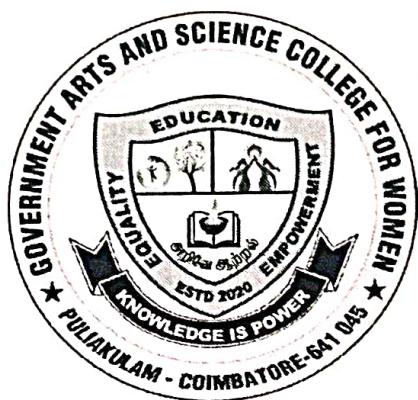


**GOVERNMENT ARTS AND SCIENCE COLLEGE FOR WOMEN**  
(AFFILIATED TO BHARATHIAR UNIVERSITY)  
PULIAKULAM  
COIMBATORE - 641 045



**DEPARTMENT OF COMPUTER SCIENCE  
B.Sc., Computer Science  
IV - SEMESTER**

**LINUX AND SHELL  
PROGRAMMING LAB  
PRACTICAL RECORD NOTE**

**Reg. No. : 2022kn887**

**Name : R.E.ANUSICA THERARAL**

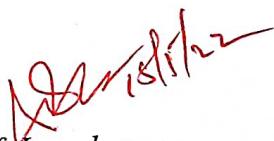
**MAY – 2022**

**GOVERNMENT ARTS AND SCIENCE COLLEGE  
FOR WOMEN**  
(AFFILIATED TO BHARATHIAR UNIVERSITY)  
**PULIAKULAM  
COIMBATORE - 641 045**

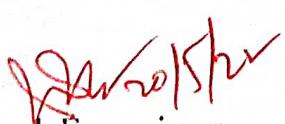
**CERTIFICATE**

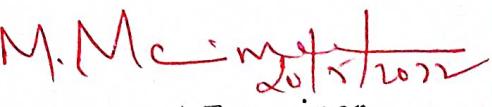
This is to certify that it is a bonafide record of practical work done by Ms. **R.E.ANNISKHA.THEBARAL.....** of **II - B.Sc., Computer Science, IV Semester** in the Computer Lab has submitted her **LINUX AND SHELL PROGRAMMING LAB (43P)** - during the academic year 2021-2022.

  
Head of the Department  
கலை மன்ற அறிவியல் கல்லூரி  
அந்திரா மாநில கல்லூரி குழுமம்  
புதுச்சேரி, புதுச்சேரி - 641045.

  
Staff In-charge

Submitted to the Bharathiar University practical Examination held on **20.05.2022** at Government Arts and Science College for Women, Puliakulam – 641 045 for the academic year 2021-2022.

  
Internal Examiner

  
External Examiner

## INDEX

S. No.	DATE	TITLE	PAGE No.	FACULTY SIGN
1.	11/3/22	File Manipulation Commands	1	
2.	16/3/22	System Configuration Commands	2	
3.	18/3/22	Implementation of Pipe, Redirection and Tee Commands	3	
4.	21/3/22	Shell Script for Displaying Date, Username, Listing of Files and Directories	4	
5.	25/3/22	Implementation of Filter Commands	5	
6.	11/4/22	Deleting File Which Has Zero File Size	6	
7.	20/4/22	Finding the Greatest Among Given Numbers Using Command Line Arguments	7	
8.	27/4/22	Finding the Sum of Individual Digits of Given Number	8	
9.	29/4/22	Shell Script for Checking the Given String or Number is a Palindrome or Not	9	
10.	02/5/22	Printing the Multiplication Table Using For Loop	10	

## FILE MANIPULATION COMMANDS

### AIM :

To write a shell script to stimulate the basic Linux Commands : rm, cmp, cat, cp, mv, wc, split diff.

### ALGORITHM :

- Step 1: Start the process
- Step 2: In the shell Prompt window perform the required commands
- Step 3: Use ls Command to list the files and directories in the current directory.
- Step 4: Copy the contents of the files to file 3, Use Cp Command.
- Step 5: Use Cat Command to view the copied Content in the file 3.
- Step 6: Use mv Command to move the contents in file 2 to file 4. And view the contents in file 4.
- Step 7: Use wc Command to display Count of the lines, words, character in the file file4.txt
- Step 8: Use wc -l, wc -w, wc -c to display the Count of lines, word, character respectively.

## 1.FILE MANIPULATION COMMANDS

```
echo "list"
ls
echo "word count"
wc file1.txt
echo "compare"
cmp file1.txt file2.txt
echo "content of file1.txt"
cat file1.txt
echo "copy"
cp file1.txt file4.txt
echo "content of file4.txt"
cat file4.txt
echo "split"
split-2 file2.txt
echo "split files"
ls
echo "content of xaa"
cat xaa
echo "different"
echo "difference of file1.txt and file2.txt"
diff file1.txt file2.txt
echo "list"
ls
echo "remove"
rm file3.txt
cat file3.txt
ls
echo "concatinate"
cat file1.txt file2.txt > x.txt
cat x.txt
echo "display the number at character in file1.txt"
wc-c file1.txt
echo "splitting the file1.txt by 3lines in each"
split-3 file1.txt
```

**OUTPUT:**

```
Press ENTER or type command to continue

dhars@DESKTOP-20BG88M ~
$ chmod +x prog1.txt

dhars@DESKTOP-20BG88M ~
$ ./prog1.txt
list
dee.txt deel.txt file1.txt file2.txt file4.txt file6.txt file7.txt prog1.txt prog4.txt prog5.txt
word count
2 5 36 file1.txt
compare
file1.txt file2.txt differ: char 1, line 1
content of file1.txt
welcome to linux shell programming

copy
content of file4.txt
welcome to linux shell programming

split
./prog1.txt: line 14: split-2: command not found
split files
dee.txt deel.txt file1.txt file2.txt file4.txt file6.txt file7.txt prog1.txt prog4.txt prog5.txt
content of xaa
cat: xaa: No such file or directory
different
difference of file1.txt and file2.txt
1,2c1
< welcome to linux shell programming
<
---
> linux is open source operating system
list
dee.txt deel.txt file1.txt file2.txt file4.txt file6.txt file7.txt prog1.txt prog4.txt prog5.txt
remove
rm: cannot remove 'file3.txt': No such file or directory
cat: file3.txt: No such file or directory
dee.txt deel.txt file1.txt file2.txt file4.txt file6.txt file7.txt prog1.txt prog4.txt prog5.txt
concative
welcome to linux shell programming

linux is open source operating system
display the number at character in file1.txt
./prog1.txt: line 32: wc-c: command not found
splitting the file1.txt by 3lines in each
./prog1.txt: line 34: split-3: command not found

dhars@DESKTOP-20BG88M ~
```

Result : Thus the given program was executed successfully.

## SYSTEM CONFIGURATION COMMANDS

### AIM :

To write a shell script to implement the user and the System information by commands.

### ALGORITHM :

Step 1: Start the process

Step 2: In the Shell Script prompt the following command : use LOGINNAME to check loginname of the user, SHELL to check the following shell information.

Step 3: Type OSTYPE Command to check the Ostype of the Linux OS.

Step 4: Type path to check the path for particular directories

Step 5: lscpu command to check the CPU information.

Step 6: Type pwd command to view the Present working directory.

Step 7: Stop the process.



## 2.SYSTEM CONFIGURATION COMMANDS

```
echo "user name" $USER  
echo "login name" $LOGNAME  
echo "current shell" $SHELL  
echo "lists at shell"  
echo "home directory:" $HOME  
echo "our PC OS is:" $OSTYPE  
echo "current path" $PATH  
echo "current directory"  
pwd  
echo "list of logged users"  
who| wc-l  
echo system configuration (or) PC configuration  
echo "free memory space"  
echo "showing all memory information"  
cat|proc|meminfo  
echo "kernal name is"  
uname-s  
echo "processor type is"  
uname-p  
echo "release is"  
uname-r  
echo "USB information"  
echo "H/W list with memory info"
```



## OUTPUT:

```
sh-4.3$ ./prog2.txt
ELIDE: Failed to source defaults.vim
Press ENTER or type command to continue

sh-4.3$ cd /root/prog2.txt
sh-4.3$ ./prog2.txt
user name char
login name
current shell /bin/bash
lists at shell
one directory; three others
for K in $(cat)
current path /usr/local/bin:/usr/bin:/cygdrive/c/Program Files/Common Files/Oracle/Java/javapath:/cygdrive/c/windows/system32:/cygdrive/c/windows/cygdrive/c/windows/System32/Maven:/cygdrive/c/
finds System32/windows/shell/1.0:/cygdrive/c/windows/System32/SpoolSS:/cygdrive/c/Users/char/AppData/Local/Microsoft/Windows/Apps
current directory
three others
list of logged users
prog2.txt has 11: no file or directory not found
prog2.txt has 11: syntax error near unexpected token `('
prog2.txt has 11: icon system configuration (or) K configuration
```

Result: Thus the given Program was executed  
Successfully ✓

## REDIRECTION AND TEE COMMANDS

### AIM:

To write a Shell Script to Implementation  
the following Pipe, redirection and tee Commands.

### ALGORITHM :

Step 1: Start the process

Step 2: use (|) Pipe Command to link one  
command with another (or) one operation

Step 3: use (>>) Command to transfer the file  
from one part into another.

Step 4: use more command to check the  
information on the screen.

Step 5: Display the result on the Screen

Step 6: Display the process

Step 7: Stop the process



### 3. IMPLEMENTATION OF PIPE,REDIRECTION AND TEE COMMANDS

```
echo "exploring pipes and redirection commands"
echo "using pipe command"
ls -l|wc
echo "using tee command:"
ls -l|tee new.txt
echo "content in new.txt"
cat new.txt
echo "using redirection commands:"
echo "content in file1.txt before redirecting:"
cat file1.txt
echo "redirecting commands is to be executed...."
cat>>file1.txt>>new.txt
echo "the content in file2.txt after appending:"
cat file2.txt
```



## OUTPUT:

```

S vi prog3.txt
E1187: Failed to source defaults.vim
Press ENTER or type command to continue

dhars@DESKTOP-20BG88M ~
S chmod +x prog3.txt

dhars@DESKTOP-20BG88M ~
S ./prog3.txt
exploring pipes and redirection commands
using pipe command
      15      128      728
using tee command:
total 14
-rw-r--r--  1 dhars  dhars   32 Apr  28 12:29 dee.txt
-rw-r--r--  1 dhars  dhars   55 Apr  28 12:29 deel.txt
-rw-r--r--  1 dhars  dhars   36 May   6 19:30 file1.txt
-rw-r--r--  1 dhars  dhars   38 May   6 19:30 file2.txt
-rw-r--r--  1 dhars  dhars   36 May   6 19:32 file4.txt
-rwxr-xr-x  1 dhars  dhars  223 Apr  29 13:35 file6.txt
-rwxr-xr-x  1 dhars  dhars  439 May   2 12:24 file7.txt
-rw-r--r--  1 dhars  dhars   60 May   6 20:09 new.txt
-rwxr-xr-x  1 dhars  dhars   625 May   6 19:32 prog1.txt
-rwxr-xr-x  1 dhars  dhars   525 May   6 20:00 prog2.txt
-rwxr-xr-x  1 dhars  dhars   399 May   6 20:10 prog3.txt
-rwxr-xr-x  1 dhars  dhars   413 Apr   25 11:15 prog4.txt
-rwxr-xr-x  1 dhars  dhars   666 Apr   28 11:57 prog5.txt
-rw-r--r--  1 dhars  dhars   74 May   6 19:32 x.txt
content in new.txt
total 14
-rw-r--r--  1 dhars  dhars   32 Apr  28 12:29 dee.txt
-rw-r--r--  1 dhars  dhars   55 Apr  28 12:29 deel.txt
-rw-r--r--  1 dhars  dhars   36 May   6 19:30 file1.txt
-rw-r--r--  1 dhars  dhars   38 May   6 19:30 file2.txt
-rw-r--r--  1 dhars  dhars   36 May   6 19:32 file4.txt
-rwxr-xr-x  1 dhars  dhars  223 Apr  29 13:35 file6.txt
-rwxr-xr-x  1 dhars  dhars  439 May   2 12:24 file7.txt
-rw-r--r--  1 dhars  dhars   60 May   6 20:09 new.txt
-rwxr-xr-x  1 dhars  dhars   625 May   6 19:32 prog1.txt
-rwxr-xr-x  1 dhars  dhars   525 May   6 20:00 prog2.txt
-rwxr-xr-x  1 dhars  dhars   399 May   6 20:10 prog3.txt
-rwxr-xr-x  1 dhars  dhars   413 Apr   25 11:15 prog4.txt
-rwxr-xr-x  1 dhars  dhars   666 Apr   28 11:57 prog5.txt
-rw-r--r--  1 dhars  dhars   74 May   6 19:32 x.txt
using redirection commands:
content in file1.txt before redirecting:
welcome to linux shell programming
redirecting commands is to be executed....
```

Result: ✓  
 Thus the given program was executed successfully.

# EX NO:4 SHELL SCRIPT FOR DISPLAYING

Date : 21/3

17

DATE, USERNAME, FILES AND DIRECTORIES.

## AIM

To write a shell script to display the current date, username and list of files and directories by getting user's choice.

## ALGORITHM

Step 1: Start the process

Step 2: Use a case statement for performing a different action into a single prompt.

Step 3: Declare case variable and case command for the program.

Step 4: If the choice is 1 then the current date will be displayed on the screen.

Step 5: If the choice is 2 then username will be shown if the choice is 3 then file can be listed along with the directories

Step 6: If name of the choice is met finally default case get executed on the screen.

Step 7: Stop the process.



#### 4.SHELL SCRIPT FOR DISPLAYING DATE,USERNAME,LISTING OF FILES AND DIRECTORIES

```
echo "1. current date:"  
echo "2. your user name:"  
echo "3. List files and directories:"  
echo "4. calender:"  
echo "5. current directory:"  
echo "enter your choice."  
read option  
case $option in  
1) echo "current date is :" $(date);;  
2) echo "user name is :" $(whoami);;  
3) echo "To listout all files and directories :" $(ls);;  
4) echo "calender is :" $(cal);;  
5) echo "current directory is :" $(pwd);;  
esac
```



## OUTPUT:

```
dhars@DESKTOP-20BG68M ~
$ chmod +x prog4.txt

dhars@DESKTOP-20BG68M ~
$ ./prog4.txt
1. current date:
2. your user name:
3. List files and directories:
4. calender:
5. current directory:
enter your choice:
1
current date is : Fri May 6 20:12:04 IST 2022

dhars@DESKTOP-20BG68M ~
$ ./prog4.txt
1. current date:
2. your user name:
3. List files and directories:
4. calender:
5. current directory:
enter your choice:
2
user name is : dhars

dhars@DESKTOP-20BG68M ~
$ ./prog4.txt
1. current date:
2. your user name:
3. List files and directories:
4. calender:
5. current directory:
enter your choice:
4
calender is : May 2022 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

dhars@DESKTOP-20BG68M ~
$ ./prog4.txt
1. current date:
2. your user name:
3. List files and directories:
4. calender:
5. current directory:
enter your choice:
4
calender is : May 2022 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

Result: Thus the given Program was executed successfully.

Ex NO: 5  
Date : 25/3

## IMPLEMENTATION OF FILTER COMMANDS

### AIM:

To write a shell script to implementation filter commands.

### ALGORITHM:

Step 1: Start the process

Step 2: Create a file using vi editor

Step 3: Copy the file/etc/passwd to passwd file

Step 4: To display the lines containing the word root use grep -c "root" passwd.

Step 5: To display all the no. of lines containing the word root use grep -c "root" passwd.

Step 6: To display all lines, words, characters, in passwd file use wc passwd.

Step 7: To display all the lines that do not match with the lines root use grep -v "root" passwd.

Step 8: To replace ":" with "\*" in the file passwd use tr ":" "\*" < passwd.

Step 9: To display the first column of the file passwd use cut -d ":" -f1 passwd.



## 5. IMPLEMENTATION OF FILTER COMMANDS

```
touch dee.txt deel.txt
echo "welcome to the world of computer"
echo "hello good morning how are you?" >dee.txt
echo "one two three four five alpha beta gamma delta epsilon" >deel.txt
echo "display the lines that match the specified pattern "
grep -n "good" dee.txt
echo "display the number of lines that matched the pattern "
grep -c "good" dee.txt
echo "display the lines that do not match the pattern "
grep -v "good" dee.txt
echo "count of words,lines and character "
wc dee.txt
echo "to extract a column "
cut -f 2-4 deel.txt
echo -c 6 deel.txt
echo "display the contents of file"
cat dee.txt
echo "Display in uppercase "
tr "[lower:]" "[upper:]< dee.txt
```



**OUTPUT:**

```
charan@DESKTOP-20BG68M ~
$ chmod +x prog5.txt

charan@DESKTOP-20BG68M ~
$ ./prog5.txt
welcome to the world of computer
display the lines that match the specified pattern
Hello good morning how are you?
display the number of lines that matched the pattern
1
display the lines that do not match the pattern
count of words, lines and character
1 6 32 dee.txt
to extract a column
one two three four five alpha beta gamma delta epsilon
wc 6 deel.txt
display the contents of file
Hello good morning how are you?
Display in uppercase
HELLO GOOD MORNING HOW ARE YOU?

charan@DESKTOP-20BG68M ~
$
```

✓ ✓

Result: Thus the given program was executed  
Successfully.

# DELETING FILE WHICH HAS ZERO FILE SIZE

## AIM :

To write a shell script to remove the file which has zero file size.

## ALGORITHM

- Step 1: Start the process
- Step 2: Create a file using vi editor
- Step 3: Get a file name as input from the user
- Step 4: use the if....else statement check the Condition
- Step 5: if the file exists then check the size of the file
- Step 6: if the size of file is zero then remove the file
- Step 7: Remove the file using rm Command if not leave it as it is.
- Step 8: Stop the process.



## 6.DELETING FILE WHICH HAS ZERO FILE SIZE

```
echo "enter the filename:"  
read fnm  
if [ -e $fnm]  
then  
echo $fnm "file exist"  
if [ -s $fnm]  
then  
echo $fnm "file has size>0"  
else  
rm $fnm  
echo $fnm "file is deleted which has size=0"  
fi  
else  
echo "file not exist"  
fi
```



## OUTPUT:

```
dhars@DESKTOP-20BG88M ~
$ chmod +x prog6.txt

dhars@DESKTOP-20BG88M ~
$ ./prog6.txt
enter the filename:
prog2.txt
prog2.txt file exist
prog2.txt file has size > 0

dhars@DESKTOP-20BG88M ~
$ ./prog6.txt
enter the filename:
prog11.txt
file not exist

dhars@DESKTOP-20BG88M ~
$
```

Result: Thus the given program was executed successfully.

AIM:

To write a shell script to find the greatest among the given number using command line argument.

ALGORITHM:

Step 1: Start the process.

Step 2: Create a file using vi editor

Step 3: Using echo print the statement

Step 4: get the file name as input from the user

Step 5: Using the for loop check the condition and Print the numbers stored in the array.

Step 6: Using if print the statement as greater and smaller numbers.

Step 7: Print the smallest numbers and largest numbers.

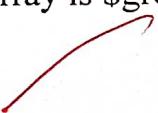
Step 8: The output will be displayed on the screen.

Step 9: Read the total number count for an array Can be got from user using Command line arguments.

Step 10: Stop the process.

## 7.FINDING THE GREATEST AMONG GIVEN NUMBERS USING COMMAND LINE ARGUMENTS

```
echo "Enter the size of array "
read n
for((i=0;i<n;i++))
do
echo "Enter ${((i+1))} number"
read nos[$i]
done
echo "Number is"
for((i=0;i<n;i++))
do
echo ${nos[$i]}
done
small=${nos[0]}
greatest=${nos[0]}
for((i=0;i<n;i++))
do
if [ ${nos[$i]} -lt $small ];
then
small=${nos[$i]}
elif [ ${nos[$i]} -gt $greatest ];
then
greatest=${nos[$i]}
fi
done
echo " smallest number of array is $small "
echo " Greatest number of array is $greatest "
```



**OUTPUT:**

```
dhar@DESKTOP-20BG88M ~
$ vi prog7.txt
E1187: Failed to source defaults.vim
Press ENTER or type command to continu
dhar@DESKTOP-20BG88M ~
$ chmod +x prog7.txt

dhar@DESKTOP-20BG88M ~
$ ./prog7.txt
Enter the size of array
3
Enter 1 number
12
Enter 2 number
54
Enter 3 number
31
Number is
12
54
31
smallest number of array is 12
Greatest number of array is 54
dhar@DESKTOP-20BG88M ~
$
```

Result:

This the given program was executed  
Successfully.

AIM :

To write a shell script to find the sum of individual digit.

ALGORITHM :

Step 1: Start the process

Step 2: Create a file using vi editor

Step 3: Using echo print the statement "Enter the number".

Step 4: Read the value from user whose summation of individual digits can be found.

Step 5: Declare & initialize the variable S<sub>et</sub> = 0 and Sum = 0.

Step 6: Find n%10, n/10, Sum = Sum + S<sub>d</sub> and Store it in S<sub>d</sub>, n, Sum respectively

Step 7: Repeat the step 6 until n greater than 0 using while loop.

Step 8: Display the output

Step 9: Stop the process.



## 8.FINDING THE SUM OF INDIVIDUAL DIGITS OF GIVEN NUMBER

```
echo -n "Enter a number:"  
read n  
sd=0  
sum=0  
while [ $n -gt 0 ]  
do  
    sd=$(( $n % 10 ))  
    n=$(( $n / 10 ))  
    sum=$(( $sum + $sd ))  
done  
echo "Sum of all digits is "$sum
```



## OUTPUT:

```
dhars@DESKTOP-20BG88M ~
$ vi prog8.txt
E1187: Failed to source defaults.vim
Press ENTER or type command to continue

dhars@DESKTOP-20BG88M ~
$ chmod +x prog8.txt

dhars@DESKTOP-20BG88M ~
$ ./prog8.txt
Enter a number:45
Sum of all digits is 9

dhars@DESKTOP-20BG88M ~
$
```



Result: Thus the given program was executed successfully.



Ex No: 9

Date : 29/4

## SHELL SCRIPT FOR CHECKING

THE GIVEN STRING OR NUMBER  
IS A PALINDROME OR NOT

43.

### AIM:

To write a shell Script for palindrome checking.

### ALGORITHM:

Step 1: Start the process

Step 2: Get the String from the user

Step 3: Reverse the String using rev<<string

Step 4: if the reversed String and the given String is same display it as palindrome

Step 5: else display not a palindrome for this use if else

Step 6: Stop the process.



## 9. SHELL SCRIPT FOR CHECKING THE GIVEN STRING OR NUMBER IS A PALINDROME OR NOT

```
echo "Enter a string"
read name
name1=$( echo $name | rev )
if [ $name = $name1 ]
then
echo "$name is a palindrome"
else
echo "$name is not a palindrome"
fi
```



**OUTPUT:**

```
dhars@DESKTOP-20BG88M ~
$ chmod +x prog9.txt

dhars@DESKTOP-20BG88M ~
$ ./prog9.txt
Enter a string
madam
madam is a palindrome

dhars@DESKTOP-20BG88M ~
$ ./prog9.txt
Enter a string
welcome
welcome is not a palindrome

dhars@DESKTOP-20BG88M ~
$
```

Result: Thus the given program was  
executed successfully.

EX NO:10  
Date : 21/5

29

## PRINTING THE MULTIPLICATION TABLE USING FOR LOOP.

### AIM

To write a Shell Script to print multiplication tables.

### ALGORITHM

Step 1 : Start the process

Step 2 : Get the table number and range from the user

Step 3 : Use for loop to find the table of the given number and range.

Step 4 : Display the calculated table in the standard table format

Step 5 : Stop the process.



## 10.PRINTING THE MULTIPLICATION TABLE USING FOR LOOP

```
echo "Enter the table number"
read n
echo "Multiplication table for $n upto the range $range"
for((i=1;i<=10;i++))
{
echo "$i X $n = ` expr $n \* $i` "
}
```

## OUTPUT:

```
dhans@DESKTOP-20BG8SM ~
$ vi prog10.txt
E1187: Failed to source defaults.vim
Press ENTER or type command to continue

dhans@DESKTOP-20BG8SM ~
$ chmod +x prog10.txt

dhans@DESKTOP-20BG8SM ~
$ ./prog10.txt
Enter the table number
3
Multiplication table for 3 upto the range
1   X   3 = 3
2   X   3 = 6
3   X   3 = 9
4   X   3 = 12
5   X   3 = 15
6   X   3 = 18
7   X   3 = 21
8   X   3 = 24
9   X   3 = 27
10  X  3 = 30

dhans@DESKTOP-20BG8SM ~
$
```

✓

Result: Thus the given program was  
executed successfully.

✓