Program No : 18

AIM **:** Write a script to accept a filename while running the script and check it has the write permission, if yes prompt the user to enter a text and append the text to the given filename.

```
#!/bin/bash
filename=$1
if [ ! -e “$filename” ]; then
echo “File does not exist”;
exit 1
fi
if [ ! -w “$filename” ]; then
echo “File does not have write permission”;
exit 1
fi
echo “Enter text to append to $filename”;
read text
echo “$text”>>”$filename”;
echo “Text appended to $filename”;
```

Output

```
user@user:~$ bash perm.sh f1
Enter text to append to f1:
arpitha
Text appended to f1
user@user:~$ cat f1
Anju
Aleena
Manu
Akhil
Joseph
```

Program : 19

Aim **:** Write a shell script which displays a list of all files in the current directory to which you have read, write &amp; execute permissions.

```
#!/bin/bash
echo “List of files with read, write, and execute permissions:”;
for file in *
do
if [ -r “$file” ] && [ -w “$file” ] && [ -x “$file” ]
then
echo “$file”;
fi
done
```

Output

```
user@user:~$ bash rwe.sh
List of files with read, write, and execute permissions:
----------------------------------------------------
3A
Desktop
dir1
dir2
Documents
Downloads
file6
MBA
Music

NetBeansProjects
Pictures
Public
sif
snap
Templates
Videos
```

Program : 20

Aim : Write a shell script which receives two file names as arguments.It should check whether the two file’s contents are the same or not .If they are the same, delete the second file.

```
#!/bin/bash
if [ $# -ne 2 ]; then
echo “Usage: $0 <file1><file2>”
exit 1
fi
file1=”$1”
file2=”$2”;
if cmp –s “$file1” “$file2”;; then
echo “$file1 and $file2 have the same contents”;
rm “$file2”
else
echo “$file1 and $file2 have different contents”;
fi
```

Output

```
user@user-H510M-S2:~$ ls
20_linux.sh Downloads ‘lab installation’ Pictures Templates
Desktop file1 Music Public Videos
Documents file2 my_app snap
user@user-H510M-S2:~$ bash 20_linux.sh file1 file2
file1 and file2 have the same contents
user@user-H510M-S2:~$ ls
20_linux.sh Downloads Music Public Videos
Desktop file1 my_app snap
Documents ‘lab installation’ Pictures Templates
```

Program : 21

Aim **:** Write a shell script, which will receive any number of filenames as arguments.The shell script should check whether such files already exist.

```
#!/bin/bash
for file in “$@”;; do
if [ -e “$file” ]; then
echo “$file exists”
else
echo “$file does not exist”
fi
done
```

Output

```
user@user-H510M-S2:~$ bash 20_linux.sh file1 file
file1 exists
file does not exist
```

Program : 22

Aim **:** Write a shell script to perform operations for student data like view, add and delete records.

```
#!/bin/bash
DATABASE=”students.txt”;
function view_records {
if [ ! –f “$DATABASE” ]; then
echo “No records found”;
return 1
fi
echo “ID | NAME | GRADE”;
cat “$DATABASE”;
}
function add_record {
echo “Enter student ID:”
read id
echo “Enter student name:”;
read name
echo “Enter student grade:”;
read grade
echo “$id | $name | $grade” >> “$DATABASE”;
echo “Record added successfully”;
}
function delete_record {
if [ ! -f “$DATABASE” ]; then
echo “No records found”
return 1
fi
echo “Enter student ID to delete:”;
 read id
if grep -q “^$id” “$DATABASE”;; then sed -i
“/^$id /d” “$DATABASE”
 echo “Record deleted successfully”
 else
echo “Record not found”
fi
}
while true; do
echo “1. View records”
echo “2. Add record”
echo “3. Delete record”
echo “4. Exit”
read choice
case $choice in
1) view_records;;
2) add_record;;
3) delete_record;;
4) exit;;
*) echo “Invalid choice”;;
esac
echo
done
```

Output

```
user@user:~$ bash student.sh 1.
View records
2. Add record
3. Delete record
4. Exit
1
ID | NAME | GRADE
------------------
102 | akash | B
103 | namitha | A
101 | amal | A

1. View records
2. Add record
3. Delete record
4. Exit

2
Enter student ID:
104
Enter student name:
binoy
Enter student grade:
B
Record added successfully 1.
View records
2. Add record
3. Delete record
4. Exit
1
```

```
ID | NAME | GRADE ---
---------------
102 | akash | B
103 | namitha | A
101 | amal | A
104 | binoy | B
1. View records
2. Add record
3. Delete record
4. Exit
3
Enter student ID to delete:
104
Record deleted successfully 1.
View records

2. Add record
3. Delete record
4. Exit
1
ID | NAME | GRADE ---
---------------
102 | akash | B
103 | namitha | A
101 | amal | A
1. View records
2. Add record 3.
Delete record 4.
Exit
4
```

Program : 23

Aim : Write a shell script to sort the given numbers in descending order using Bubble sort .

```
#!/bin/bash
echo “Enter the numbers to sort (space-separated):”
read -a numbers
length=${#numbers[@]}
for (( i=0; i<length-1; i++ ))
do
for (( j=0; j<$length-i-1; j++ ))
do
if (( ${numbers[j]}< ${numbers[j+1]} ))
then
temp=${numbers[j]}
numbers[j]=${numbers[j+1]}
numbers[j+1]=$temp
fi
done
done
echo “Sorted Array (Descending Order):”
echo “${numbers[@]}”
```

Output

```
user@user:~$ bash sort.sh
Enter the numbers to sort (space-separated):
10 20 15 30 50 40
Sorted Array (Descending Order):
50 40 30 20 15 10
```

Program : 24

Aim **:** Write a shell program to find the factorial of a number using function.

```
#!/bin/bash
function fact
{
f=1
echo “enter the number”
read a
for((i=1;i<=$a;i++))
{
f=$((f*i))
}
echo “factorial is: $f”
}
Fact
```

Output

```
user@user:~$ bash facto.sh
enter the number:
5
factorial is:120
```

Program : 25

Aim **:** Write a shell program to determine whether the given string is palindrome or not using function.

```
#!/bin/bash
function is_palindrome
{
reverse=$(echo $1 | rev)
if [[ $1 == $reverse ]]
then
echo “Palindrome”
else
echo “Not Palindrome”
fi
}
echo “Enter a string to check if it is a palindrome:”
read str
result=$(is_palindrome “$str”)
echo “The given string is $result”
```

Output

```
user@user:~$ bash pal1.sh
Enter a string to check if it is a palindrome:
malayalam
The given string is Palindrome
```

Program : 26

Aim **:** Write a script to rename all c files to cpp files.

```
#!/bin/bash
ls *.c>new.temp
while read line
do
echo $line
new=${line}pp
echo $new
mv $line $new>/dev/null
echo “All .c files have been renamed to .cpp files”
 done
<new.temp
```

Output

```
user@user:~$ ls -a | grep *.c
new.c
user@user:~$ bash ctocpp.sh
new.c
new.cpp
All .c files have been renamed to .cpp files
```

PROGRAM : 27

Aim : The word “mca” is present in some of the files supplied as arguments. Write a script to search each of these files, and to stop at the first file containing the word “mca” and report it.

```
#!/bin/bash
while [ $# -ne 0 ]
do
grep -l “mca” $1
shift
done
```

OUTPUT :

```
user@user:~$ bash mca.sh file1 file2 file3
file2
```

PROGRAM : 28

Aim **:** Write a script to receive any number of filenames as arguments and to check whether the arguments supplied is a file or directory. If it is directory, it should be appropriately reported. if it is a filename then name of the file as well as the number of lines present in it should be reported.

```
#!/bin/bash
for arg in “$@”
do
if [ -d “$arg” ]; then
echo “$arg is a directory”
elif [ -f “$arg” ]; then
num_lines=$(wc -l <”$arg&”)
echo “$arg has $num_lines line(s)”
else
echo “$arg is neither a file nor a directory”
fi
done
```

Output

```
user@user:~$ ls
3c cost.sh dir2 even.sh file2 file.txt.save mark.sh
new_directory perm.sh reverse.sh snap sumdig.sh text
argu.sh cp direc.sh excp.py file3 first MBA
new_folder1 Pictures rev.sh sort.sh sum.sh textfile
char.sh ctocpp.sh dir.sh f1 file4 fruit.py mca.sh
new.temp pow.sh rwe.sh student.sh temp this
combin.sh data.db Documents f2 file5 fruit.txt mul.sh
odd.sh prgm1.sh second students.txt Templates touch
command.sh data.py Downloads facto.sh file6 hello.cpp Music pal1.sh
prgm2.sh series.sh stud.py term.sh Videos Copied Desktop echo file
file.py is NetBeansProjects pali.sh prime.sh sif stud.sh test1.py year.sh
‘Copied successfully’ dir1 edited file1 file.sh jannah new.cpp pal.sh
Public sif.txt successfully test.py
user@user:~$ bash argu.sh MBA
MBA is a directory
```

Program : 29

Aim **:** Write a script to read from a file which is supplied as a command line argument and count the number of lines and words. If there is no filename supplied, the script should accept text from the keyboard.

```
 #!/bin/bash
if [ $# -eq 1 ]; then
filename=$1
if [ -f “$filename” ]; then
num_lines=$(wc -l < “$filename”)
num_words=$(wc -w < “$filename”)
echo “File ‘$filename’ has $num_lines line(s) and $num_words word(s).” else
echo “Error: ‘$filename’ is not a valid file.”
exit 1
fi
else
echo “Enter some text (EOF to end):”
text=$(cat)
num_lines=$(echo “$text” | wc -l)
num_words=$(echo “$text” | wc -w)
echo “Input has $num_lines line(s) and $num_words word(s).”
 fi
```

Output

```
user@user:~$ bash arg.sh file2
File ‘file2’  has 2 line(s) and 3 word(s).
```

Program : 30

Aim **:** Write a shell script which receives an even number of file names. Suppose four file names are supplied then the first file should get copied into the second file, the third file should get copied into the fourth file, and so on. If odd numbers of file names are supplied then no copying should take place and an error message should be displayed.

```
#!/bin/bash
if [ $# -eq 0 ] || [ $(($# % 2)) -ne 0 ]; then
echo “Error: an even number of file names is required.”
exit 1
fi
for (( i=1; i<=$#; i+=2 ))
do
cp “${i}” “${i+1}”
echo “Copied ‘${i}’ to ‘{i+1}’”
done
```

Output

```
user@user:~$ bash evn.sh file1 file2 file3
Error: an even number of file names is required.
user@user:~$ bash evn.sh file1 file2 file3 file4
Copied ‘file1’ to ‘1’
Copied ‘file3’ to ‘1’
```

Program : 31

Aim **:** Write a script to wish the user “Good Morning, Good Afternoon and Good Evening” when he logs in to the system based on the time.

```
#!/bin/bash
hour=$(date + “%H”)
if [ $hour -ge 5 ] && [ $hour -lt 12 ]; then
echo “Good morning!”
elif [ $hour -ge 12 ] && [ $hour -lt 18 ]; then
echo “Good afternoon!”
else
echo “Good evening!”
fi
```

Output

```
user@user:~$ bash time.sh
Good evening!
```