1. echo -e "\”God! Bless us\n We are starting Shell Scripting \”"

2.man ls

3.read -p "enter your name:" n
 echo $n

4.l

5.[Vim is an editor to create or edit a text file

creating a file using vim:
Start vim by typing vim filename
To insert text press i
Now start editing text. Add new text or delete unwanted text.
One can press Esc and type :wq to save changes to a file and exit from vim]

syntax:vim redblack.c

6. pwd

7. a) [-R, --recursive ; list subdirectories recursively]
ls -R

b) ls -l
ls -lt
ls -S
ls -r
ls -lSr
ls -h

8. ls -lr > lsoutput
cat lsoutput

9. a) head -10 redblack.c
b) tail -n +10 redblack.c >> out2.txt
less out2.txt

10. ls -l >> lsoutput.txt
cat lsoutput.txt

11. ls -l |less

12. [cat command comes from its functionality to concatenate files. It can read, concatenate, and write file contents to the standard output.

Syntax: cat [OPTIONS] [FILE_NAMES]]

a) cat >> file1.txt
b) cat >> file2.txt
c) cat - file2 >> file1
cat file1

13. a) cp file1 newfile
b) cp -n file2 newfile
c) cp -i file1 newfile
d) cp -u file1 newfile
e)
f) [-s, --symbolic-link ;make symbolic links instead of copying]
cp -s file3.txt file4.txt

g) [-R, -r, --recursive ; copy directories recursively]
cp -R dir1 dir5

14. mkdir -p dir1/dir6
a) mv dir5 dir1/dir6
b) mv newfile.txt oldfile.txt
c) mv file1 dir6/file3 [syntax: mv filename destfolder/newfilename]
d) rm [a,e,i,o,u,A,E,I,O,U]*
e) rm ???*
f) rm -rf *

15. vim testfile1
a) vim testfile1
Press esc
Type :set number
b)
c)
d)

16. a)
i) chmod o+r testfile2
ii) chmod u-w testfile2
iii) chmod g=x testfile2
iv) chmod u+w,o-r,g=r testfile2

v) chmod a=rw testfile2

b) i) chmod 666 testfile3
   ii) chmod 754 testfile3
c) chmod --reference=testfile3 testfile2
d) chmod -R 664 dir1

[Permission setting in symbolic mode:
`Syntax: chmod [OPTIONS] [ugoa…][-+=]perms…[,…] FILE…`

- u - The file owner.
- g - The users who are members of the group.
- o - All other users.
- – Removes the specified permissions.
- + Adds specified permissions.
- = Changes the current permissions to the specified permissions. If no permissions are specified after the = symbol, all permissions from the specified user class are removed.
- a - All users, identical to ugo.

Permission setting in Numeric mode(3 digit):

- Owner: rwx=4+2+1=7
- Group: r-x=4+0+1=5
- Others: r=4+0+0=4]

17. [`cut` is a command-line utility that allows you to cut parts of lines from specified files or piped data and print the result to standard output. It can be used to cut parts of a line by delimiter, byte position, and character.

Syntax: `cut OPTION... [FILE]...]`

a) cut -c 50|ls -l
b) cut -d: -f 1,3 /etc/passwd

18. tr command : Translate, squeeze, and/or delete characters from standard input, writing to standard output.

a) cat > sample
   cat sample|tr [:lower:] [:upper:]
b) ls -l|tr -s ' '

c) cat file1 | tr -d "\n"

19. cat > F1
cat > F2
cat > F3
paste -d "," F1 F2 F3

20. a) find a*|wc -l
b)find -name "*.py" -type f -delete [incomplete]
c) find a* -exec cp {} /dir1/dir2 \; [incomplete]
d) find -type f -mmin 30

21. a) cat/etc/passwd|head -12
a) cat /etc/passwd|tail -7
b) cat /etc/passwd|head -3
c) cat /etc/passwd|head -5 -N
d) cat /etc/passwd|head -9|tail -1

22. a) grep abc testfile
b) grep -v abc testfile [-v : --invert-match Invert the sense of matching, to select non-matching lines.]
c) grep -l "print" *.py [-l, --files-with-matches]
d) grep -L "print" *.py [-L, --files-without-match]
e) grep -c "import" *.py [-c, --count]
f) grep -n "print" *.py [-n, --line-number Prefix each line of output with the 1-based line number within its input file]
g) grep -lR "print"
h) grep "print" *.py -A 10
grep "print" *.py -B 10

i) **ls -l | grep "^d" | cut -d" " -f 9**

23. a) read -p "enter a number:" x
read -p "enter a number:" y
echo "sum is : `expr $x + $y`"
echo "difference is : `expr $x - $y`"
echo "product is : `expr $x \* $y`"
echo "quotient is : `expr $x / $y`"
echo "reminder is : `expr $x % $y`"

b) read -p "enter a string:" s
read -p "enter the position:" p

```
read -p "enter the length:" l
echo "substring: `expr substr $s $p $l`"
```

24. a) useradd -s /bin/bash -m user2
    b) chage -E 2022-05-02 -I 5 user2
    c) chown -R user2:user2 dir2
    d)
        i) userdel user2
        ii) userdel -r user2

25. a) tar -czvf mydir.tar.gz dir1 (ls -l mydir.tar.gz)
    b) tar -xvzf mydir.tar.gz -C dir1/dir6
    c) [display and update info about the CPU processes]
        i) type top then
        ii) type top then
    d) [ps stands for process status]
        i) ps
        ii) ps -A
    e) df -h
        [df stands for disk filesystem]

Shell script :

1.

```
read -p "Enter a number: " num
if test $((num % 2)) == 0
then
echo "$num is even"
else
echo "$num is odd"
fi
```

2.

```
read -p "Enter three marks out of 100 each : " m1 m2 m3
s=$(($m1+$m2+$m3))
avg=$(echo  "scale=2;$s / 3"|bc)
echo -e "Average: $avg"
if [[ $(echo "if (${avg} >= 90) 1 else 0" | bc) -eq 1 ]]
then
echo "Grade: S"
elif [[ $(echo "if (${avg} < 90) 1 else 0" | bc) -eq 1 ]] && [[ $(echo "if (${avg} >= 80) 1 else
0" | bc) -eq 1 ]]
then
echo "Grade: A"
elif [[ $(echo "if (${avg} < 80) 1 else 0" | bc) -eq 1 ]] && [[ $(echo "if (${avg} >= 60) 1 else
0" | bc) -eq 1 ]]
then
echo "Grade: B"
elif [[ $(echo "if (${avg} < 60) 1 else 0" | bc) -eq 1 ]] && [[ $(echo "if (${avg} >= 40) 1 else
0" | bc) -eq 1 ]]
then
echo "Grade: P"
else
echo "Grade: F"
fi
```

3.

```
read -p "Enter the Indian state: " state
state=$(echo $state | tr '[:upper:]' '[:lower:]')
case $state in
"andhra pradesh")
        echo "Language: Telugu";;
"assam")
        echo "Language: Assamese";;
```

```
"bihar"|"himachal pradesh")
        echo "Language: Hindi";;
"karnataka")
        echo "Language: Kannada";;
"kerala"|"lakshadweep")
        echo "Language: Malayalam";;
"tamil nadu")
        echo "Language: Tamil";;
*)
        echo "Language: Unknown";;
esac
```

4.

```
for file in /etc/passwd
do
        result=$(grep stud* /etc/passwd)
        echo $result
        if [[ "$file1 == $result" ]]
        then
                result=$(echo "${result}"| cut -d: -f 1)
                echo $result
                sudo usermod -d /usr $result
                echo "successful"
                break
        else
                echo "unsuccessful"
        fi
done

usermod --password $(echo MY_NEW_PASSWORD | openssl passwd -1 -stdin)
USERNAME
```

5.

```
read -p "Enter a number: " num
echo "Multiplication table of $num : "
for (( i=1; i<=10; i++))
do
        val=$(( num * i ))
        echo "$i * $num = $val"
done
```

6.

```
read -p "Enter a decimal number: " n
val=0
power=1
while [ $n  -ne  0 ]
do
r=`expr $n % 2`
val=`expr $r \* $power + $val`
power=`expr $power \* 10`
n=`expr $n \/ 2`
done
echo "Binary equivalent : $val"
```