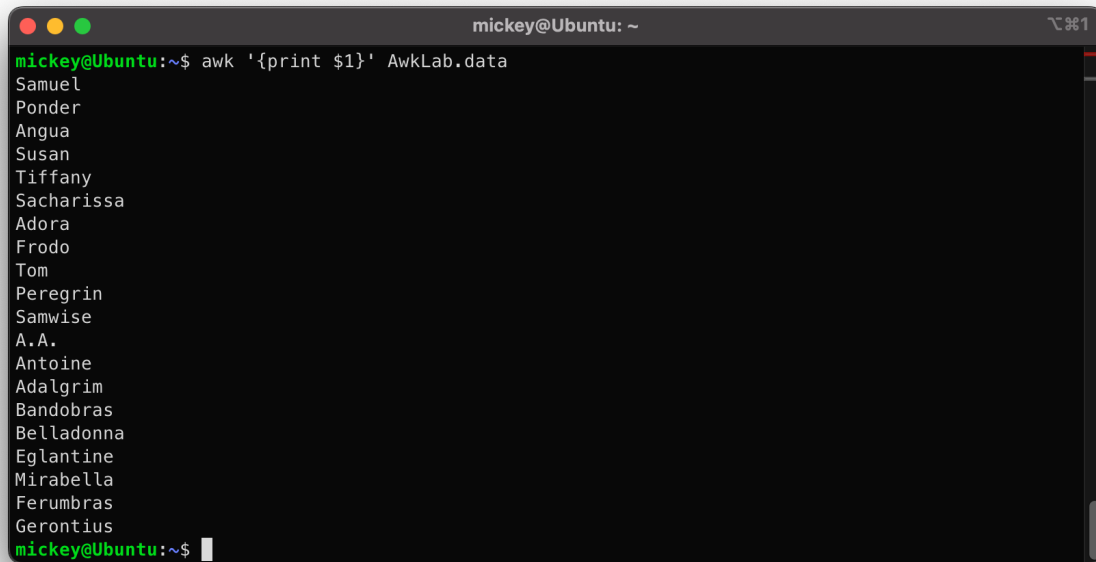


AWK Command

1. Print all the First Names.

Command

```
awk '{print $1}' AwkLab.data
```



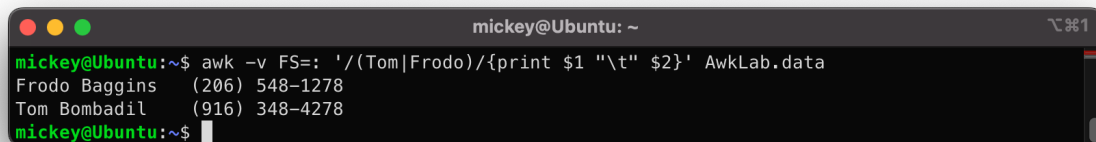
```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk '{print $1}' AwkLab.data  
Samuel  
Ponder  
Angua  
Susan  
Tiffany  
Sacharissa  
Adora  
Frodo  
Tom  
Peregrin  
Samwise  
A.A.  
Antoine  
Adalgrim  
Bandobras  
Belladonna  
Eglantine  
Mirabella  
Ferumbras  
Gerontius  
mickey@Ubuntu:~$
```

Explanation: By default **awk** uses spaces as the delimiter and we can get everything upto the first space (which in here is the **first name**).

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

Command

```
awk -v FS=: '/(Tom|Frodo)/{print $1 "\t" $2}' AwkLab.data
```



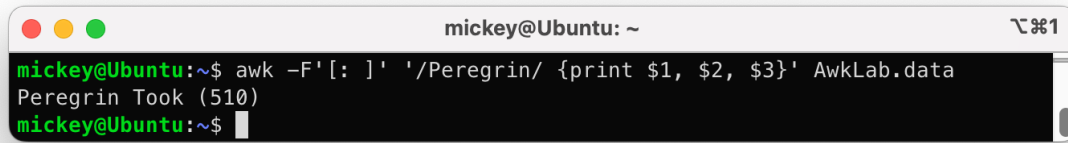
```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -v FS=: '/(Tom|Frodo)/{print $1 "\t" $2}' AwkLab.data  
Frodo Baggins (206) 548-1278  
Tom Bombadil (916) 348-4278  
mickey@Ubuntu:~$
```

Explanation: Given the data format, to print phone numbers it is better to change the default delimiter **space** to **:** which separates the lines based off the **:** delimiter. To achieve the same, we have to pass the argument **FS=:**. To reduce the search space to just **Tom** or **Frodo**, we use the regex **(Tom|Frodo)** and print the name and the phone number represented by **\$1** and **\$2** respectively.

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

Command

```
awk -F'[: ]' '/Peregrin/ {print $1, $2, $3}' AwkLab.data
```



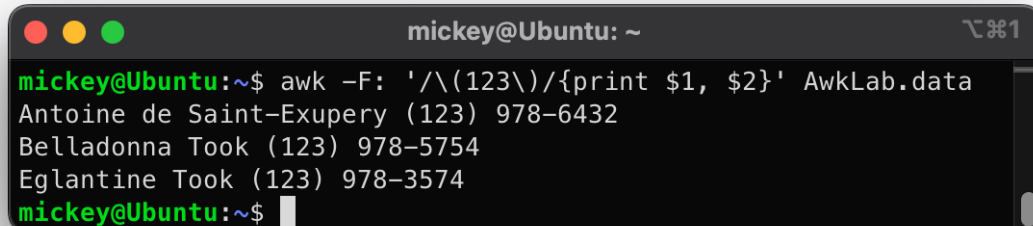
```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -F'[: ]' '/Peregrin/ {print $1, $2, $3}' AwkLab.data  
Peregrin Took (510)  
mickey@Ubuntu:~$
```

Explanation: Given the data format, it will be best to break the lines based on both **space** and **:** as the delimiters. To achieve the same, we pass the argument flag `-F'[:]'`. Using the regex `/Peregrin/` we reduce the search space for **Peregrin** only and print the first name, last name and area code represented by `$1`, `$2` and `$3` respectively.

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
```



```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -F: '/\ (123\)/ {print $1, $2}' AwkLab.data  
Antoine de Saint-Exupery (123) 978-6432  
Belladonna Took (123) 978-5754  
Eglantine Took (123) 978-3574  
mickey@Ubuntu:~$
```

Explanation: Given the data format, to print phone numbers it is better to change the default delimiter **space** to **:** which separates the lines based off the **:** delimiter. To achieve the same, we have to pass the argument `-F:`. To reduce the search space to (123) area code, we use the regex `/\ (123\)/` to match the area-code. We can print the full name and complete phone number using `$1` and `$2` accordingly.

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

Command

```
awk -F'[: ]' '/([a-zA-Z]+)( ([a-zA-Z]+) | )((T|D)[a-zA-Z]+):.*/ {print $1}' AwkLab.data
```

```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -F: '/([a-zA-Z]+)( ([a-zA-Z]+) | )((T|D)[a-zA-Z]+):.*/{print $1}' AwkLab.data  
Adora Belle Dearheart  
Peregrin Took  
Adalgrim Took  
Belladonna Took  
Eglantine Took  
Mirabella Took  
Ferumbas III Took  
Gerontius Took  
mickey@Ubuntu:~$
```

Explanation: Given the data format, to print phone numbers it is better to change the default delimiter `space` to `:` which separates the lines based off the `:` delimiter. To achieve the same, we have to pass the argument `-F:`. To reduce the search space to last names starting with either `T` or `D` we use the regex `/([a-zA-Z]+)(([a-zA-Z]+) |)((T|D)[a-zA-Z]+):.*/`.

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

Command

```
awk '/^.{1,4} .*/{print $1}' AwkLab.data
```

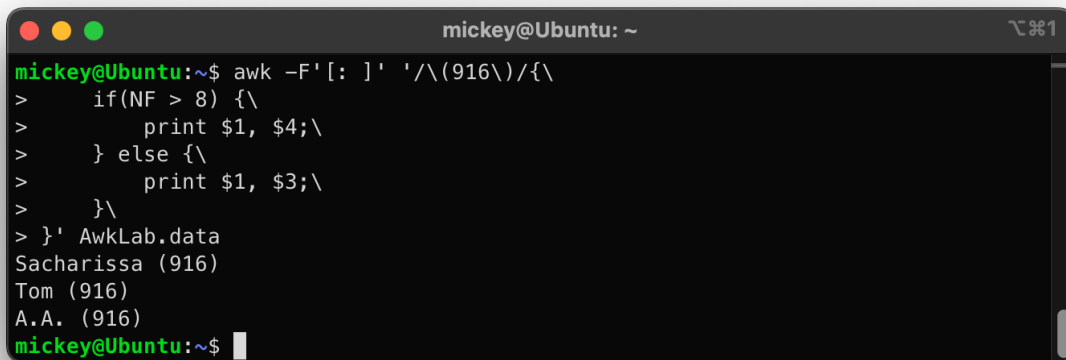
```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk '/^.{1,4} .*/{print $1}' AwkLab.data  
Tom  
A.A.  
mickey@Ubuntu:~$
```

Explanation: To reduce the search space to all the first names containing four or less, we can use the regex `/^.{1,4} .*` which looks for first names of length 1 to 4. The period (`.`) represents any character and `{1,4}` represents the length of the pattern. So here we are looking for any string of length 1 to 4 with a following space and any characters afterwards. We can use the default delimiter and print the first component (i.e. the first name) with `$1`.

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

Command

```
awk -F'[: ]' ' /\(916\)\/{\n  
    if(NF > 8) {\n  
        print $1, $4;\n  
    } else {\n  
        print $1, $3;\n  
    }\n  
' AwkLab.data
```



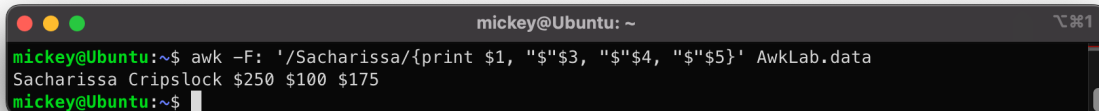
```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -F'[: ]' '/^(916\)/{\n  
>     if(NF > 8) {\n  
>         print $1, $4;\n  
>     } else {\n  
>         print $1, $3;\n  
>     }\n  
> }' AwkLab.data  
Sacharissa (916)  
Tom (916)  
A.A. (916)  
mickey@Ubuntu:~$
```

Explanation: To reduce the search space to lines containing the area code 916 we can use the regex `/^(916\)/` and use both space and colon as the delimiters (by passing the argument `-F'[:]'`). We can print the first name with `$1` and area code with `$3` or `$4` depending if the name contains a middle name. We can check the presence of a middle name by checking the total number of components. If is greater than 8, we can be sure that the name contains a middle name.

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 -F: '/Sacharissa/{print $1, "$"$3, "$"$4, "$"$5}' AwkLab.data
```



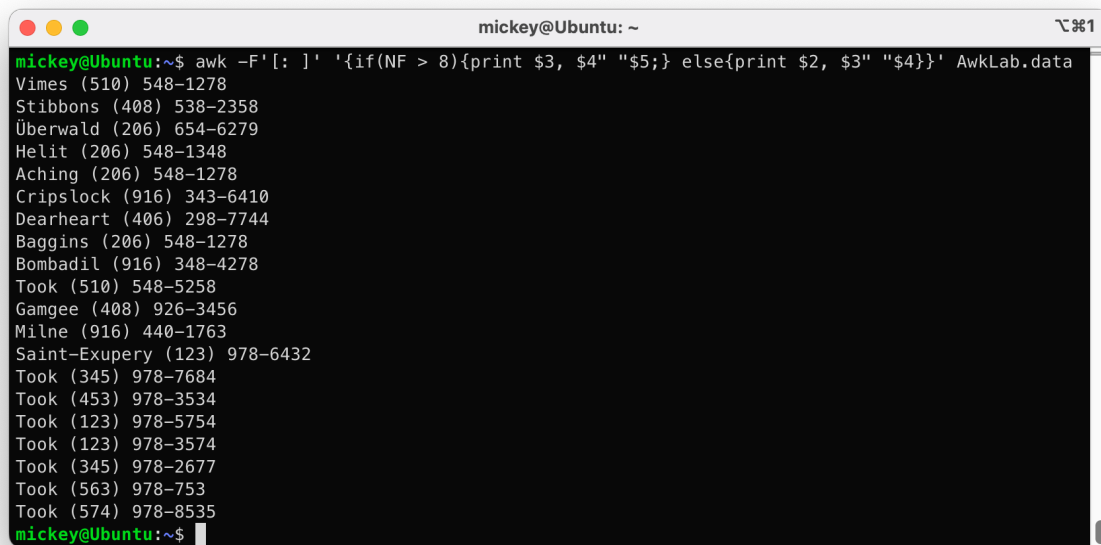
```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -F: '/Sacharissa/{print $1, "$"$3, "$"$4, "$"$5}' AwkLab.data  
Sacharissa Cripslock $250 $100 $175  
mickey@Ubuntu:~$
```

Explanation: Use colon as the delimiter and reduce the search space using the regex `/Sacharissa/`. Print the details accordingly and use the `"$"` for printing inlined characters.

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

Command

```
awk -F'[: ]' '{if(NF > 8){print $3, $4" "$5;} else{print $2, $3" "$4}}' AwkLab.data
```

A terminal window titled 'mickey@Ubuntu: ~' with a terminal icon. The prompt is 'mickey@Ubuntu:~\$'. The command entered is 'awk -F'[:]' '{if(NF > 8){print \$3, \$4 " "\$5;} else{print \$2, \$3 " "\$4}}' AwkLab.data'. The output lists 20 entries, each with a name, a number in parentheses, and a phone number. The entries are: Vimes (510) 548-1278, Stibbons (408) 538-2358, Überwald (206) 654-6279, Helit (206) 548-1348, Aching (206) 548-1278, Cripslock (916) 343-6410, Dearheart (406) 298-7744, Baggins (206) 548-1278, Bombadil (916) 348-4278, Took (510) 548-5258, Gamgee (408) 926-3456, Milne (916) 440-1763, Saint-Exupery (123) 978-6432, Took (345) 978-7684, Took (453) 978-3534, Took (123) 978-5754, Took (123) 978-3574, Took (345) 978-2677, Took (563) 978-753, and Took (574) 978-8535. The prompt 'mickey@Ubuntu:~\$' is visible at the bottom.

```
mickey@Ubuntu:~$ awk -F'[: ]' '{if(NF > 8){print $3, $4 " "$5;} else{print $2, $3 " "$4}}' AwkLab.data
Vimes (510) 548-1278
Stibbons (408) 538-2358
Überwald (206) 654-6279
Helit (206) 548-1348
Aching (206) 548-1278
Cripslock (916) 343-6410
Dearheart (406) 298-7744
Baggins (206) 548-1278
Bombadil (916) 348-4278
Took (510) 548-5258
Gamgee (408) 926-3456
Milne (916) 440-1763
Saint-Exupery (123) 978-6432
Took (345) 978-7684
Took (453) 978-3534
Took (123) 978-5754
Took (123) 978-3574
Took (345) 978-2677
Took (563) 978-753
Took (574) 978-8535
mickey@Ubuntu:~$
```

Explanation: Use both colon and space as the delimiters. Check for the presence of middle name. If middle name is present, print last name using \$3 else print last name using \$2. The area code and phone number follows.

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

Command

```
awk -F'[: ]' '{if(NF > 8){\
  if($8 > 110){\
    print $1, $3, "$"$8;\
  }\
} else {\
  if($7 > 110) {\
    print $1, $2, "$"$7\
  }\
}\
}' AwkLab.data
```

```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -F[: ] ' {if(NF > 8){if($8 > 110){print $1, $3, "$"$8} else{if($7 > 110){print $1, $2, "$"$7}}}' AwkLab.data  
Samuel Vimes $175  
Ponder Stibbons $201  
Susan Helit $175  
Tiffany Aching $150  
Sacharissa Cripslock $175  
Adora Dearheart $275  
Tom Bombadil $175  
Peregrin Took $135  
Samwise Gamgee $200  
A.A. Milne $300  
Antoine Saint-Exupery $175  
Adalgrim Took $467  
Bandobras Took $4673  
Belladonna Took $175  
Eglantine Took $4367  
Mirabella Took $175  
Ferumbas Took $3457  
Gerontius Took $4562  
mickey@Ubuntu:~$
```

Explanation: Use both colon and space as the delimiters. Check for the presence of middle name. If middle name is present, print last name using \$3 and last month's contribution using \$8 else print last name using \$2 last month's contribution using \$7. In between, we need to check if the last month's contribution is more than \$110. If not, we will not print anything.

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

Command

```
awk -F'[: ]' '{\n  if(NF > 8) {\n    if($6 < 150) {\n      print $3, $4, $5, "$"$6\n    }\n  } else {\n    if($5 < 150) {\n      print $2, $3, $4, "$"$5\n    }\n  }\n}' AwkLab.data
```

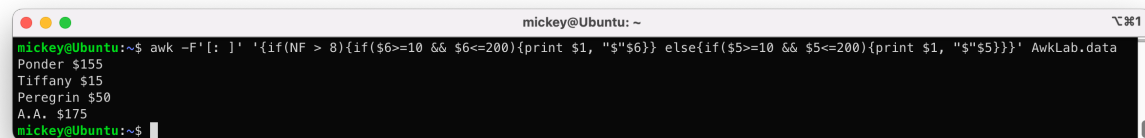
```
mickey@Ubuntu: ~  
mickey@Ubuntu:~$ awk -F'[: ]' '{if(NF > 8){if($6 < 150){print $3, $4, $5, "$"$6} else{if($5 < 150){print $2, $3, $4, "$"$5}}}' AwkLab.data  
Aching (206) 548-1278 $15  
Took (510) 548-5258 $50  
mickey@Ubuntu:~$
```

Explanation: Use both colon and space as the delimiters. Check for the presence of middle name. If middle name is present, print last name using \$3, phone number using \$4 and \$5, and first month's contribution using \$6 else print last name using \$2, phone number using \$3 and \$4, and first month's contribution using \$. In between, we need to check if the first month's contribution is less than \$150 using the if statement. If not, we will not print anything.

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

Command

```
awk -F'[: ]' '{\
    if(NF > 8) {\
        if($6>=10 && $6<=200) {\
            print $1, "$"$6\
        }\
    } else {\
        if($5>=10 && $5<=200) {\
            print $1, "$"$5\
        }\
    }\
}' AwkLab.data
```



```
mickey@Ubuntu: ~
mickey@Ubuntu:~$ awk -F'[: ]' '{if(NF > 8){if($6>=10 && $6<=200){print $1, "$"$6}} else{if($5>=10 && $5<=200){print $1, "$"$5}}}' AwkLab.data
Ponder $155
Tiffany $15
Peregrin $50
A.A. $175
mickey@Ubuntu:~$
```

Explanation: Use both colon and space as the delimiters. Check for the presence of middle name. check for the first month's contribution if it is between [\$10, \$200] and print it accordingly (either using \$5 or \$6).

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'[: ]' '{\
    if(NF > 8) {\
        if($6+$7+$8<700) {\
            print $1, $3, "$"($6+$7+$8)\
        }\
    } else {\
        if($5+$6+$7<700) {\
            print $1, $2, "$"($5+$6+$7)\
        }\
    }\
}' AwkLab.data
```



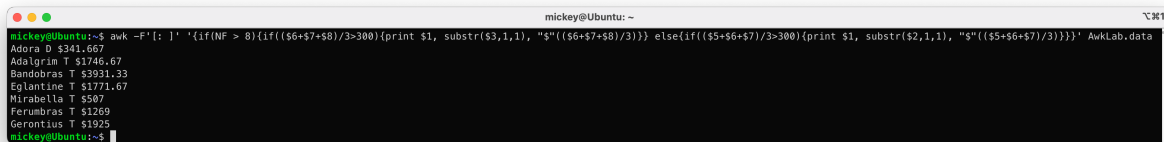
```
mickey@Ubuntu:~$
mickey@Ubuntu:~$ awk -F'[: ]' '{if(NF > 8){if($6+$7+$8<700){print $1, $3, "$"($6+$7+$8)}} else{if($5+$6+$7<700){print $1, $2, "$"($5+$6+$7)}}}' AwkLab.data
Samuel Vimes $525
Ponder Stibbons $446
Angua Überwald $360
Susan Helit $525
Tiffany Aching $353
Sacharissa Crisplock $525
Frodo Baggins $405
Tom Bombadil $525
Peregrin Took $280
Samwise Gamgee $618
A.A. Milne $550
Antoine Saint-Exupery $525
mickey@Ubuntu:~$
```

Explanation: Use both colon and space as the delimiters. Check for the presence of middle name. check for the total contribution if it is less than \$700 and print it accordingly (either using \$6+\$7+\$8 or \$5+\$6+\$7) along with the first name and last name.

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'[: ]' '{\
    if(NF > 8) {\
        if(($6+$7+$8)/3>300) {\
            print $1, substr($3,1,1), "$"((($6+$7+$8)/3))\
        }\
    } else {\
        if(($5+$6+$7)/3>300) {\
            print $1, substr($2,1,1), "$"((($5+$6+$7)/3))\
        }\
    }\
}' AwkLab.data
```



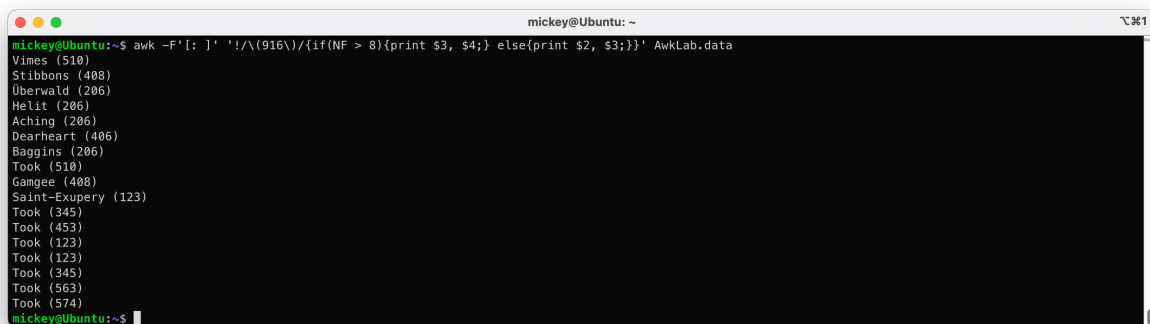
```
mickey@Ubuntu: ~
mickey@Ubuntu:~$ awk -F'[: ]' '{\
    if(NF > 8) {\
        if(($6+$7+$8)/3>300) {\
            print $1, substr($3,1,1), "$"((($6+$7+$8)/3))\
        }\
    } else {\
        if(($5+$6+$7)/3>300) {\
            print $1, substr($2,1,1), "$"((($5+$6+$7)/3))\
        }\
    }\
}' AwkLab.data
Adora D $341.667
Adalgrim T $1746.67
Bandobras T $3031.33
Eglantine T $1771.67
Hirabella T $507
Perumbras T $1269
Gerontius T $1925
mickey@Ubuntu:~$
```

Explanation: Use both colon and space as the delimiters. Check for the presence of middle name. check for the average contribution (by dividing the total contribution by 3) if it is greater than \$300 and print it accordingly (either using \$6+\$7+\$8 or \$5+\$6+\$7) along with the first name and print use substr(\$X,1,1) to get the first character of the last name where \$X is the string.

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

Command

```
awk -F'[: ]' '{ !/\(916\)}/{\
    if(NF > 8) {\
        print $3, $4;\
    } else {\
        print $2, $3;\
    }\
}' AwkLab.data
```



```
mickey@Ubuntu: ~
mickey@Ubuntu:~$ awk -F'[: ]' '{ !/\(916\)}/{\
    if(NF > 8) {\
        print $3, $4;\
    } else {\
        print $2, $3;\
    }\
}' AwkLab.data
Vimes (510)
Stibbons (408)
Überwald (206)
Helit (206)
Aching (206)
Dearheart (406)
Baggins (206)
Took (510)
Gamgee (408)
Saint-Exupery (123)
Took (345)
Took (453)
Took (123)
Took (123)
Took (345)
Took (563)
Took (574)
mickey@Ubuntu:~$
```

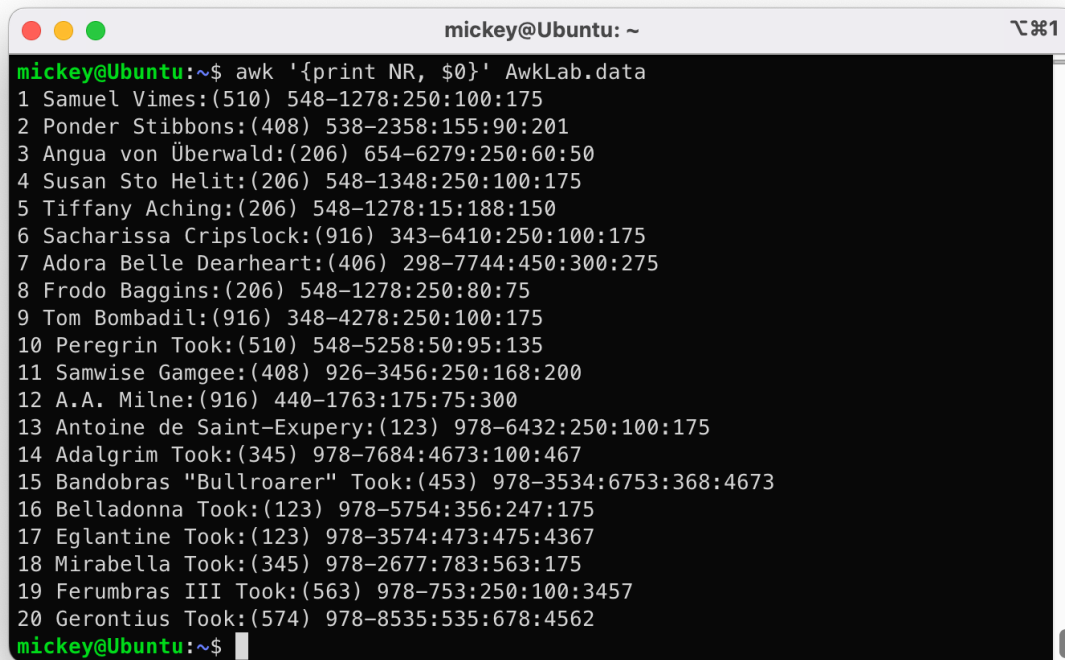
Explanation: To reduce the search space to strings not containing /\(916\) / use the ! before

the regex which tell us the to inverse match the space. Check for the presence of middle name and print accordingly.

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

Command

```
awk '{print NR, $0}' AwkLab.data
```

A terminal window titled 'mickey@Ubuntu: ~' showing the output of the command 'awk '{print NR, \$0}' AwkLab.data'. The output lists 20 records, each starting with a line number followed by a space and the record content. The records are: 1 Samuel Vimes:(510) 548-1278:250:100:175, 2 Ponder Stibbons:(408) 538-2358:155:90:201, 3 Angua von Überwald:(206) 654-6279:250:60:50, 4 Susan Sto Helit:(206) 548-1348:250:100:175, 5 Tiffany Aching:(206) 548-1278:15:188:150, 6 Sacharissa Cripslock:(916) 343-6410:250:100:175, 7 Adora Belle Dearheart:(406) 298-7744:450:300:275, 8 Frodo Baggins:(206) 548-1278:250:80:75, 9 Tom Bombadil:(916) 348-4278:250:100:175, 10 Peregrin Took:(510) 548-5258:50:95:135, 11 Samwise Gamgee:(408) 926-3456:250:168:200, 12 A.A. Milne:(916) 440-1763:175:75:300, 13 Antoine de Saint-Exupery:(123) 978-6432:250:100:175, 14 Adalgrim Took:(345) 978-7684:4673:100:467, 15 Bandobras "Bullroarer" Took:(453) 978-3534:6753:368:4673, 16 Belladonna Took:(123) 978-5754:356:247:175, 17 Eglantine Took:(123) 978-3574:473:475:4367, 18 Mirabella Took:(345) 978-2677:783:563:175, 19 Ferumbras III Took:(563) 978-753:250:100:3457, 20 Gerontius Took:(574) 978-8535:535:678:4562. The prompt 'mickey@Ubuntu:~\$' is visible at the bottom.

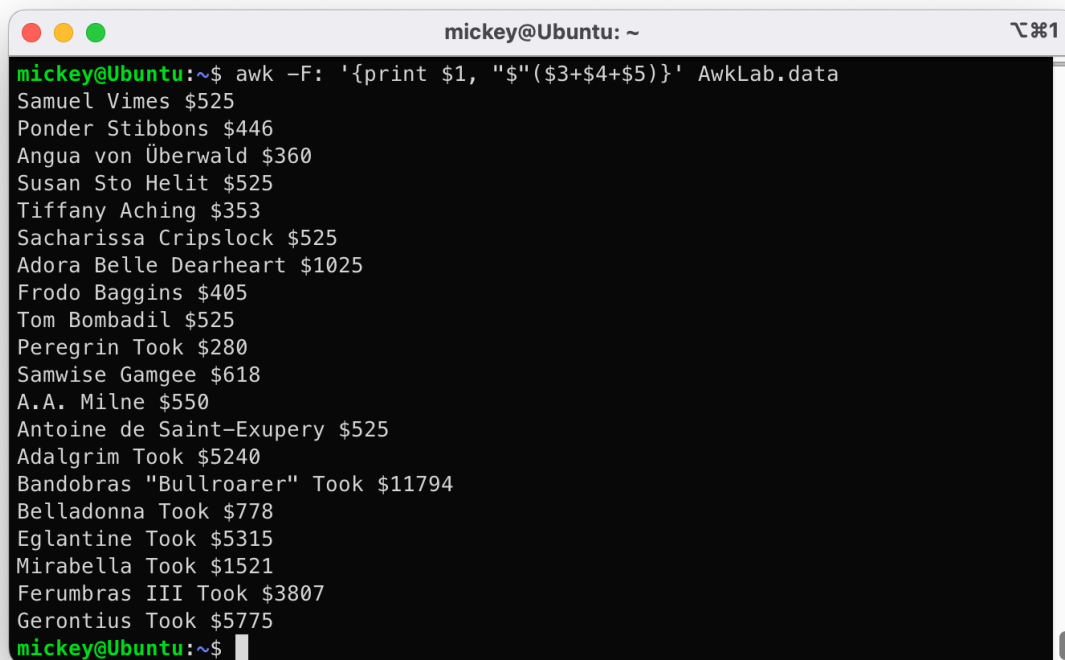
```
mickey@Ubuntu:~$ awk '{print NR, $0}' AwkLab.data
1 Samuel Vimes:(510) 548-1278:250:100:175
2 Ponder Stibbons:(408) 538-2358:155:90:201
3 Angua von Überwald:(206) 654-6279:250:60:50
4 Susan Sto Helit:(206) 548-1348:250:100:175
5 Tiffany Aching:(206) 548-1278:15:188:150
6 Sacharissa Cripslock:(916) 343-6410:250:100:175
7 Adora Belle Dearheart:(406) 298-7744:450:300:275
8 Frodo Baggins:(206) 548-1278:250:80:75
9 Tom Bombadil:(916) 348-4278:250:100:175
10 Peregrin Took:(510) 548-5258:50:95:135
11 Samwise Gamgee:(408) 926-3456:250:168:200
12 A.A. Milne:(916) 440-1763:175:75:300
13 Antoine de Saint-Exupery:(123) 978-6432:250:100:175
14 Adalgrim Took:(345) 978-7684:4673:100:467
15 Bandobras "Bullroarer" Took:(453) 978-3534:6753:368:4673
16 Belladonna Took:(123) 978-5754:356:247:175
17 Eglantine Took:(123) 978-3574:473:475:4367
18 Mirabella Took:(345) 978-2677:783:563:175
19 Ferumbras III Took:(563) 978-753:250:100:3457
20 Gerontius Took:(574) 978-8535:535:678:4562
mickey@Ubuntu:~$
```

Explanation: “NR” is a special built-in variable of AWK that stands for “number of records”. It keeps count of the number of line and we can print the NR variable to get the current line and print it accordingly.

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

Command

```
awk -F: '{print $1, "$"($3+$4+$5)}' AwkLab.data
```

A terminal window titled 'mickey@Ubuntu: ~' with a standard Ubuntu window header (red, yellow, green buttons). The terminal shows a command: `mickey@Ubuntu:~$ awk -F: '{print $1, "$"($3+$4+$5)}' AwkLab.data`. The output lists names and their total contributions: Samuel Vimes \$525, Ponder Stibbons \$446, Angua von Überwald \$360, Susan Sto Helit \$525, Tiffany Aching \$353, Sacharissa Cripslock \$525, Adora Belle Dearheart \$1025, Frodo Baggins \$405, Tom Bombadil \$525, Peregrin Took \$280, Samwise Gamgee \$618, A.A. Milne \$550, Antoine de Saint-Exupery \$525, Adalgrim Took \$5240, Bandobras "Bullroarer" Took \$11794, Belladonna Took \$778, Eglantine Took \$5315, Mirabella Took \$1521, Ferumbras III Took \$3807, Gerontius Took \$5775. The prompt `mickey@Ubuntu:~$` is visible at the bottom.

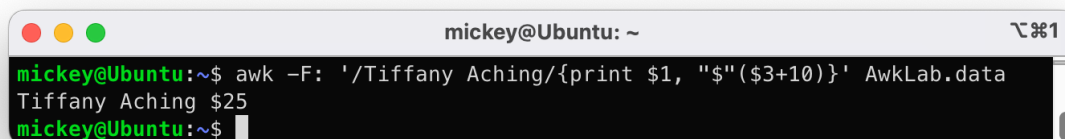
```
mickey@Ubuntu:~$ awk -F: '{print $1, "$"($3+$4+$5)}' AwkLab.data
Samuel Vimes $525
Ponder Stibbons $446
Angua von Überwald $360
Susan Sto Helit $525
Tiffany Aching $353
Sacharissa Cripslock $525
Adora Belle Dearheart $1025
Frodo Baggins $405
Tom Bombadil $525
Peregrin Took $280
Samwise Gamgee $618
A.A. Milne $550
Antoine de Saint-Exupery $525
Adalgrim Took $5240
Bandobras "Bullroarer" Took $11794
Belladonna Took $778
Eglantine Took $5315
Mirabella Took $1521
Ferumbras III Took $3807
Gerontius Took $5775
mickey@Ubuntu:~$
```

Explanation: Using colon as the only delimiter, we can print the full name represented by \$1 and add up \$3+\$4+\$5 to get the total contribution.

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
```

A terminal window titled 'mickey@Ubuntu: ~' with a standard Ubuntu window header. The terminal shows a command: `mickey@Ubuntu:~$ awk -F: '/Tiffany Aching/{print $1, "$"($3+10)}' AwkLab.data`. The output is: `Tiffany Aching $25`. The prompt `mickey@Ubuntu:~$` is visible at the bottom.

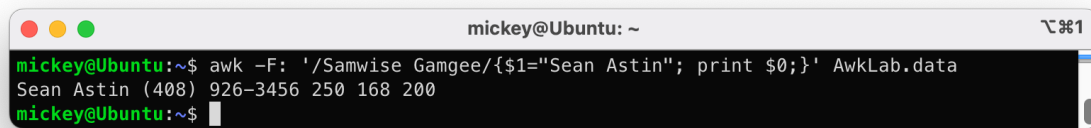
```
mickey@Ubuntu:~$ awk -F: '/Tiffany Aching/{print $1, "$"($3+10)}' AwkLab.data
Tiffany Aching $25
mickey@Ubuntu:~$
```

Explanation: Reduce the search space using the regex `/Tiffany Aching/`. Using the colon delimiter, we can get the full name and print the first contribution using \$3 after adding 10 to the value.

19. Change Samwise Gamgee's name to Sean Astin

Command

```
awk -F: '/Samwise Gamgee/{$1="Sean Astin"; print $0;}' AwkLab.data
```

A terminal window titled 'mickey@Ubuntu: ~' with a terminal icon and a zoom icon. The prompt is 'mickey@Ubuntu:~\$'. The command entered is 'awk -F: '/Samwise Gamgee/{\$1="Sean Astin"; print \$0;}' AwkLab.data'. The output is 'Sean Astin (408) 926-3456 250 168 200'. The prompt is now 'mickey@Ubuntu:~\$' with a cursor.

```
mickey@Ubuntu:~$ awk -F: '/Samwise Gamgee/{$1="Sean Astin"; print $0;}' AwkLab.data
Sean Astin (408) 926-3456 250 168 200
mickey@Ubuntu:~$
```

Explanation: Reduce the search space using the regex `/Samwise Gamgee/`. Using the colon delimiter, we can change the var `$1` to `Sean Astin` and print the entire record using `$0`.

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 Took's 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 Command

```
#!/usr/bin/awk -f
```

```
BEGIN {FS="[: ]"}{
    if(NF > 8){
        if($3=="Took"){
            print $1, "$"($6+$7+$8);
        }

        if($8 >= 10 && $8 <= 200){
            print $1, $2, $3, "| Last Contribution: $"$8;
        }

        if(($8+$7+$6)/3 < 300){
            print $1, $2, $3, "| Average: $"((($8+$7+$6)/3));
        }
    } else {
        if($2 == "Took"){
            print $1, "$"($5+$6+$7);
        }

        if($7 >= 10 && $7 <= 200){
            print $1, $2, "| Last Contribution: $"$7;
        }

        if(($7+$6+$5)/3 < 300){
            print $1, $2, "| Average: $"((($7+$6+$5)/3));
        }
    }
}
```

```
# Make it executable
chmod +x awk_script.sh

# Run the script
./awk_script.sh AwkLab.data
```

A terminal window titled "mickey@Ubuntu: ~" with a standard Ubuntu window header (red, yellow, green buttons). The terminal shows the command `./awk_script.sh AwkLab.data` and its output. The output lists names and their contribution and average values. The names are: Samuel Vimes, Ponder Stibbons, Angua von Überwald, Susan Sto Helit, Tiffany Aching, Sacharissa Cripslock, Frodo Baggins, Tom Bombadil, Peregrin Took, Samwise Gamgee, A.A. Milne, Antoine de Saint-Exupery, Adalgrim, Bandobras, Belladonna, Eglantine, Mirabella, Ferumbras, and Gerontius. The output format is: Name | Last Contribution: \$X | Average: \$Y. The terminal has a dark background and a light-colored cursor at the end of the last line.

```
mickey@Ubuntu:~$ ./awk_script.sh AwkLab.data
Samuel Vimes | Last Contribution: $175
Samuel Vimes | Average: $175
Ponder Stibbons | Average: $148.667
Angua von Überwald | Last Contribution: $50
Angua von Überwald | Average: $120
Susan Sto Helit | Last Contribution: $175
Susan Sto Helit | Average: $175
Tiffany Aching | Last Contribution: $150
Tiffany Aching | Average: $117.667
Sacharissa Cripslock | Last Contribution: $175
Sacharissa Cripslock | Average: $175
Frodo Baggins | Last Contribution: $75
Frodo Baggins | Average: $135
Tom Bombadil | Last Contribution: $175
Tom Bombadil | Average: $175
Peregrin $135
Peregrin Took | Last Contribution: $135
Peregrin Took | Average: $93.3333
Samwise Gamgee | Last Contribution: $200
Samwise Gamgee | Average: $206
A.A. Milne | Average: $183.333
Antoine de Saint-Exupery | Last Contribution: $175
Antoine de Saint-Exupery | Average: $175
Adalgrim $467
Bandobras $4673
Belladonna $175
Belladonna Took | Last Contribution: $175
Belladonna Took | Average: $259.333
Eglantine $4367
Mirabella $175
Mirabella Took | Last Contribution: $175
Ferumbras $3457
Gerontius $4562
mickey@Ubuntu:~$
```

Explanation: NF is a predefined variable whose value is the number of fields in the current record.

The script uses both colon and space as the delimiters and checks for the presence of a middle name by comparing the `NF` variable. Firstly, it checks for the last name if it is `Took` and if true, prints the first name. Then checks for the last contribution and finally checks for average and prints it.

References

1. <https://marinegeo.github.io/2018-08-10-adding-images-markdown/>
2. <https://stackoverflow.com/questions/8257865/can-field-separator-in-awk-encompass-multiple-characters>
3. <https://www.tecmint.com/write-shell-scripts-in-awk-programming/>
4. <https://www.unix.com/shell-programming-and-scripting/143768-use-awk-print-first-6-characters.html>
5. <https://www.aholdengouveia.name/LinuxAdmin/Awk.html>
6. <https://flylib.com/books/en/4.356.1.52/1/>
7. <https://unix.stackexchange.com/questions/318846/inverse-regex-in-awk>
8. <https://linuxhint.com/nr-built-in-awk>