COMMANDS	***USES***	***EXAMPLE***
1. ls	It is used to list information about files and directories within the file system.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main) $ ls Autodesk/ Bing.url desktop.ini Links/

2. Mkdir	The command mkdir stands for “make directory”. It creates each directory specified on the command line in the order given.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ mkdir hello.c
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ ls
		Autodesk/ Bing.url desktop.ini hello.c/ Links/

3.chdir	In Linux, the chdir command is used to **change the current working directory**. This command is similar to the cd command in Unix.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main) $ cd hello.c ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/hello.c (main) $ ^C

4.rmdir	The rmdir command is used to remove empty directories from our Linux Operating System. Every directory that needs to be removed should not contain any files or subdirectories. Otherwise, the rmdir command	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ rmdir hello.c
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ ls
		Autodesk/ Bing.url desktop.ini Links/

	cannot remove the directory from our file system.	
5.cat	The cat command is termed “concatenate”.It performs three main roles related to manipulation of text files: creating them, displaying them & combining them.	*

6.rm	rm command in UNIX stands for remove and by default is used for removing files. It is simple but a powerful command especially when used with options such as -rf which allow it to delete non-empty directories forcefully.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ mkdir A
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ touch a.txt
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ ls
		A/ a.txt Autodesk/ Bing.url desktop.ini Links/
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ rm a.txt
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites

		(main)
		$ rm A
		rm: cannot remove 'A': Is a directory
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites (main)
		$ ls
		A/ Autodesk/ Bing.url desktop.ini Links/
		ully.

7.mv	**mv** stands for **move**. mv is used to move one or more files or directories from one place to another in a file system.It has two distinct functions:
	(i) It renames a file or folder.
	(ii) It moves a group of files to a different directory.
		rename:
		$ ls
		a.txt b.txt c.txt
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
		$ mv a.txt hr.txt
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
		$ ls
		b.txt c.txt hr.txt
		
		_____
		move:
		$ ls
		a.txt b.txt c.txt
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
		$ mv a.txt hr.txt
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Trial (main)
		$ ls
		b.txt c.txt hr.txt

8.Cp		
9.head	The head command,	ROHINI@DESKTOP-AAJD063

	as the name implies, print the top N number of data of the given input. By default, it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.	MINGW64 ~/Favorites/Links/Trial (main)
		$ cat > b.txt
		riya
		jaiswal
		krishna
		xyz
		aarti
		myself
		os
		linux
		unix
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
		$ head b.txt
		riya
		jaiswal
		krishna
		xyz
		aarti
		myself
		os
		linux
		unix
		
		ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main)
		$ head -n 6 b.txt
		riya
		jaiswal
		krishna
		xyz
		aarti
		myself

10.tail	It is the complementary of head command.The tail command, as the name implies, print the last N number of data of the given input. By default it prints the last 10 lines	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ tail b.txt riya jaiswal krishna xyz aarti myself os

	of the specified files. If more than one file name is provided then data from each file is precedes by its file name.	linux unix _____ ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ tail -n 3 b.txt os linux unix
11.sort	Sort is a standard command-line program that prints the lines of its input or concatenation of all files listed in its argument list in sorted order. The sort command is a command-line utility for sorting lines of text files. It supports sorting alphabetically, in reverse order, by number, by month, and can also remove duplicates.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ sort b.txt aarti jaiswal krishna linux myself os riya unix xyz
12.wc	wc command in Linux with examples. wc stands for word count. As the name implies, it is mainly used for counting purpose. It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments. By default it displays four-columnar output.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ wc b.txt 9 9 52 b.txt
13.chown	The basic chown	ROHINI@DESKTOP-AAJD063

	command syntax consists of a few segments. The help file shows the following format: `chown [OPTIONS] USER[:GROUP] FILE(s)` ● [OPTIONS] – the command can be used with or without additional options. ● [USER] – the username or the numeric user ID of the new owner of a file. ● [:] – use the colon when changing a group of a file. ● [GROUP] – changing the group ownership of a file is optional. ● FILE – the target file.	MINGW64 ~/Favorites/Links/Trial (main) $ chown --version chown (GNU coreutils) 8.32 Copyright (C) 2020 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>. This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law. Written by David MacKenzie and Jim Meyering.
15.chgrp	**chgrp command** in Linux is used to change the group ownership of a file or directory. All files in Linux belong to an owner and a group. You can set the owner by using "chown" command, and the group by the "chgrp"	

	command.**Syntax:**
	`chgrp [OPTION]… GROUP FILE…`
	`chgrp [OPTION]… –reference=RFILE FILE…`	

16.umask	The umask command in Linux is used to set default permissions for files or directories the user creates.The umask command specifies the permissions that the user does not want to be given out to the newly created file or directory. umask works by doing a Bitwise AND with the bitwise complement(where the bits are inverted, i.e. 1 becomes 0 and 0 becomes 1) of the umask.The bits which are set in the umask value, refer to the permissions, which are not assigned by default, as these values are subtracted from the maximum permission for files/directories.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ umask 0022
17.Ps	The ps command, short for Process Status, is a command line utility that is used to display or view information related to the processes	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ PS PID PPID PGID WINPID TTY UID STIME

	running in a Linux system. As we all know, Linux is a multitasking and multiprocessing system. Therefore, multiple processes can run concurrently without affecting each other.	COMMAND 1340 1 1340 12148 cons0 197609 19:27:56 /usr/bin/bash 2123 1340 2123 7996 cons0 197609 22:41:52 /usr/bin/PS
18.pipe	A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing. The Unix/Linux systems allow stdout of a command to be connected to stdin of another command. You can make it do so by using the pipe character '|'. Pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ sort b.txt | uniq aarti jaiswal krishna linux myself os riya unix xyz
19.Redirection operators	Redirection is a feature in Linux such that when executing a command, you can	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ date > c.txt

	change the standard input/output devices. The basic workflow of any Linux command is that it takes an input and give an output. ● The standard input (stdin) device is the keyboard. ● The standard output (stdout) device is the screen.	ROHINI@DESKTOP-AAJD063 MINGW64 ~/Favorites/Links/Trial (main) $ cat c.txt Fri, Jan 27, 2023 11:16:57 PM

20.

a)Display top 10 processes in descending order:

```
tsec11@ubuntu:~$ ps -e | head -n 10
   PID TTY          TIME CMD
     1 ?        00:00:03 systemd
     2 ?        00:00:00 kthreadd
     3 ?        00:00:00 rcu_gp
     4 ?        00:00:00 rcu_par_gp
     5 ?        00:00:00 kworker/0:0-eve
     6 ?        00:00:00 kworker/0:0H-kb
     7 ?        00:00:00 kworker/u256:0-
     8 ?        00:00:00 mm_percpu_wq
     9 ?        00:00:00 ksoftirqd/0
    10 ?        00:00:00 rcu_sched
```

b)Display the processor with highest memo usage

```
tsec11@ubuntu:~$ ps -eo pid,ppid,%mem,%cpu --sort=-%mem | head -n 10
   PID   PPID %MEM %CPU
  2128   1980  7.9  1.9
  1181   1112  6.0  0.5
  2479   1951  5.9  0.6
  2645      1  5.2  1.3
  2555      1  2.8  0.1
  1971   1969  2.8  0.5
  2347   1951  2.7  0.1
  2395   2347  2.5  0.0
```

```
  1496   1112  2.4  0.0
```
c)Display current user logged in and logname

```
tsec11@ubuntu:~$ echo "your Logname:$(echo $LOGNAME)"
your Logname:tsec11
```
d)

```
Display current shell, home directory, operating system type, current path
setting, current working directory,


tsec11@ubuntu:~$ echo "Current shell:$echo($SHELL)"
Current shell:(/bin/bash)

tsec11@ubuntu:~$ echo "your username:$echo($USER)"
your username:(tsec11)
//cd pwd (gives present working directory)
//cd ~(gives home / main directory)
```

e)Display OS version, release number, kernel version.

```
tsec11@ubuntu:~$ uname -a #displays information about the current system
hardware platform ,name of OS and its version and so on
Linux ubuntu 5.4.0-137-generic #154~18.04.1-Ubuntu SMP Tue Jan 10 16:58:20
UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```
2.Write shell program

1.Add 2 numbers

```
echo "ADD 2 numbers"
echo "Enter number a"
read a
echo "Enter number b"
read b
c=$(($a+$b))
echo "Addition is:$c"

/* $-reads and display number stored */
```
Output:

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Desktop/Shell Progr (main)
$ sh ADD.sh
ADD 2 numbers
Enter number a
2
Enter number b
3
Addition is:5
```

2.Check if a numbered entered is even or odd

```
echo "Even odd"
echo "Enter number to be checked"
read a
if [ `expr $a % 2` == 0 ];
then
        echo "Even"
else
        echo "odd"
fi
```

Output:

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Desktop/Shell Progr (main)
$ sh EO.sh
Even odd
Enter number to be checked
4
Even

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Desktop/Shell Progr (main)
$ sh EO.sh
Even odd
Enter number to be checked
3
odd
```

3.Find the sum of n natural numbers

```
echo "Shell scripting to find sum of n Numbers"
echo "Enter value of n"
read n
i=1
sum=0
while [ $i -le $n ]
do
sum=$((sum+i))
i=$((i+1))
done
echo "Sum is:" $sum
```

Output:

```
echo "Shell scripting to find sum of n Numbers"
echo "Enter value of n"
read n
i=1
sum=0
```

```
while [ $i -le $n ]
do
sum=$((sum+i))
i=$((i+1))
done
echo "Sum is:" $sum
```

4.Determine if a person is eligible to vote.

```
echo perform to determine eligibility to vote
echo enter age
read age
if [ $age -ge 18 ];
then
        echo eligible to vote
else
        echo not eligible to vote
fi
```

Output:

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Desktop/Shell Progr (main)
$ sh vote.sh
perform to determine eligibility to vote
enter age
30
eligible to vote

ROHINI@DESKTOP-AAJD063 MINGW64 ~/Desktop/Shell Progr (main)
$ sh vote.sh
perform to determine eligibility to vote
enter age
2
not eligible to vote
```

5)Display all filenames beginning with character a and displays its contents.

```
for k in a *
do
        echo "file name is $k"
        cat $k
done
```

Output:

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Desktop/Shell Progr (main)
$ sh filea.sh
file name is a
cat: a: No such file or directory
```

```
file name is ADD,sh.txt
file name is ADD.sh
echo "ADD 2 numbers"
echo "Enter number a"
read a
echo "Enter number b"
read b
c=$(($a+$b))
echo "Addition is:$c"
file name is Add.sh.txt
echo "Enter number a"
read a
echo "Enter number b"
read b
file name is EO.sh
echo "Even odd"
echo "Enter number to be checked"
read a
if [ `expr $a % 2` == 0 ];
then
	echo "Even"
else
	echo "odd"
fifile name is fact.sh
echo program to find factorial
echo enter number n
read n
fact=1

for((i=1;i<=n;i++))
do
	fact=$((fact * i))
done

echo "factorial of $n is $fact"
file name is filea.sh
for k in a *
do
	echo "file name is $k"
	cat $k
donefile name is New Text Document.txt
cat: New: No such file or directory
cat: Text: No such file or directory
cat: Document.txt: No such file or directory
```

file name is Sn.sh

```
echo "Shell scripting to find sum of n Numbers"
echo "Enter value of n"
read n
i=1
sum=0
while [ $i -le $n ]
do
sum=$((sum+i))
i=$((i+1))
done
echo "Sum is:" $sumfile name is vote.sh
echo perform to determine eligibility to vote
echo enter age
read age
if [ $age -ge 18 ];
then
        echo eligible to vote
else
        echo not eligible to vote
fi
```

file name is vote.txt

6)Find factorial of a number

```
echo program to find factorial
echo enter number n
read n
fact=1

for((i=1;i<=n;i++))
do
           fact=$((fact * i))
done

echo "factorial of $n is $fact"
```

Output:

```
ROHINI@DESKTOP-AAJD063 MINGW64 ~/Desktop/Shell Progr (main)
$ sh fact.sh
program to find factorial
enter number n
4
factorial of 4 is 24
```

7)Check validity of a username and password with a function defined in the code .