National Technical University of Ukraine

“Igor Sikorsky Kyiv Polytechnic Institute”

Faculty of Informatics and Computer Science

Department of Information Systems and Technologies

Laboratory work № __4

from the discipline «LINUX »

Subject: « *Linux streams*»

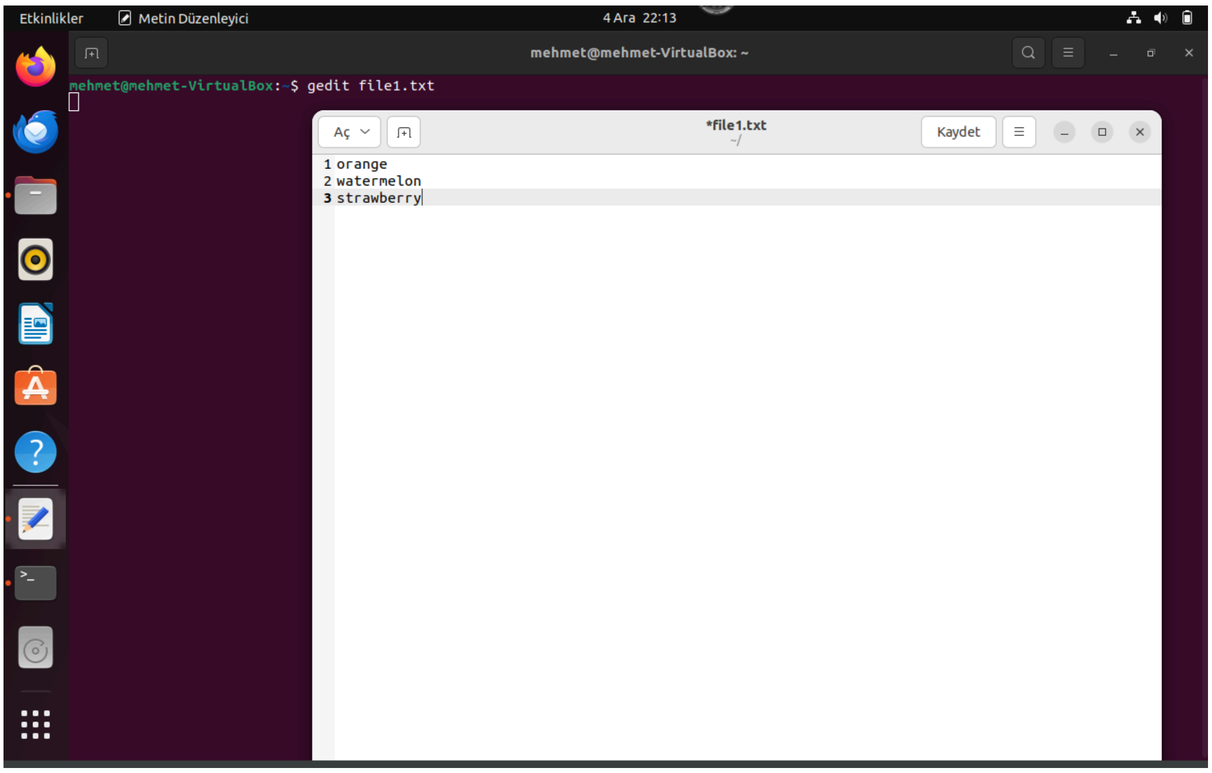
Performed by: student of group IM-14 FIOT Full name Mehmet KULUBECİOGLU	Checked:: Senior lecturer of Department ST Maryna Khmeliuk

KYIV 2023

<u>TASK :</u>

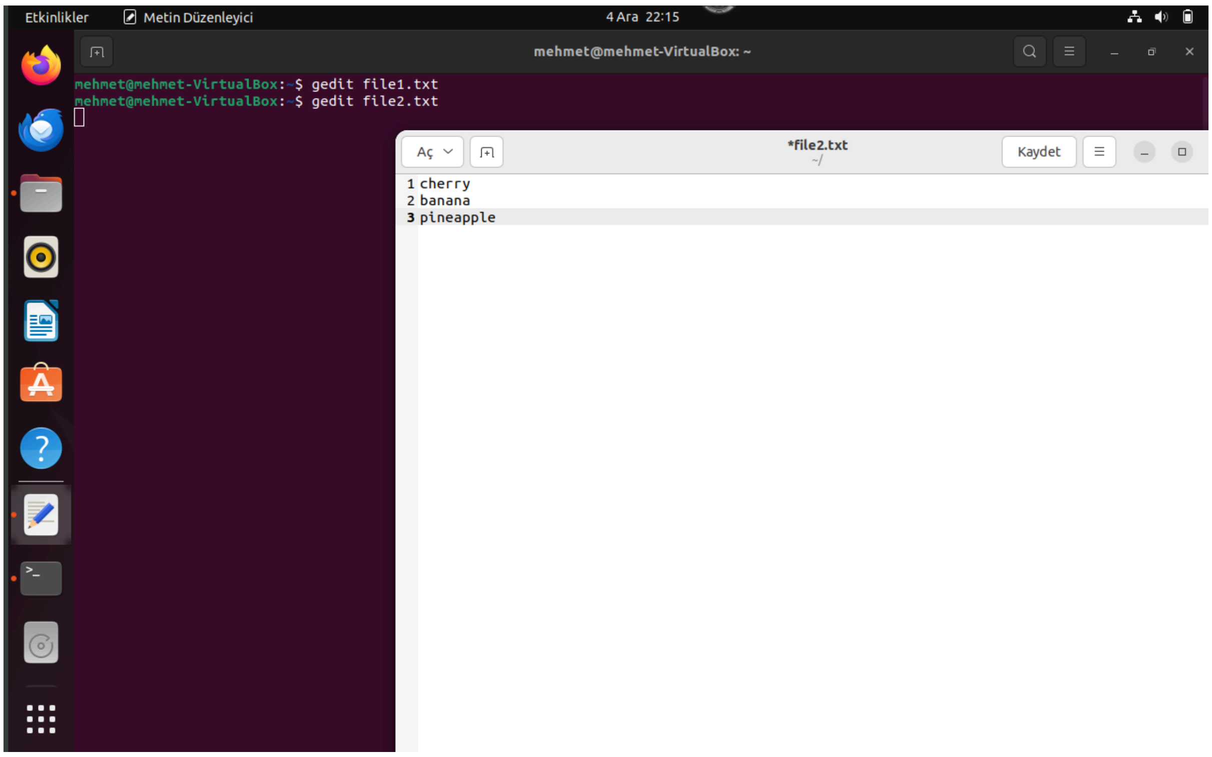
Log in to the system using the user created in the first work. Familiarize yourself with Linux streams (stdin, stdout, stderr), redirecting data to/from streams. Learn to use data redirection using data redirection symbols; grep search command using regular expressions.

Firstly we create and edit file1.txt , file2.txt with its content



```
mehmet@mehmet-VirtualBox:~$ gedit file1.txt
```

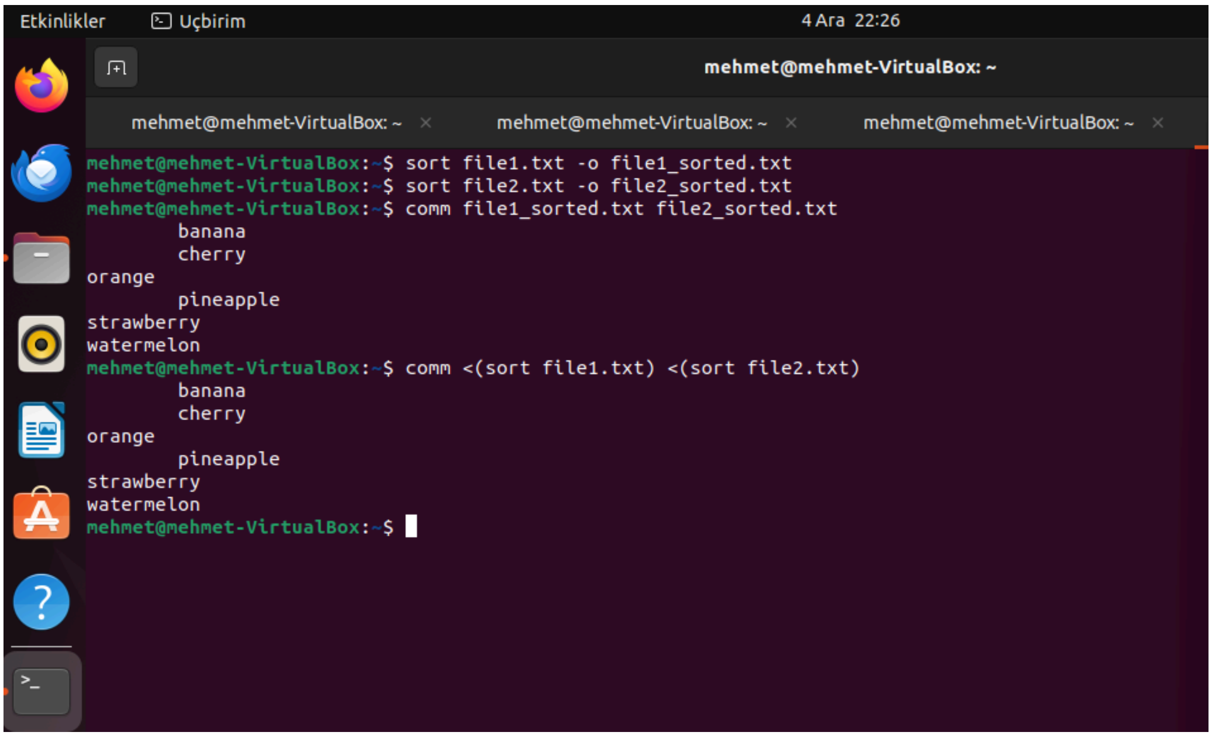
```
orange
watermelon
strawberry
```



```
mehmet@mehmet-VirtualBox:~$ gedit file1.txt
mehmet@mehmet-VirtualBox:~$ gedit file2.txt
```

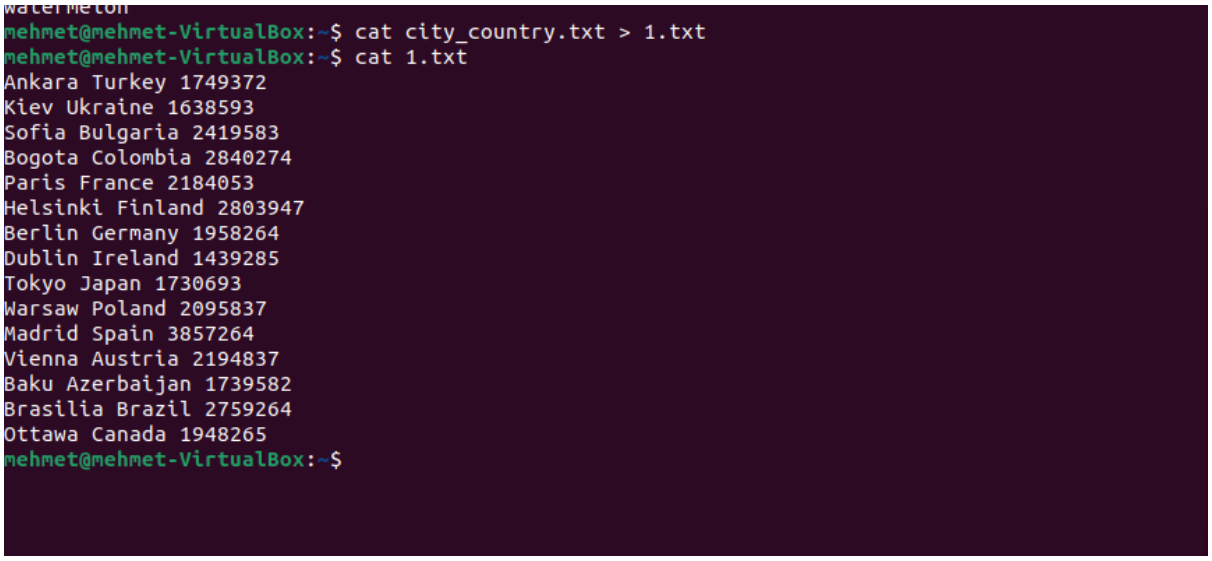
```
cherry
banana
pineapple
```

COMM : We use this command to compare two lines as we see here. We also use sort to sort out the lines of the files.


Etkinlikler
Uçbirim
4 Ara 22:26
mehmet@mehmet-VirtualBox: ~
mehmet@mehmet-VirtualBox: ~
mehmet@mehmet-VirtualBox: ~
mehmet@mehmet-VirtualBox: ~

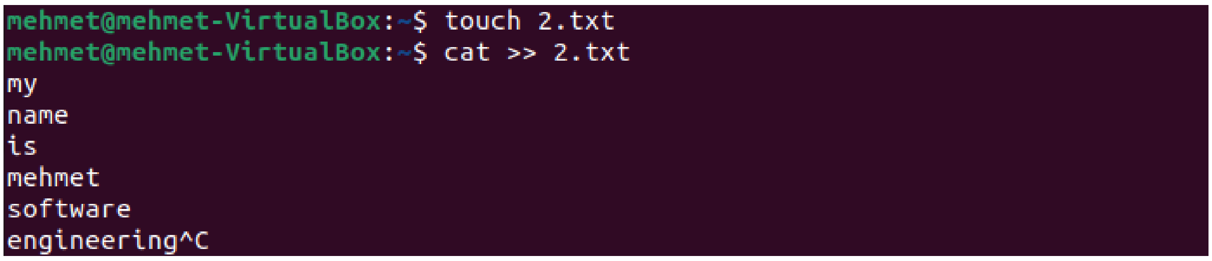
```
mehmet@mehmet-VirtualBox:~$ sort file1.txt -o file1_sorted.txt
mehmet@mehmet-VirtualBox:~$ sort file2.txt -o file2_sorted.txt
mehmet@mehmet-VirtualBox:~$ comm file1_sorted.txt file2_sorted.txt
        banana
        cherry
orange
        pineapple
strawberry
watermelon
mehmet@mehmet-VirtualBox:~$ comm <(sort file1.txt) <(sort file2.txt)
        banana
        cherry
orange
        pineapple
strawberry
watermelon
mehmet@mehmet-VirtualBox:~$ 
```

> : We use Stdin here to load the concatenation of city_country.txt to 1.txt, we see the results.

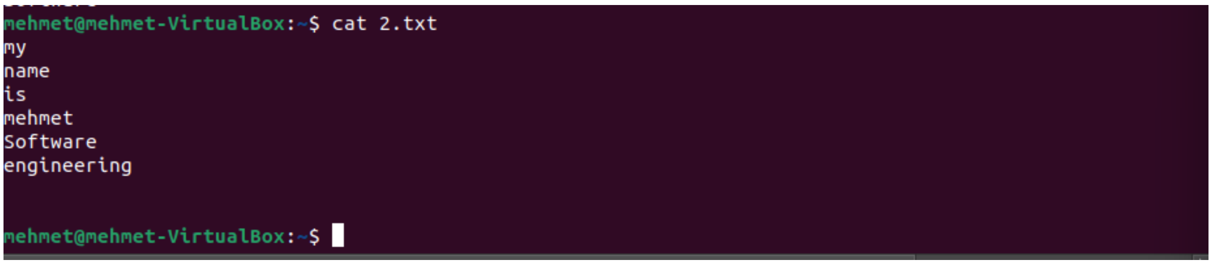


```
watermelon
mehmet@mehmet-VirtualBox:~$ cat city_country.txt > 1.txt
mehmet@mehmet-VirtualBox:~$ cat 1.txt
Ankara Turkey 1749372
Kiev Ukraine 1638593
Sofia Bulgaria 2419583
Bogota Colombia 2840274
Paris France 2184053
Helsinki Finland 2803947
Berlin Germany 1958264
Dublin Ireland 1439285
Tokyo Japan 1730693
Warsaw Poland 2095837
Madrid Spain 3857264
Vienna Austria 2194837
Baku Azerbaijan 1739582
Brasilia Brazil 2759264
Ottawa Canada 1948265
mehmet@mehmet-VirtualBox:~$
```

Here we use '>>' to create a new file that contains the data from the standard output stream. If the specified file exists, it is appended.

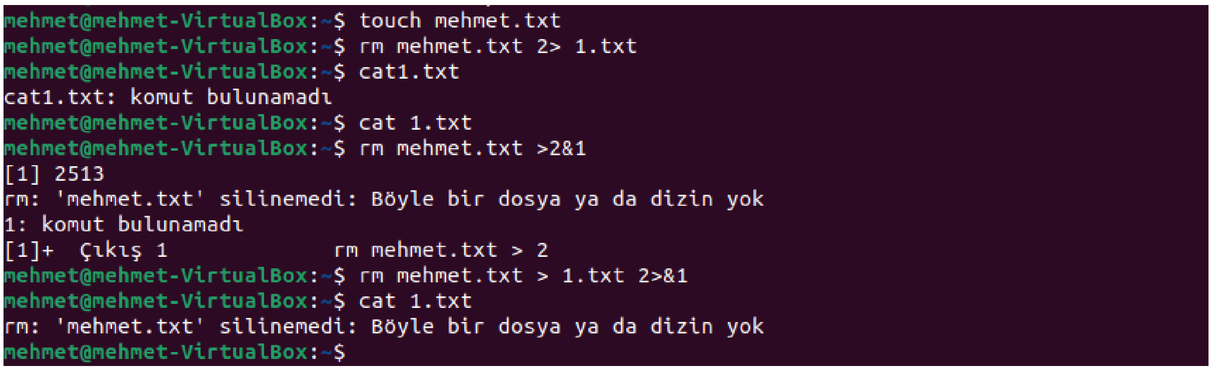


```
mehmet@mehmet-VirtualBox:~$ touch 2.txt
mehmet@mehmet-VirtualBox:~$ cat >> 2.txt
my
name
is
mehmet
software
engineering^C
```



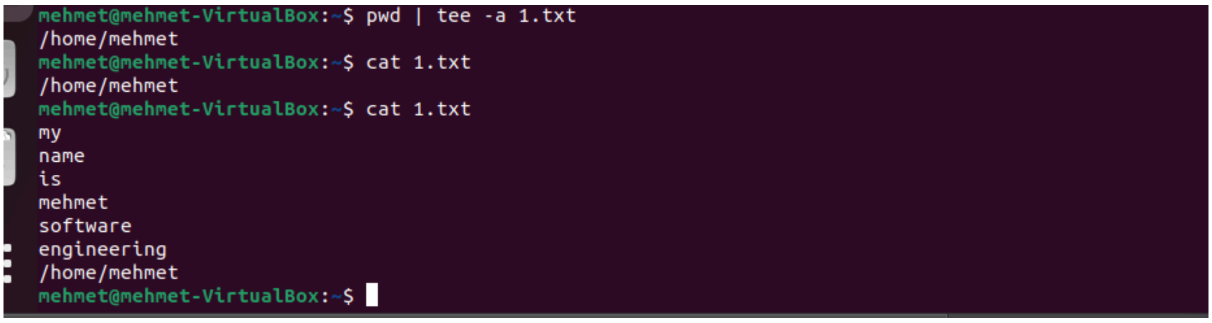
```
mehmet@mehmet-VirtualBox:~$ cat 2.txt
my
name
is
mehmet
Software
engineering


mehmet@mehmet-VirtualBox:~$ █
```

Here we redirect the stderr of ‘rm’ to1.txt and in the 2nd next line redirect the stdout of ‘rm’ to 1.txt & stderr to stdout.



```
mehmet@mehmet-VirtualBox:~$ touch mehmet.txt
mehmet@mehmet-VirtualBox:~$ rm mehmet.txt 2> 1.txt
mehmet@mehmet-VirtualBox:~$ cat1.txt
cat1.txt: komut bulunamadı
mehmet@mehmet-VirtualBox:~$ cat 1.txt
mehmet@mehmet-VirtualBox:~$ rm mehmet.txt >2&1
[1] 2513
rm: 'mehmet.txt' silinemedi: Böyle bir dosya ya da dizin yok
1: komut bulunamadı
[1]+  Çıkış 1                 rm mehmet.txt > 2
mehmet@mehmet-VirtualBox:~$ rm mehmet.txt > 1.txt 2>&1
mehmet@mehmet-VirtualBox:~$ cat 1.txt
rm: 'mehmet.txt' silinemedi: Böyle bir dosya ya da dizin yok
mehmet@mehmet-VirtualBox:~$
```

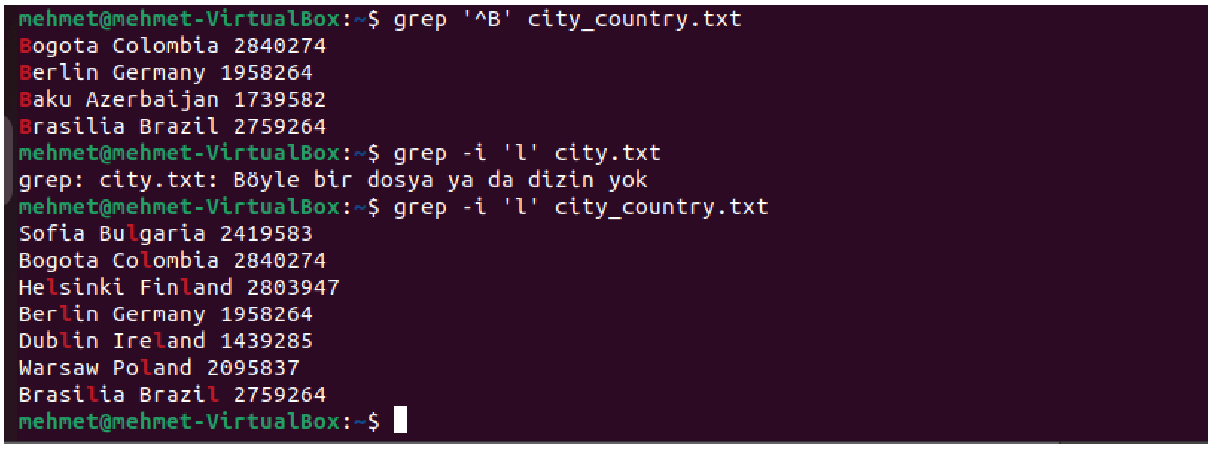
We use ‘tee’ command to save output and pipe to another command



```
mehmet@mehmet-VirtualBox:~$ pwd | tee -a 1.txt
/home/mehmet
mehmet@mehmet-VirtualBox:~$ cat 1.txt
/home/mehmet
mehmet@mehmet-VirtualBox:~$ cat 1.txt
my
name
is
mehmet
software
engineering
/home/mehmet
mehmet@mehmet-VirtualBox:~$
```

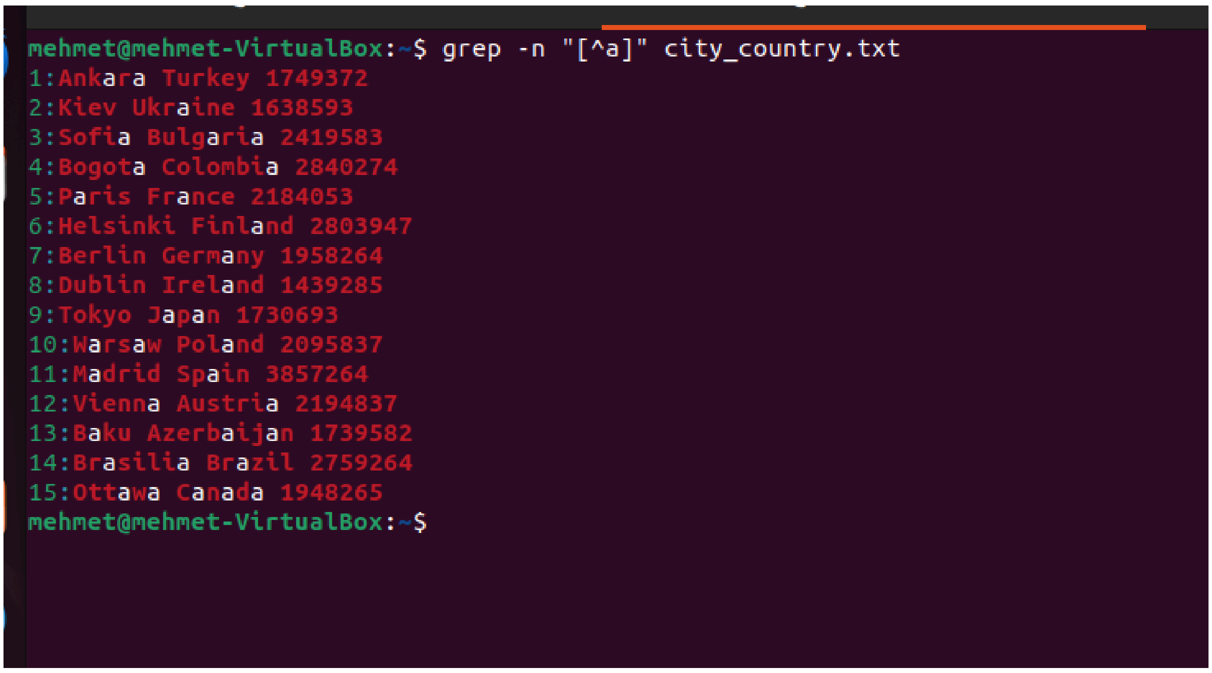
Here we use grep for pattern matching.
First, we use ‘^B’ to search for strings starting with ‘B’ from city.txt

Second, we search for ‘l’ strings with case insensitivity ‘-i’ from city.txt



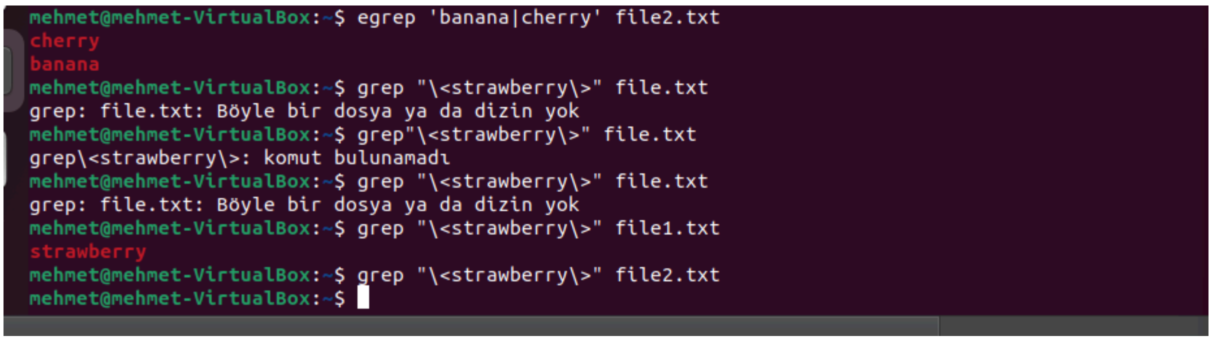
```
mehmet@mehmet-VirtualBox:~$ grep '^B' city_country.txt
Bogota Colombia 2840274
Berlin Germany 1958264
Baku Azerbaijan 1739582
Brasilia Brazil 2759264
mehmet@mehmet-VirtualBox:~$ grep -i 'l' city.txt
grep: city.txt: Böyle bir dosya ya da dizin yok
mehmet@mehmet-VirtualBox:~$ grep -i 'l' city_country.txt
Sofia Bulgaria 2419583
Bogota Colombia 2840274
Helsinki Finland 2803947
Berlin Germany 1958264
Dublin Ireland 1439285
Warsaw Poland 2095837
Brasilia Brazil 2759264
mehmet@mehmet-VirtualBox:~$
```

Continuing, we search for strings without ‘a’.



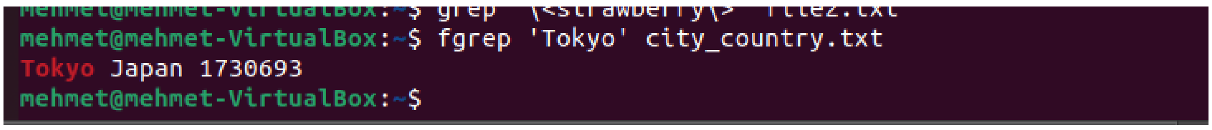
```
mehmet@mehmet-VirtualBox:~$ grep -n "[^a]" city_country.txt
1:Ankara Turkey 1749372
2:Kiev Ukraine 1638593
3:Sofia Bulgaria 2419583
4:Bogota Colombia 2840274
5:Paris France 2184053
6:Helsinki Finland 2803947
7:Berlin Germany 1958264
8:Dublin Ireland 1439285
9:Tokyo Japan 1730693
10:Warsaw Poland 2095837
11:Madrid Spain 3857264
12:Vienna Austria 2194837
13:Baku Azerbaijan 1739582
14:Brasilia Brazil 2759264
15:Ottawa Canada 1948265
mehmet@mehmet-VirtualBox:~$
```

egrep : We use this command here for extended regular expression syntax.Like, here we search for ‘banana’ or ‘cherry’ in file2.txt
Next, we search for ‘strawberry’ by exactly coinciding with the string in file2.txt



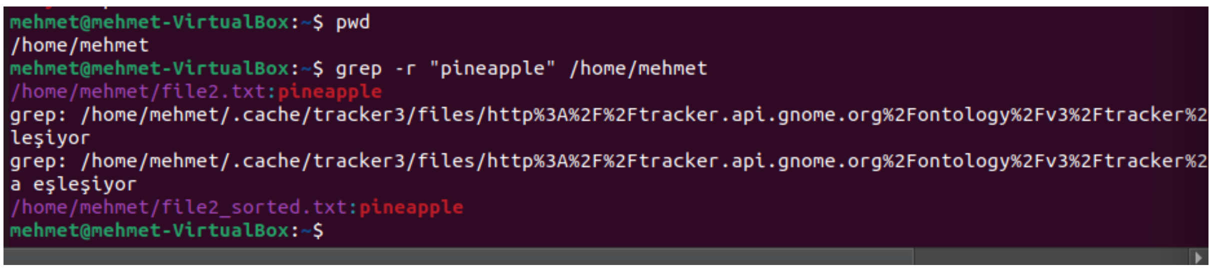
```
mehmet@mehmet-VirtualBox:~$ egrep 'banana|cherry' file2.txt
cherry
banana
mehmet@mehmet-VirtualBox:~$ grep "\<strawberry\>" file.txt
grep: file.txt: Böyle bir dosya ya da dizin yok
mehmet@mehmet-VirtualBox:~$ grep"\<strawberry\>" file.txt
grep\<strawberry\>: komut bulunamadı
mehmet@mehmet-VirtualBox:~$ grep "\<strawberry\>" file.txt
grep: file.txt: Böyle bir dosya ya da dizin yok
mehmet@mehmet-VirtualBox:~$ grep "\<strawberry\>" file1.txt
strawberry
mehmet@mehmet-VirtualBox:~$ grep "\<strawberry\>" file2.txt
mehmet@mehmet-VirtualBox:~$ 
```

fgrep : we use fgrep for fixed string search , like we do search here for “Tokyo”



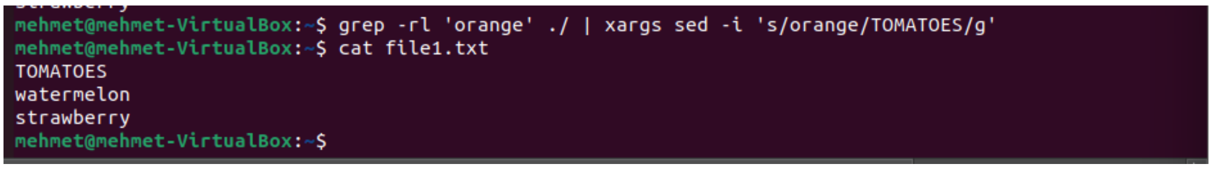
```
mehmet@mehmet-VirtualBox:~$ grep "\<strawberry\>" file2.txt
mehmet@mehmet-VirtualBox:~$ fgrep 'Tokyo' city_country.txt
Tokyo Japan 1730693
mehmet@mehmet-VirtualBox:~$
```

Well again we can see the use case for grep.
Here, we search for “pineapple” string in the directory “/home/mehmet”...we see the results as given - ‘file1.txt’.


```
mehmet@mehmet-VirtualBox:~$ pwd
/home/mehmet
mehmet@mehmet-VirtualBox:~$ grep -r "pineapple" /home/mehmet
/home/mehmet/file2.txt:pineapple
grep: /home/mehmet/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%2
leşiyor
grep: /home/mehmet/.cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%2
a eşleşiyor
/home/mehmet/file2_sorted.txt:pineapple
mehmet@mehmet-VirtualBox:~$
```

Here we change the string1 with string2.
‘rl’ - recursive,display only matching names
‘xargs’ - for arguments
sed - for transforming and filtering
‘s’ - substitution

We change “orange” with “TOMATOES” and see the result…


```
mehmet@mehmet-VirtualBox:~$ grep -rl 'orange' ./ | xargs sed -i 's/orange/TOMATOES/g'
mehmet@mehmet-VirtualBox:~$ cat file1.txt
TOMATOES
watermelon
strawberry
mehmet@mehmet-VirtualBox:~$
```

CONCLUSION:

In this lab work, we explored essential concepts in Linux command-line operations, focusing on stream redirection, regular expressions, and powerful text-processing utilities. We learned about stdin, stdout, and stderr streams, understanding how to redirect and manipulate them using symbols like `>`, `>>`, `2>`, `2>>`, and `<`. We applied these concepts in conjunction with commands like `grep`, `egrep`, `fgrep`, and `sed` for text search and manipulation, employing regular expressions to define complex patterns. The use of `tee` was demonstrated to split and redirect command output. Furthermore, we practiced combining commands, such as recursively searching for files and employing `sed` for in-place text substitution. This lab provided hands-on experience in leveraging fundamental Linux tools for efficient data manipulation and stream management, enhancing our understanding of command-line capabilities for system administration and data processing tasks.