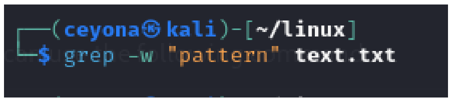
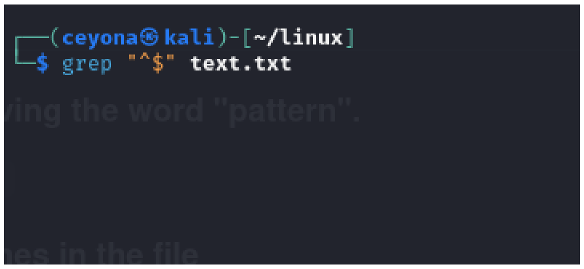
More on Grep

1. Print all the lines having the word "pattern".



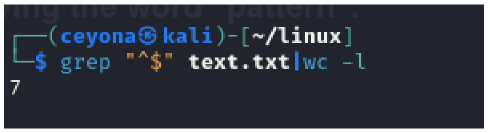
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -w "pattern" text.txt
```

2. Pick out the blank lines in the file



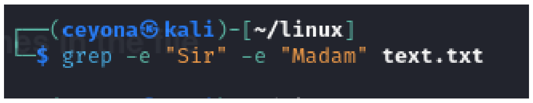
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep "^$" text.txt
```

3. Count total number of empty lines in the file.



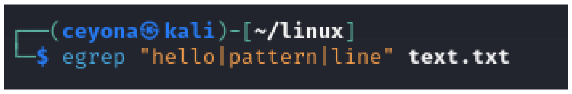
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep "^$" text.txt|wc -l
7
```

4. Print the line which have both “Sir and Madam”.



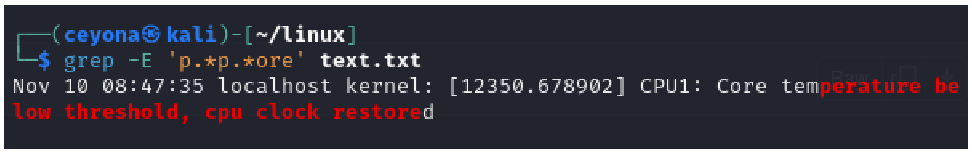
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -e "Sir" -e "Madam" text.txt
```

5. pick out lines with “pattern1” “pattern2” or “pattern3”. (use the alternator |)



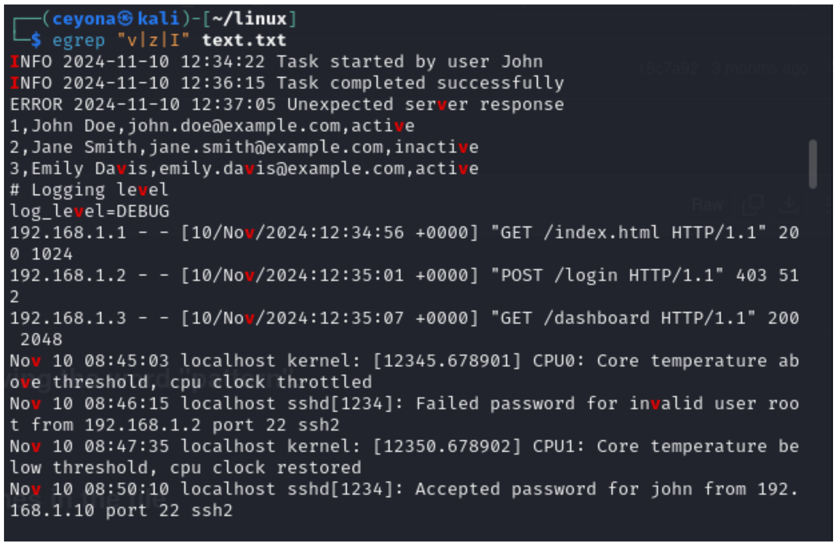
```
┌──(ceyona㉿kali)-[~/linux]
└─$ egrep "hello|pattern|line" text.txt
```

6. pick out lines that have at least two p's followed by any number of letters followed by 'ore'. The p's do not have to be next to each other.



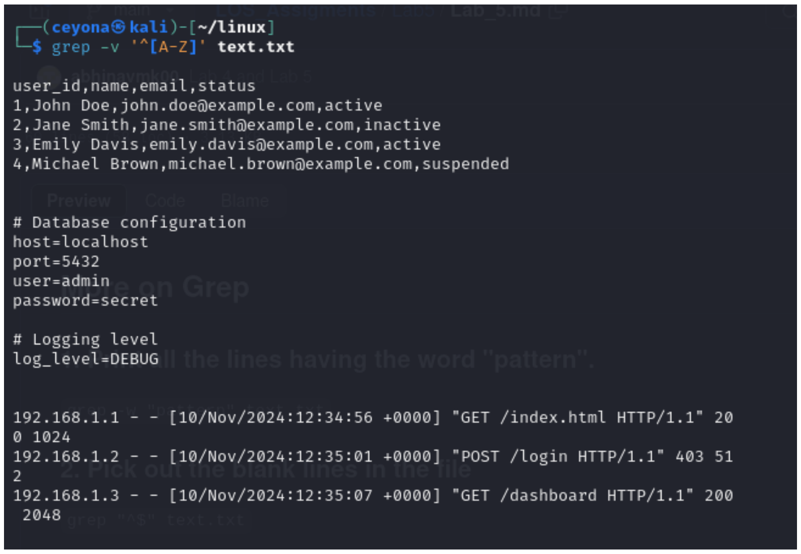
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -E 'p.*p.*ore' text.txt
Nov 10 08:47:35 localhost kernel: [12350.678902] CPU1: Core temperature be
low threshold, cpu clock restored
```

7. pick out all the lines with v, z or I in them



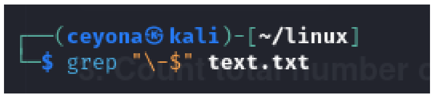
```
┌──(ceyona㉿kali)-[~/linux]
└─$ egrep "v|z|I" text.txt
INFO 2024-11-10 12:34:22 Task started by user John
INFO 2024-11-10 12:36:15 Task completed successfully
ERROR 2024-11-10 12:37:05 Unexpected server response
1,John Doe,john.doe@example.com,active
2,Jane Smith,jane.smith@example.com,inactive
3,Emily Davis,emily.davis@example.com,active
# Logging level
log_level=DEBUG
192.168.1.1 - - [10/Nov/2024:12:34:56 +0000] "GET /index.html HTTP/1.1" 20
0 1024
192.168.1.2 - - [10/Nov/2024:12:35:01 +0000] "POST /login HTTP/1.1" 403 51
2
192.168.1.3 - - [10/Nov/2024:12:35:07 +0000] "GET /dashboard HTTP/1.1" 200
 2048
Nov 10 08:45:03 localhost kernel: [12345.678901] CPU0: Core temperature ab
ove threshold, cpu clock throttled
Nov 10 08:46:15 localhost sshd[1234]: Failed password for invalid user roo
t from 192.168.1.2 port 22 ssh2
Nov 10 08:47:35 localhost kernel: [12350.678902] CPU1: Core temperature be
low threshold, cpu clock restored
Nov 10 08:50:10 localhost sshd[1234]: Accepted password for john from 192.
168.1.10 port 22 ssh2
```

8. pick out all the lines that do not start with an uppercase letter.



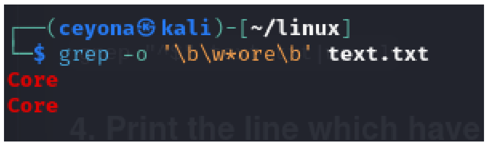
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -v '^[A-Z]' text.txt

user_id,name,email,status
1,John Doe,john.doe@example.com,active
2,Jane Smith,jane.smith@example.com,inactive
3,Emily Davis,emily.davis@example.com,active
4,Michael Brown,michael.brown@example.com,suspended

# Database configuration
host=localhost
port=5432
user=admin
password=secret

# Logging level
log_level=DEBUG


192.168.1.1 - - [10/Nov/2024:12:34:56 +0000] "GET /index.html HTTP/1.1" 20
0 1024
192.168.1.2 - - [10/Nov/2024:12:35:01 +0000] "POST /login HTTP/1.1" 403 51
2
192.168.1.3 - - [10/Nov/2024:12:35:07 +0000] "GET /dashboard HTTP/1.1" 200
 2048
```

9. pick out all the lines that end with a dash –



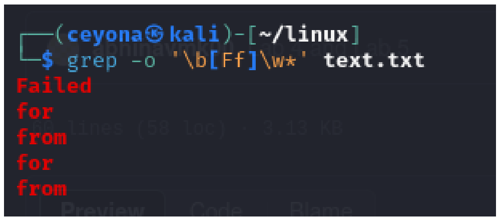
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep "\-$" text.txt
```

10. pick out all the words that end with ore



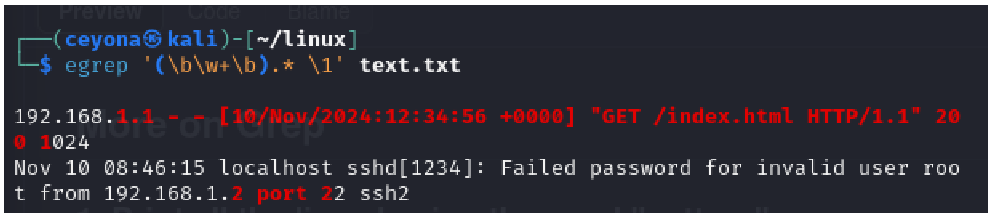
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -o '\b\w*ore\b' text.txt
Core
Core
```

11. pick out all the words that start with f or F



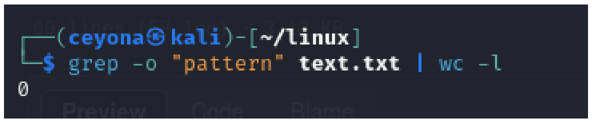
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -o '\b[Ff]\w*' text.txt
Failed
for
from
for
from
```

12. pick out lines that uses first letter alliteration - starting two words with the same letter.



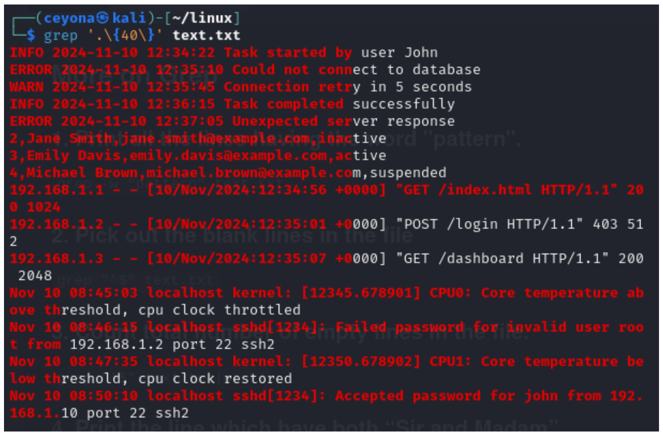
```
┌──(ceyona㉿kali)-[~/linux]
└─$ egrep '(\b\w+\b).* \1' text.txt

192.168.1.1 - - [10/Nov/2024:12:34:56 +0000] "GET /index.html HTTP/1.1" 20
0 1024
Nov 10 08:46:15 localhost sshd[1234]: Failed password for invalid user roo
t from 192.168.1.2 port 22 ssh2
```

13. determine how many times contains the word "pattern".



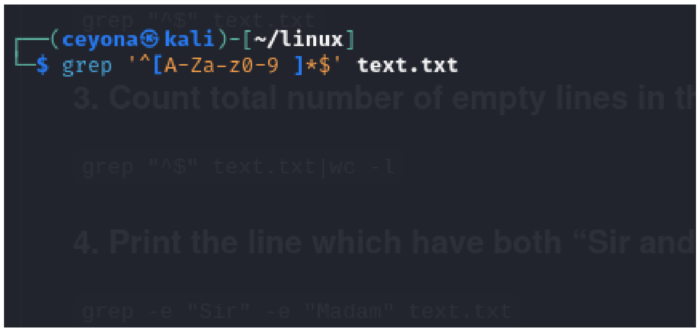
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -o "pattern" text.txt | wc -l
0
```

14. to pick out lines with at least 40 characters:



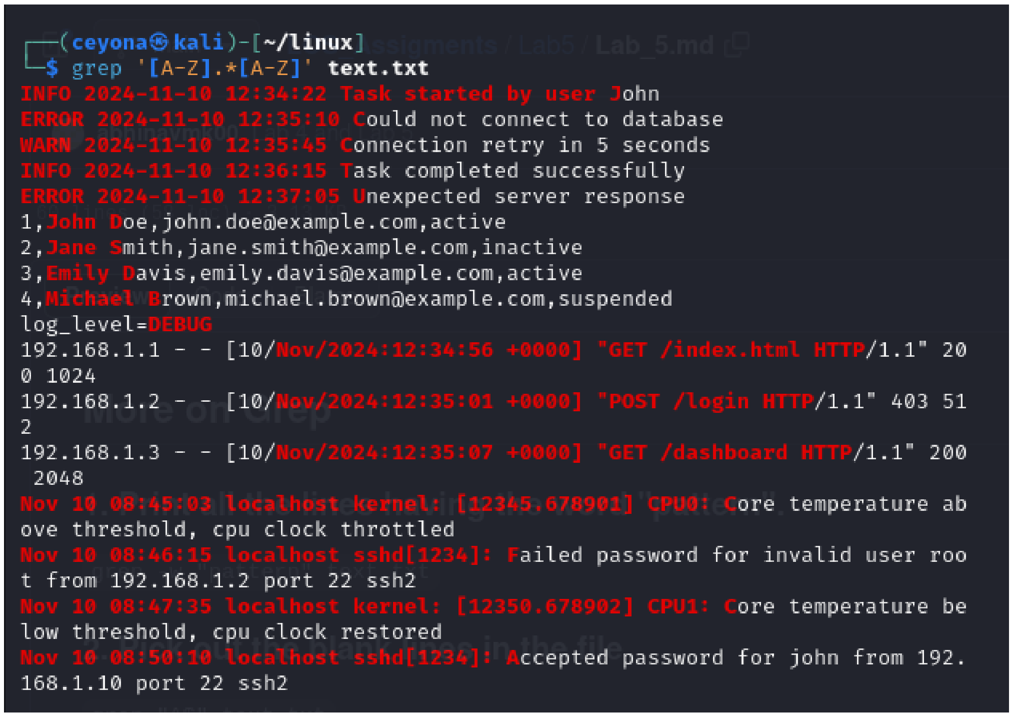
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep '.\{40\}' text.txt
INFO 2024-11-10 12:34:22 Task started by user John
ERROR 2024-11-10 12:35:10 Could not connect to database
WARN 2024-11-10 12:35:45 Connection retry in 5 seconds
INFO 2024-11-10 12:36:15 Task completed successfully
ERROR 2024-11-10 12:37:05 Unexpected server response
2,Jane Smith,jane.smith@example.com,inactive
3,Emily Davis,emily.davis@example.com,active
4,Michael Brown,michael.brown@example.com,suspended
192.168.1.1 - - [10/Nov/2024:12:34:56 +0000] "GET /index.html HTTP/1.1" 20
0 1024
192.168.1.2 - - [10/Nov/2024:12:35:01 +0000] "POST /login HTTP/1.1" 403 51
2
192.168.1.3 - - [10/Nov/2024:12:35:07 +0000] "GET /dashboard HTTP/1.1" 200
 2048
Nov 10 08:45:03 localhost kernel: [12345.678901] CPU0: Core temperature ab
ove threshold, cpu clock throttled
Nov 10 08:46:15 localhost sshd[1234]: Failed password for invalid user roo
t from 192.168.1.2 port 22 ssh2
Nov 10 08:47:35 localhost kernel: [12350.678902] CPU1: Core temperature be
low threshold, cpu clock restored
Nov 10 08:50:10 localhost sshd[1234]: Accepted password for john from 192.
168.1.10 port 22 ssh2
```

15. to pick out lines with no punctuation



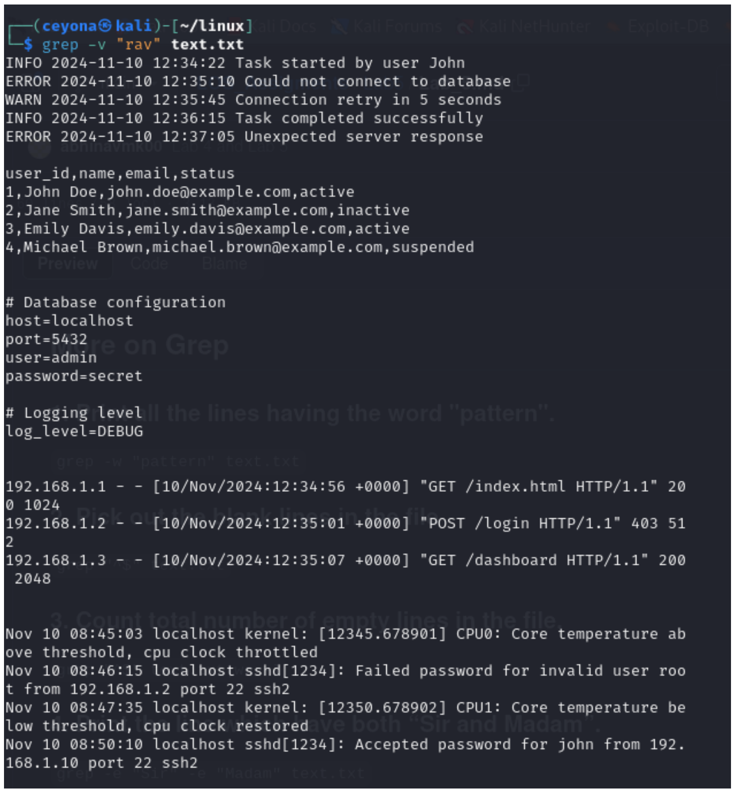
```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep '^[A-Za-z0-9 ]*$' text.txt
```

16. to pick out lines with an uppercase letter other than the first character. (The first character on the line does not count.)



```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep '[A-Z].*[A-Z]' text.txt
INFO 2024-11-10 12:34:22 Task started by user John
ERROR 2024-11-10 12:35:10 Could not connect to database
WARN 2024-11-10 12:35:45 Connection retry in 5 seconds
INFO 2024-11-10 12:36:15 Task completed successfully
ERROR 2024-11-10 12:37:05 Unexpected server response
1,John Doe,john.doe@example.com,active
2,Jane Smith,jane.smith@example.com,inactive
3,Emily Davis,emily.davis@example.com,active
4,Michael Brown,michael.brown@example.com,suspended
log_level=DEBUG
192.168.1.1 - - [10/Nov/2024:12:34:56 +0000] "GET /index.html HTTP/1.1" 20
0 1024
192.168.1.2 - - [10/Nov/2024:12:35:01 +0000] "POST /login HTTP/1.1" 403 51
2
192.168.1.3 - - [10/Nov/2024:12:35:07 +0000] "GET /dashboard HTTP/1.1" 200
 2048
Nov 10 08:45:03 localhost kernel: [12345.678901] CPU0: Core temperature ab
ove threshold, cpu clock throttled
Nov 10 08:46:15 localhost sshd[1234]: Failed password for invalid user roo
t from 192.168.1.2 port 22 ssh2
Nov 10 08:47:35 localhost kernel: [12350.678902] CPU1: Core temperature be
low threshold, cpu clock restored
Nov 10 08:50:10 localhost sshd[1234]: Accepted password for john from 192.
168.1.10 port 22 ssh2
```

17. To pick out lines without rav

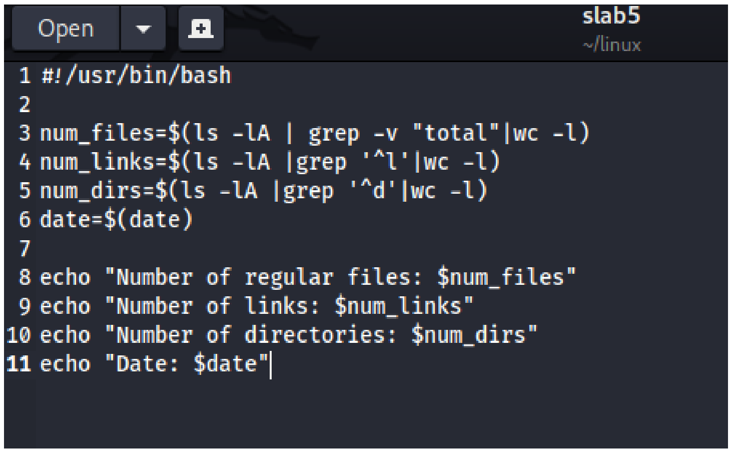


```
┌──(ceyona㉿kali)-[~/linux]
└─$ grep -v "rav" text.txt
INFO 2024-11-10 12:34:22 Task started by user John
ERROR 2024-11-10 12:35:10 Could not connect to database
WARN 2024-11-10 12:35:45 Connection retry in 5 seconds
INFO 2024-11-10 12:36:15 Task completed successfully
ERROR 2024-11-10 12:37:05 Unexpected server response

user_id,name,email,status
1,John Doe,john.doe@example.com,active
2,Jane Smith,jane.smith@example.com,inactive
3,Emily Davis,emily.davis@example.com,active
4,Michael Brown,michael.brown@example.com,suspended


# Database configuration
host=localhost
port=5432
user=admin
password=secret

# Logging level
log_level=DEBUG


192.168.1.1 - - [10/Nov/2024:12:34:56 +0000] "GET /index.html HTTP/1.1" 20
0 1024
192.168.1.2 - - [10/Nov/2024:12:35:01 +0000] "POST /login HTTP/1.1" 403 51
2
192.168.1.3 - - [10/Nov/2024:12:35:07 +0000] "GET /dashboard HTTP/1.1" 200
 2048


Nov 10 08:45:03 localhost kernel: [12345.678901] CPU0: Core temperature ab
ove threshold, cpu clock throttled
Nov 10 08:46:15 localhost sshd[1234]: Failed password for invalid user roo
t from 192.168.1.2 port 22 ssh2
Nov 10 08:47:35 localhost kernel: [12350.678902] CPU1: Core temperature be
low threshold, cpu clock restored
Nov 10 08:50:10 localhost sshd[1234]: Accepted password for john from 192.
168.1.10 port 22 ssh2
```

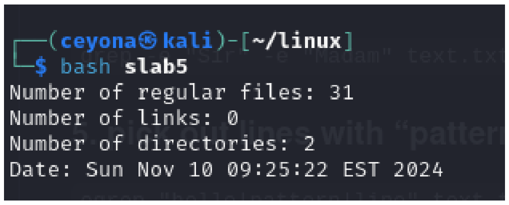
Quotes:

18. Write a shell script to generate a report with the following details.

- Number of regular files
- Number of links
- Number of directories
- Print the date when it was processed!

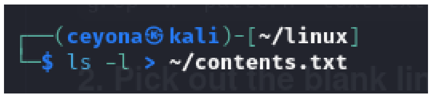


```
#!/usr/bin/bash

num_files=$(ls -lA | grep -v "total"|wc -l)
num_links=$(ls -lA |grep '^l'|wc -l)
num_dirs=$(ls -lA |grep '^d'|wc -l)
date=$(date)

echo "Number of regular files: $num_files"
echo "Number of links: $num_links"
echo "Number of directories: $num_dirs"
echo "Date: $date"
```

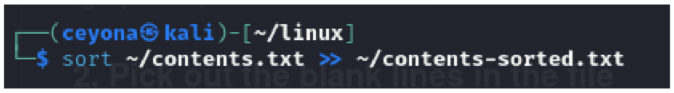

```
┌──(ceyona㉿kali)-[~/linux]
└─$ bash slab5
Number of regular files: 31
Number of links: 0
Number of directories: 2
Date: Sun Nov 10 09:25:22 EST 2024
```

Redirection

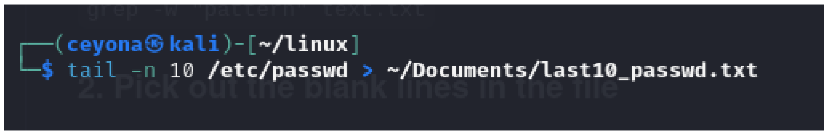
18.
19. List the contents of your current directory, including the ownership and permissions, and store the output to a file called contents.txt within your home directory.


```
┌──(ceyona㉿kali)-[~/linux]
└─$ ls -l > ~/contents.txt
```

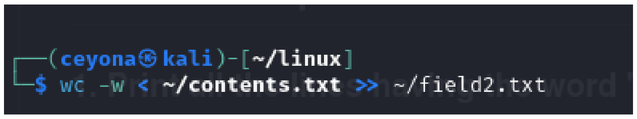
20. Sort the contents of the contents.txt file from your current directory and append it to the end of a new file named contents-sorted.txt.


```
┌──(ceyona㉿kali)-[~/linux]
└─$ sort ~/contents.txt >> ~/contents-sorted.txt
```

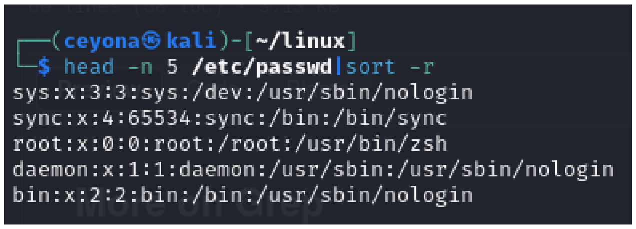
21. Display the last 10 lines of the /etc/passwd file and redirect it to a new file in the your user's Documents directory.


```
┌──(ceyona㉿kali)-[~/linux]
└─$ tail -n 10 /etc/passwd > ~/Documents/last10_passwd.txt
```

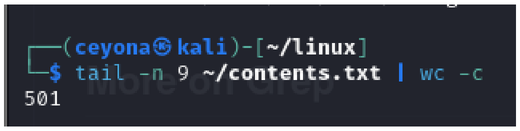
22. Count the number of words within the contents.txt file and append the output to the end of a file field2.txt in your home directory. You will need to use both input and output redirection.


```
┌──(ceyona㉿kali)-[~/linux]
└─$ wc -w < ~/contents.txt >> ~/field2.txt
```

23. Display the first 5 lines of the /etc/passwd file and sort the output reverse alphabetically.

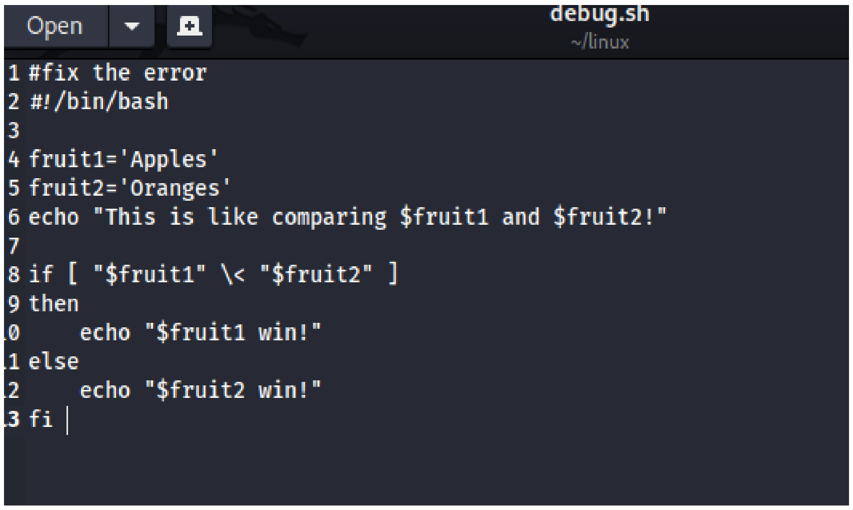

```
┌──(ceyona㉿kali)-[~/linux]
└─$ head -n 5 /etc/passwd|sort -r
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

24. Using the previously created contents.txt file, count the number of characters of the last 9 lines.

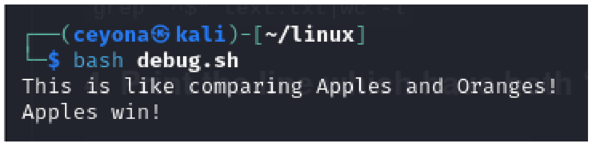

```
┌──(ceyona㉿kali)-[~/linux]
└─$ tail -n 9 ~/contents.txt | wc -c
501
```

Debug

25. Debug the script 1_debug.sh



```
#fix the error
#!/bin/bash

fruit1='Apples'
fruit2='Oranges'
echo "This is like comparing $fruit1 and $fruit2!"

if [ "$fruit1" \< "$fruit2" ]
then
    echo "$fruit1 win!"
else
    echo "$fruit2 win!"
fi
```



```
┌──(ceyona㉿kali)-[~/linux]
└─$ bash debug.sh
This is like comparing Apples and Oranges!
Apples win!
```

Success is no accident. It is hard work, perseverance, learning, studying, sacrifice and most of all, love of what you are doing or learning to do.