Programs based on python

1) Run some python programs on Pi like:

a) Read your name and print Hello message with name
b) Read two numbers and print their sum, difference, product and division.
c) Word and character count of a given string.
d) Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.

a) Read your name and print Hello messag

Code:

```
name = input("Enter your name: ")

print("Hello, " + name + "!")
```

Explanation:
input() reads your name from the keyboard.
The name is stored in the variable name.
print() displays a greeting message using that name.

Example:
Input: Alice
Output: Hello, Alice!

b) Read two numbers and print their sum, difference, product and division.

Code:

```
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))

print("Sum:", a + b)
print("Difference:", a - b)
print("Product:", a * b)

if b != 0:
    print("Division:", a / b)
else:
    print("Division: Cannot divide by zero")
```

Explanation:
float() is used to accept decimal numbers.

We perform:

+ for sum

- for difference

* for multiplication

/ for division (only if second number is not zero).

Example:
Input: 8, 2
Output: Sum: 10, Difference: 6, Product: 16, Division: 4.0

c) Word and character count of a given string.

Code:

```
text = input("Enter a string: ")

word_count = len(text.split())
char_count = len(text)

print("Word count:", word_count)
print("Character count:", char_count)
```

Explanation:
text.split() breaks the sentence into words.

len() counts:

Words in the list

Characters in the string (including spaces)

Example:
Input: "Hello world"
Output: Word count: 2, Character count: 11

d) Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.

Code:

```
shape = input("Enter the shape (rectangle/triangle/circle): ").lower()

if shape == "rectangle":
    length = float(input("Enter length: "))
    width = float(input("Enter width: "))
    area = length * width

elif shape == "triangle":
    base = float(input("Enter base: "))
    height = float(input("Enter height: "))
```

```
    area = 0.5 * base * height

elif shape == "circle":
    radius = float(input("Enter radius: "))
    area = 3.14159 * radius * radius

else:
    area = None
    print("Invalid shape!")

if area is not None:
    print("Area of the", shape, "is:", area)
```

Explanation:
You enter the shape name.

Depending on the shape:

Rectangle: area = length × width

Triangle: area = ½ × base × height

Circle: area = π × radius²

Uses if-elif-else to choose the formula.

3.14159 is used as an approximate value of π.

Example:
Input: rectangle, 5, 4
Output: Area of the rectangle is: 20.0

2) Run some python programs on Pi like:

a) Print a name 'n' times, where name and n are read from standard input, using for and while loops.
b) Handle Divided by Zero Exception.
c) Print current time for 10 times with an interval of 10 seconds.
d) Read a fileline byline and print the word count of each line.

a) Print a name 'n' times, where name and n are read from standard input, using for and while loops.

Code:

```
name = input ("Enter your name: ")
n = int (input("How many times to print your name? "))
for i in range(n):
    print(name)
```

Code using while loop:

```
name = input ("Enter your name: ")
n = int(input("How many times to print your name? "))

count = 0
while count < n:
    print(name)
    count += 1
```

Explanation:
input () gets the name and how many times to print it.
The for loop runs from 0 to n-1 and prints the name.
The while loop keeps printing until count reaches n.

Example:
Enter your name: Alice
Enter how many times to print the name: 3
Output:

Alice
Alice
Alice

b) Handle Divided by Zero Exception.

Code:

```
try:
    num1 = float (input ("Enter numerator: "))
    num2 = float (input ("Enter denominator: "))
    result = num1 / num2
    print ("Result:", result)

except ZeroDivisionError:
    print ("Error: Cannot divide by zero!")
```

Explanation:
User enters two numbers.
If the second number is **zero**, dividing by it causes an error.
The program catches this using try-except and shows a friendly message.

Example:

Enter numerator: 10
Enter denominator: 0

Output:

User enters two numbers.
If the second number is **zero**, dividing by it causes an error.
The program catches this using try-except and shows a friendly message.

Example:

Enter numerator: 10
Enter denominator: 0

Output:

Error: Cannot divide by zero!

c) Print current time for 10 times with an interval of 10 seconds.

Code:

```
import time
from datetime import datetime

for i in range(10):
    current_time = datetime.now().strftime("%H:%M:%S")
    print("Time:", current_time)
    time.sleep(10)  # wait 10 seconds
```

Explanation:

Uses datetime to get current time.
Uses time.sleep(10) to wait for 10 seconds between each print.
Repeats the process 10 times.

d) Read a fileline byline and print the word count of each line.

Code:

```
# Create or use an existing text file, e.g., sample.txt

filename = "sample.txt"

with open(filename, "r") as file:
    line_number = 1
    for line in file:
        words = line.split()  # Split line into words
        count = len(words)
        print(f"Line {line_number} has {count} words.")
        line_number += 1
```

Explanation:
open(filename, "r") opens the file in read mode.

for line in file: reads the file line by line.

line.split() splits each line into a list of words.

len(words) gives the word count of that line.

Example File (sample.txt):

Python is easy to learn
It runs on Raspberry Pi

Output:
Line 1 has 5 words.
Line 2 has 5 words.

Raspberry Pi Linux Commands – Practical Worksheet (with Explanations)

Start Raspberry Pi and try various Linux commands in command terminal window: ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, pingetc. explain use of each command in simple language also explain hoe to run the same on raspbian os terminal.

Objective

To learn and practice basic **Linux commands** on the **Raspbian OS terminal** in Raspberry Pi.
This helps understand how files, folders, and processes are managed using the command line.

How to Start

1. Start your **Raspberry Pi**.
2. Open the **Terminal** (black icon) or press **Ctrl + Alt + T**.
3. You'll see a prompt like:
4. `pi@raspberrypi:~ $`
5. Type each command and press **Enter** to run.

Commands, Explanations & Practical Tasks

1 ls – List Files and Folders

Use: Displays all files and folders in the current directory.
Syntax: ls [options]

Example:

ls

Output:

Documents Downloads Pictures test.txt

Extra options:

- ls -l → shows details like size, date, and permissions
- ls -a → shows hidden files too

Task:
Run ls -l and ls -a and observe the difference.

2 cd – Change Directory

Use: Move between folders.
Syntax: cd [foldername]

Example:

cd Documents

Output:
You move inside the *Documents* folder.

Other examples:

- cd .. → go back one folder
- cd / → go to the root folder
- cd ~ → go to your home directory

Task:
Try cd / then cd ~ and note what happens.

3 touch – Create an Empty File

Use: Makes a new blank file.
Syntax: touch [filename]

Example:

touch hello.txt

Output:
Creates a new empty file named hello.txt.

Task:
Run ls to confirm the file was created.

4 mv – Move or Rename Files

Use: Moves or renames files.
Syntax: mv [source] [destination]

Examples:

mv hello.txt Documents
Moves file into “Documents” folder.
mv oldname.txt newname.txt
Renames a file.

Task:
Rename one file and move another into a folder.

5 rm – Remove Files or Folders

Use: Deletes files or folders permanently.
Syntax: rm [filename]

Examples:

rm hello.txt
rm -r foldername

- rm → deletes files
- rm -r → deletes folders and contents

Be careful – deleted files cannot be recovered!

Task:
Create a file, delete it using rm, and confirm with ls.

6 man – Manual (Help Command)

Use: Shows detailed help for any command.
Syntax: man [command]

Example:

man ls

Output:
Displays help for ls. Press **q** to exit.

Task:
Run man cd and read what it says.

7 mkdir – Make Directory

Use: Creates a new folder.
Syntax: mkdir [foldername]

Example:

mkdir testfolder

Output:
Creates a folder named “testfolder”.

Task:
Check with ls to see the new folder.

8 rmdir – Remove Empty Directory

Use: Deletes an empty folder.
Syntax: rmdir [foldername]

Example:

rmdir testfolder

Note: Works only if the folder is empty.

Task:
Try deleting a folder using rmdir.

9 tar – Archive Files

Use: Combines multiple files into one archive (backup).
Syntax: tar -cvf [filename.tar] [foldername]

Example:

tar -cvf backup.tar Documents/

Output:
Creates backup.tar containing all files in *Documents*.

Extracting:

tar -xvf backup.tar

Task:
Create and extract your own tar file.

10 gzip – Compress Files

Use: Compresses files to save space.
Syntax: gzip [filename]

Example:

gzip test.txt

→ Creates test.txt.gz

Decompress:

gunzip test.txt.gz

Task:
Compress and decompress a text file.

11 cat – View File Content

Use: Displays file content on the screen.
Syntax: cat [filename]

Example:

cat hello.txt

Output:
Shows the content of hello.txt.

Task:
Create a file with text and view it using cat.

12 more – View File Page by Page

Use: Shows large file content one screen at a time.
Syntax: more [filename]

Example:

more /etc/passwd

Press **Spacebar** for next page, **q** to quit.

13 less – Scroll Through File

Use: Similar to more but allows scrolling up/down.
Syntax: less [filename]

Example:

less /etc/passwd

Use ↑ ↓ keys to navigate, **q** to quit.

14 ps – Process Status

Use: Lists currently running programs/processes.
Syntax: ps

Example Output:

```
PID TTY          TIME CMD
1234 pts/0    00:00:00 bash
1250 pts/0    00:00:00 python
```

Task:
Run ps to see your terminal processes.

15 sudo – Superuser Command

Use: Runs a command as an administrator (root).
Syntax: sudo [command]

Example:

sudo apt-get update

Output:
Starts a system update (requires internet).

Task:
Try using sudo ls /root (lists root folder).

16 cron – Task Scheduler

Use: Runs commands automatically at set times.
Syntax: crontab -e

Example entry:

* * * * * echo "Hello from cron" >> /home/pi/cron_test.txt

Adds text every minute to the file cron_test.txt.

Task:
Set a cron job and check if the file updates.

17 chown – Change File Owner

Use: Changes who owns the file.
Syntax: sudo chown [user]:[group] [file]

Example:

sudo chown pi:pi test.txt

Task:
View ownership before and after using ls -l.

18 chgrp – Change Group Ownership

Use: Changes the group that owns a file.
Syntax: sudo chgrp [group] [file]

Example:

sudo chgrp students test.txt

Task:
Check file's group before and after with ls -l.

19 ping – Test Internet Connection

Use: Tests if a network or website is reachable.
Syntax: ping [hostname or IP]

Example:

ping -c 4 google.com

Output:

PING google.com (142.250.194.14): 56 data bytes
64 bytes from ... time=25.4 ms

Press **Ctrl + C** to stop.

Task:
Ping your college website or google.com.

Sample Practice Session

Try this full sequence to understand how commands work together:

```
mkdir mylab
cd mylab
touch file1.txt file2.txt
ls
echo "Hello Raspberry Pi" > file1.txt
cat file1.txt
tar -cvf mylab.tar file1.txt file2.txt
gzip mylab.tar
ps
ping -c 3 google.com
```

Observation Table

Sr. No	Command	Purpose	Example Output / Observation
1	ls	Lists directory contents	Displays list of files
2	cd	Change directory	Changes working folder
3	touch	Create empty file	File created successfully
4	mv	Move or rename file	File moved/renamed
5	rm	Delete file	File removed
6	man	Get help for a command	Help page opened
7	mkdir	Make new directory	Folder created
8	rmdir	Remove empty folder	Folder deleted
9	tar	Archive files	Archive created
10	gzip	Compress file	File compressed
11	cat	View file content	Text displayed
12	more	View long file pagewise	Text shown page by page
13	less	Scroll through file	Scroll up/down possible
14	ps	View running processes	Process list shown
15	sudo	Run as admin	Command runs as root
16	cron	Schedule task	Task added to cron
17	chown	Change file owner	Owner changed
18	chgrp	Change file group	Group changed
19	ping	Test network	Ping replies received