

More on Grep

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

```
(kali㉿kali)-[~]  
$ grep 'pattern' file.txt
```

The **pattern** is complex but manageable.
Every **pattern** should be tested thoroughly.
Here are some **patterns**: **pattern1**, **pattern2**, **pattern3**.
Hope you find these **patterns** useful.

2. Pick out the blank lines in the file

```
(kali㉿kali)-[~]
$ grep '^$' file.txt
```

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

```
(kali㉿kali)-[~]  
$ grep -c '^$' file.txt
```

4. Print the line which have both "Sir and Madam".

```
(kali㉿kali)-[~]
$ grep -i "Sir" file.txt | grep "Madam"
```

assist? **Madam**, could you also help? Sir
 Sir **Madam**

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

```
(kali㉿kali)-[~]  
$ grep -E "pattern1|pattern2|pattern3" file.txt
```

Here are some patterns: **pattern1, pattern2, pattern3.**

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.

```
(kali㉿kali)-[~]
$ grep -E "p.*p.*ore" file.txt
Two peas in a pod. Three p's before the core.
```

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

```
(kali㉿kali)-[~]
$ grep "[vziI]" file.txt
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.
Patterns are essential for understanding the task.
The pattern is complex but manageable.
Every pattern should be tested thoroughly.
Sometimes you need a bit of help. Sir, could you assist? Madam, could you also help? Sir
Hope you find these patterns useful.
sir
Sir
Sir Madam
Two peas in a pod. Three p's before the core.
This line ends with a dash -
This line does not end with anything special
Sir Forget the details. Focus on the core.
sir Finally, some lines do not match the requirements.
```

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

```
(kali㉿kali)-[~]
$ grep "^[^A-Z]" file.txt
assist? Madam, could you also help? Sir
madam
sir
sir Finally, some lines do not match the requirements.
```

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

```
(kali㉿kali)-[~]
$ grep -- "-$" file.txt
Sir Madam -
This line ends with a dash -
This line does not end with anything special -
```

10. pick out all the words that end with ore

```
(kali㉿kali)-[~]
$ grep -o "\b\w*ore\b" file.txt
before
core
core
```

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

```
(kali@kali)-[~]
$ grep -o "\b[Ff]\w*" file.txt

for
find
Forget
Focus
Finally
```

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

```
(kali@kali)-[~]
$ grep -E '^(\b([A-Za-z]))\w*\b \1\w*' file.txt

MAdam Madam
This This line ends with a dash -
```

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

```
(kali@kali)-[~]
$ grep -o -i 'pattern' file.txt | wc -l

8
```

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

```
(kali@kali)-[~]
$ awk 'length($0) >= 40' file.txt

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.
Patterns are essential for understanding the task.
Every pattern should be tested thoroughly.
Here are some patterns: pattern1, pattern2, pattern3.
Sometimes you need a bit of help. Sir, could you
Two peas in a pod. Three p's before the core.
This this line does not end with anything special -
Sir Forget the details. Focus on the core.
sir Finally, some lines do not match the requirements.
```

15. to pick out lines with no punctuation

```
(kali@kali)-[~]
$ grep -v '[:punct:]' file.txt

madam
sir
SIR
MAdam Madam
Mad
```

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

```
(kali@kali)-[~]
└─$ grep '^A-Z][A-Z]' file.txt
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.
Sometimes you need a bit of help. Sir, could you
assist? Madam, could you also help? Sir
Sir Madam -
MAdam Madam
Two peas in a pod. Three p's before the core.
This This line ends with a dash -
Sir Forget the details. Focus on the core.
sir Finally, some lines do not match the requirements.
```

17. To pick out lines without rav

```
(kali@kali)-[~]
└─$ grep -v 'rav' file.txt
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.
Patterns are essential for understanding the task.
The pattern is complex but manageable.

Every pattern should be tested thoroughly.
Here are some patterns: pattern1, pattern2, pattern3.
Sometimes you need a bit of help. Sir, could you

assist? Madam, could you also help? Sir
Hope you find these patterns useful.

madam
sir
SIR
Sir Madam -
MAdam Madam
Mad

Two peas in a pod. Three p's before the core.
This This line ends with a dash -
This this line does not end with anything special -
Sir Forget the details. Focus on the core.

sir Finally, some lines do not match the requirements.
```

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!

```
1 #!/bin/bash
2
3 reg_files=$(find . -type f | wc -l)
4
5 links=$(find . -type l | wc -l)
6
7 dirs=$(find . -type d | wc -l)
8
9 date=$(date)
10
11 echo "Report generated on $date"
12 echo "Number of regular files: $reg_files"
13 echo "Number of links: $links"
14 echo "Number of directories: $dirs"
15
```



```

(kali㉿kali)-[~]
$ gedit script1.sh

(kali㉿kali)-[~]
$ chmod +x script1.sh

(kali㉿kali)-[~]
$ ./script1.sh
Report generated on Sun Sep  1 09:24:37 AM EDT 2024
Number of regular files: 6238
Number of links: 5
Number of directories: 385

```

Redirection

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.

```

(kali㉿kali)-[~]
$ ls -l > ~/contents.txt

(kali㉿kali)-[~]
$ cat contents.txt
total 1680
-rw-rw-r-- 1 kali kali    78 Aug  6 10:09 10.bash
-rwxrw-rw- 1 kali kali 1028743 Aug 19 10:45 241059027.zip
-rwxrw-rw- 1 kali kali  524042 Aug 19 11:56 241059028.zip
-rw-rw-r-- 1 kali kali    52 Aug 13 09:05 anu1.txt
-rwx--x--x 1 kali kali   214 Aug 11 12:18 anusha
drwxrwxr-x 2 kali kali  4096 Aug 21 13:49 anushapatil
-rw-rw-r-- 1 kali kali    94 Aug  5 14:54 anusha.sh
-rw-rw-r-- 1 kali kali   855 Aug 14 09:46 assingment2.py
drwxrwxr-x 2 kali kali  4096 Aug 22 11:05 bhavana
-rw-rw-r-- 1 kali kali    53 Aug 13 09:02 bhavana.py
-rw-rw-r-- 1 kali kali    50 Aug 13 09:39 bhavana.txt
-rw-rw-r-- 1 kali kali  2219 Sep  1 09:28 contents-sorted.tx
-rw-rw-r-- 1 kali kali     0 Sep  1 09:30 contents.txt
-rw-rw-r-- 1 kali kali   251 Aug 22 10:53 content.txt
-rwxrwxr-x 1 kali kali   121 Aug 22 09:15 count.sh
-rw-rw-r-- 1 kali kali    98 Aug 22 09:08 count.txt
drwxrwxr-x 3 kali kali  4096 Aug 11 06:51 CYS
drwxr-xr-x 2 kali kali  4096 Aug  4 12:56 Desktop
drwxr-xr-x 2 kali kali  4096 Aug  4 12:56 Documents
drwxr-xr-x 2 kali kali  4096 Aug 30 12:36 Downloads
-rw-rw-r-- 1 kali kali     0 Aug  5 21:23 file
-rw-rw-r-- 1 kali kali     1 Aug  5 16:06 file1.json
-rw-rw-r-- 1 kali kali    49 Aug  5 14:57 file1.sh
-rw-rw-r-- 1 kali kali    19 Aug 20 09:02 file1.txt
-rw-rw-r-- 1 kali kali   404 Aug 22 11:58 file.gz
-rw-rw-r-- 1 kali kali  2219 Sep  1 09:25 file.txt
-rw-rw-r-- 1 kali kali    26 Aug  5 09:46 hello.sh
drwxrwxr-x 5 kali kali  4096 Aug 11 03:18 LS
drwxr-xr-x 2 kali kali  4096 Aug  4 12:56 Music
-rw-rw-r-- 1 kali kali   666 Aug 21 14:44 outputFile.txt
drwxr-xr-x 2 kali kali  4096 Aug 22 09:43 Pictures
-rwxrwxr-x 1 kali kali   666 Aug 21 14:10 poem.txt
drwxr-xr-x 2 kali kali  4096 Aug  4 12:56 Public
-rw-rw-r-- 1 kali kali  4206 Aug 11 13:08 python1.py
-rw-rw-r-- 1 kali kali  4667 Aug 20 09:02 python2.py
-rw-rw-r-- 1 kali kali  2315 Aug 21 12:53 python3.py
-rw-rw-r-- 1 kali kali     0 Aug 21 14:50 redirection.sh
-rwxrwxr-x 1 kali kali   380 Sep  1 09:20 script1.sh
-rwxrwxr-x 1 kali kali   141 Aug 22 09:42 script_patt.sh
-rwxrwxr-x 1 kali kali   274 Aug 22 09:51 search_patt.sh

```

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.

```
(kali@kali)-[~]
$ sort ~/contents.txt >> ~/contents-sorted.txt

(kali@kali)-[~]
$ cat contents-sorted.txt
drwxrwxr-x 2 kali kali 4096 Aug 11 07:30 work
drwxrwxr-x 2 kali kali 4096 Aug 21 13:49 anushapatil
drwxrwxr-x 2 kali kali 4096 Aug 22 11:05 bhavana
drwxrwxr-x 3 kali kali 4096 Aug 11 06:51 CYS
drwxrwxr-x 5 kali kali 4096 Aug 11 03:18 LS
drwxr-xr-x 2 kali kali 4096 Aug 22 09:43 Pictures
drwxr-xr-x 2 kali kali 4096 Aug 30 12:36 Downloads
drwxr-xr-x 2 kali kali 4096 Aug 4 12:56 Desktop
drwxr-xr-x 2 kali kali 4096 Aug 4 12:56 Documents
drwxr-xr-x 2 kali kali 4096 Aug 4 12:56 Music
drwxr-xr-x 2 kali kali 4096 Aug 4 12:56 Public
drwxr-xr-x 2 kali kali 4096 Aug 4 12:56 Templates
drwxr-xr-x 2 kali kali 4096 Aug 4 12:56 Videos
-rw-rw-r-- 1 kali kali 0 Aug 21 14:50 redirection.sh
-rw-rw-r-- 1 kali kali 0 Aug 5 21:23 file
-rw-rw-r-- 1 kali kali 0 Sep 1 09:25 file.txt
-rw-rw-r-- 1 kali kali 19 Aug 20 09:02 file1.txt
-rw-rw-r-- 1 kali kali 1 Aug 5 16:06 file1.json
-rw-rw-r-- 1 kali kali 2315 Aug 21 12:53 python3.py
-rw-rw-r-- 1 kali kali 251 Aug 22 10:53 content.txt
-rw-rw-r-- 1 kali kali 26 Aug 5 09:46 hello.sh
-rw-rw-r-- 1 kali kali 404 Aug 22 11:58 file.gz
-rw-rw-r-- 1 kali kali 4206 Aug 11 13:08 python1.py
-rw-rw-r-- 1 kali kali 4667 Aug 20 09:02 python2.py
-rw-rw-r-- 1 kali kali 49 Aug 5 14:57 file1.sh
-rw-rw-r-- 1 kali kali 50 Aug 13 09:39 bhavana.txt
-rw-rw-r-- 1 kali kali 52 Aug 13 09:05 anu1.txt
-rw-rw-r-- 1 kali kali 53 Aug 13 09:02 bhavana.py
-rw-rw-r-- 1 kali kali 666 Aug 21 14:44 outputFile.txt
-rw-rw-r-- 1 kali kali 78 Aug 6 10:09 10.bash
-rw-rw-r-- 1 kali kali 855 Aug 14 09:46 assingment2.py
-rw-rw-r-- 1 kali kali 94 Aug 5 14:54 anusha.sh
-rw-rw-r-- 1 kali kali 98 Aug 22 09:08 count.txt
-rwxrwxrwx 1 kali kali 1028743 Aug 19 10:45 241059027.zip
-rwxrwxrwx 1 kali kali 524042 Aug 19 11:56 241059028.zip
-rwxrwxr-x 1 kali kali 121 Aug 22 09:15 count.sh
-rwxrwxr-x 1 kali kali 141 Aug 22 09:42 script_patt.sh
-rwxrwxr-x 1 kali kali 274 Aug 22 09:51 search_patt.sh
-rwxrwxr-x 1 kali kali 380 Sep 1 09:20 script1.sh
-rwxrwxr-x 1 kali kali 666 Aug 21 14:10 poem.txt
-rwx--x--x 1 kali kali 214 Aug 11 12:18 anusha
```

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.

```

(kali㉿kali)-[~]
$ tail -n 10 /etc/passwd > ~/Documents/passwd10.txt

(kali㉿kali)-[~]
$ cat passwd10.txt
cat: passwd10.txt: No such file or directory

(kali㉿kali)-[~]
$ cat Documents/passwd10.txt
_gophish:x:124:130::/var/lib/gophish:/usr/sbin/nologin
_iodine:x:125:65534::/run/iodine:/usr/sbin/nologin
_miredo:x:126:65534::/var/run/miredo:/usr/sbin/nologin
_statd:x:127:65534::/var/lib/nfs:/usr/sbin/nologin
_redis:x:128:131::/var/lib/redis:/usr/sbin/nologin
_postgres:x:129:132:PostgreSQL administrator,,:/var/lib/postgresql:/bin/bash
_mosquitto:x:130:133::/var/lib/mosquitto:/usr/sbin/nologin
_inetsim:x:131:134::/var/lib/inetsim:/usr/sbin/nologin
_gym:x:132:135::/var/lib/openvas:/usr/sbin/nologin
kali:x:1000:1000::,/home/kali:/usr/bin/zsh

```

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.

```

(kali㉿kali)-[~]
$ cat content.txt
total 20
-rwxrwxr-x 1 kali kali 130 Aug 22 10:38 1.sh
-rwxrwxr-x 1 kali kali 117 Aug 22 10:45 2.sh
-rw-rw-r-- 1 kali kali 590 Aug 22 10:51 outputfile.txt
-rw-rw-r-- 1 kali kali 590 Aug 22 10:24 poem.txt

(kali㉿kali)-[~]
$ wc -w < ~/content.txt >> ~/field2.txt

(kali㉿kali)-[~]
$ cat field2.txt
38

```

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

```

(kali㉿kali)-[~]
$ 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.

```

(kali㉿kali)-[~]
$ tail -n 9 ~/content.txt | wc -m
203

```

Debug

25. Debug the script 1_debug.sh

```
1 #fix the error
2 #!/bin/bash
3 fruit1 = Apples
4 fruit2 = Oranges
5 if [ $1 -lt $# ]
6 then
7 echo "This is like comparing $fruit1 and $fruit2!"
8 elif test [$1 -gt $2 ]
9 then
10 echo '$fruit1 win!'
11 else
12 echo "Fruit2 win!"
13 done|
```

```
1 #fix the error
2 #!/bin/bash
3 fruit1="Apples"
4 fruit2="Oranges"
5 if [ "$1" -lt "$#" ]
6 then
7 echo "This is like comparing '$fruit1' and '$fruit2'!"
8 elif [ "$1" -gt "$2" ]
9 then
10 echo "'$fruit1' win!"
11 else
12 echo "'$fruit2' win!"
13 fi
```



```

(kali@kali)-[~]
$ gedit debug.sh

(kali@kali)-[~]
$ bash ./debug.sh
./debug.sh: line 2: /!bin/bash: No such file or directory
./debug.sh: line 3: fruit1: command not found
./debug.sh: line 4: fruit2: command not found
./debug.sh: line 13: syntax error near unexpected token `done'
./debug.sh: line 13: `done'

(kali@kali)-[~]
$ bash ./debug.sh
./debug.sh: line 2: /!bin/bash: No such file or directory
./debug.sh: line 3: fruit1: command not found
./debug.sh: line 4: fruit2: command not found
./debug.sh: line 13: syntax error near unexpected token `done'
./debug.sh: line 13: `done'

(kali@kali)-[~]
$ bash ./debug.sh
./debug.sh: line 3: fruit1: command not found
./debug.sh: line 4: fruit2: command not found
./debug.sh: line 13: syntax error near unexpected token `done'
./debug.sh: line 13: `done'

(kali@kali)-[~]
$ bash ./debug.sh
./debug.sh: line 13: syntax error near unexpected token `done'
./debug.sh: line 13: `done'

(kali@kali)-[~]
$ bash ./debug.sh
./debug.sh: line 14: syntax error: unexpected end of file

(kali@kali)-[~]
$ bash ./debug.sh
./debug.sh: line 5: [: : integer expression expected
./debug.sh: line 8: test: [: integer expression expected
'Oranges' win!

(kali@kali)-[~]
$ bash ./debug.sh 1 2
This is like comparing 'Apples' and 'Oranges'!

```

```

(kali@kali)-[~]
$ bash ./debug.sh 5 3
./debug.sh: line 8: test: [5: integer expression expected
'Oranges' win!

(kali@kali)-[~]
$ bash ./debug.sh
./debug.sh: line 5: [: : integer expression expected
./debug.sh: line 8: test: [: integer expression expected
'Oranges' win!

(kali@kali)-[~]
$ bash ./debug.sh 5 3
./debug.sh: line 8: [5: command not found
'Oranges' win!

(kali@kali)-[~]
$ bash ./debug.sh 5 3
'Apples' win!

(kali@kali)-[~]
$ bash ./debug.sh 5 5
'Oranges' 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.