# Labs - Linux’s advanced notions

Astier Guillaume, Lefebvre Loic, Morit Luca

09/10/2026



## Contents

**Math's commands** **1**

**The flows** **1**

**ReGex / Sed** **2**

Text file . . . . . 2
Grep . . . . . 2
Sed . . . . . 3
Text file . . . . . 3
Advanced ReGex . . . . . 4
Bonus . . . . . 4

## Math's commands

- Perform the following calculation and redirect the stdout to a file named "calculation1.txt" : 10 + 24 - 5 / 11
- Perform the previous calculation with 4 numbers after the decimal point and redirect the output to a file named "calculation2.txt"
- Compare the result
- Perform a sequence of numbers from 1 to 100 with a step of 6 and redirect the output to a file named "sequence1.txt"
- Perform the previous sequence and adapt the output format to separe numbers with ';' and redirect the output on the "sequence1.txt" file (without deleting the previous sequence in the file)

## The flows

- Redirect the stdout of the following command `echo -e "un\ndeux\ntrois\nquatre"` in a file named Count.txt
- Redirect the stdout of the following command `echo -e "cinq\nsix"` at the end of Count.txt file (don't erase the content of the file !)
- Display the content of Count.txt in stdout
- Combine the two SHELL COMMAND cat and wc (redirect stdout of cat command in stdin of wc command) to count the number of lines in Count.txt
- With the SHELL command find, locate Count.txt

- With the precedent command redirect the stdout flow in LocateCountFile.txt file
- List files in a non existing directory and redirect stderr flow in /dev/null to get a clear result
- List files in a non existing directory and redirect stdout and stderr flows in KoList.txt
- Use the SHELL cat command to populate a file named Myletter.txt with input (stdin) from the user and ending by the string "FIN" Example :

```
username@hostname:~$ cat [...]
>Hello,
>comment vas tu
>FIN
```

## ReGex / Sed

### Text file

- Put this content in the /home/isen/regex.txt file.

> Toto loves titi
> or so
> TOTO LOVE TITI
> and again
> TOTO or Mr TOTO loves TITI Ms TITI
> /NAME/ likes bash
> I do not know what to say *To_Delete*
> and you what do you mean?
> /NAME/ toto /NAME/ titi /NAME/
> toto titi toto titi toto titi to-ti

### Grep

- Show lines contains the string "toto" with non case sensitivity
- Show lines that start with '/'
- Show lines that end with '?'
- Show lines that contain capital letters
- Show lines that contain an 'A' OR a 'b'
- Show lines that contain an 'A' AND a 'b'
- Show lines that have 2 't'

### Sed

- Replace the first occurrence in each line of the word TOTO by TITI
- Replace all '/NAME/' with your first name
- Delete all lines that contain '*To_Delete*'
- Add a line containing '*ERROR*' before each line which does not have a capital letter at the beginning
- Replace '?' by '!'
- Replace 'titi' by 'toto' and 'toto' by 'titi'

### Text file

- Put this content in the /home/isen/advancedRegex.txt file.

> Antoine Dupont
>
> Léon Marchand
>
> 0687548965
>
> Jean NEYMAR
>
> BobMarley
>
> 6985478954
>
> Baptiste& Serîn
>
> jean.neymar@gmail.com
>
> vincentlagaffe.outlook/com
>
> 078565321454

> 192.168.0.1

> 456.567.0.1

> 192.168.0.1.2

> d8:43:ae:2e:03:01

> 987:43:ae:2e:03:01

### Advanced ReGex

- Write the regex to check lines that contains Prename and Name (Prename and Name are strings separating with a space, containing only letters and strating with a capital letters)
- Write the regex to check lines that contains a phone number (Phone number is a string containing only 10 numbers and starting with "06" or "07")
- Write the regex to check lines that contains an email address (Email address is a string containing identifier, provider and a domain respectingly separated by a "@" and ".". Exemple : toto83@gmail.com)

### Bonus

- Write the regex to check if a string is a @IP (4 digit separated with '.', each digit is between 0 and 255)
- Write the regex to check if a string is a @MAC (6 digit separated with ':', each digit is coded in hexa)