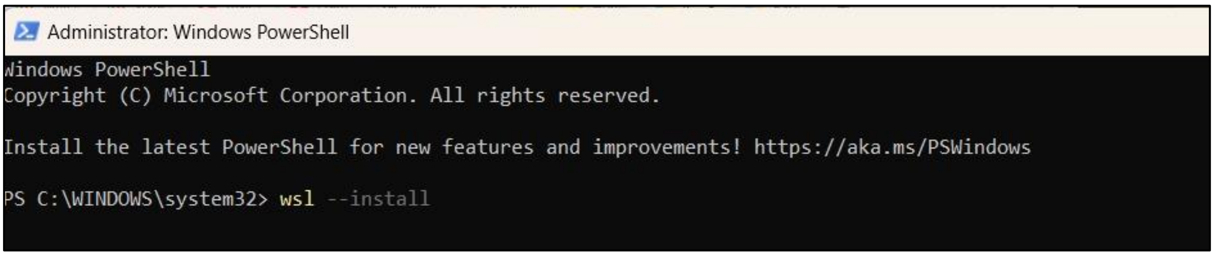
Linux lecture-1 notes:

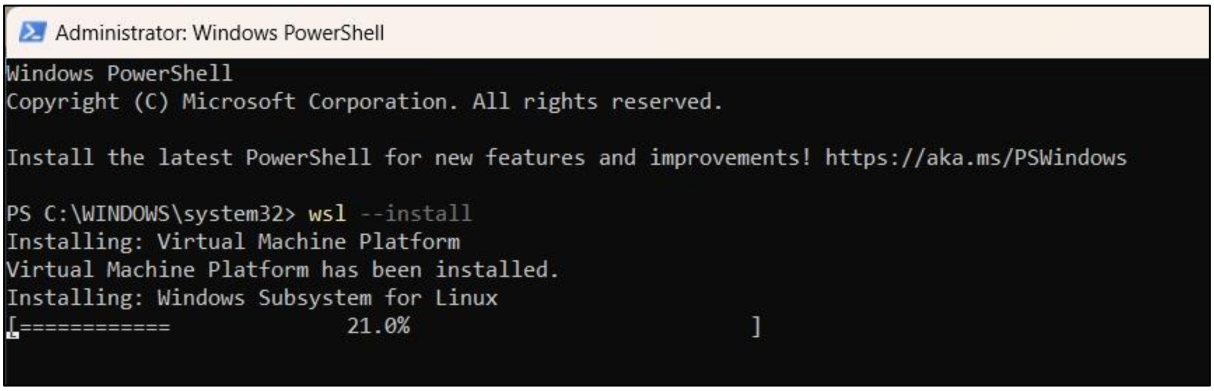
Firstly to install the working environment, we need to follow following steps:

- Open power shell with administrator permissions.
- Then write “wsl --install” . And then press enter.
- And it would start installing ubuntu on your windows.
- And then you need to restart your pc.
- And then u can simply open ubuntu app. And set your username and password. And then you are ready to go.



Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
```



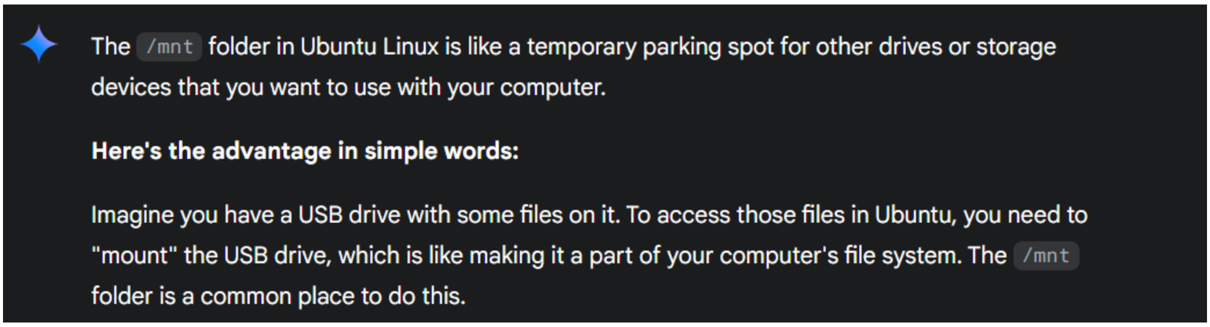
Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
[============               21.0%                          ]
```

And then we can also install sql and other file operations using apt update command (sudo apt update is used for updating package list) like:

```
sudo apt update
sudo apt install mysql-server -y
sudo service mysql start
sudo mysql -u root -p

CREATE DATABASE dbname
```

Now after this, we can go into mnt directory, so that to access every file or folder on our personal computer:

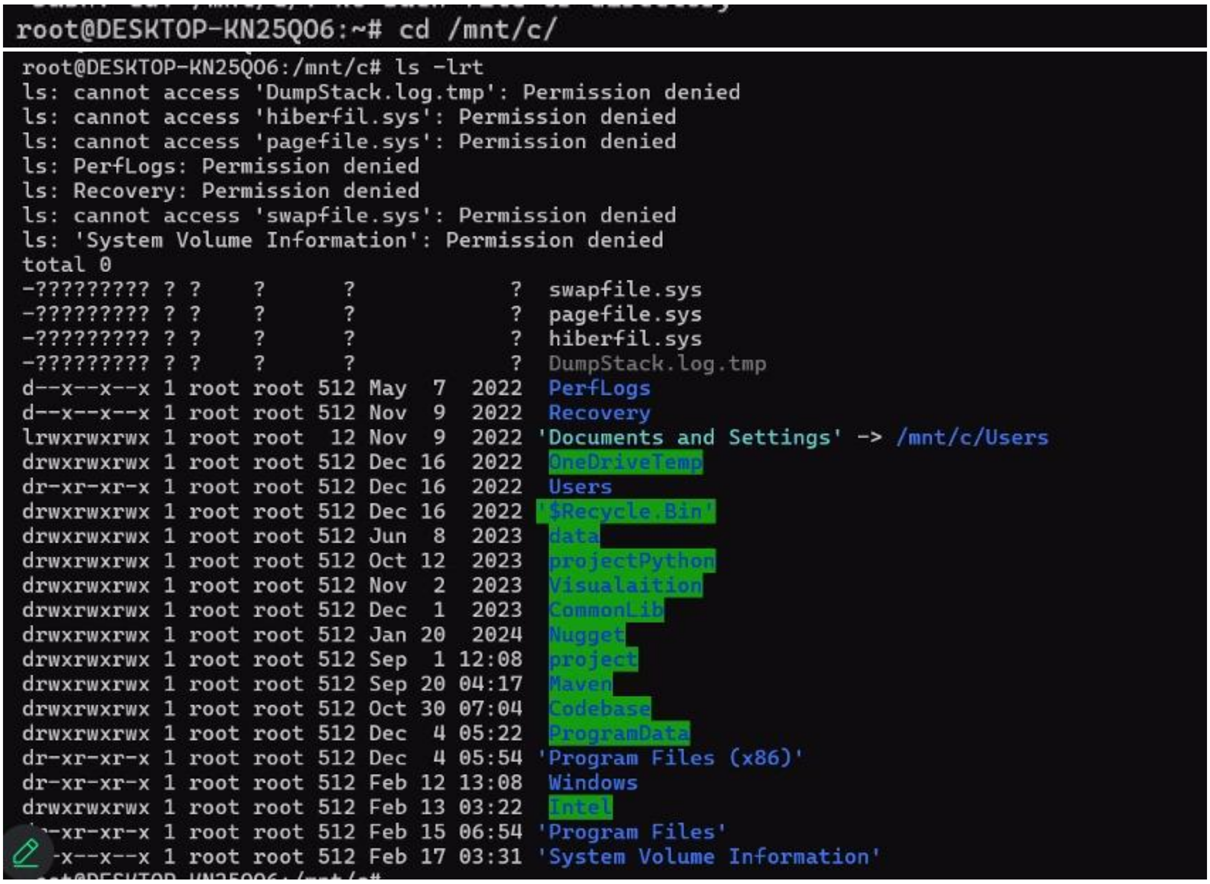


The `/mnt` folder in Ubuntu Linux is like a temporary parking spot for other drives or storage devices that you want to use with your computer.

Here's the advantage in simple words:

Imagine you have a USB drive with some files on it. To access those files in Ubuntu, you need to "mount" the USB drive, which is like making it a part of your computer's file system. The `/mnt` folder is a common place to do this.

Now to actually go to mount directory to see all the files and folder of our windows we use :



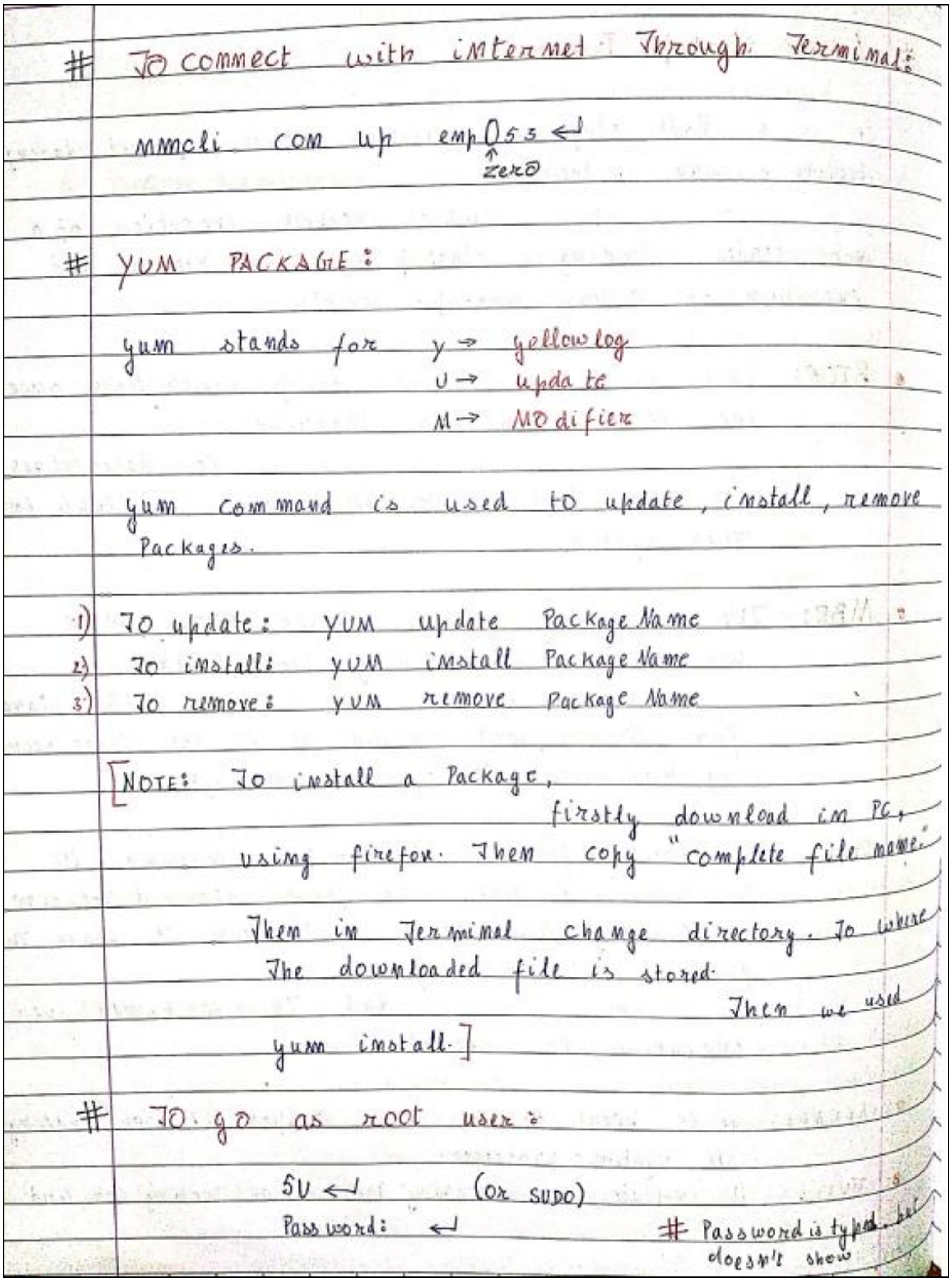
```
root@DESKTOP-KN25Q06:~# cd /mnt/c/
```

```
root@DESKTOP-KN25Q06:/mnt/c# ls -lrt
ls: cannot access 'DumpStack.log.tmp': Permission denied
ls: cannot access 'hiberfil.sys': Permission denied
ls: cannot access 'pagefile.sys': Permission denied
ls: PerfLogs: Permission denied
ls: Recovery: Permission denied
ls: cannot access 'swapfile.sys': Permission denied
ls: 'System Volume Information': Permission denied
total 0
-????????? ? ?    ?      ?            ?  swapfile.sys
-????????? ? ?    ?      ?            ?  pagefile.sys
-????????? ? ?    ?      ?            ?  hiberfil.sys
-????????? ? ?    ?      ?            ?  DumpStack.log.tmp
d--x--x--x 1 root root 512 May  7  2022  PerfLogs
d--x--x--x 1 root root 512 Nov  9  2022  Recovery
lrwxrwxrwx 1 root root  12 Nov  9  2022 'Documents and Settings' -> /mnt/c/Users
drwxrwxrwx 1 root root 512 Dec 16  2022  OneDriveTemp
dr-xr-xr-x 1 root root 512 Dec 16  2022  Users
drwxrwxrwx 1 root root 512 Dec 16  2022 '$Recycle.Bin'
drwxrwxrwx 1 root root 512 Jun  8  2023  data
drwxrwxrwx 1 root root 512 Oct 12  2023  projectPython
drwxrwxrwx 1 root root 512 Nov  2  2023  Visualaition
drwxrwxrwx 1 root root 512 Dec  1  2023  CommonLib
drwxrwxrwx 1 root root 512 Jan 20  2024  Nugget
drwxrwxrwx 1 root root 512 Sep  1 12:08  project
drwxrwxrwx 1 root root 512 Sep 20 04:17  Maven
drwxrwxrwx 1 root root 512 Oct 30 07:04  Codebase
drwxrwxrwx 1 root root 512 Dec  4 05:22  ProgramData
dr-xr-xr-x 1 root root 512 Dec  4 05:54 'Program Files (x86)'
dr-xr-xr-x 1 root root 512 Feb 12 13:08  Windows
drwxrwxrwx 1 root root 512 Feb 13 03:22  Intel
dr-xr-xr-x 1 root root 512 Feb 15 06:54 'Program Files'
d--x--x--x 1 root root 512 Feb 17 03:31 'System Volume Information'
```

Note: ls-lrt is used to return all the files based on modification time. Where the latest file is showed at the last or bottom.

NOTE: whatsoever installations and everything we do are in c folder as wsl was installed their and if we go to d drive and try all this. Then it may generate error and to resolve that we need to install wsl on d drive. As all saves are stored into root directory,

NOW LETS START WITH BASIC LINUX COMMANDS:



To connect with internet Through Terminal:

nmcli con up enp0s3 ↵
zero

YUM PACKAGE:

yum stands for y → yellow dog
u → update
m → Modifier

yum command is used to update, install, remove Packages.

1) To update: yum update Package Name
2) To install: yum install Package Name
3) To remove: yum remove Package Name

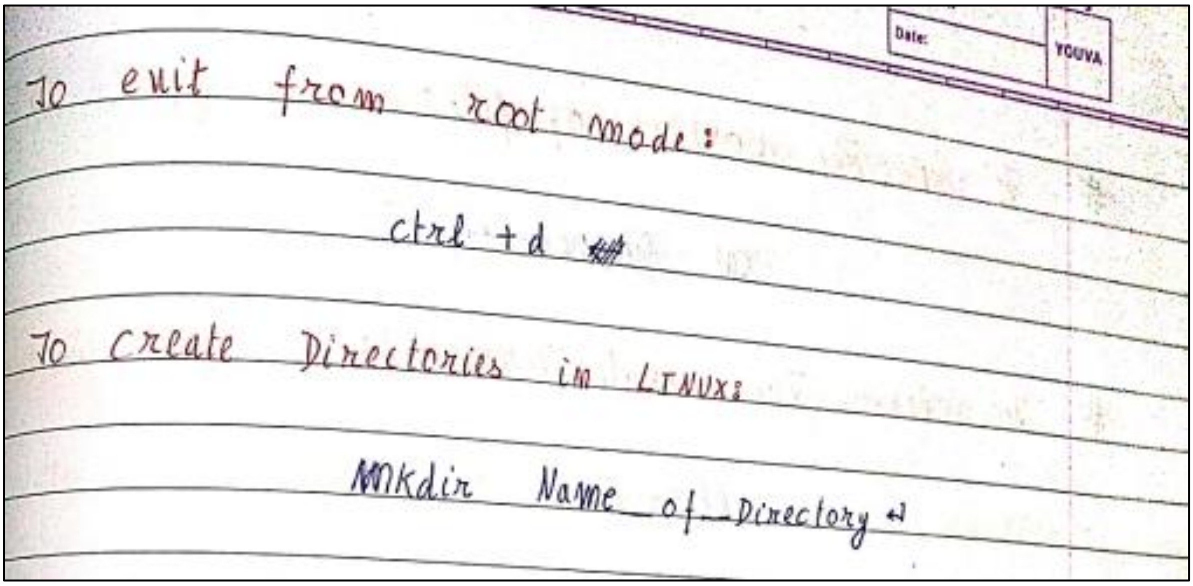
[NOTE: To install a Package, firstly download in PC, using firefox. Then copy "complete file name."

Then in Terminal change directory. To where The downloaded file is stored. Then we used yum install.]

To go as root user:

su ↵ (or sudo)
Password: ↵

Password is typed but doesn't show



Date: YOUVA

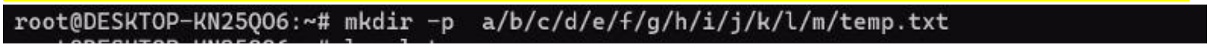
To exit from root mode:

ctrl + d

To create Directories in LINUX:

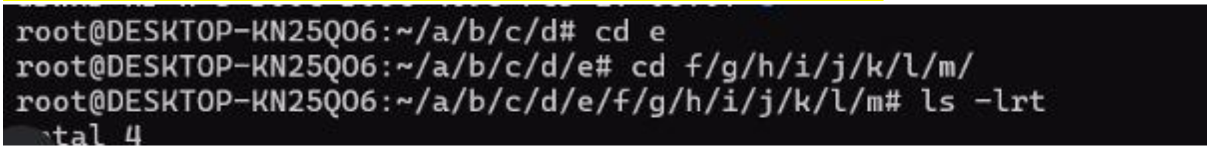
Mkdir Name_of_Directory ↵

And now say if we want to create multiple directories in command:

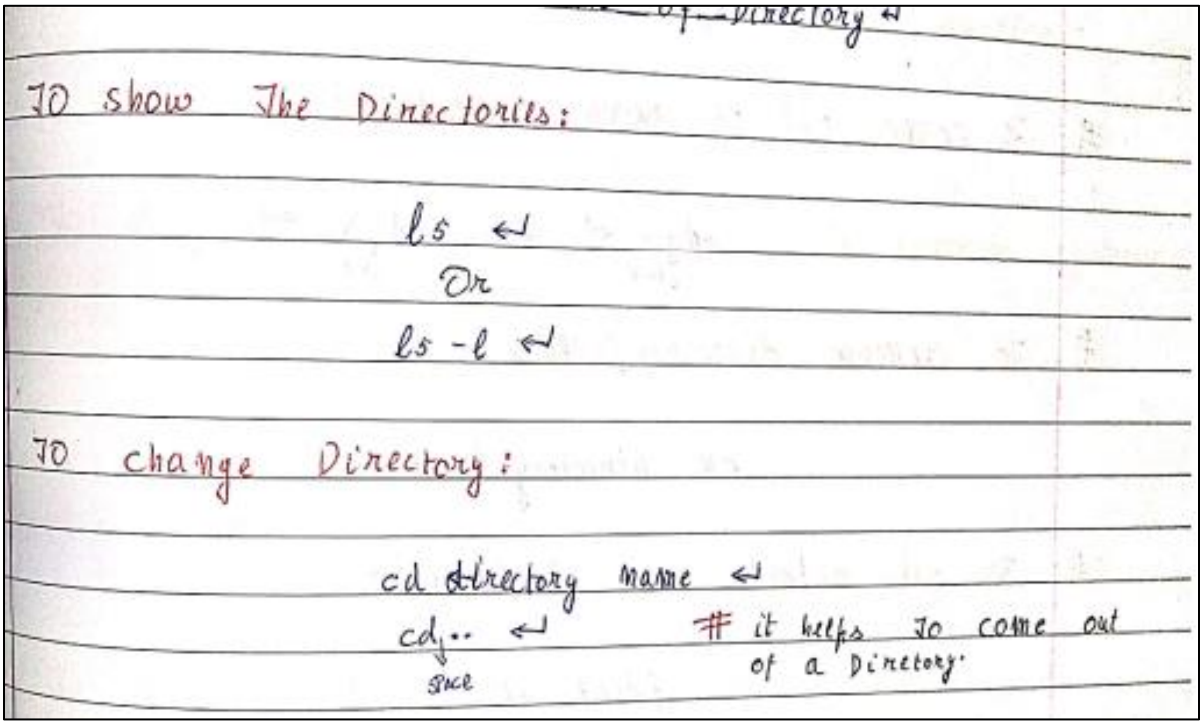


```
root@DESKTOP-KN25Q06:~# mkdir -p  a/b/c/d/e/f/g/h/i/j/k/l/m/temp.txt
```

And we can also verify by traversing through them:



```
root@DESKTOP-KN25Q06:~/a/b/c/d# cd e
root@DESKTOP-KN25Q06:~/a/b/c/d/e# cd f/g/h/i/j/k/l/m/
root@DESKTOP-KN25Q06:~/a/b/c/d/e/f/g/h/i/j/k/l/m# ls -lrt
tal 4
```



of_Directory ↵

To show The Directories:

ls ↵

Or

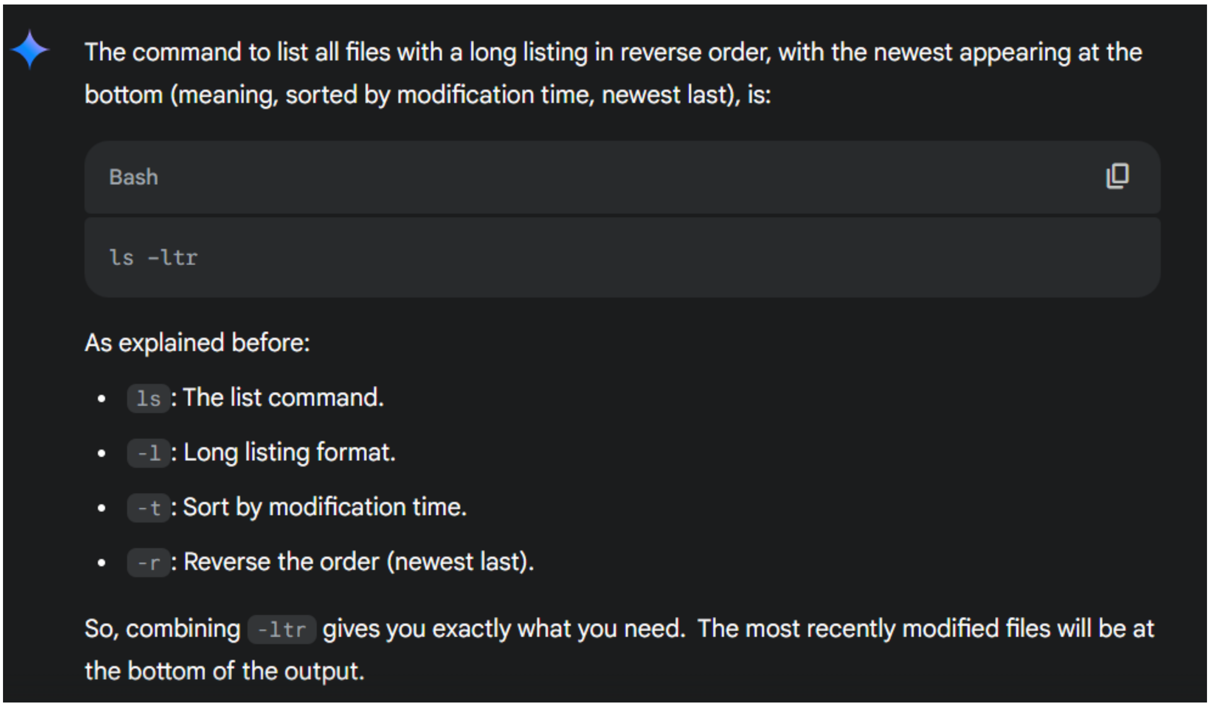
ls -l ↵

To change Directory:

cd directory name ↵

cd.. ↵ # it helps To come out of a Directory.

space



The command to list all files with a long listing in reverse order, with the newest appearing at the bottom (meaning, sorted by modification time, newest last), is:

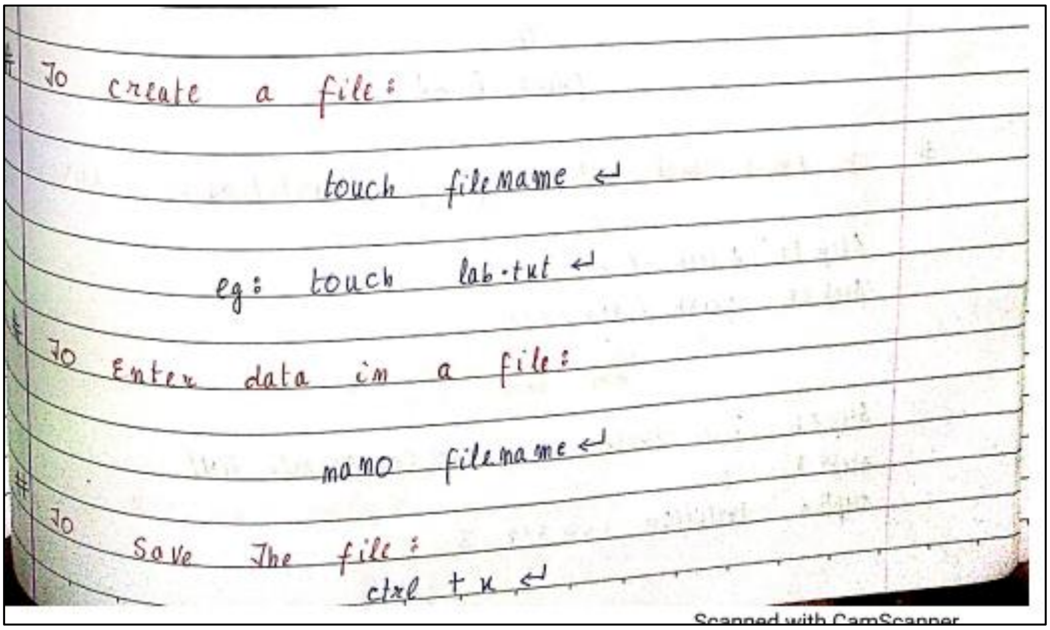
```bash
ls -ltr
```

As explained before:

- `ls`: The list command.
- `-l`: Long listing format.
- `-t`: Sort by modification time.
- `-r`: Reverse the order (newest last).

So, combining `-ltr` gives you exactly what you need. The most recently modified files will be at the bottom of the output.

And now to create a file:



To create a file:

touch filename ↵

eg: touch lab.txt ↵

To Enter data in a file:

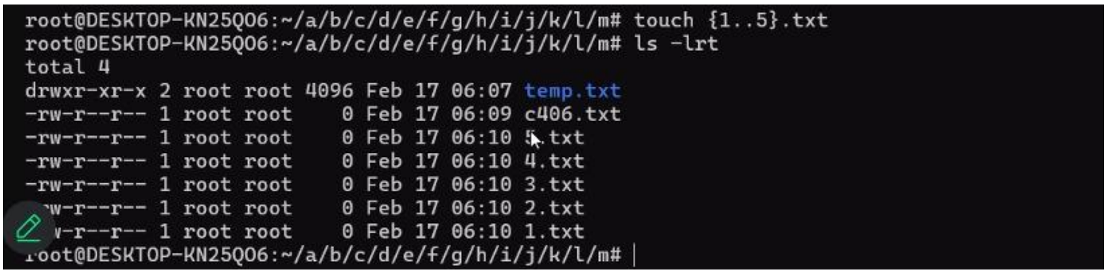
nano filename ↵

To Save the file:

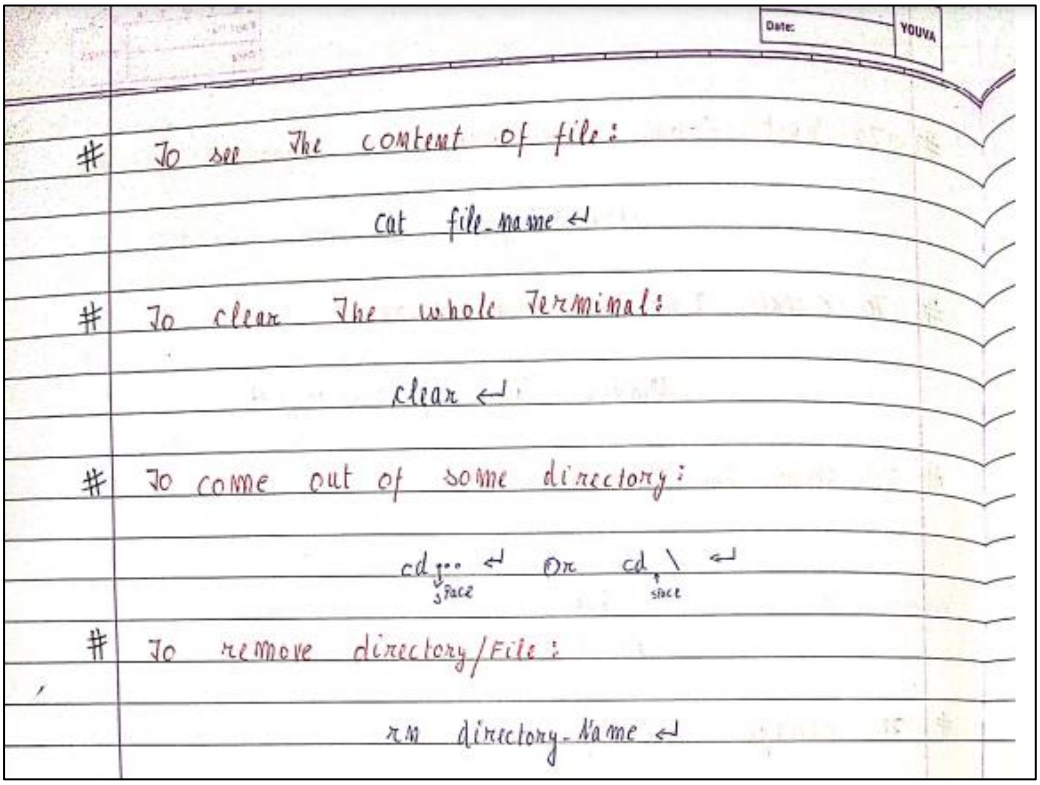
ctrl + x ↵

Scanned with CamScanner

And to create multiple files in one go:



```
root@DESKTOP-KN25Q06:~/a/b/c/d/e/f/g/h/i/j/k/l/m# touch {1..5}.txt
root@DESKTOP-KN25Q06:~/a/b/c/d/e/f/g/h/i/j/k/l/m# ls -lrt
total 4
drwxr-xr-x 2 root root 4096 Feb 17 06:07 temp.txt
-rw-r--r-- 1 root root    0 Feb 17 06:09 c406.txt
-rw-r--r-- 1 root root    0 Feb 17 06:10 5.txt
-rw-r--r-- 1 root root    0 Feb 17 06:10 4.txt
-rw-r--r-- 1 root root    0 Feb 17 06:10 3.txt
-rw-r--r-- 1 root root    0 Feb 17 06:10 2.txt
-rw-r--r-- 1 root root    0 Feb 17 06:10 1.txt
root@DESKTOP-KN25Q06:~/a/b/c/d/e/f/g/h/i/j/k/l/m#
```



# To see The content of file:

cat file_name ↵

# To clear The whole Terminal:

clear ↵

# To come out of some directory:

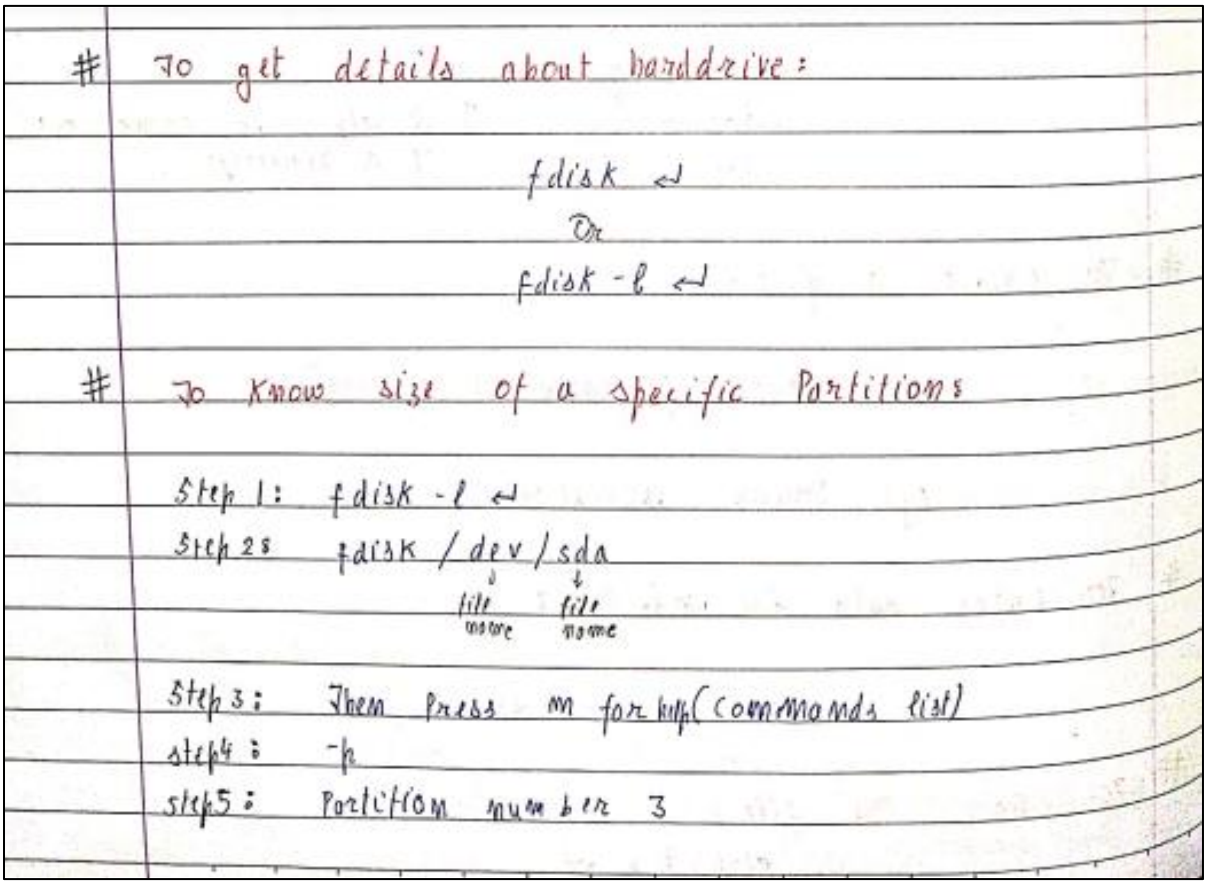
cd .. ↵ Or cd \ ↵

space space

# To remove directory/File:

rm directory_Name ↵

TO GET DETAILS ABOUT HARDDRIVE:



# To get details about harddrive:

fdisk ↵

Or

Fdisk -l ↵

# To Know size of a specific Portitions

Step 1: fdisk -l ↵

Step 2: fdisk /dev/sda

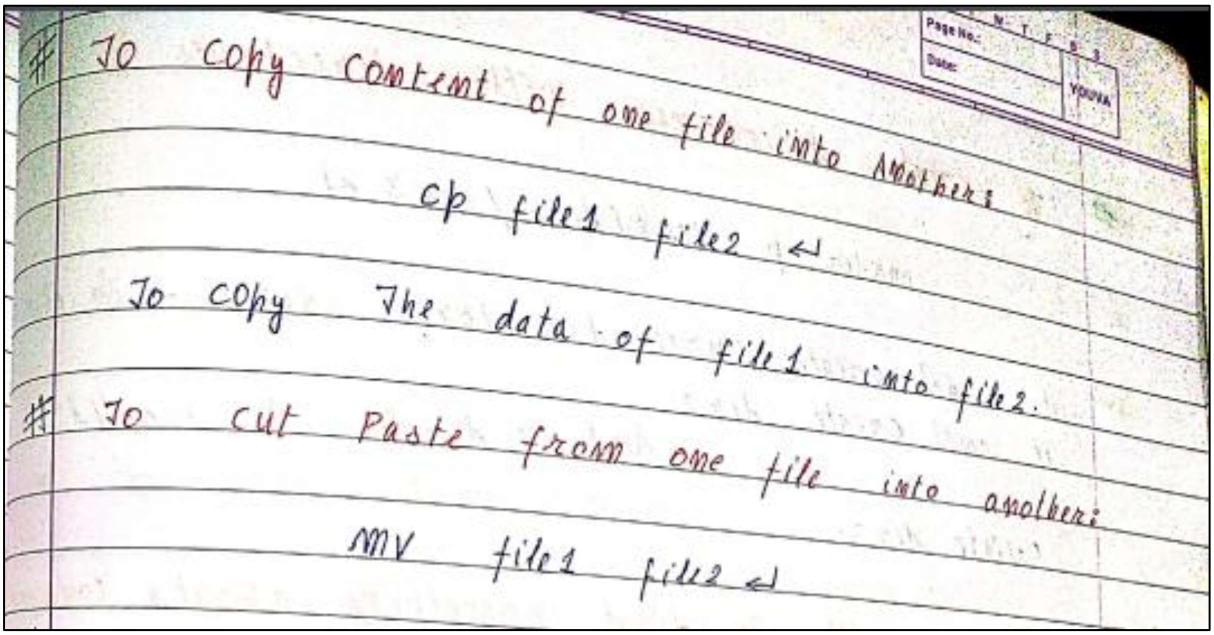
file store file name

Step 3: Them Press m for help(commands list)

step4: -p

step5: Portition number 3

To copy file from one location to other:



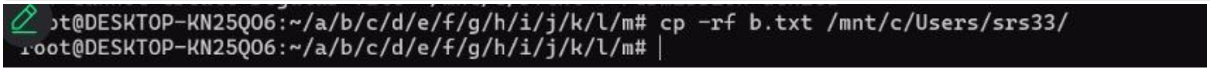
To copy Content of one file into Another:

cp file1 file2 ↵

To copy The data of file1 into file2.

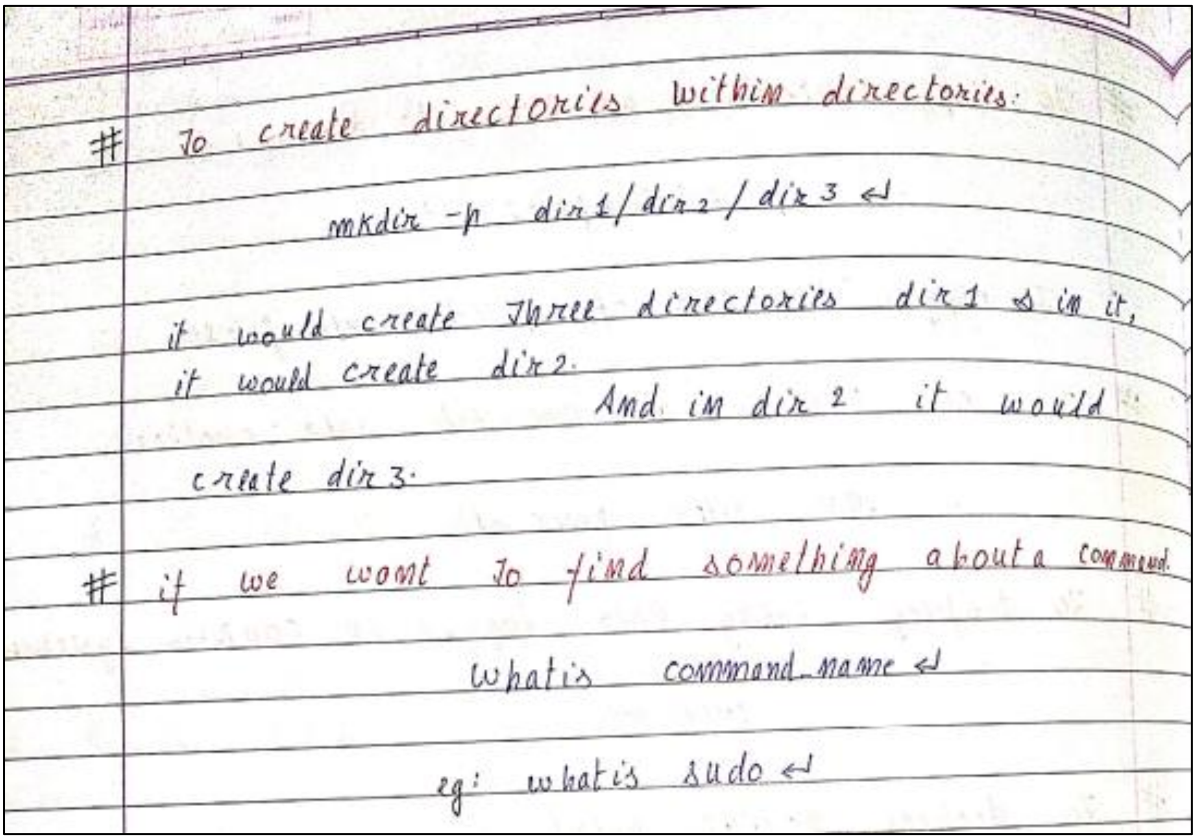
To cut Paste from one file into another:

mv file1 file2 ↵



```
ot@DESKTOP-KN25Q06:~/a/b/c/d/e/f/g/h/i/j/k/l/m# cp -rf b.txt /mnt/c/Users/srs33/
root@DESKTOP-KN25Q06:~/a/b/c/d/e/f/g/h/i/j/k/l/m# |
```

- **Now it would copy b.txt in sre file.**
- **“rf” stands for recursively folder, that is if we want to copy folder recursively.**
- **so it would also copy all the folders and files within b as well to new location.**



To create directories within directories:

mkdir -p dir1/dir2/dir3 ↵

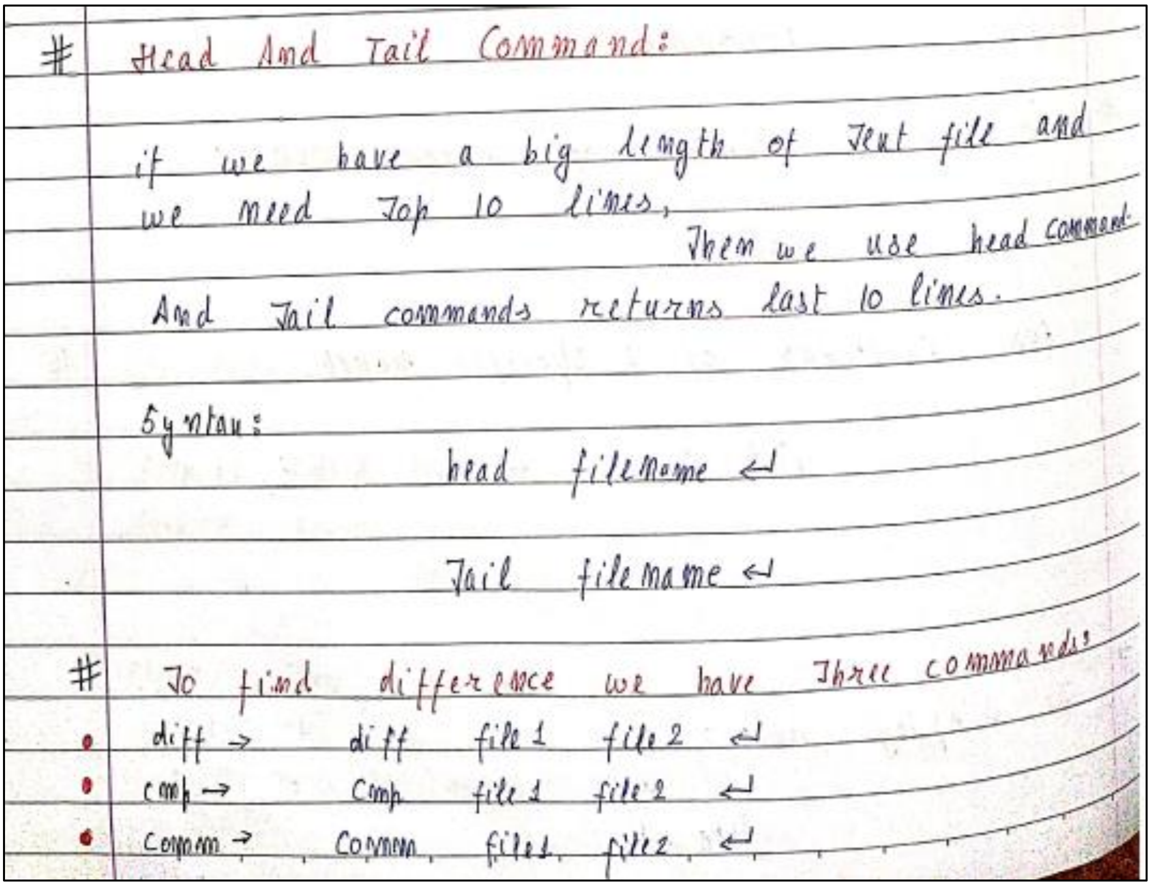
it would create Three directories dir1 & in it, it would create dir2. And in dir 2 it would create dir3.

if we want To find something about a command.

whatis command_name ↵

eg: whatis sudo ↵

And in addition to “whatis command” we can also use “man command”.



Head And Tail Command:

if we have a big length of Text file and we need Top 10 lines,
Then we use head command

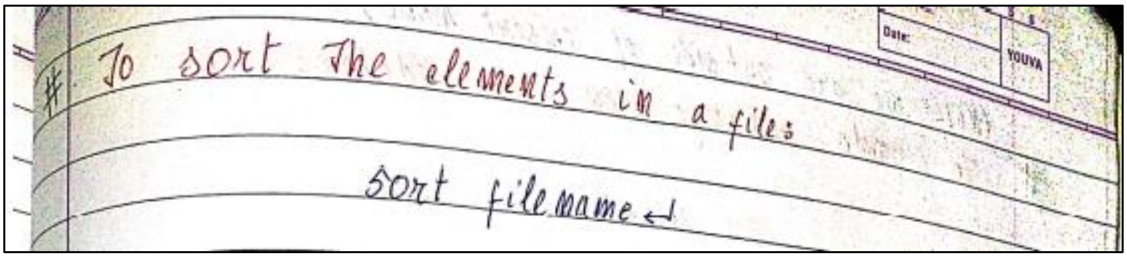
And Tail commands returns last 10 lines.

Syntax:

```
head filename ↵

Tail filename ↵
```

To find difference we have Three commands:

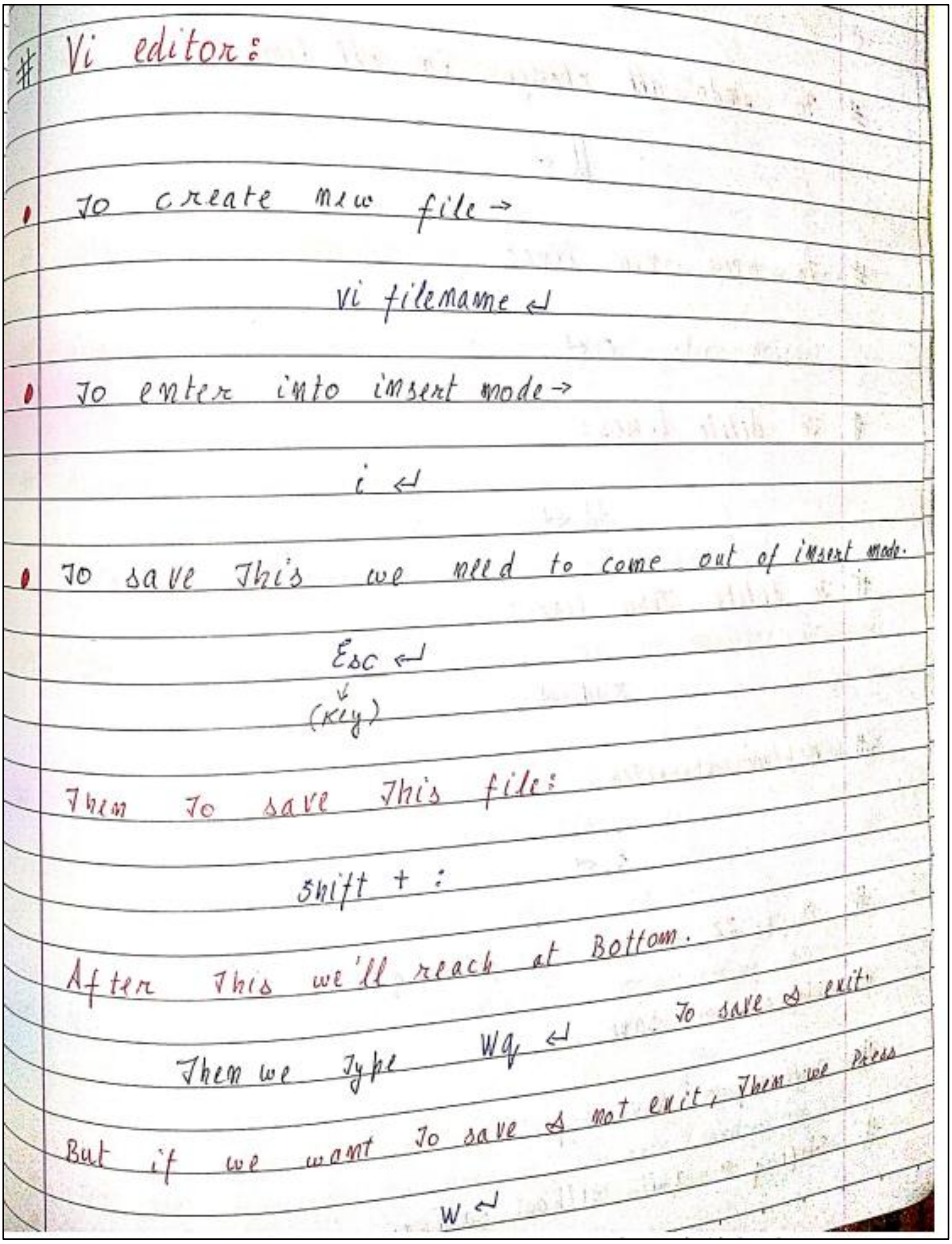
- diff → diff file1 file2 ↵
- cmp → Cmp file1 file2 ↵
- comm → comm file1 file2 ↵



To sort The elements in a file:

```
sort filename ↵
```

TO CREATE FILE USING VIM EDITOR:



Vi editor:

- To create new file →

vi filename ↵

- To enter into insert mode →

i ↵

- To save This we need to come out of insert mode.

Esc ↵
↓
(Key)

Then To save This file:

shift + :

After This we'll reach at Bottom.

Then we Type wq ↵ To save & exit.

But if we want To save & not exit, Then we Press

w ↵

(NOTE: we are outside of Insert Mode.)

To undo change in vi editor:

u ↵

To undo all changes in all line:

U ↵

To open new line:

o ↵

To delete lines:

dd ↵

To delete Three lines:

3dd ↵

replace character:

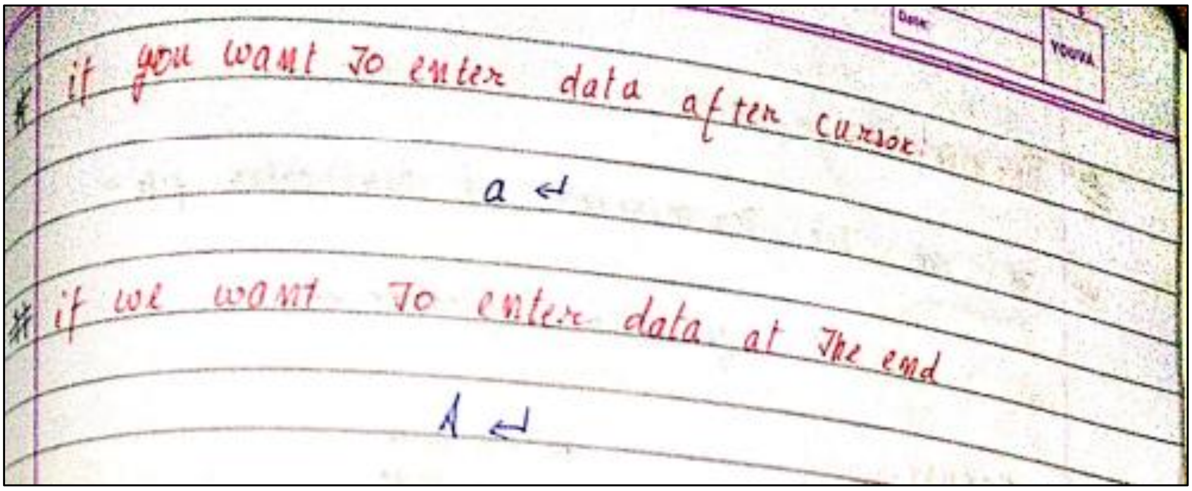
r ↵

Shift: ZZ → Save The file & quit

Shift: w → save

Shift: wq → save & quit

Shift: q → quit without saving.



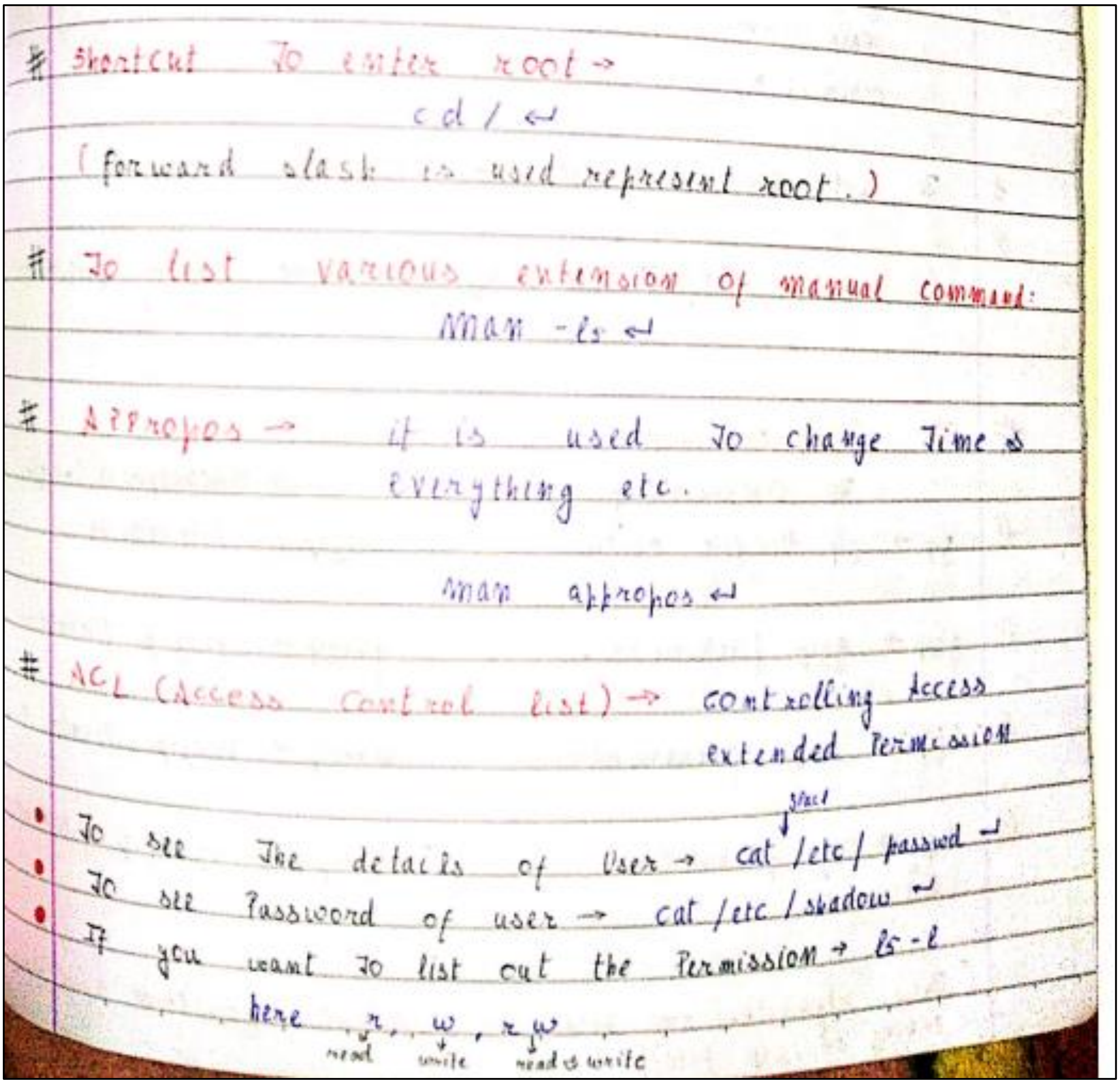
if you want To enter data after cursor.

a ↵

if we want To enter data at The end

A ↵

NOW SOME MORE IMPORTANT COMMANDS :



Shortcut To enter root →

cd / ↵

(forward slash is used represent root.)

To list various extension of manual command:

man -ls ↵

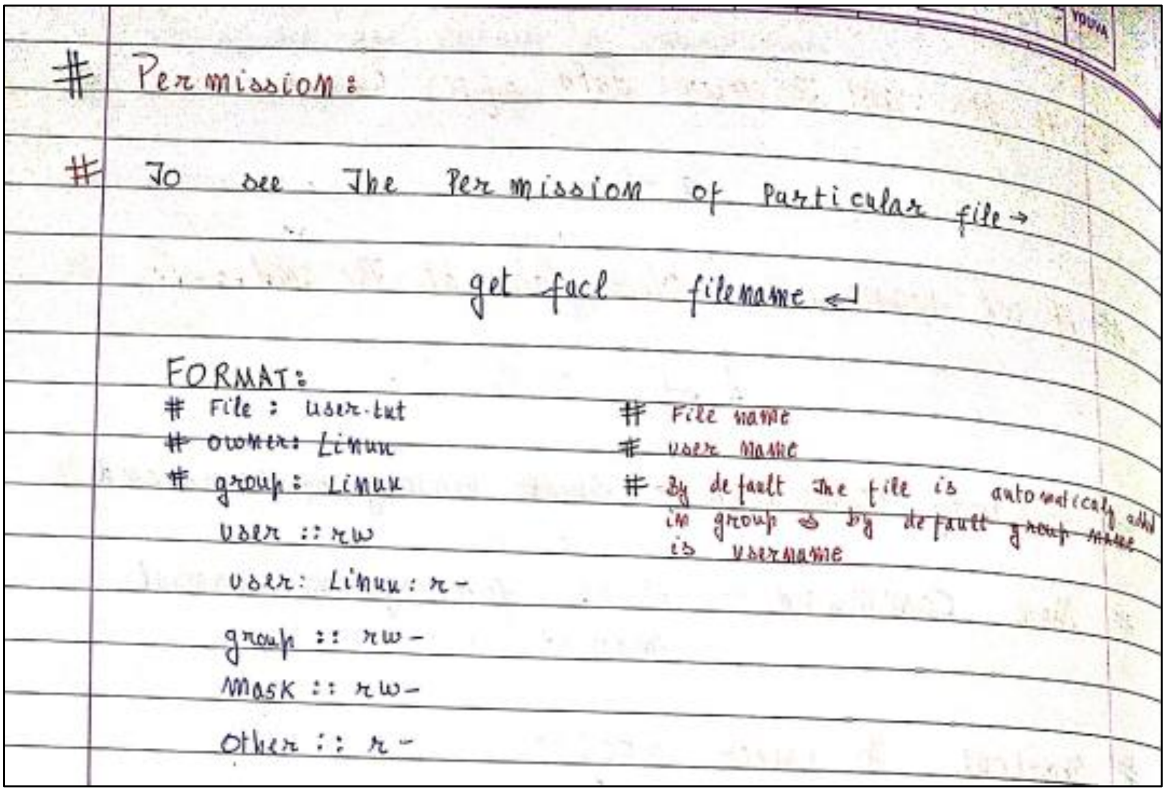
Appropos → it is used To change Time & everything etc.

man appropos ↵

ACL (Access Control list) → controlling Access extended Permission

- To see The details of User → cat /etc/ passwd ↵ (space)
- To see Password of user → cat /etc / shadow ↵
- If you want To list out the Permission → ls -l
  here r, w, rw
  read write read & write

PERMISSIONS:



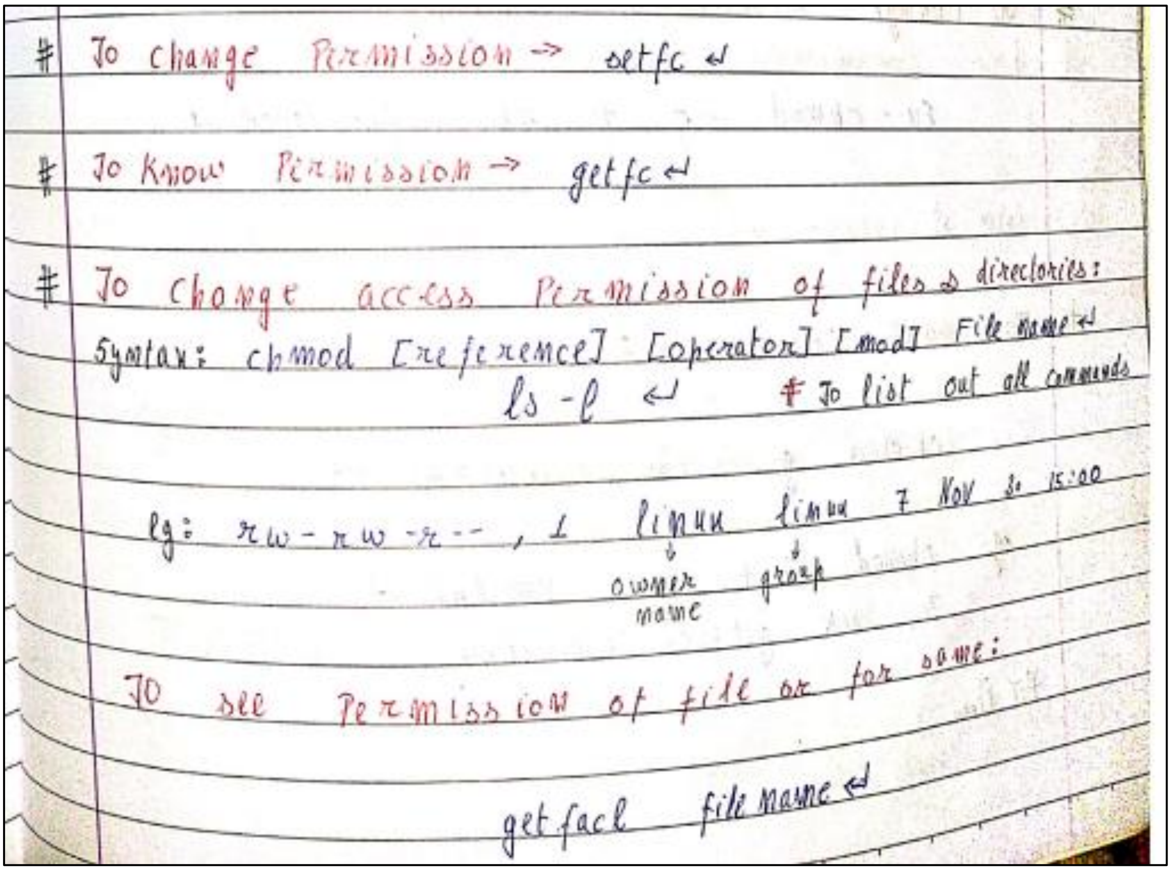
Permission:

To see The Permission of Particular file →

get facl filename ↵

FORMAT:

```
# File : user.txt                  # File name
# owner: Linux                     # user name
# group: Linux                     # By default The file is automatically add
   user ::rw                         in group & by default group name
   user: Linux: r-                   is username
   group :: rw-
   mask :: rw-
   other :: r-
```



To change Permission → setfc ↵

To Know Permission → getfc ↵

To change access Permission of files & directories:

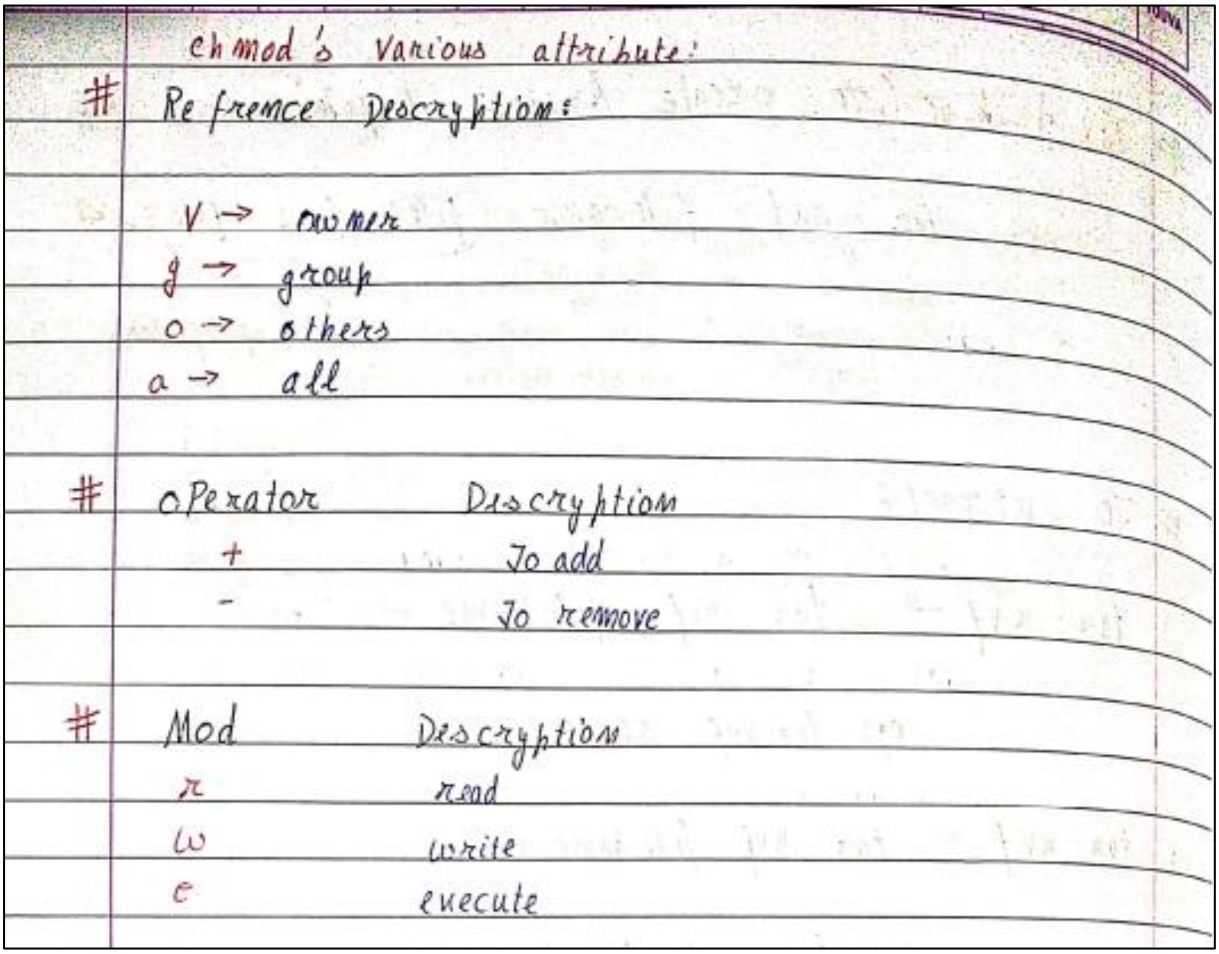
Syntax: chmod [reference] [operator] [mod] File name ↵

ls -l ↵ # To list out all commands

```
eg: rw-rw-r--, 1 linux    linux  7 Nov 30 15:00
                  ↓         ↓
                owner     group
                name
```

To see Permission of file or for same:

getfacl file name ↵



chmod's Various attribute:

Refrence Description:

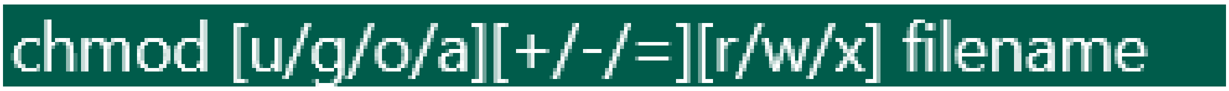
v →	owner
g →	group
o →	others
a →	all

#

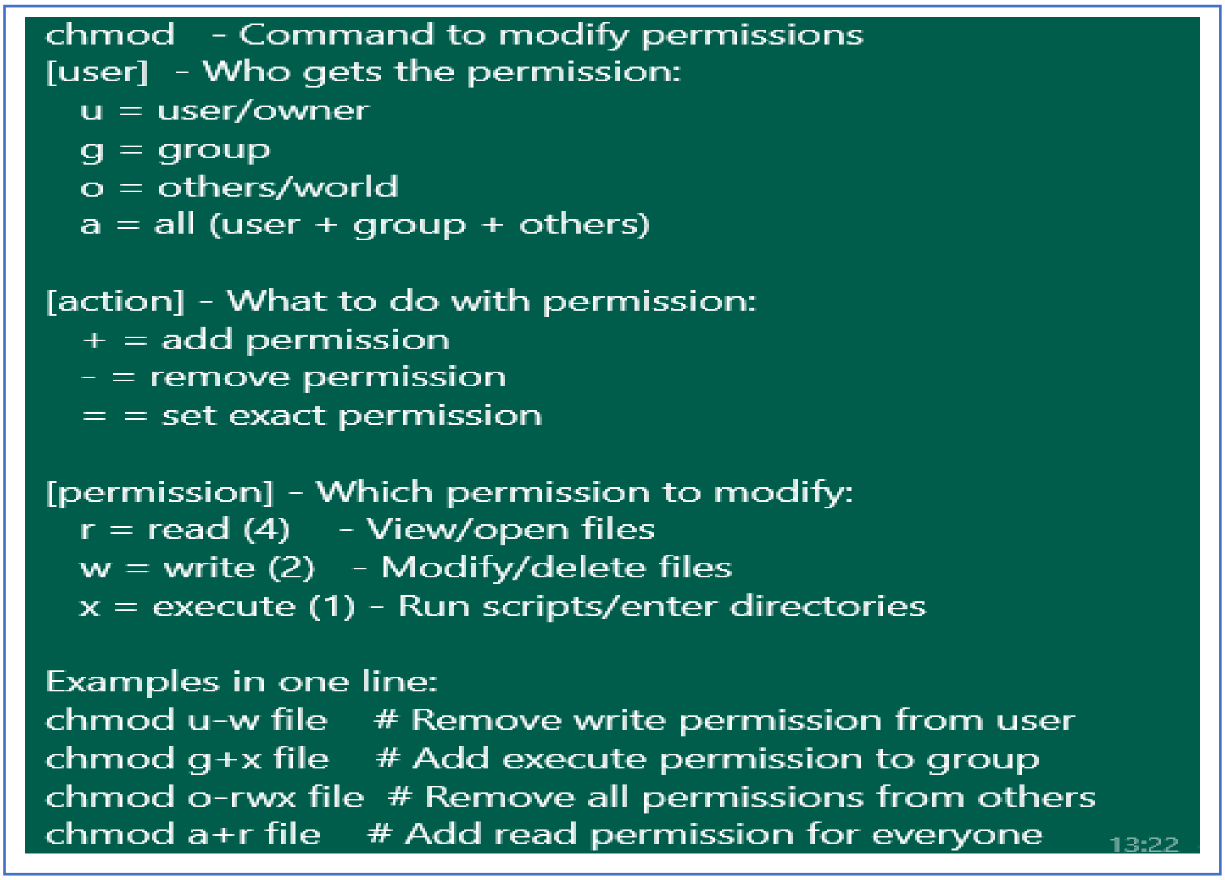
Operator	Description
+	To add
-	To remove

#

Mod	Description
r	read
w	write
e	execute

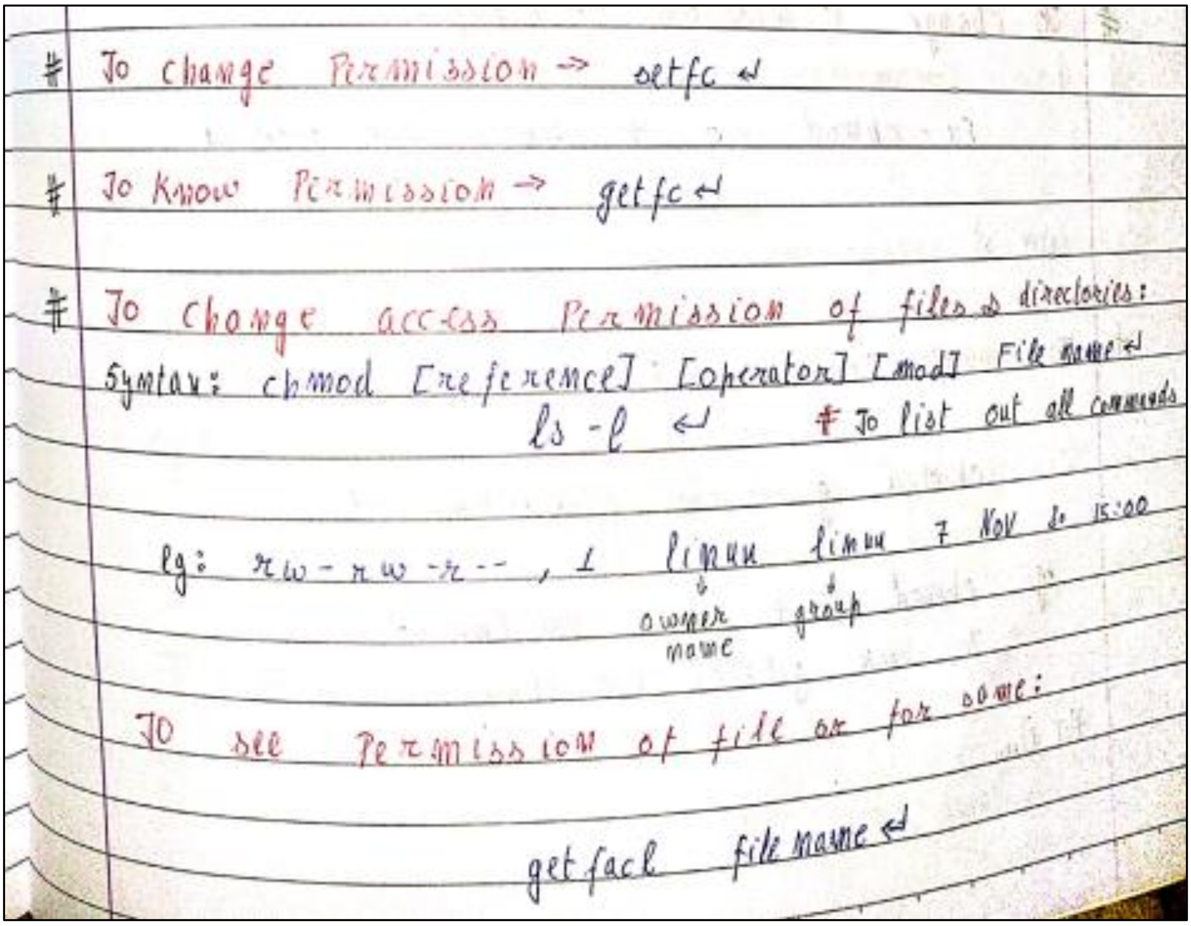


chmod [u/g/o/a][+/-/=][r/w/x] filename



```
chmod   - Command to modify permissions
[user]  - Who gets the permission:
   u = user/owner
   g = group
   o = others/world
   a = all (user + group + others)

[action] - What to do with permission:
   + = add permission
   - = remove permission
   = = set exact permission

[permission] - Which permission to modify:
   r = read (4)     - View/open files
   w = write (2)    - Modify/delete files
   x = execute (1) - Run scripts/enter directories

Examples in one line:
chmod u-w file     # Remove write permission from user
chmod g+x file     # Add execute permission to group
chmod o-rwx file  # Remove all permissions from others
chmod a+r file     # Add read permission for everyone
```

13:22



To change Permission -> setfc ↵

To Know Permission -> getfc ↵

To Change access Permission of files & directories:

Syntax: chmod [reference] [operator] [mod] File name ↵

ls -l ↵ # To list out all commands

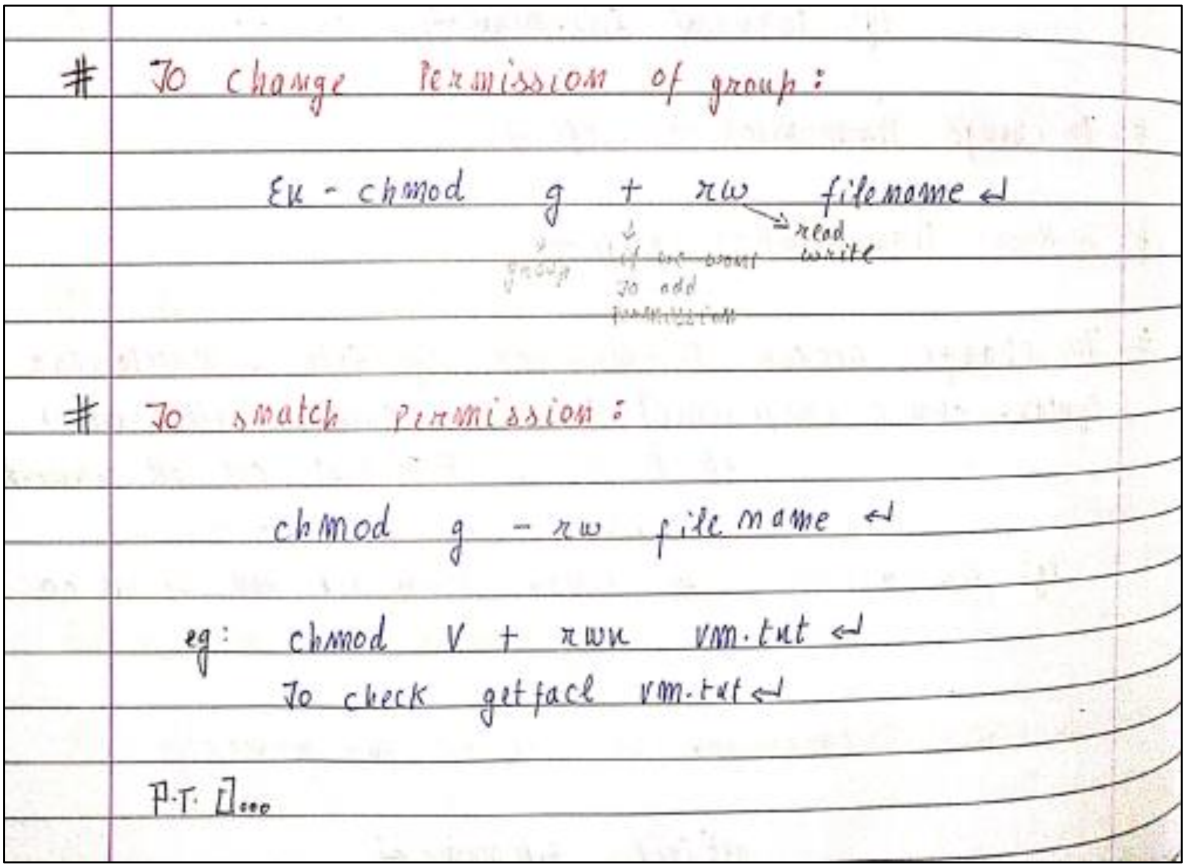
eg: rw-rw-r--, 1 linux linux 7 Nov 8 15:00

owner name

group

To see Permission of file or for some:

getfacl file name ↵



To change Permission of group:

Ex - chmod g + rw filename ↵

group

if we want To add permission

read write

To smatch Permission:

chmod g - rw file name ↵

eg: chmod V + rwx vm.txt ↵

To check getfacl vm.txt ↵

P.T.O...

chmod (in numeric) →

0 → no permission
1 → execute
2 → write
4 → read

if 3, That is 1+2 or 2+1
This means write & execute

if 5, That is 4+1 or 1+4
This means read + execute

if 6, That is 4+2
This means read + write

if 7, That is 4+2+1
This means read + write + execute

Syntax : chmod NNN Filename ↵

NNN: owner ↓ group ↘ other

∴ In one command we can give Permission of all other

eg : Ex - chmod 2 7 3 abc.txt ↵

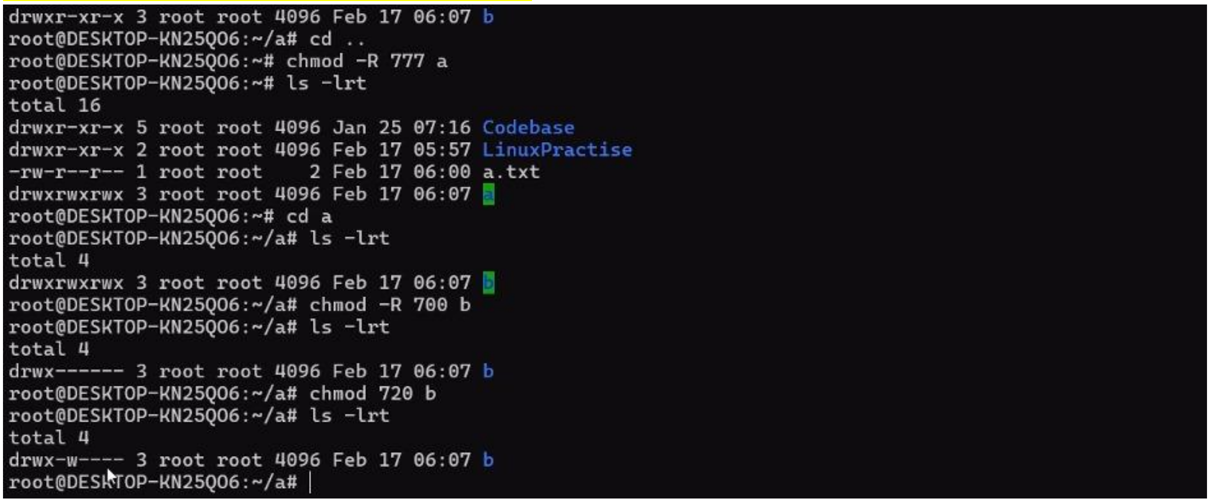
2 ↓ Permission for owner

7 ↓ Permission for group

3 → Permission for write & execute

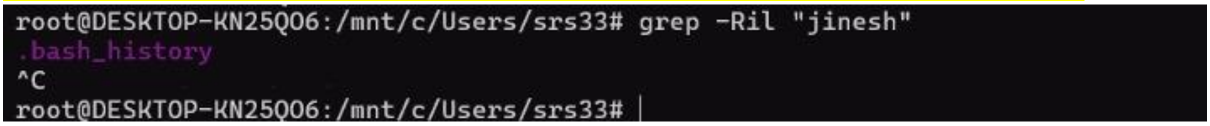
P.T.□ ...

To give or snatch permissions:



```
drwxr-xr-x 3 root root 4096 Feb 17 06:07 b
root@DESKTOP-KN25Q06:~/a# cd ..
root@DESKTOP-KN25Q06:~# chmod -R 777 a
root@DESKTOP-KN25Q06:~# ls -lrt
total 16
drwxr-xr-x 5 root root 4096 Jan 25 07:16 Codebase
drwxr-xr-x 2 root root 4096 Feb 17 05:57 LinuxPractise
-rw-r--r-- 1 root root    2 Feb 17 06:00 a.txt
drwxrwxrwx 3 root root 4096 Feb 17 06:07 a
root@DESKTOP-KN25Q06:~# cd a
root@DESKTOP-KN25Q06:~/a# ls -lrt
total 4
drwxrwxrwx 3 root root 4096 Feb 17 06:07 b
root@DESKTOP-KN25Q06:~/a# chmod -R 700 b
root@DESKTOP-KN25Q06:~/a# ls -lrt
total 4
drwx------ 3 root root 4096 Feb 17 06:07 b
root@DESKTOP-KN25Q06:~/a# chmod 720 b
root@DESKTOP-KN25Q06:~/a# ls -lrt
total 4
drwx-w---- 3 root root 4096 Feb 17 06:07 b
root@DESKTOP-KN25Q06:~/a# |
```

To search for specific content in file we use grep. grep is done to do recursive search. (use man grep to search about all the flags.)

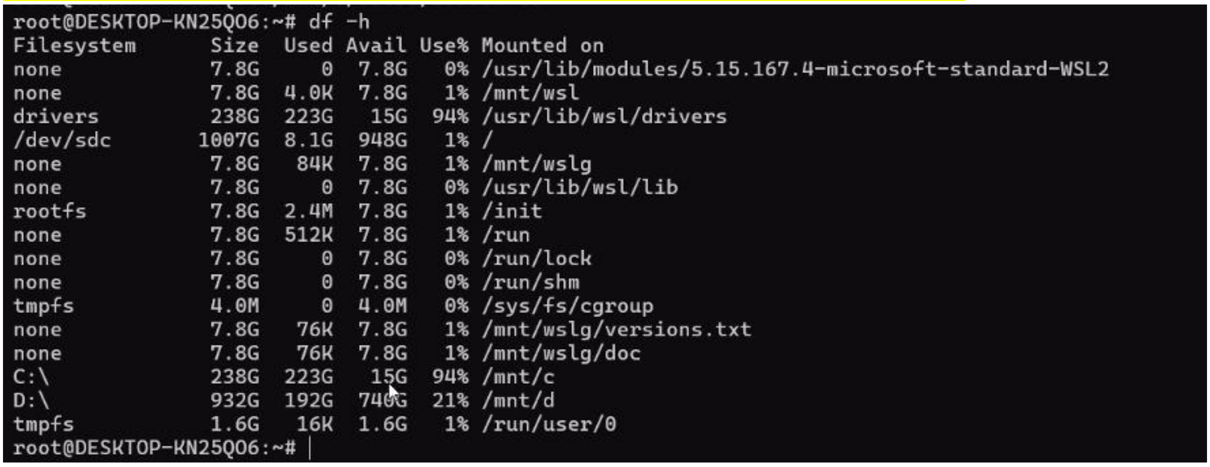


```
root@DESKTOP-KN25Q06:/mnt/c/Users/srs33# grep -Ril "jinesh"
.bash_history
^C
root@DESKTOP-KN25Q06:/mnt/c/Users/srs33# |
```

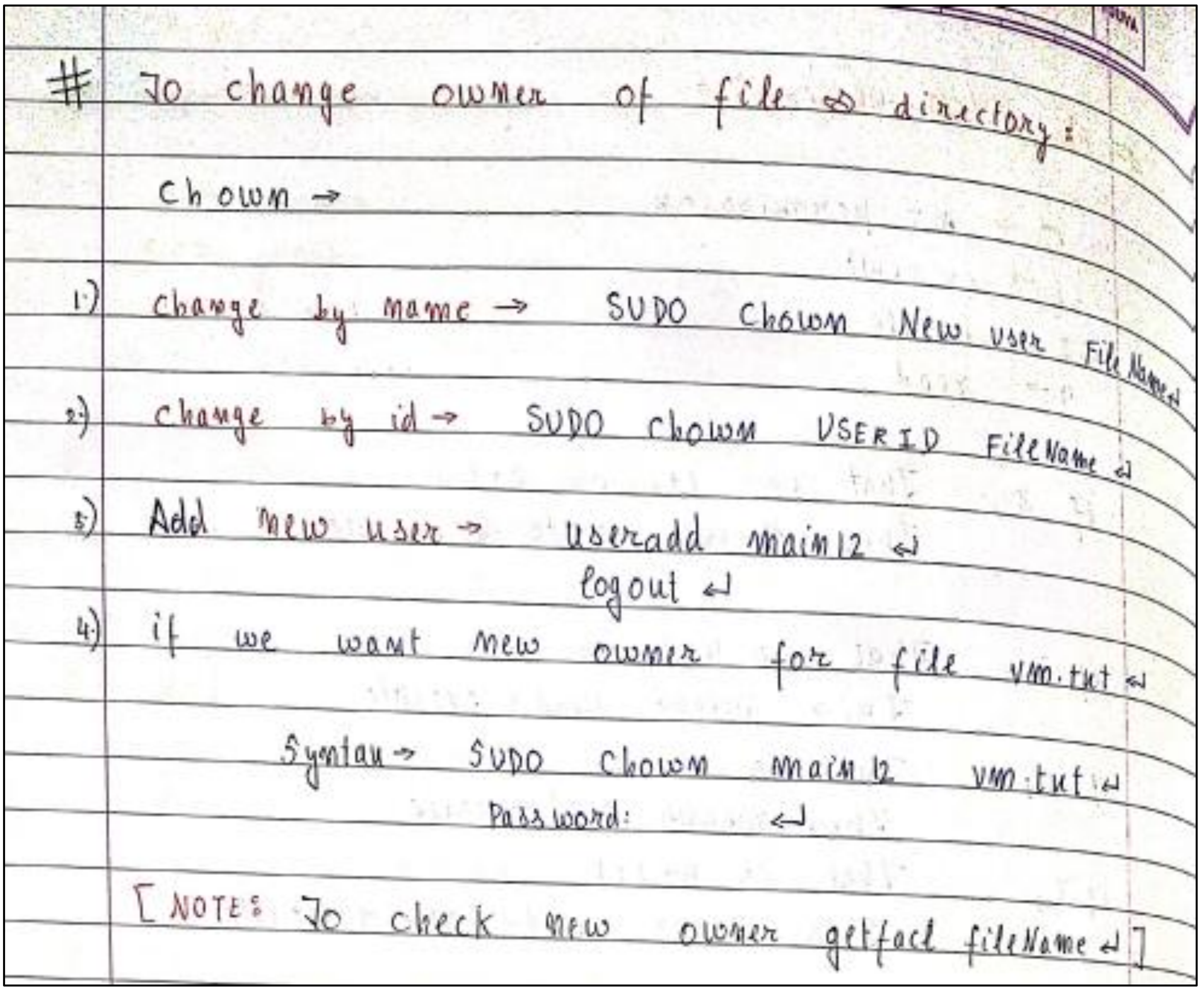
and if u want to search something in man help we write:

"/ <and then whatsoever we want to search>".

To see how much space we have used on our windows: "df -h"



```
root@DESKTOP-KN25Q06:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
none            7.8G     0  7.8G   0% /usr/lib/modules/5.15.167.4-microsoft-standard-WSL2
none            7.8G  4.0K  7.8G   1% /mnt/wsl
drivers         238G  223G   15G  94% /usr/lib/wsl/drivers
/dev/sdc       1007G  8.1G  948G   1% /
none            7.8G   84K  7.8G   1% /mnt/wslg
none            7.8G     0  7.8G   0% /usr/lib/wsl/lib
rootfs          7.8G  2.4M  7.8G   1% /init
none            7.8G  512K  7.8G   1% /run
none            7.8G     0  7.8G   0% /run/lock
none            7.8G     0  7.8G   0% /run/shm
tmpfs           4.0M     0  4.0M   0% /sys/fs/cgroup
none            7.8G   76K  7.8G   1% /mnt/wslg/versions.txt
none            7.8G   76K  7.8G   1% /mnt/wslg/doc
C:\             238G  223G   15G  94% /mnt/c
D:\             932G  192G  740G  21% /mnt/d
tmpfs           1.6G   16K  1.6G   1% /run/user/0
root@DESKTOP-KN25Q06:~# |
```



To change owner of file & directory:

chown →

1.) change by name → SUDO Chown New user FileName ↵

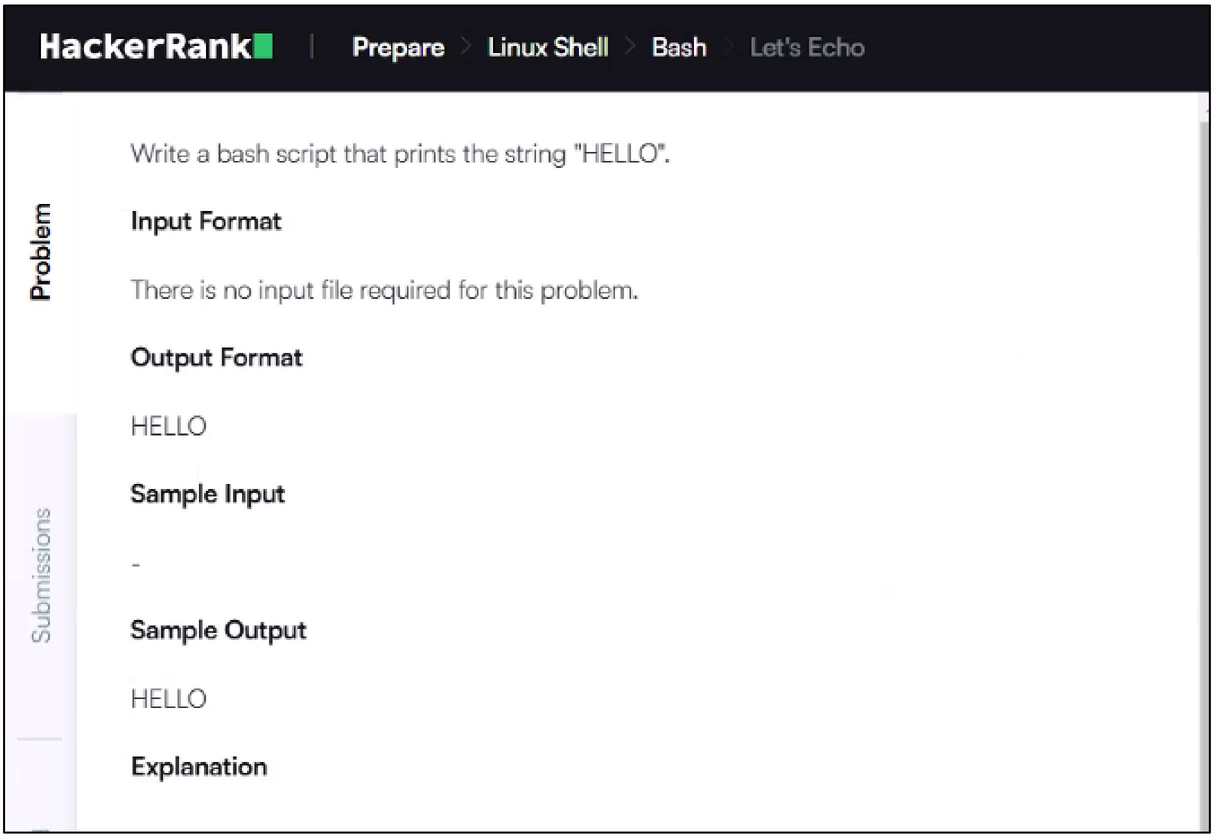
2.) change by id → SUDO chown USERID FileName ↵

3.) Add new user → useradd main12 ↵
logout ↵

4.) if we want new owner for file vm.txt ↵

Syntax → SUDO Chown main12 vm.txt ↵
Password: ↵

[NOTE: To check new owner getfacl fileName ↵]



Write a bash script that prints the string "HELLO".

Input Format

There is no input file required for this problem.

Output Format

HELLO

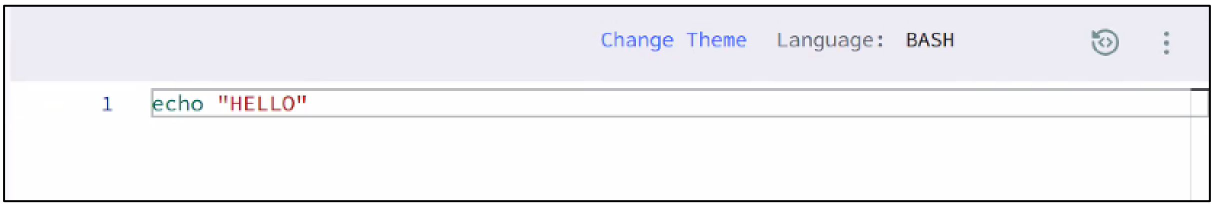
Sample Input

-

Sample Output

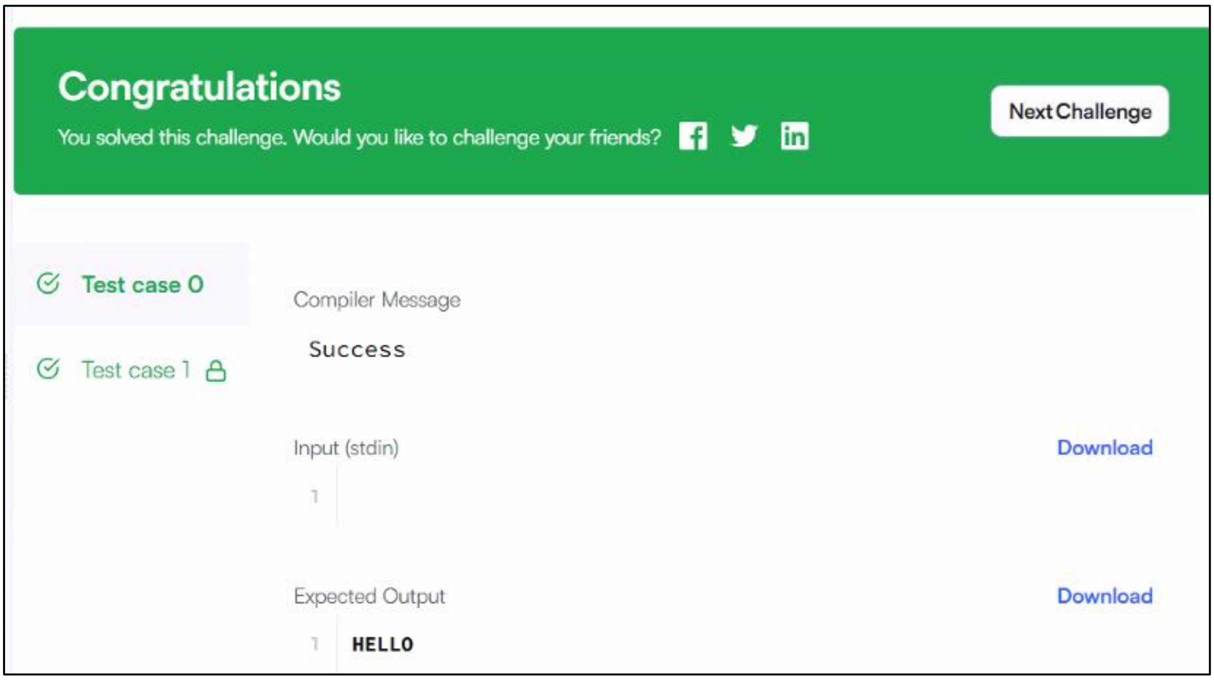
HELLO

Explanation



Language: BASH

```
echo "HELLO"
```



Congratulations

You solved this challenge. Would you like to challenge your friends?

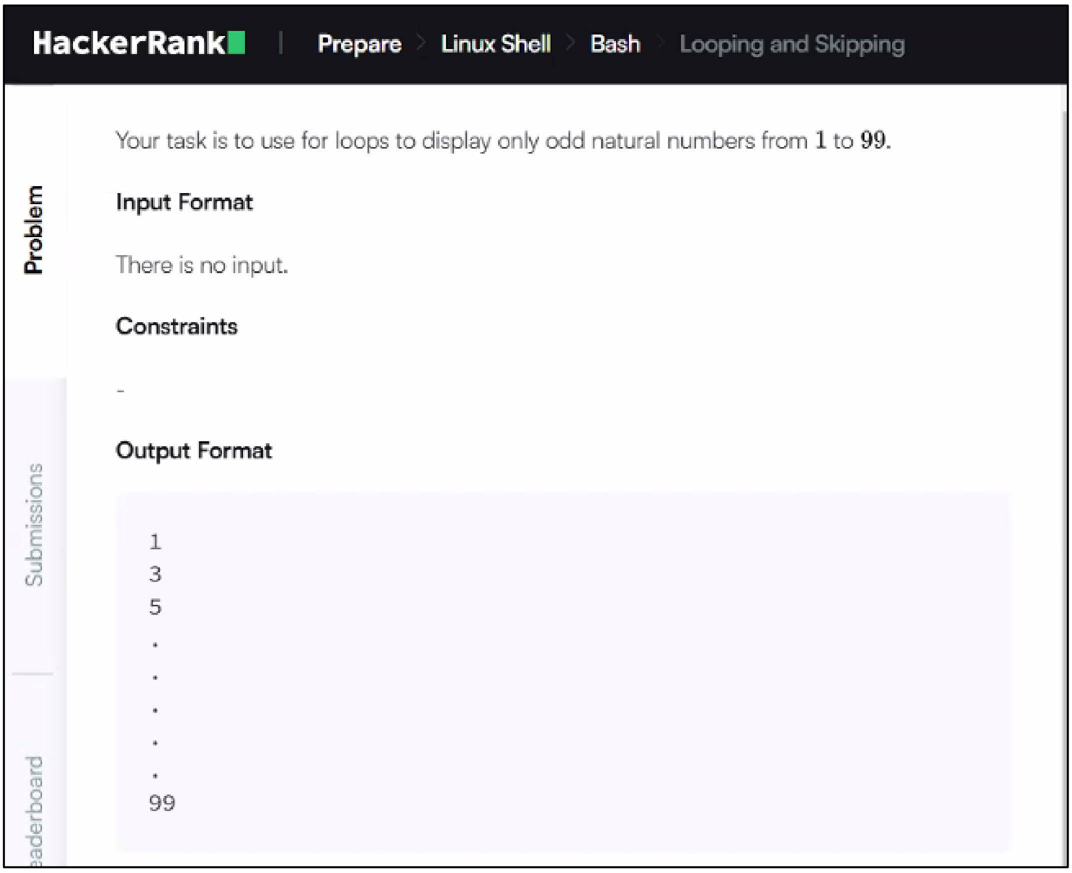
Compiler Message

```
Success
```

Input (stdin)

Expected Output

```
HELLO
```



Your task is to use for loops to display only odd natural numbers from 1 to 99.

Input Format

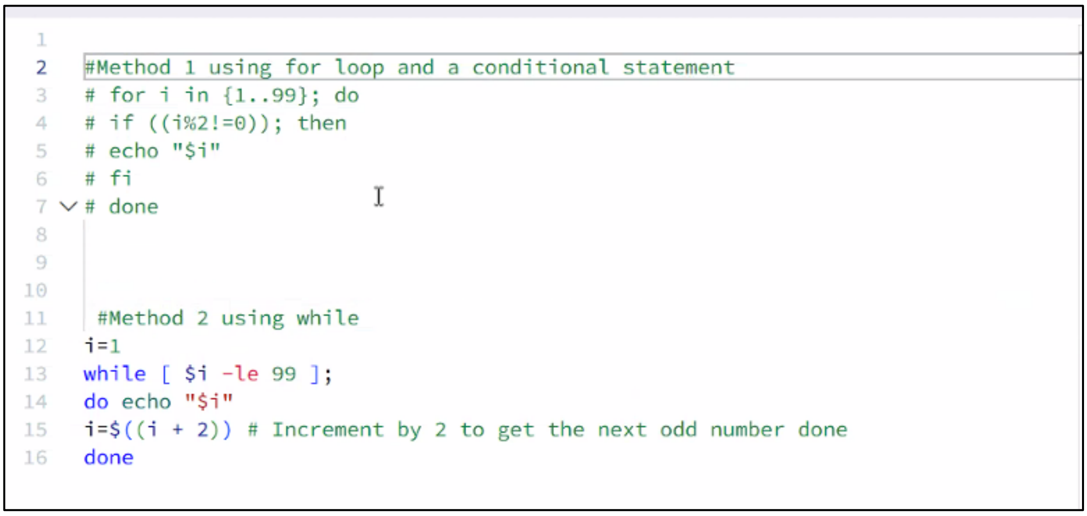
There is no input.

Constraints

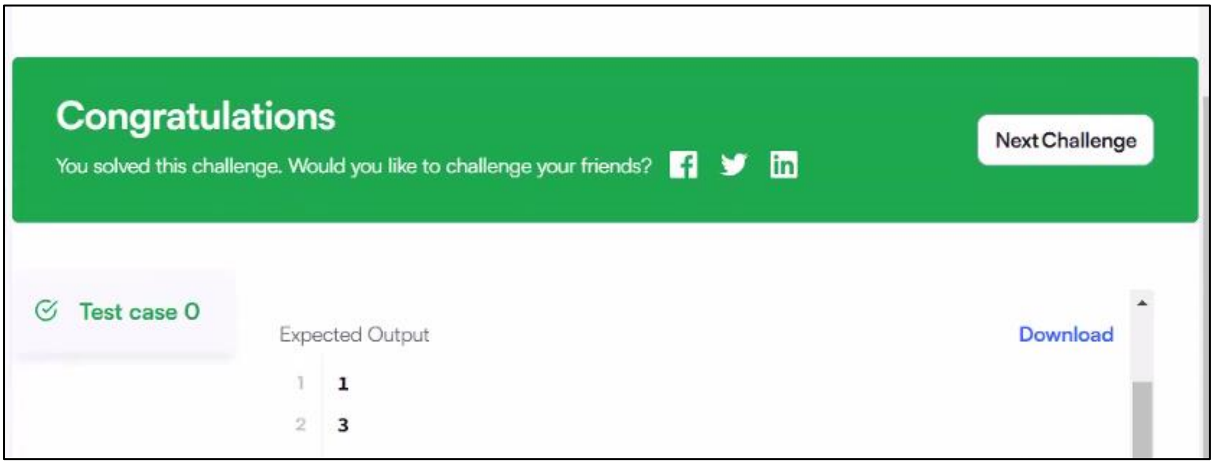
-

Output Format

```
1
3
5
.
.
.
.
.
99
```



```

#Method 1 using for loop and a conditional statement
# for i in {1..99}; do
# if ((i%2!=0)); then
# echo "$i"
# fi
# done



 #Method 2 using while
i=1
while [ $i -le 99 ];
do echo "$i"
i=$((i + 2)) # Increment by 2 to get the next odd number done
done
```

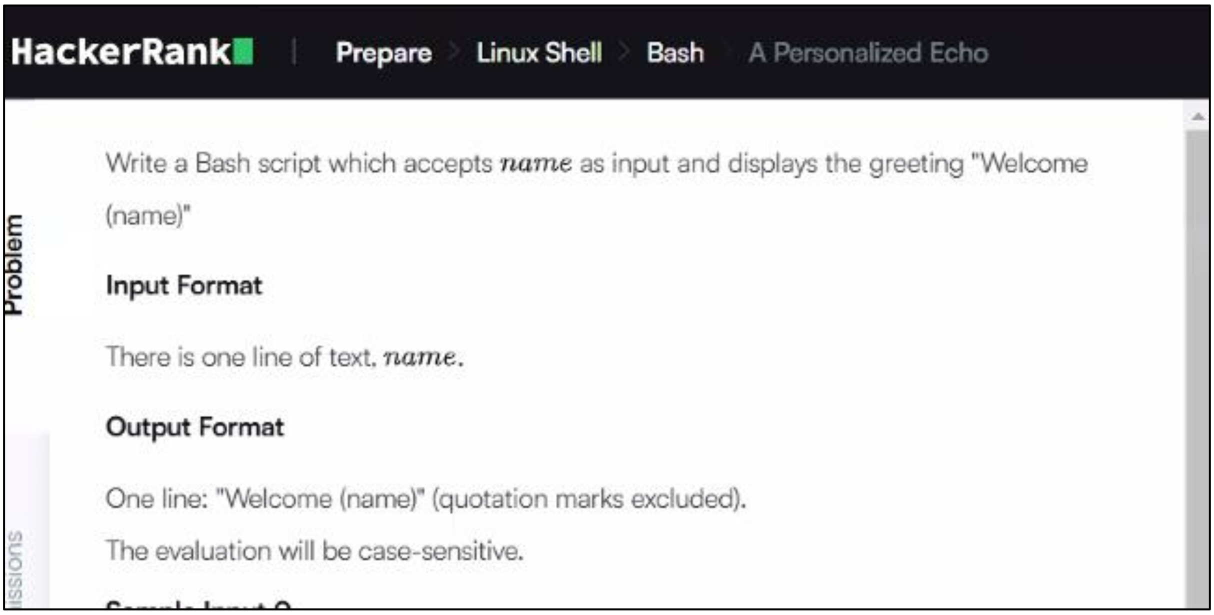


Congratulations

You solved this challenge. Would you like to challenge your friends?

Expected Output

```
1
3
```



Write a Bash script which accepts *name* as input and displays the greeting "Welcome (name)"

Input Format

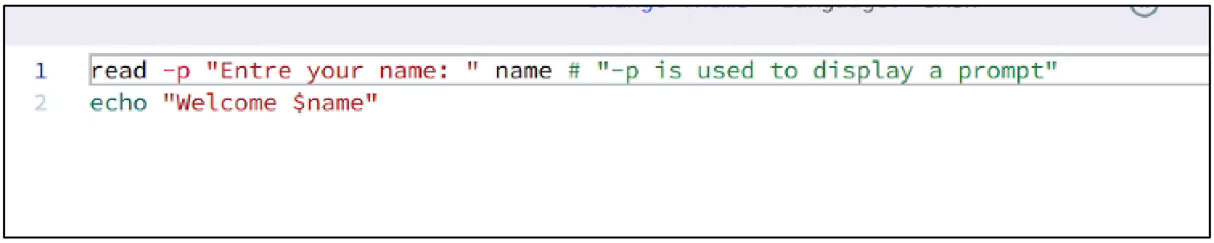
There is one line of text, *name*.

Output Format

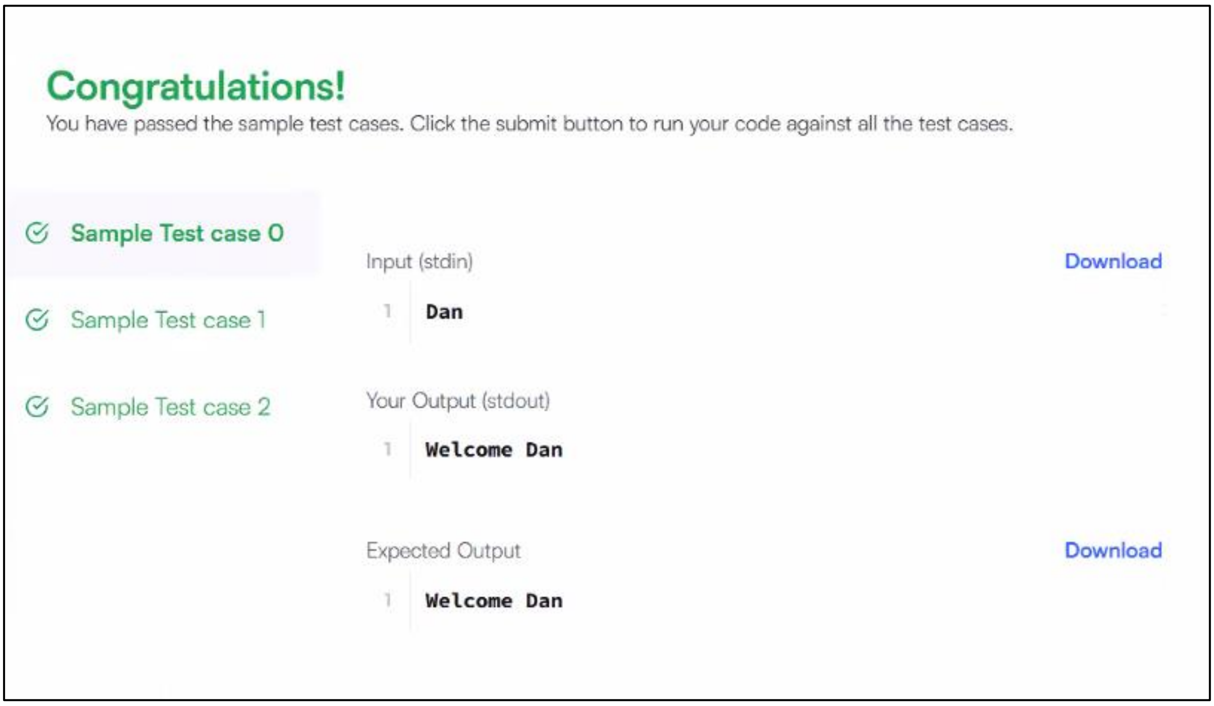
One line: "Welcome (name)" (quotation marks excluded).

The evaluation will be case-sensitive.

Sample Input 0



```
read -p "Entre your name: " name # "-p is used to display a prompt"
echo "Welcome $name"
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

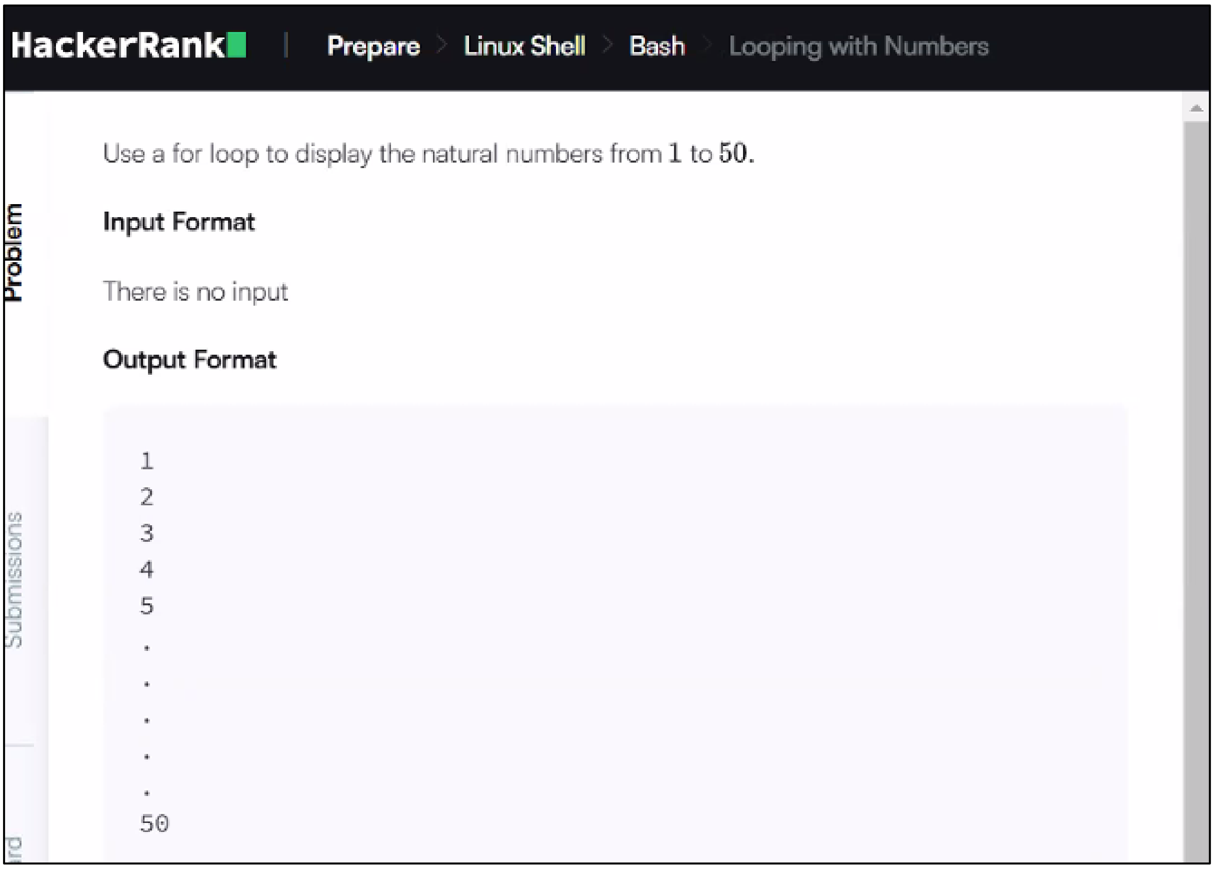
```
Dan
```

Your Output (stdout)

```
Welcome Dan
```

Expected Output

```
Welcome Dan
```



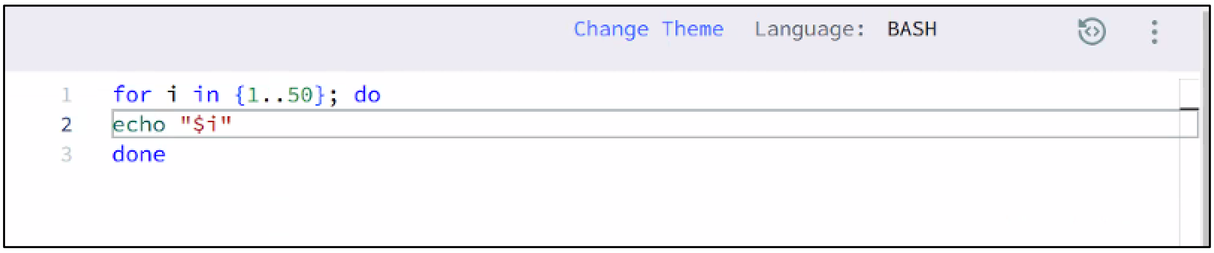
Use a for loop to display the natural numbers from 1 to 50.

Input Format

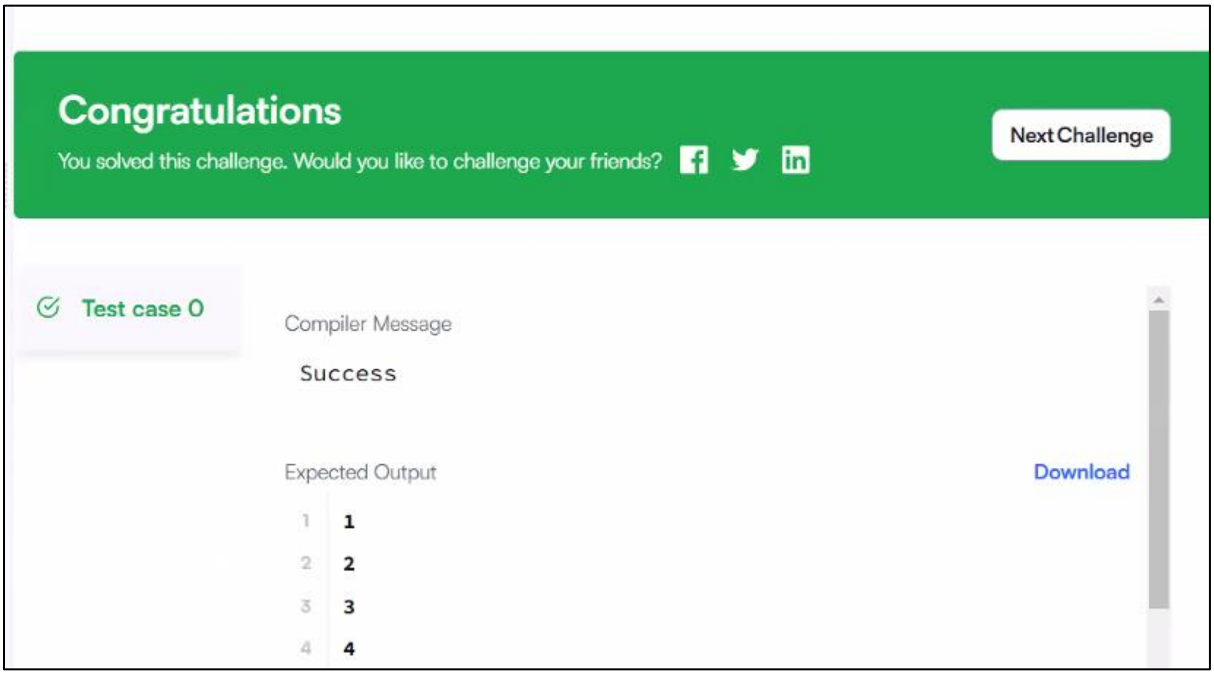
There is no input

Output Format

```
1
2
3
4
5
.
.
.
.
.
50
```



```bash
for i in {1..50}; do
echo "$i"
done
```



Congratulations

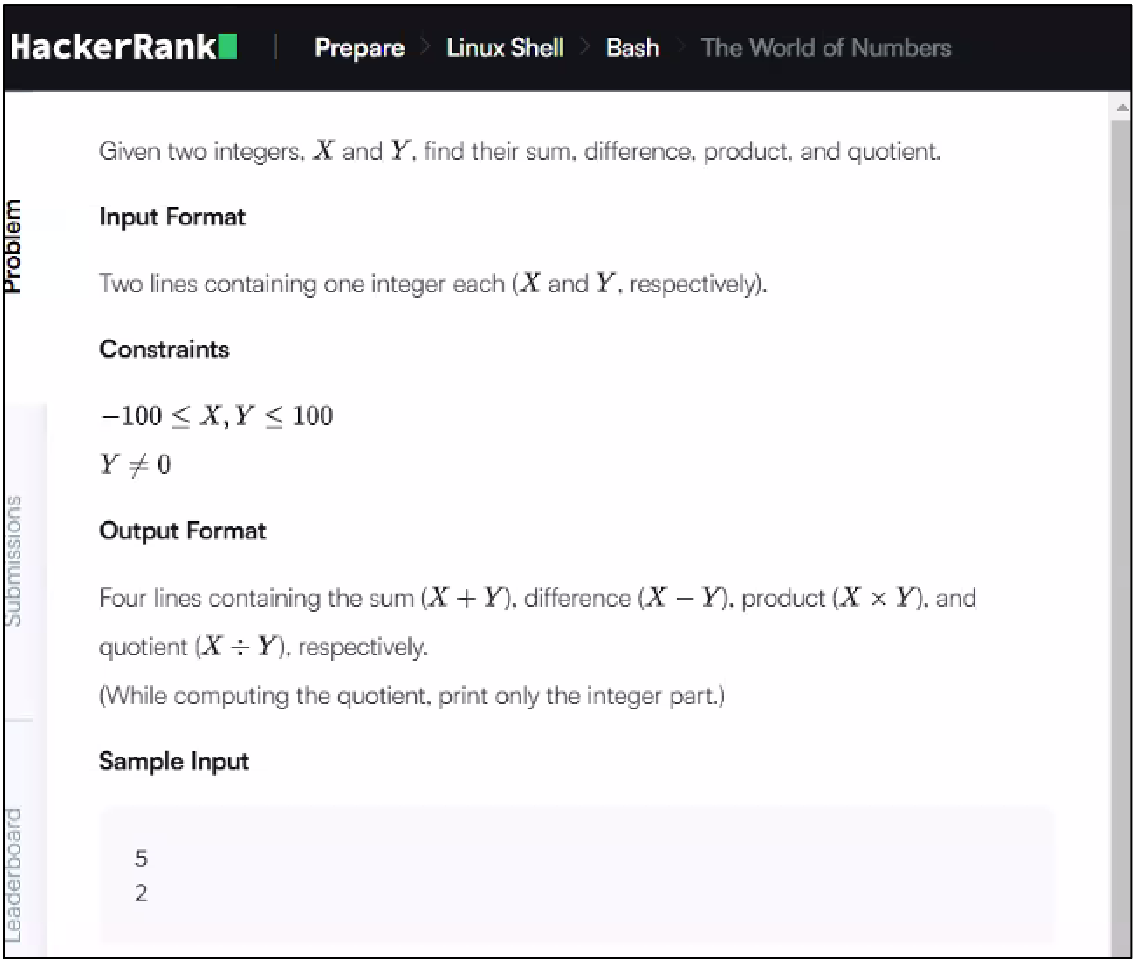
You solved this challenge. Would you like to challenge your friends?

Compiler Message

```
Success
```

Expected Output

```
1
2
3
4
```



Given two integers, X and Y, find their sum, difference, product, and quotient.

Input Format

Two lines containing one integer each (X and Y, respectively).

Constraints

$-100 \leq X, Y \leq 100$

$Y \neq 0$

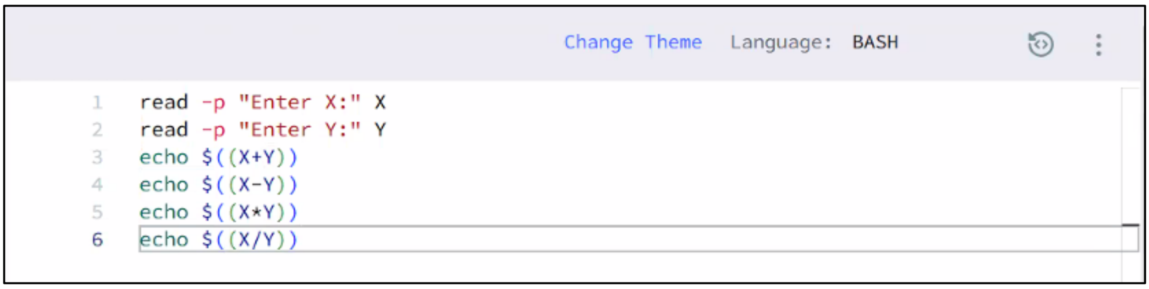
Output Format

Four lines containing the sum ($X + Y$), difference ($X - Y$), product ($X \times Y$), and quotient ($X \div Y$), respectively.

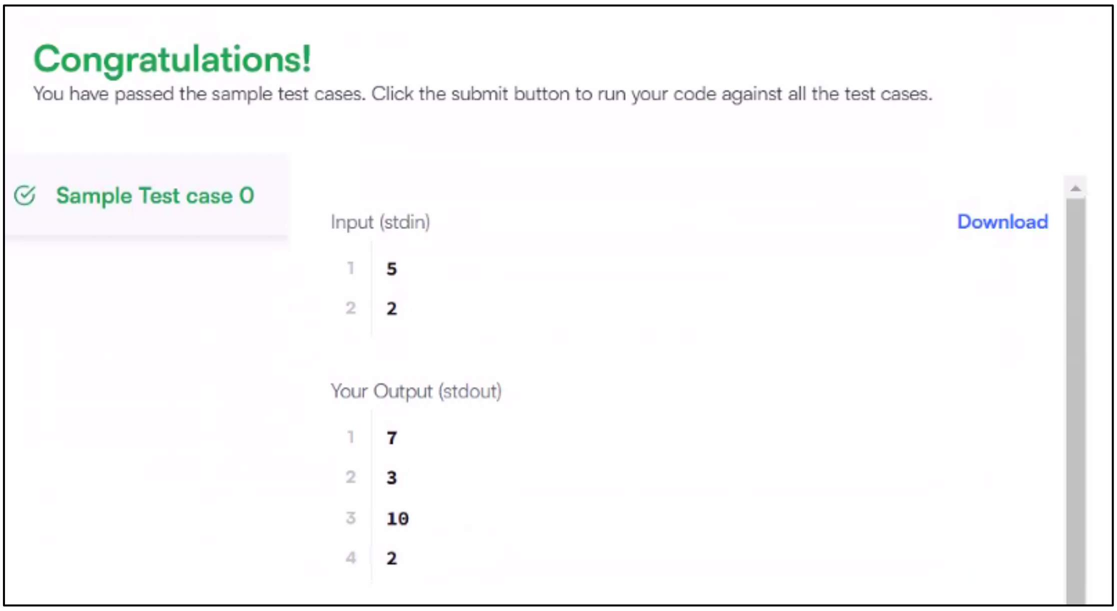
(While computing the quotient, print only the integer part.)

Sample Input

```
5
2
```



```bash
read -p "Enter X:" X
read -p "Enter Y:" Y
echo $((X+Y))
echo $((X-Y))
echo $((X*Y))
echo $((X/Y))
```



Congratulations!

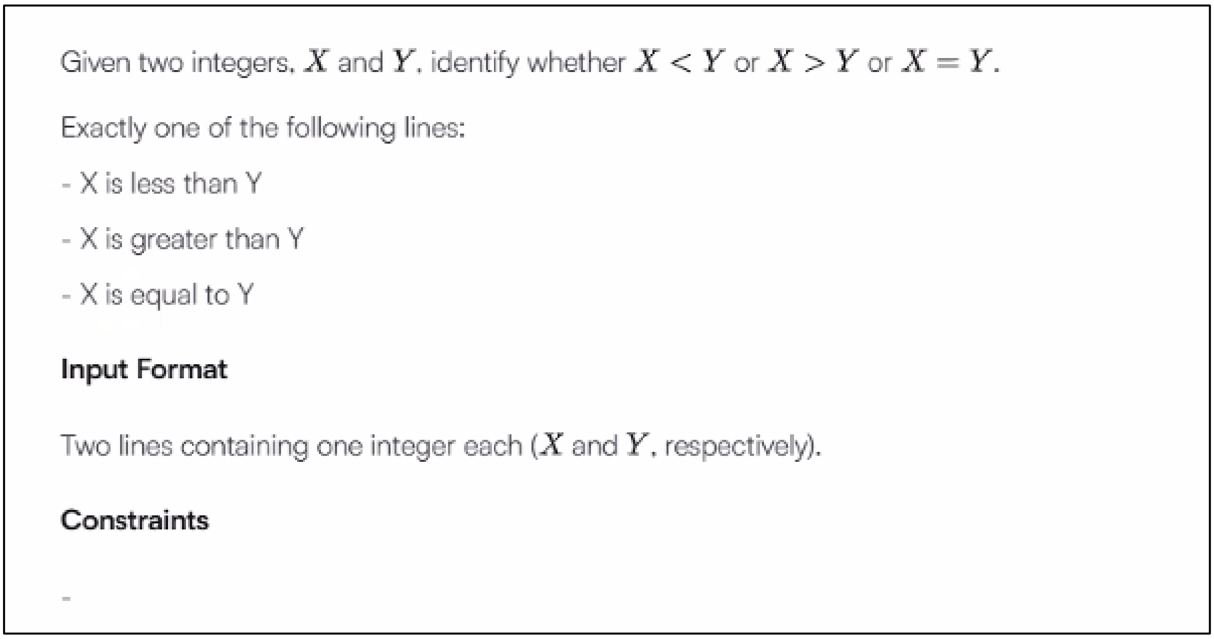
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
5
2
```

Your Output (stdout)

```
7
3
10
2
```



Given two integers, X and Y, identify whether $X < Y$ or $X > Y$ or $X = Y$.

Exactly one of the following lines:

- X is less than Y

- X is greater than Y

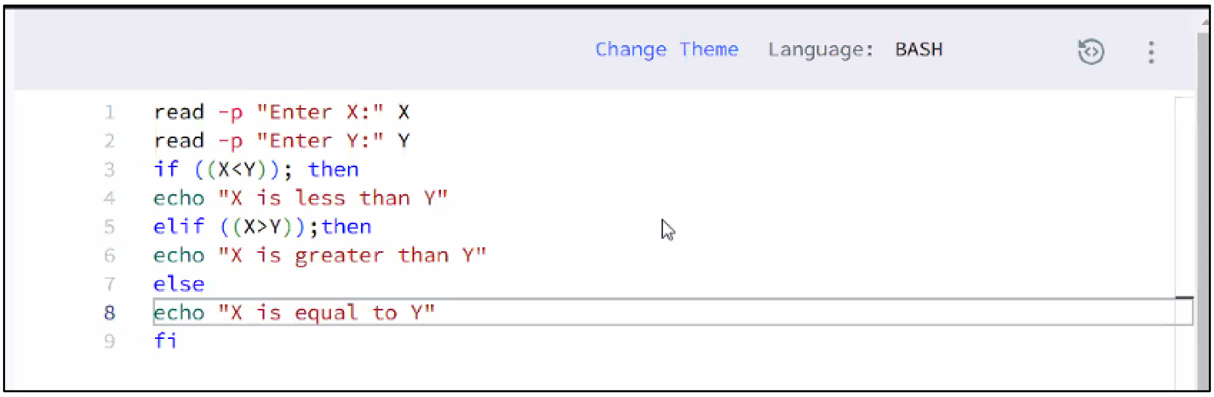
- X is equal to Y

Input Format

Two lines containing one integer each (X and Y, respectively).

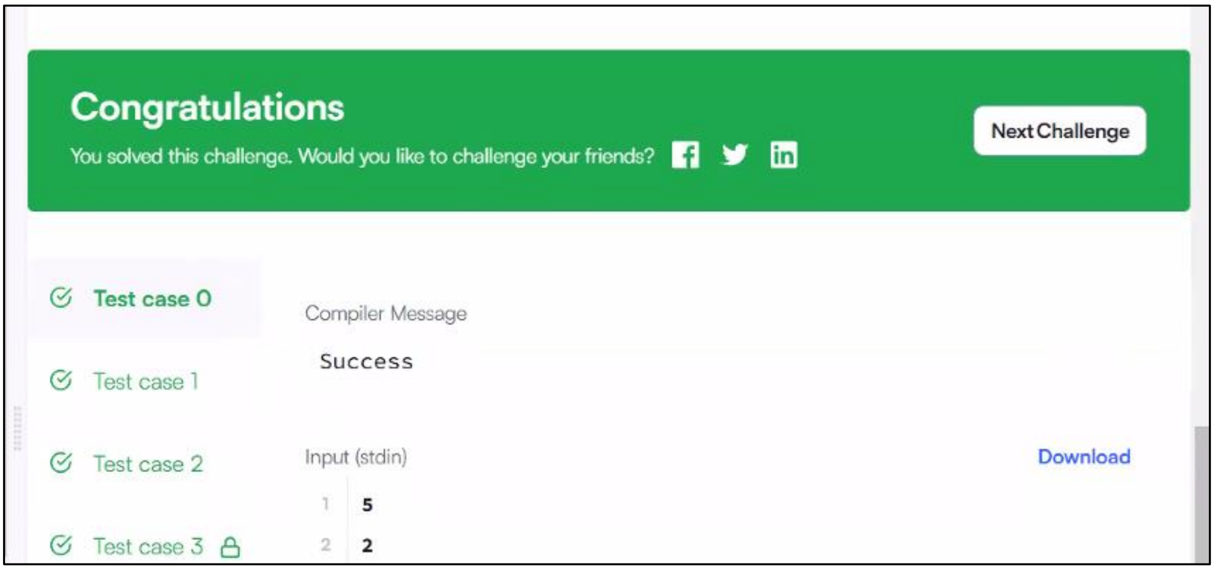
Constraints

-



Language: BASH

```
read -p "Enter X:" X
read -p "Enter Y:" Y
if ((X<Y)); then
echo "X is less than Y"
elif ((X>Y));then
echo "X is greater than Y"
else
echo "X is equal to Y"
fi
```



Congratulations

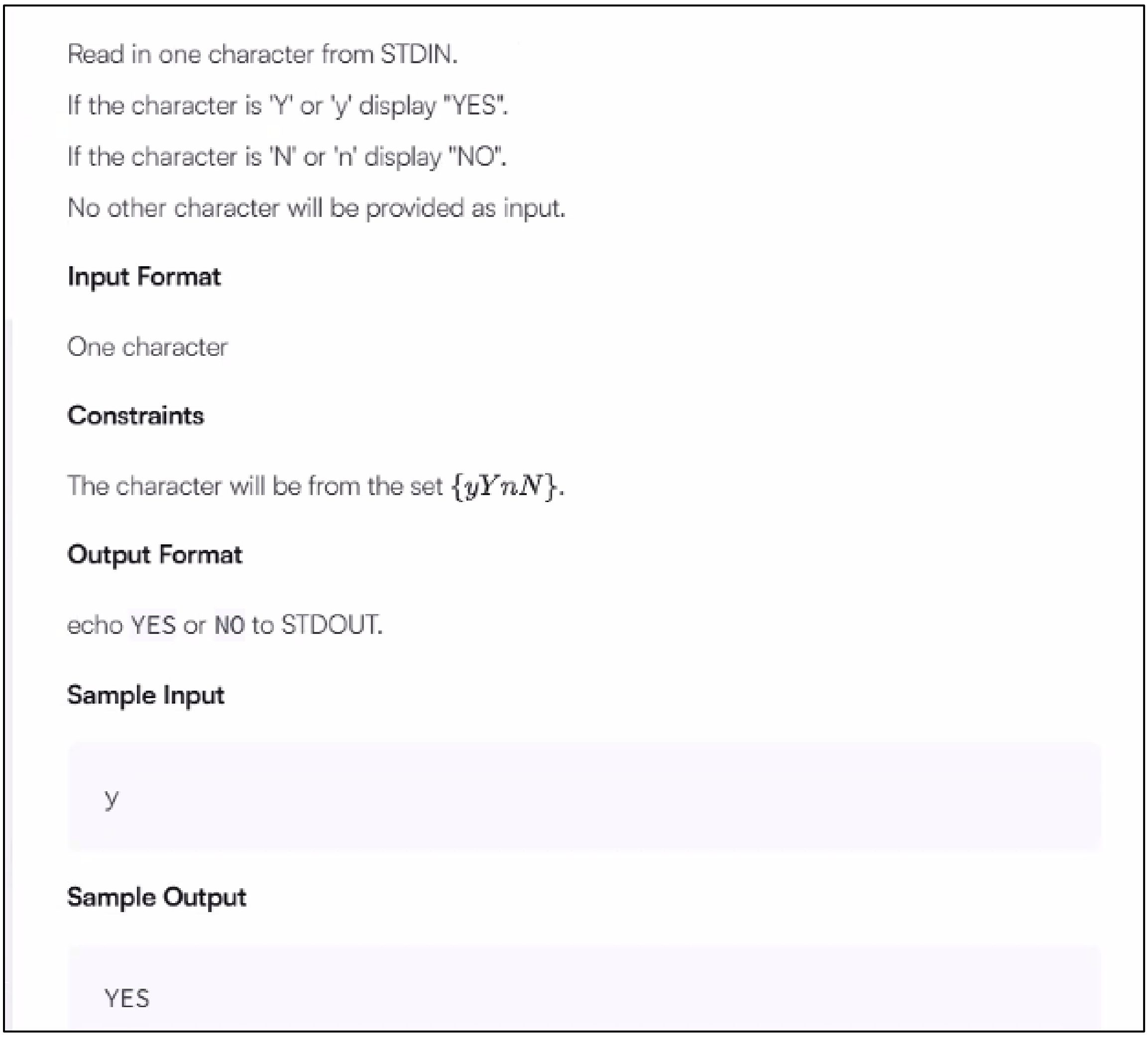
You solved this challenge. Would you like to challenge your friends?

Compiler Message

Success

Input (stdin)

```
5
2
```



Read in one character from STDIN.

If the character is 'Y' or 'y' display "YES".

If the character is 'N' or 'n' display "NO".

No other character will be provided as input.

Input Format

One character

Constraints

The character will be from the set $\{yYnN\}$.

Output Format

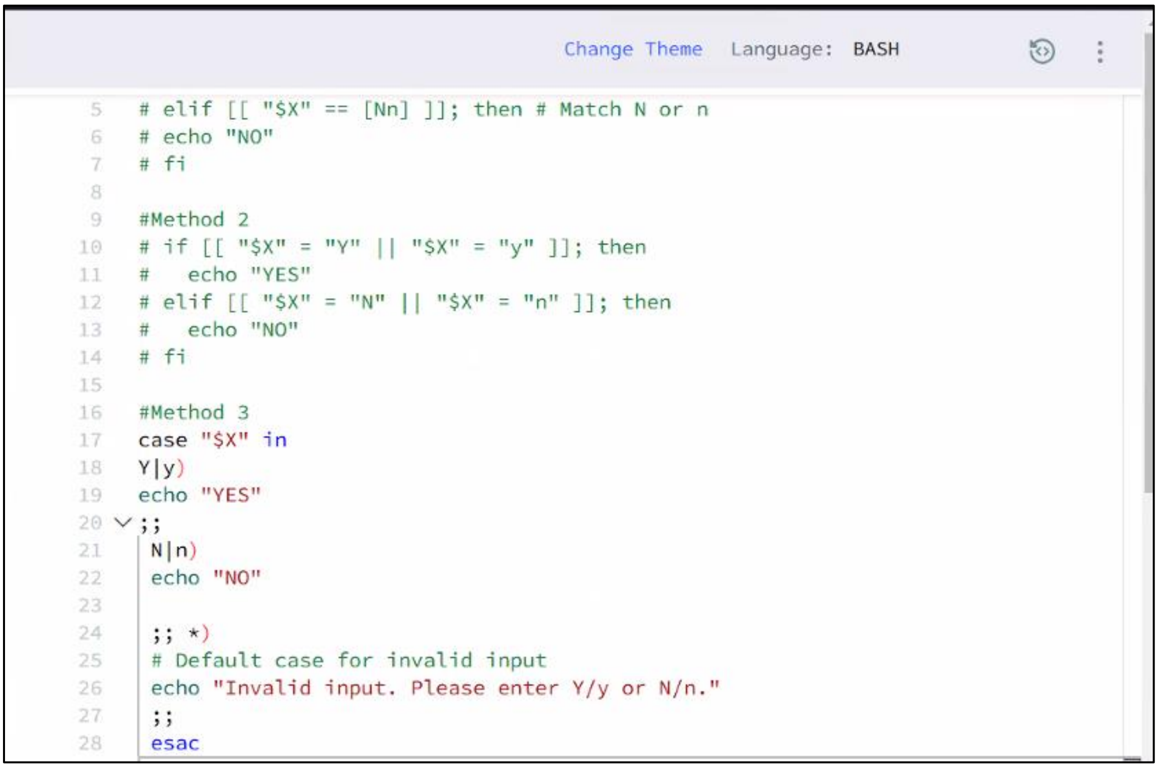
echo `YES` or `NO` to STDOUT.

Sample Input

```
y
```

Sample Output

```
YES
```



Language: BASH

```
# elif [[ "$X" == [Nn] ]]; then # Match N or n
# echo "NO"
# fi

#Method 2
# if [[ "$X" = "Y" || "$X" = "y" ]]; then
#   echo "YES"
# elif [[ "$X" = "N" || "$X" = "n" ]]; then
#   echo "NO"
# fi

#Method 3
case "$X" in
Y|y)
echo "YES"
;;
 N|n)
 echo "NO"

 ;; *)
 # Default case for invalid input
 echo "Invalid input. Please enter Y/y or N/n."
 ;;
 esac
```