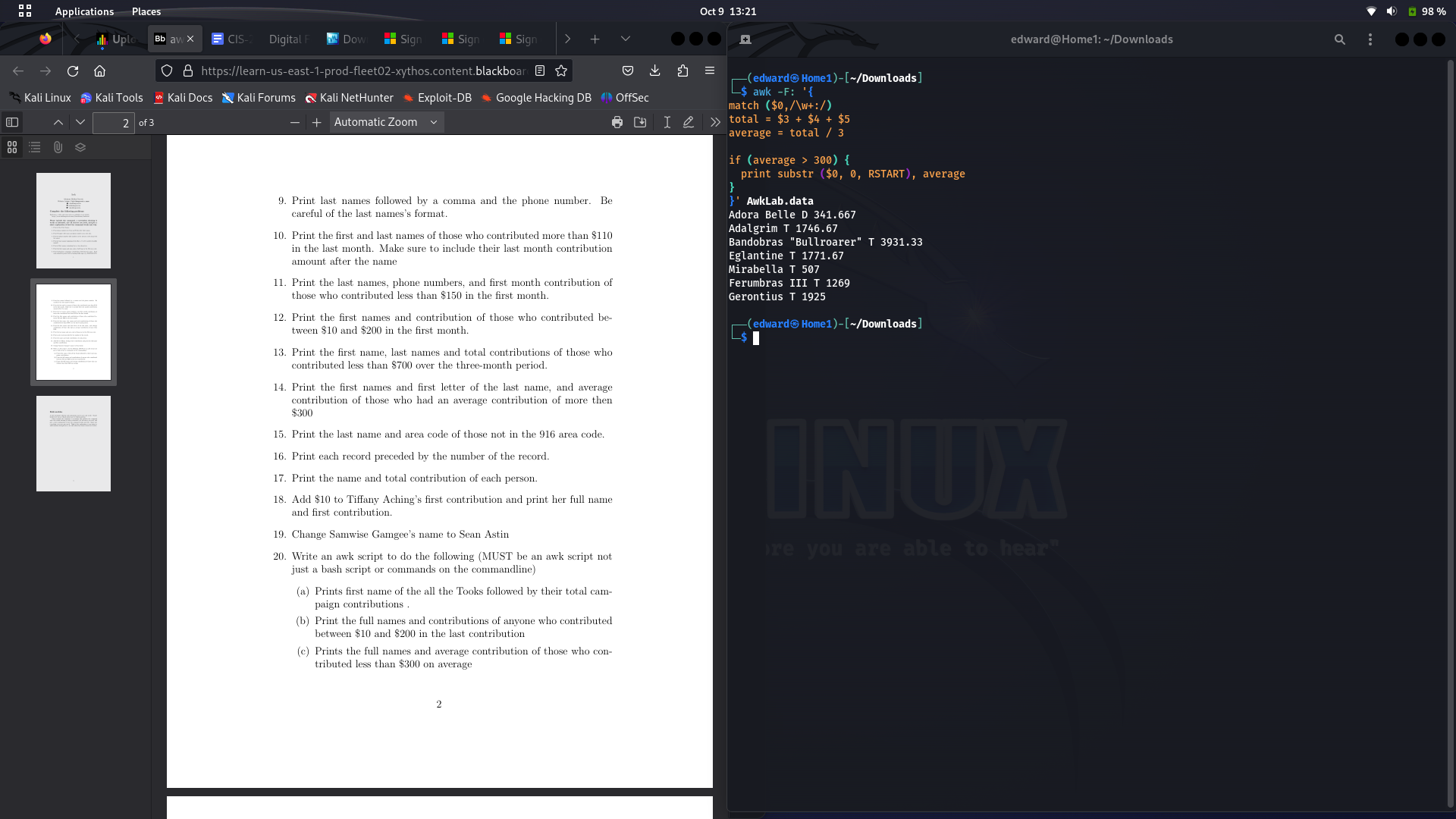
Indention is a little weird on this one, but we basically created the **total** variable which will store the values of all the total contributions. **If** the sum of these is less than 700, then we tell **awk** to print the full name and the **total** variable.



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

**Command: awk -F: '{match ($0,/\w+:/) total = $3 + $4 + $5 average = total / 3 if (average > 300) { print substr ($0, 0, RSTART), average}}' AwkLab.data**

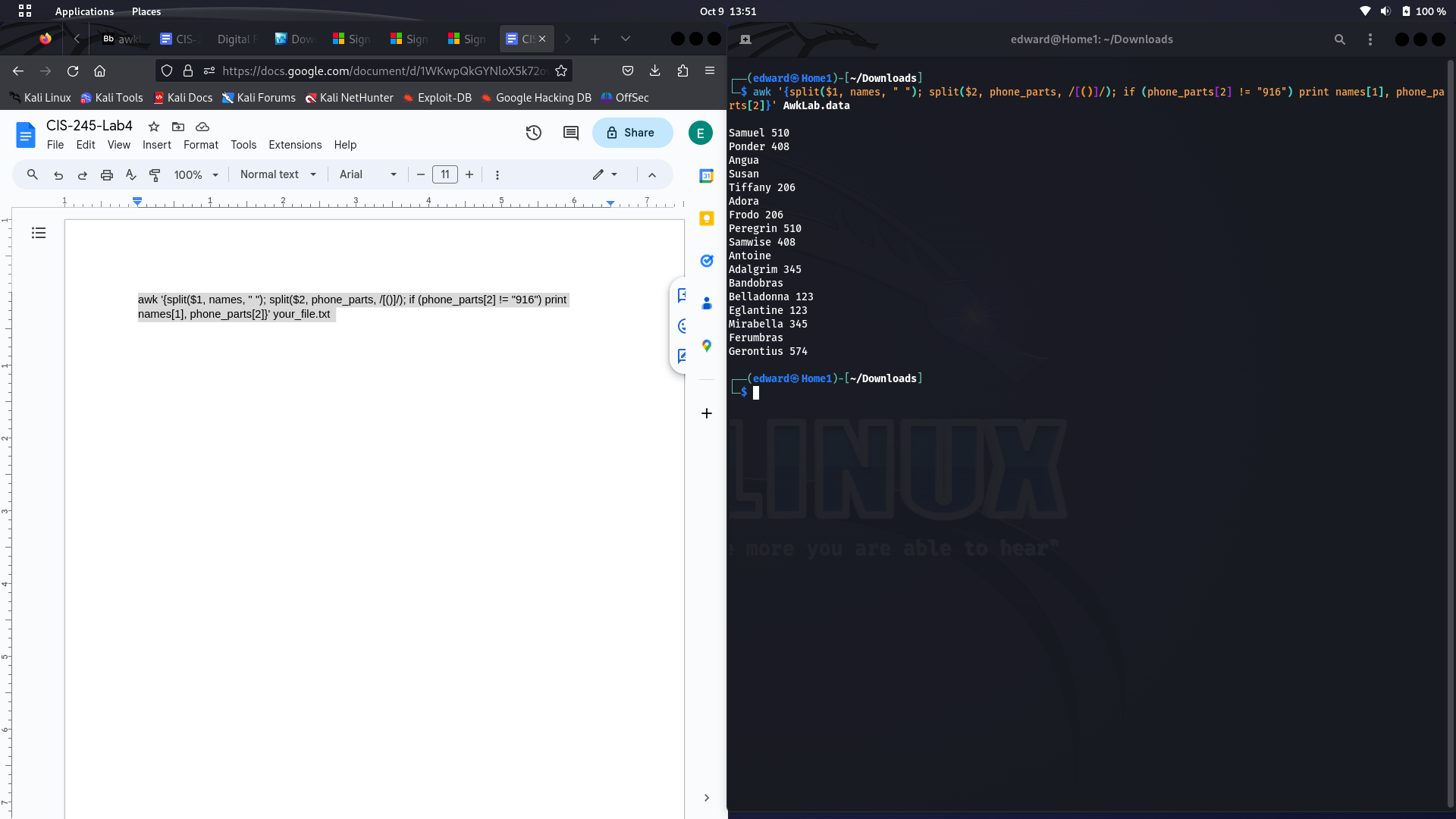
So here as with the previous problems, we create the **total** variable to store the values of all the contributions, but we then create the **average** variable which will divide this by 3 and create our **average.** We use **RSTART** to match the first character of our last name and with the **substr** we can pull that exact string.



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

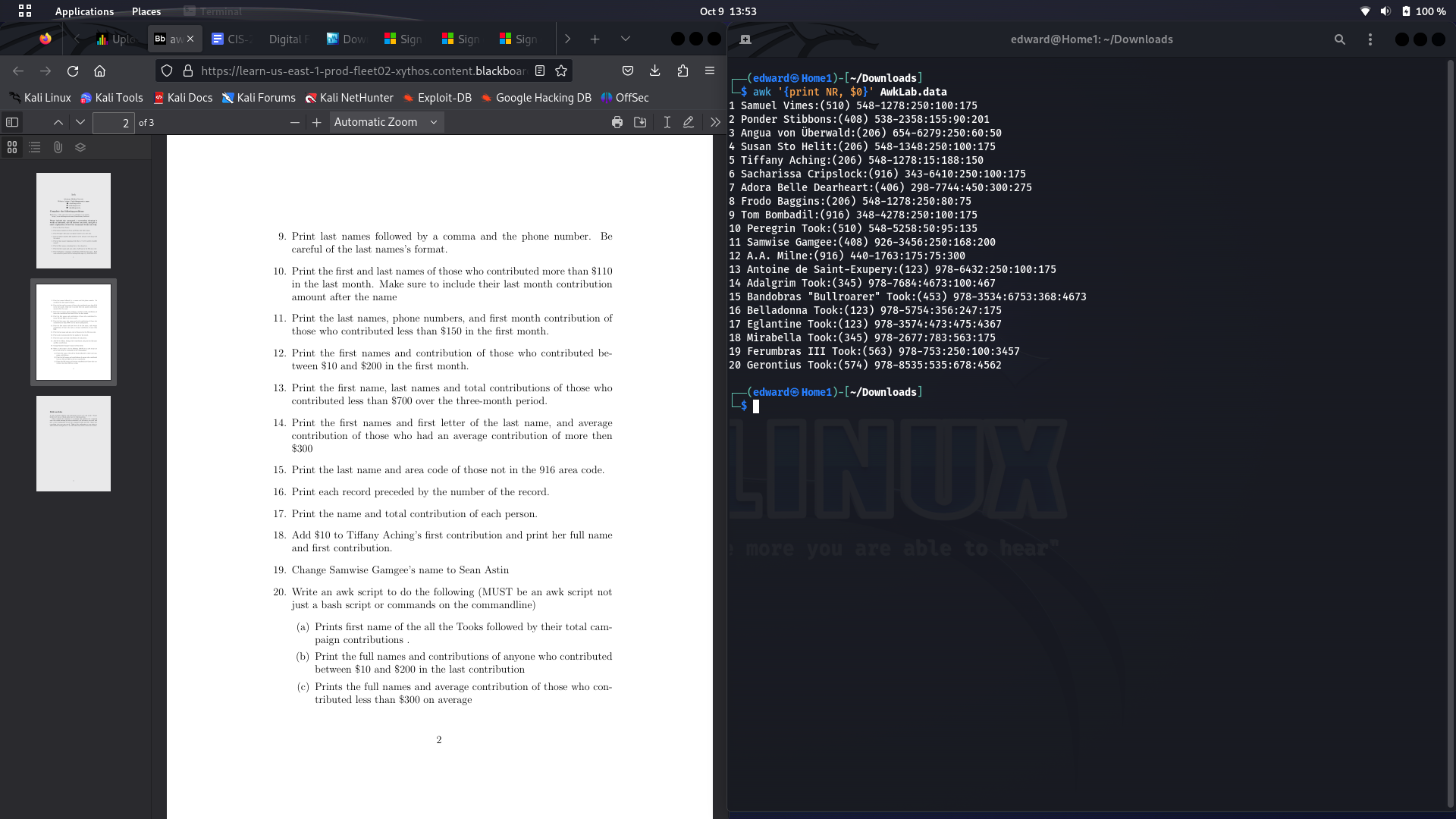
**Command:awk '{split($1, names, " "); split($2, phone\_parts, /[()]/); if (phone\_parts[2] != "916") print names[1], phone\_parts[2]}' AwkLab.data**

Here since our first field contains both our first and last name we create the variable **names** with the help of **split** that just splits our first field using a space as a delimiter. The second split **split($2, phone\_parts, /[()]/)** then stores the area code into our variable **phone\_parts** with the delimiter of **/[()]/).** If phone\_parts is different than 916, the we print the first names along with their area code



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

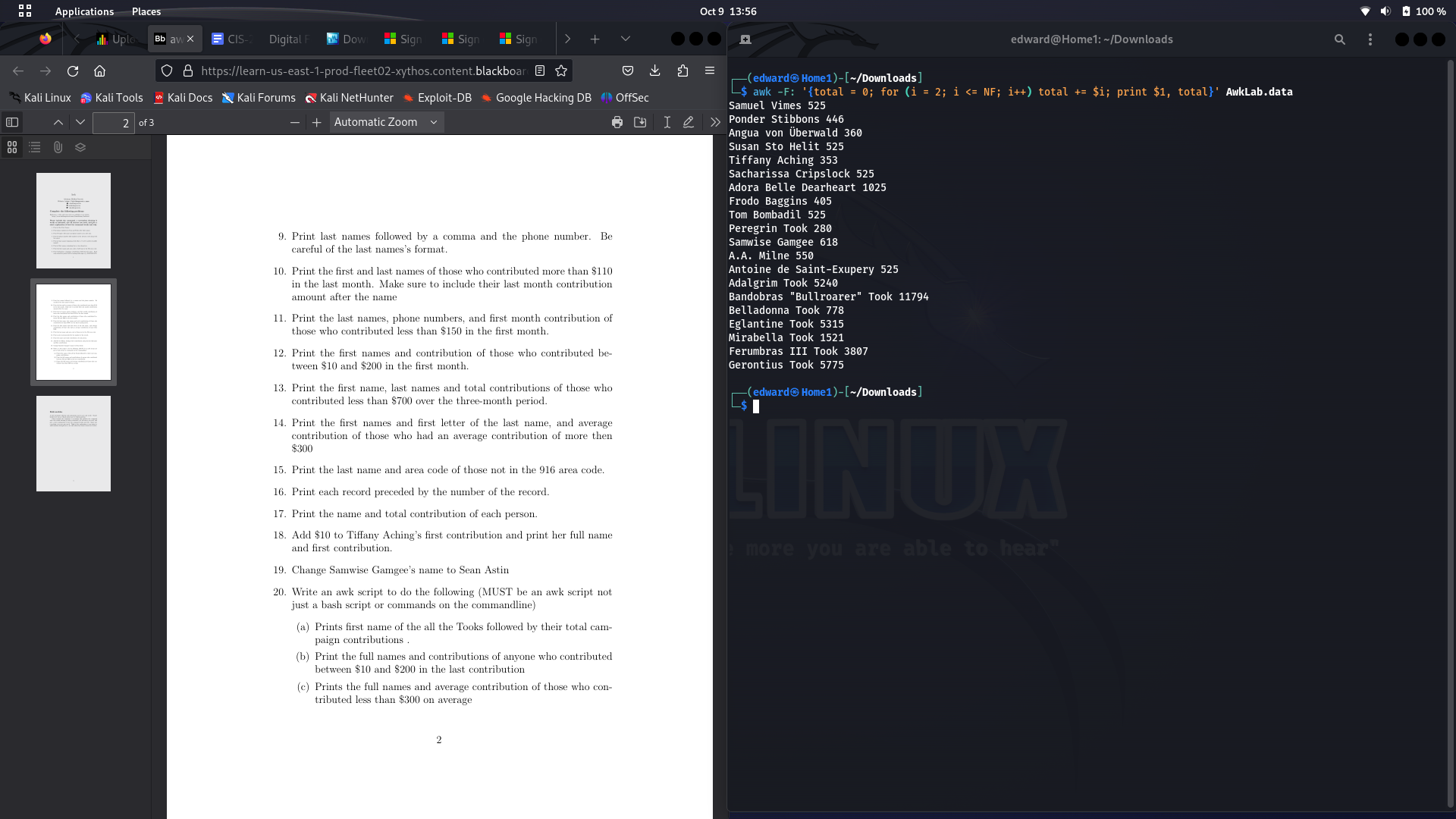
**Command: awk ‘{print NR, $0}’ AwkLab.data  
NR** is a built in variable in awk that represents the current record line, so we mix that with **$0** which prints out the whole line to get the full records



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

**awk -F: '{total = 0; for (i = 2; i <= NF; i++) total += $i; print $1, total}' AwkLab.data**

Here we again create the total variable which will get us the sum of each person’s contribution.  
**for (i = 2; i <= NF; i++) total += $i;** This loop is used to tell **awk** that starting from field 2 to add the candidates contribution and store it in **total.** Then we just **print** out the names and **total**



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

**Command: awk -F: '/Tiffany Aching/ {print $1, $3 += 10}' AwkLab.data**

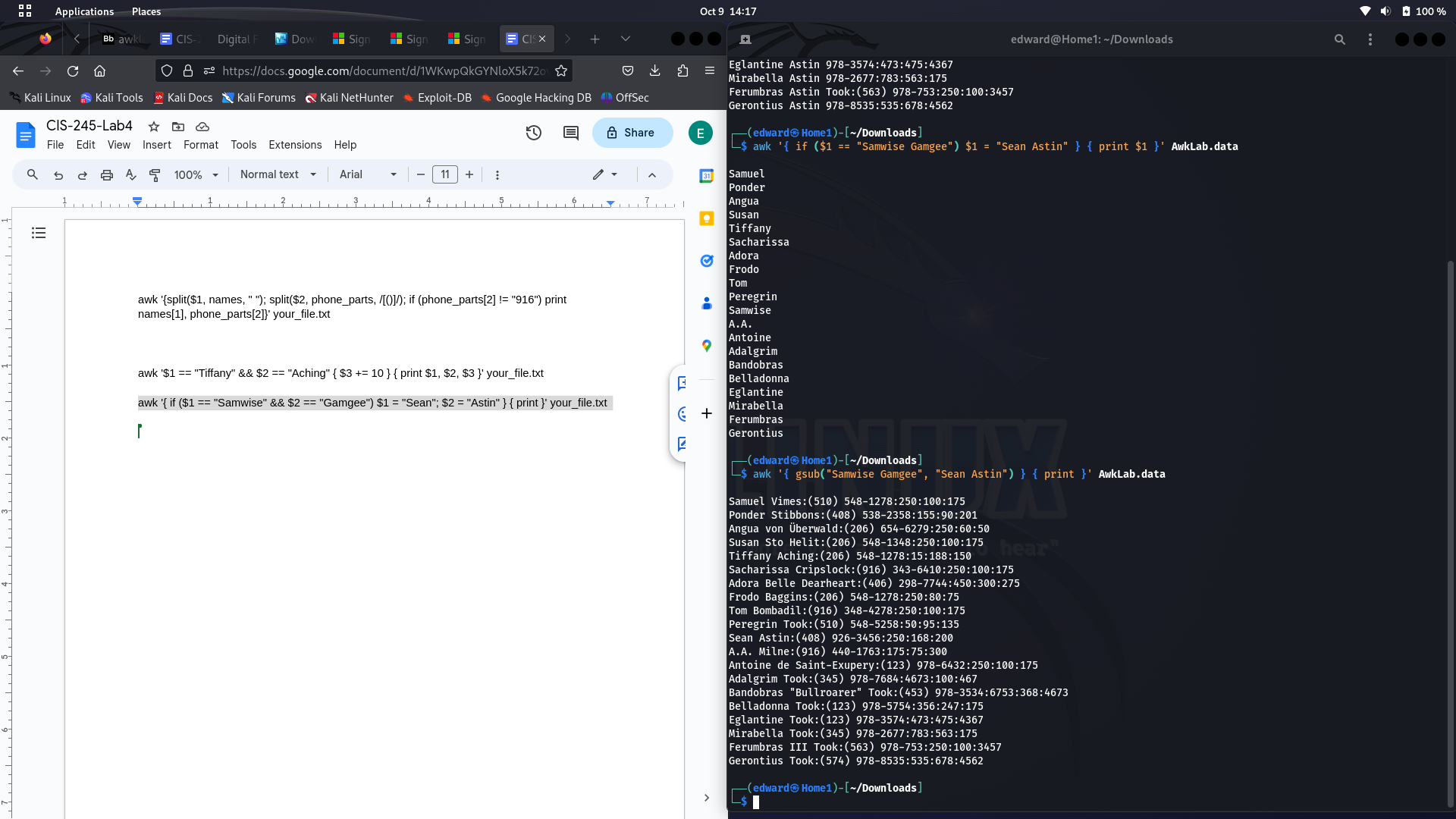
Here we just specify to **awk** to look for lines with this name and then add $10 to the 3rd field with  **$3 += 10**



19. Change Samwise Gamgee’s name to Sean Astin

**Command: awk '{ gsub("Samwise Gamgee", "Sean Astin") } { print }' AwkLab.data**

Here we used the built in variable **gsub** to globally replace all lines with **"Samwise Gamgee"** to **"Sean Astin". { print }'** is just used to print the updated records



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 cam- paign 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 con- tributed less than $300 on average

**Could not get script to work**

**RESOURCES**

**Man awk**

[**https://www.gnu.org/software/gawk/manual/html\_node/String-Functions.html**](https://www.gnu.org/software/gawk/manual/html_node/String-Functions.html)

[**https://www.geeksforgeeks.org/awk-command-unixlinux-examples/**](https://www.geeksforgeeks.org/awk-command-unixlinux-examples/)

<https://www.howtogeek.com/562941/how-to-use-the-awk-command-on-linux/>

<https://linuxhandbook.com/awk-command-tutorial/>