PROGRAM 1: FILE MANIPULATION

AIM:

To write a shell script to stimulate the following command rm, cp, cat, mv, cmp, wc, split, diff.

ALGORITHM:

Step 1: Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201
Step 4: Enter the user name and password then create a program using vi filename.sh
Step 5: Create two new files using cat command
Step 6: Copy the contents from one file to another file using cp command
Step 7: Move the content from one file to another file using mv command
Step 8: Remove the file using rm command
Step 9: Compare two files using rmp command
Step 10: Use diff command to view the difference in two files
Step 11: Count the number of words in two files using wc command.
Step 12: Use split command to split the content of the file.
Step 13: stop the process.

RESULT:

Thus the Shell script has been executed and verified successfully.

PROGRAM:

```
echo "FILE MANIPULATION"
echo "********************"
while :
do
echo "Manipulation List"
echo "***************"
echo "1.Create."
echo "2.Display."
echo "3.Copy."
echo "4.Move."
echo "5.Remove."
echo "6.Compare."
echo "7.Difference."
echo "8.Word count."
echo "9.Split."
echo "Enter your choice:"
read ch
case $ch in
1)
echo "Creating file"
echo "Enter the file name:"
read filepgm1
cat $filepgm1
;;
2)
echo "Display file"
ls
;;
3)
echo "Copying file"
echo "Enter the file name:"
read filepgm1
read filepgm2
```

```
cp $filepgm1 $filepgm2
echo "File copied"
;;
4)
echo "Moving file"
echo "Enter the file name:"
read filepgm1
read filepgm2
mv $file1 $file2
echo "File Moved"
;;
5)
echo "Removing files"
echo "Enter the file name:"
read filepgm1
rm $filepgm1
echo "File Removed"
;;
6)
echo "Comparing files"
echo "Enter the file name:"
read filepgm1
read filepgm2
cmp $filepgm1 $filepgm2
echo "File compared"
;;
7)
echo "Comparing the different files"
echo "Enter the file name:"
read filepgm1
read filepgm2
diff  $filepgm1 $filepgm2
echo "File  differed"
;;
```

```
8)
echo "Counting Words"
echo "Enter the file name:"
read  filepgm1
read filepgm2
wc $filepgm1 $filepgm2
echo "Words counted"
;;
9)
echo "Splitting files"
echo "Enter the file name:"
read filepgm1
split $filepgm1
echo "File Splitted"
;;
esac
echo "Do you want to continue[Y/N]:"
read s
if [[ $s == "N" || $s == "n" ]]
then
echo "Exited"
exit
fi
done
```

OUTPUT.

```
Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
8.Word count
9.Split
Enter your choice: 1
```

Read a file

```
Enter the file name:
filepgm1.sh
kongu arts and science college
kongu nadu arts and science collge
Do you want to continue[Y/N]: Y
Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
8.Word count
9.Split
Enter your choice:  2
```

Display file

```
          filepgm2.sh    newpgm5.sh      pgm4.sh   program10.sh
internal10.sh  newpgm6.sh      pgm5.sh   program66.sh   new_directory
newprogram6.sh
Do you want to continue[Y/N]: Y

Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
8.Word count
9.Split
Enter your choice: 3
```

Copying file

```
Enter the file name:
filepgm1.sh
filepgm2.sh
File copied
Do you want to continue[Y/N]: Y

Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
```

```
8.Word count
9.Split
Enter your choice: 6
```

Comparing files

```
enter the file name:
filepgm1.sh
filepgm2.sh
filepgm1.sh filepgm2.sh differ: byte 32, line 2
File comapred
Do you want to continue[Y/N]: Y

Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
8.Word count
9.Split
Enter your choice:  7
```

Comparing difference files

```
Enter the file name:
filepgm1.sh
filepgm2.sh
2c2
< kongu nadu arts and science collge
---
>
File Differed
Do you want to continue[Y/N]: Y
```

```
Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
8.Word count
9.Split
Enter your choice: 8
```

Counting words

```
Enter the file name:
filepgm1.sh
filepgm2.sh
 2 11 66 filepgm1.sh
 2  5 32 filepgm2.sh
 4 16 98 total
Words counted
Do you want to continue[Y/N]: Y
Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
8.Word count
9.Split
Enter your choice: 9
```

```
Splitting files
Enter the file name:
filepgm1.sh
File splitted
Do you want to continue[Y/N]: Y

Manipulation List
*****************
1.Read a file
2.Display
3.Copy
4.Move
5.Remove
6.Compare
7.Difference
8.Word count
9.Split
Enter your choice: 5

Removing files
Enter the file name:
filepgm1.sh
File removed
Do you want to continue[Y/N]:N
```

PROGRAM 2: SYSTEM CONFIGURATION DETAILS

AIM:

To write a shell script to show the following system configuration A: currently logged user and logname. B=Current shell, home directory, OS type, Current path setting, Current working directory. C= Currently logged number of users and available shells. D= CPU and memory information.

ALGORITHM:

Step1 : Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201
Step 4: Enter the user name and password then create a program using vi filename.sh
Step 5: Use the logname command to find the currently logged user.
Step 6: Use $o,$HOME to find the current shell and home directory.
Step 7: Use the command uname –a, uname –r,pwd, who –q to find the operating systems, Kernal
name , Current working directoryand number of users.
Step 8: Use the following command cat\etc\shell for all available shells,lscpu for CPU information,
free –m for memory information.
Step 9: Stop the process.

RESULT:

Thus the Shell script has been executed and verified successfully.

PROGRAM

```
echo "SYSTEM CONFIGURATION DETAILS"
echo "**********************************"
echo "1.Displaying Current user."
echo "*********************"
logname
echo "2.Current shells."
echo "*************"
echo $0
echo "3.Home directory."
echo "**************"
echo $HOME
echo "4.Operating System."
echo "****************"
uname -a
echo "5.kernal name."
echo "************"
uname -r
echo "6.Current Working directories."
echo "*************************"
pwd
echo "7. Number of users."
echo "****************"
who -q
echo "8.Available Shells."
echo "***************"
cat /etc/shells
echo "9.CPU information."
echo "****************"
lscpu
echo "10.Memory information."
echo "********************"
free –m
```

OUTPUT:

```
SYSTEM CONFIGURATION DETAILS
***************************
1. Displaying current user
**************************
        ct239
2. CURRENT SHELLS
*****************
        prg2.sh
3. HOME DIRECTORY
*****************
        /home/ct239
4.OPERATING SYSTEMS
******************
            Linux  localhost.localdomain  3.10.0-1062.el7.x86_64  #1  SMP
Wed        Aug 7 18:08:02 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
5. KERNAL NAME
**************
        3.10.0-1062.el7.x86_64
6. CURRENT WORKING DIRECTORY
        /home/ct239
7. NUMBER OF USERS
******************
        ct239
        # users=1
8. AVAILABLE USERS
******************
        prg2.sh: line 26: cat/etc/shells: No such file or directory
9. CPU INFORMATION
******************
        Architecture:          x86_64
        CPU op-mode(s):        32-bit, 64-bit
        Byte Order:            Little Endian
        CPU(s):                4
        On-line CPU(s) list:   0-3
```

```
    Thread(s) per core:    2
    Core(s) per socket:    2
    Socket(s):             1
    NUMA node(s):          1
    Vendor ID:             GenuineIntel
    CPU family:            6
    Model:                 60
    Model name:            Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz
    Stepping:              3
    CPU MHz:               910.888
    CPU max MHz:           3500.0000
    CPU min MHz:           800.0000
    BogoMIPS:              6983.99
    Virtualization:        VT-x
    L1d cache:             32K
    L1i cache:             32K
    L2 cache:              256K
    L3 cache:              3072K
    NUMA node0 CPU(s):     0-3
    Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx
pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64
monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1
sse4_2 movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand
lahf_lm abm epb ssbd ibrs ibpb stibp tpr_shadow vnmi flexpriority ept
vpid fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid xsaveopt
dtherm arat pln pts md_clear spec_ctrl intel_stibp flush_l1d
```

10. MEMORY IONFORMATION

****

```
          total     used     free     shared  buff/cache   available
Mem:       3702      402     2996         41         303        3045
Swap:      3967        0     3967
```

PROGRAM 3: REDIRECTION AND PIPES COMMAND

AIM:

To write a shell script for pipes, redirection and the tee commands.

ALGORITHM

Step1 : Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201
Step 4: Enter the user name and password then create a program using vi filename.sh
Step 5: Pipe command ls is used to list out the files in direction.
Step 6: The redirection command >> is used to append the contents in already created file.
Step 7: The tee command read standard input and write it to both output of one or more files.
Step 8: stop the process.

PROGRAM :

```
echo "REDIRECTION,PIPES,TEE COMMAND"
echo "***********************************"
while :
do
echo "1.PIPES."
echo "2.REDIRECTION."
echo "3.TEE COMMAND."
echo "Enter your choice"
read ch
case $ch in
1)
echo "pipes"
ls |sort
;;
2)
echo "REDIRECTION"
echo "This is a text file">>file1
echo "This file is successfully redirected"
echo "Redirected file"
cat file1
;;
3)
echo "TEE COMMAND"
wc file1 |tee file2
echo "Content of the file after implementing tee"
cat file2
;;
*)
echo "Invaild Output"
;;
esac
echo "Do you want to continue[y/n]:"
read s
if [[ $s == "N" || $s == "n" ]]
then
echo "Exited"
exit
fi
done
```

RESULT:

Thus the Shell script has been executed and verified successfully.

OUTPUT:

```
REDIRECTION,PIPES,TEE COMMAND
**********
1.pipes
2.Redirection
3.TEE command
Enter your choice : 1

pipes
eng.sh
file
file1
file1.sh
file2
filepgm2.sh
internal10.sh
new_directory
newfile1.sh
newpgm1.sh
newpgm4.sh
newpgm5
newpgm5.sh
newpgm6.sh
newprogram6.sh
perl5
pgm10.sh
pgm1.sh
PGM2.SH
pgm3.sh
pgm4.sh
pgm5.sh
pgm66.sh
pgm6.sh
pgm7.sh
pgm8.sh
pgm9.sh
```

```
prg2.sh
program10.sh
program2.sh
program66.sh
program6.sh
Do you want to continue [y/n]: y

REDIRECTION,PIPES,TEE COMMAND
**********
1.pipes
2.Redirection
3.TEE command
Enter your choice :2

Redirection

This is the text file
This is the banana
This is our college
Welcome to college
This is information to all
This is the text file
Do you want to continue [y/n]: y

REDIRECTION,PIPES,TEE COMMAND
**********
1.pipes
2.Redirection
3.TEE command
Enter your choice :3

TEE command
  9  31 153 file1
Content of the file after implementing tee
is text file
text file
This is the text file
This is the banana
This is our college
Welcome to college
This is information to all

This is the text file
Do you want to continue [y/n]:
```

PROGRAM 4: DISPLAYING OPTIONS

AIM:

To write a shell program for displaying current date, username, file listing and directories by getting user choice.

ALGORITHM:

Step1 : Start the process

Step 2: Open the putty tool

Step 3: Type the IP address 192.168.8.201

Step 4: Enter the user name and password then create a program using vi filename.sh

Step 5: Print the currently logged username by using uname -1 command.

Step 6: Print the current date by using date command

Step 7: List out all the files and directories using ls command.

Step 8: List out all files and directories with details using ls-ls command.

Step 9: Run the shell script by using sh filename,sh

Step 10: Stop the process.

PROGRAM

```
echo "DISPLAYING OPTIONS"
echo "*********************"
while :
do
echo "OPTIONS"
echo “*********”
echo "1.Date."
echo "2.Username."
echo "3.Listing files and directories."
echo "4.Listing files and directories with details."
echo "Enter your choice:"
read ch
case $ch in
1)
echo "Date"
date
;;
2)
echo "Username"
uname -a
;;
3)
echo "Listing files and directories"
ls -1
;;
4)
echo "Listing files and directories with details"
ls -l
;;
esac
echo "Do you want to Continue [y/n]:"
read a
if [[ $a == "N" || $a == "n" ]]
then
echo "Exited"
exit
```

```
fi
done
```

RESULT:

Thus the Shell script has been executed and verified successfully.

OUTPUT:

```
DISPLAYING OPTIONS
---------------
Options
*********
1.Date
2.Username
3.Listing files and directories
4.Listing files and directories with details
Enter your choice :1
1.Date
Date:01/01/13
Do you want to continue [y/n]; Y

DISPLAYING OPTIONS
---------------
Options
*********
1.Date
2.Username
3.Listing files and directories
4.Listing files and directories with details
Enter your choice : 2

2.Username
Linux localhost.localdomain 3.10.0-1062.el7.x86_64 #1 SMP Wed Aug 7
18:08:02 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
Do you want to continue [y/n]; Y
```

```
DISPLAYING OPTIONS
---------------
Options
*********
1.Date
2.Username
3.Listing files and directories
4.Listing files and directories with details
Enter your choice: 3

3.Listing files and directories
     file1     filepgm2.sh    newfile1.sh  newpgm5     newprogram6.sh
pgm1.sh  pgm4.sh   pgm6.sh  pgm9.sh       program2.sh   temp.sh
Do you want to continue [y/n]; Y

DISPLAYING OPTIONS
---------------
Options
*********
1.Date
2.Username
3.Listing files and directories
4.Listing files and directories with details
Enter your choice : 4

4.Listing files and directories with details
total 164K
-rw-rw-r--. 1 ct239 ct239    0 Jan  1 07:28 2
-rw-rw-r--. 1 ct239 ct239   77 Jan  1  2013 eng.sh
-rw-rw-r--. 1 ct239 ct239    0 Jan  1 07:28 file
-rw-rw-r--. 1 ct239 ct239  153 Jan  1 07:27 file1
-rw-rw-r--. 1 ct239 ct239   99 Jan  1 07:42 file1.sh
-rw-rw-r--. 1 ct239 ct239   18 Jan  1 07:31 file2
-rw-rw-r--. 1 ct239 ct239   32 Jan  1  2013 filepgm2.sh
-rw-rw-r--. 1 ct239 ct239  241 Jan  1 07:22 internal10.sh
drwxrwxr-x. 2 ct239 ct239   34 Jan  1  2013 new_directory
-rw-rw-r--. 1 ct239 ct239    0 Jan  1 07:40 newfile1.sh
```

```
-rw-rw-r--. 1 ct239 ct239 1.4K Jan  1 06:07 newpgm1.sh
-rw-rw-r--. 1 ct239 ct239    0 Jan  1 06:43 newpgm4.sh
```

PROGRAM 5:   FILTER COMMANDS

AIM:
To write a shell script to implement the filter command.

ALGORITHM:
Step1 : Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201
Step 4: Enter the user name and password then create a program using vi filename.sh
Step 5: grep command is used to search for pattern in file or standard input.
Step 6: tr command replace or remove the specific character.
Step 7: cut command is used to extract sections from each line of input file.
Step 8: wc command is used to count the word in file.
Step 9: : Run the shell script by using sh filename,sh
Step 10: Stop the process.

PROGRAM

```
echo "FILTER COMMAND"
echo "******************"
while :
do
echo "1.Search a pattern (grep)"
echo "2.Replacing the text(tr)"
echo "3.Extract a column(cut)"
echo "4.Counting a word(wc)"
echo "Enter your choice:"
read ch
if [ $ch -eq 1 ]
then
```

```
echo "  "
echo "grep command"
grep This file1
elif [ $ch -eq 2 ]
then
echo “  ”
echo "tr command"
tr  "[a-z]"  "[A-Z]" <file1
elif [ $ch -eq 3 ]
then
echo "  "
echo "cut command"
cut -c 1-5 file2
elif  [ $ch -eq 4 ]
then
echo "  "
echo "Word count "
wc file2
else
echo "Invalid output"
fi
echo "Do you want to continue[y/n]:"
read s
if [[ $s == "N" || $s == "n" ]]
then
echo "Exited"
exit
fi
done
```

RESULT:

Thus the Shell script has been executed and verified successfully.

OUTPUT

```
filter command
***********
1.Search a pattern (grep)
2.Replacing the text (tr)
```

```
3.Extract a column (wt)
4.Counting a word (wc)
Enter your choice : 1

grep command
This is the text file
This is the banana
This is our college
This is information to all
This is the text file
Do you want to continue [Y/N]; Y

filter command
***********
1.Search a pattern (grep)
2.Replacing the text (tr)
3.Extract a column (wt)
4.Counting a word (wc)
Enter your choice :2


tr command
IS TEXT FILE
TEXT FILE
THIS IS THE TEXT FILE
THIS IS THE BANANA
THIS IS OUR COLLEGE
WELCOME TO COLLEGE
THIS IS INFORMATION TO ALL

THIS IS THE TEXT FILE
Do you want to continue [Y/N]; y

filter command
***********
1.Search a pattern (grep)
2.Replacing the text (tr)
```

```
3.Extract a column (wt)
4.Counting a word (wc)
Enter your choice: 3

cut command
is te
text
This
This
This
Welco
This
This
Do you want to continue [Y/N]; y

filter command
***********
1.Search a pattern (grep)
2.Replacing the text (tr)
3.Extract a column (wt)
4.Counting a word (wc)
Enter your choice: 4

Word count
  9  31 153 file1
Do you want to continue [Y/N];
```

PROGRAM 6:

AIM:

To write a shell script to count and reports the occurrence of each word in the file using command line arguments.

ALGORITHM:

Step1 : Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201

Step 4: Enter the user name and password then create a program using vi filename.sh

Step 5: The grep command is used to search for a string in groups of files.

Step 6: awk command is used to scan a file line by line and count the occurrence of each word in the file.

Step 9: Run the shell script by using sh filename,sh

Step 10: Stop the process.

PROGRAM:

```
if [ $# -ne 1 ];

then

echo "USAGE:";

exit -1

fi

filename=$1

egrep -o "\b[[:alpha:]]+\b" $filename | \

awk '{ count[$0]++ }

END {printf("%-14s%s\n","Word","Count") ;

for(ind in count)

{ printf("%-14s%d\n", ind,count[ind]); }

}'
```

RESULT:

Thus the Shell script has been executed and verified successfully.

OUTPUT:

```
[tamil@localhost ~]$ sh newpgm66.sh temp1.sh

Word        Count

Arts          1

College      2

Science      1
```

PROGRAM 7: FACTORIAL OF A NUMBER

AIM:

To Write a shell script to find the factorial of a given number.

ALGORITHM:

Step1 : Start the process

Step 2: Open the putty tool

Step 3: Type the IP address 192.168.8.201

Step 4: Enter the user name and password then create a program using vi filename.sh

Step 5: Get a number from user and calculate the factorial using below formula fact=**$((**fact * num**))** and num=$((num - 1)).

Step 6: Run the shell script by using sh filename,sh

Step 7: Stop the process.

PROGRAM:

```
clear
echo "enter the number"
read num
fact=1
while [ $num -gt 1 ]
do
fact=$((fact*num))
num=$((num - 1))
done
echo $fact
```

RESULT:

Thus the Shell script has been executed and verified successfully.

OUTPUT:

```
Enter the number : 5
120
```

PROGRAM 8: FINDING THE GREATEST NUMBER

AIM:

To Write a shell script to find the greatest among the given set of numbers using command line arguments.

ALGORITHM

Step1 : Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201
Step 4: Enter the user name and password then create a program using vi filename.sh

Step 5: Read the 3 integer values from the user and find the which one is the biggest number.
Step 6: If all integers are zero, then print “no arguments given”.
Step 7:Use gt command to find the greatest number and print which one is the greatest.
Step 8:Run the shell script by using sh filename,sh
Step 7: Stop the process.

PROGRAM:

```
echo "GREATEST NUMBER"
echo “********************”
if [ $# -eq 0 ]
then
echo "No arguments given"
echo "Execute script with arguments"
big=0
else
big=$1
fi
for var in "$@"
do
if [ $var -gt $big ]
then
big=$var
fi
done
echo "Greatest number is :$big"
```

RESULT:

Thus the Shell script has been executed and verified successfully.

OUTPUT:

```
[tamil@localhost ~]$ sh newpgm88.sh   20 68 95
```

```
Greatest number is: 95
```

PROGRAM 9: PALINDROME CHECKING

AIM:

To write a shell script for palindrome checking.

ALGORITHM

Step1 : Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201
Step 4: Enter the user name and password then create a program using vi filename.sh
Step 5: Get the string from user for palindrome checking.
Step 6: Compare original and reverse string. If the original string is equal to the reverse string, then print the result “string is a palindrome”.
Step 7: If the original string is not equal to the reverse string, then print the result “string is not a palindrome”.
Step 8:Run the shell script by using sh filename,sh
Step 7: Stop the process.

PROGRAM

```
clear
echo "Enter a string:"
read input
reverse=""
len=${#input}
for(( i=$len-1; i>=0; i-- ))
do
reverse="$reverse${input:$i:1}"
done
if [ $input == $reverse ]
```

```
then
echo "$input is palindrome"
else
echo "$input is not palindrome"
fi
```

OUTPUT

```
Enter a string: madam
madam is palindrome

Enter a string: computer
computer is not palindrome
```

RESULT:

Thus the Shell script has been executed and verified successfully.

PROGRAM 10 : MULTIPLICATION TABLE

AIM:

To write the shell script to print the multiplication table using for loop.

ALGORITHM

Step1 : Start the process
Step 2: Open the putty tool
Step 3: Type the IP address 192.168.8.201
Step 4: Enter the user name and password then create a program using vi filename.sh
Step 5: Get the table number and row limit for print the multiplication table.
Step 6: Using for loop to calculate the multiplication table.

Step 7: Run the shell script by using sh filename,sh
Step 8: Stop the process.

PROGRAM:

```
clear
echo Enter a number:
read n
for i in {1..12}
do
val=`expr $i \* $n`
echo "$i*$n = $val"
done
```

OUTPUT

```
Enter a number:
8
1*8 = 8
2*8 = 16
3*8 = 24
4*8 = 32
5*8 = 40
6*8 = 48
7*8 = 56
8*8 = 64
9*8 = 72
10*8 = 80
11*8 = 88
12*8 = 96
```

RESULT

Thus the Shell script has been executed and verified successfully.