=> Shell Scripting :-

Assignment :-

Write a shell script to check if given number is prime number or not
write a shell script to check given number is even or odd
write a shell script to check given string is palindrome or not

sh file.sh —

=> Command Line Arguments

→ supply/pass values/Data/info to script file at the time of execution.

$ sh f1.sh 44

$# => total no of arg passed
$0 → get the script file
$1 => Read first cmd arg
$2 => Read semd cmd arg
$* => Read all cmd args

cmd arg → Data passing to script file

```
 cat 13-script.sh
#! /bin/bash

RESULT=$(($1+$2))

echo " Sum of Data passed is : $RESULT"

ubuntu@ip-172-31-41-37:~$ sh 13-script.sh 4 4
 Sum of Data passed is : 8
```

```
#! /bin/bash

echo "Total Args/info passed : $#"

echo "Script file name : $0"

echo "First Command Line Args : $1"

echo "Second Command Line Args : $2"

echo "All Command Line Args : $*"
```

```
# To get the total number of args
echo "Total Args/info passed : $#"

# To get the script file name
echo "Script file name : $0"

/--> comments are used to provide meta data about our scripts and commands
--> Meta data --> data about data --> info about info
--> By Looking at comments any one can easily understand our script
# is used to write line comments

Multi line comments
 <<COMMENT
this is multi line
in acsripting
COMMENT
```

=> Redirect output to a log file :-

Redirecting output can be achieved using
> (or) >> operators

$ command > log-file.txt -> overide
$ command >> log-file.txt -> append

```
date > telusko.txt

cat telusko.txt
Sat Mar  1 04:13:52 UTC 2025

 pwd > telusko.txt
cat telusko.txt
/home/ubuntu

date >> telusko.txt
cat telusko.txt
/home/ubuntu
Sat Mar  1 04:15:55 UTC 2025
```

```
#! /bin/bash

# Define log file path
LOG_FILE=myapp.log

#function to log messages

log_message(){
        local timestamp=$(date +"%Y-%m-%d %T")
        local message=$1
        echo "[$timestamp] $message" >> $LOG_FILE
}
# call log function
log_message "Script Execution Started"

echo "This is something regular message - 01"
echo "This is something regular message -02"

# simulate error
mkdirs java &> LOG_FILE

#call log function

log_message "Script Execution Completed"

---------
LOG_FILE=myapp.log

# redirect error mesgs to log file
exec 2>> $LOG_FILE

# redirect standard mesgs to log file
exec 1>> $LOG_FILE

#function to log messages
log_message(){
        local timestamp=$(date +"%Y-%m-%d %T")
        local message=$1
        echo "[$timestamp] $message" >> $LOG_FILE
}
# call log function
log_message "Script Execution Started"

echo "This is something regular message - 01"
echo "This is something regular message -02"

# simulate error
mkdirs java
ls abcd

#call log function

log_message "Script Execution Completed"
```

=> Write a shell script to print system info

```
#! /bin/bash

#This Script file is used to print system info

#display current date and time

echo "Date & Time : $(date)"

#display host name of the system

echo "Hostname : $(hostname)"

#display uptime of system
echo "System Uptime : $(uptime)"

#Display Disc usage
echo "Disk Usage:"
df -h

#Display memory usage
echo "Memory Usage:"
free -h
```

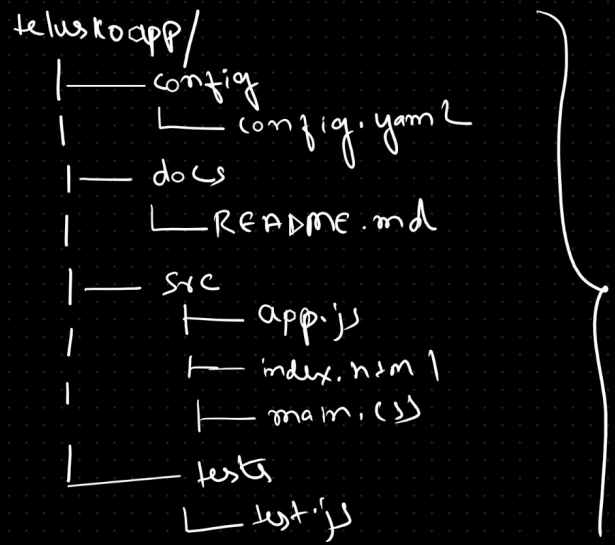
=> Shell sript for log analysis :-

```
vi log-analizer.sh
#! /bin/bash

LOG_FILE="/var/log/syslog"

#Count error messages occurances
ERROR_COUNT=$(grep -c "ERROR" "$LOG_FILE")

#printing count of error messages
#
echo "Number of error in systen log file  : $ERROR_COUNT"
```

Shell sript to creati file backup => (tar)

```
vi backup.sh
#! /bin/bash
SOURCE_DIR=/home/ubuntu/

TARGET_DIR=/home/ubuntu

echo "backup process starting..."

tar -czvf "$SOURCE_DIR/backup_$(date +%Y%m%d).tar.gz" "$TARGET_DIR"

echo "backup is completed with a tar file"
```

Write shell script to create project structure :-




```
#! /bin/bash

PROJECT_NAME="teluskoapp"
ROOT_DIR=$(pwd)


create_project(){

    mkdir $1
    mkdir $1/src
    mkdir $1/tests
    mkdir $1/docs
    mkdir $1/config
    touch $1/config/config.yaml
    touch $1/docs/README.md
    touch $1/src/app.js
    touch $1/src/main.css
    touch $1/src/index.html
    touch $1/tests/test.js
}
create_project $ROOT_DIR/$PROJECT_NAME
```

sudo apt install tree

sudo snap install tree

```
tree teluskoapp
teluskoapp
├── config
│   └── config.yaml
├── docs
│   └── README.md
├── src
│   ├── app.js
│   ├── index.html
│   └── main.css
└── tests
    └── test.js
```

=> cron : job -> cron expression crontab '+