



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
COLLEGE OF SCIENCE AND HUMANITIES**

Ramapuram, Chennai

**Department of Computer Science & Applications
(B.Sc-Computer Science)**

PRACTICAL RECORD

NAME

:

REGISTER NO

:

COURSE

:

B.Sc – COMPUTER SCIENCE

SEMESTER / YEAR

:

V / III

SUBJECT CODE

:

USA20501J

SUBJECT NAME

:

WEB PROGRAMMING LAB

OCTOBER / NOVEMBER 2024



COLLEGE OF SCIENCE AND HUMANITIES

Ramapuram, Chennai

**Department of Computer Science & Applications
(B.Sc-Computer Science)**

REGISTER NUMBER:

BONAFIDE CERTIFICATE

This is to certify that the bonafide work done by _____ in the subject
WEB PROGRAMMING [USA20501J] at, SRM Institute of Science and Technology, Ramapuram,
Chennai in October/November 2024.

STAFF IN-CHARGE

HEAD OF THE DEPARTMENT

Submitted for the University Practical Examination held at SRM Institute of Science and
Technology, Ramapuram, Chennai on _____.

INTERNAL EXAMINER 1

INTERNAL EXAMINER 2

Index

S.NO	DATE	EXPERIMENTS	SIGNATURE
1	26.06.2024	Work with Linux Server	
2	27.06.2024	Work with Files and Directory Commands	
3	03.07.2024	Work with File Commands, Creating and Modifying Files using Vi editor	
4 (a)	04.07.2024	Perform Arithmetic Operations	
4 (b)	04.07.2024	Display Bio-Data	
4 (c)	10.07.2024	Find Area of Circle	
4 (d)	10.07.2024	Find Area of Square	
4 (e)	11.07.2024	Find Sum of N Numbers	
4 (f)	11.07.2024	Find Sum of Even Numbers	
4 (g)	17.07.2024	Find Sum of Odd Numbers	
5 (a)	18.07.2024	Find given no. is Armstrong Number or not	
5 (b)	24.07.2024	Find given no. is Prime or not	
5 (c)	31.07.2024	Prepare Multiplication Table	
6	07.08.2024	Embedding PHP Script in Html	
7 (a)	08.08.2024	Implement Functions with Parameters	
7 (b)	14.08.2024	Prepare Gross Salary of Employee	
7 (c)	15.08.2024	Generate Electricity Bill using the Function	
7 (d)	28.08.2024	Find given no is Prime or not using Functions	
8	29.08.2024	Perform various String Operations in PHP	
9	4.09.2024	Find given String is Palindrome or not	
10	5.09.2024	Sort the Indexed Array elements in Ascending order and Descending order	
11	11.09.2024	Display Student's Information using Array of Objects	
12	12.09.2024	Implement Introspection	
13	18.09.2024	Demonstrate Database Connection with MySQL	
14	19.09.2024	Demonstrate MySQL queries	
15	03.10.2024	Develop simple Database Applications	

EX.NO: 1

NAME:

DATE:

REG NO:

WORK WITH LINUX SERVER

Aim:

Learn and execute basic Linux commands.

Commands:

1. **ip** – a collection of utilities for controlling TCP/IP networking and traffic control in Linux.
2. **ls** – list directory contents.
3. **df** – display disk space usage.
4. **du** – estimate file space usage.
5. **free** – display memory usage.
6. **find** – locates files based on some user-specified criteria.
7. **last** – show a listing of last logged-in users.
8. **w** – show a list of currently logged-in user sessions.
9. **grep** – Search a file for a pattern of characters, then display all matching lines.
10. **netstat** – for network statistics.
11. **ss** – utility to investigate sockets.
12. **ssh** – secure command-line access to remote Linux systems.
13. **sudo** – execute commands with administrative privilege.
14. **cd** – directory navigation.
15. **pwd** – shows your current directory location.
16. **cp** – copying files and folders.
17. **mv** – moving files and folders.
18. **rm** – removing files and folders.
19. **mkdir** – create or make new directories.
20. **touch** – used to update the access date and modification date of a computer file or directory.
21. **man** – for reading system reference manuals.
22. **apropos** – Search man page names and descriptions.
23. **rsync** – remote file transfers and syncing.
24. **tar** – an archiving utility.
25. **gzip** – file compression and decompression.
26. **b2zip** – similar to gzip. It uses a different compression algorithm.
27. **zip** – for packaging and compressing (to archive) files.
28. **locate** – search files in Linux.
29. **ps** – information about the currently running processes.
30. **dd** – convert and copy files.

31. `fdisk` – manipulate the disk partition table.
32. `blkid` – command-line utility to locate/print block device attributes.
33. `mkfs` – build a Linux file system.
34. `fsck` – tool for checking the consistency of a file system.
35. `whois` – client for the whois directory service.
36. `nc` – command-line networking utility.
37. `umask` – set file mode creation mask.
38. `chmod` – change the access permissions of file system objects.
39. `useradd` – create a new user or update default new user information.
40. `vi` – text editor.
41. `cat` – display file contents.
42. `more` – display file contents one screen/page at a time.
43. `less` – similar to the more command with additional features.
44. `tail` – used to display the tail end of a text file or piped data.
45. `killall` – Sends a kill signal to all instances of a process by name.
46. `wait` – Suspend script execution until all jobs running in the background have been terminated.
47. `passwd` – change a user's password.
48. `chpassword` –
49. `mount` / `umount` – provides access to an entire filesystem in one directory.
50. `clear` – clears the screen of the terminal.

Output:

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

labex:~/ $ cd/var/log [0:47:06]
zsh: no such file or directory: cd/var/log
labex:~/ $ cd /var/log [0:49:28]
labex:log/ $ pwd [0:49:57]
/var/log
labex:log/ $ cd.. [0:50:25]
zsh: command not found: cd..
labex:log/ $ cd [0:50:45]
labex:~/ $ pwd [0:51:12]
/home/labex
labex:~/ $ ls [0:51:26]
Code Desktop
labex:~/ $ cd Destop [0:52:00]
cd: no such file or directory: Destop
labex:~/ $ cd Destop/ [0:52:35]
cd: no such file or directory: Destop/
labex:~/ $ cd DesKtop/ [0:52:52]
cd: no such file or directory: DesKtop/
labex:~/ $ cd Desktop/ [0:53:12]
```

```
labex:~/ $ cd Desktop/ [0:53:12]
labex:Desktop/ $ [0:53:22]
```



```

Code Desktop
labex:~/ $ cd Destop [0:52:00]
cd: no such file or directory: Destop
labex:~/ $ cd Destop/ [0:52:35]
cd: no such file or directory: Destop/
labex:~/ $ cd DeskTop/ [0:52:47]
cd: no such file or directory: DeskTop/
labex:~/ $ cd Desktop/ [0:53:12]
labex:Desktop/ $ ls -lh [0:53:22]
total 24K
-rwxr-xr-x 1 labex labex 751 Jun 9 2018 gedit.desktop
-rwxr-xr-x 1 labex labex 3.6K Jun 9 2018 gvim.desktop
-rwxr-xr-x 1 labex labex 2.2K Jun 9 2018 netsurf-gtk.desktop
-rwxr-xr-x 1 labex labex 532 Jun 9 2018 sublime_text.desktop
-rwxr-xr-x 1 labex labex 5.3K Jul 8 00:46 xfce4-terminal.desktop
labex:Desktop/ $ ls -la [0:57:18]
total 32
drwxr-xr-x 2 labex labex 4096 Jun 9 2018 .
drwxr-xr-x 15 labex labex 4096 Jul 8 00:58 ..
-rwxr-xr-x 1 labex labex 751 Jun 9 2018 gedit.desktop
-rwxr-xr-x 1 labex labex 3601 Jun 9 2018 gvim.desktop
-rwxr-xr-x 1 labex labex 2225 Jun 9 2018 netsurf-gtk.desktop
-rwxr-xr-x 1 labex labex 532 Jun 9 2018 sublime_text.desktop
-rwxr-xr-x 1 labex labex 5337 Jul 8 00:46 xfce4-terminal.desktop
labex:Desktop/ $ touch file1 [0:58:36]
labex:Desktop/ $ touch file1.txt [0:59:35]

```

```

labex:Desktop/ $ touch file1 [0:58:36]
labex:Desktop/ $ touch file1.txt [0:59:35]
labex:Desktop/ $ ls [0:59:49]
file1 gedit.desktop netsurf-gtk.desktop xfce4-terminal.desktop
file1.txt gvim.desktop sublime_text.desktop
labex:Desktop/ $ mv file1.txt file2.txt [0:59:56]
labex:Desktop/ $ ls [1:04:20]
file1 gedit.desktop netsurf-gtk.desktop xfce4-terminal.desktop
file2.txt gvim.desktop sublime_text.desktop
labex:Desktop/ $ cat file1.txt [1:04:25]
cat: file1.txt: No such file or directory
labex:Desktop/ $ cat file2.txt [1:06:03]
labex:Desktop/ $ hello welcome to lab session [1:06:15]
zsh: command not found: hello
labex:Desktop/ $ cat file2.txt [1:06:44]
labex:Desktop/ $ cat > file3 [1:07:13]
hello welcome to the lab session of Linux.
labex:Desktop/ $ cat file3 [1:08:52]
hello welcome to the lab session of Linux.
labex:Desktop/ $ cat > file4 [1:09:03]
^C
labex:Desktop/ $ cat > file4 [1:10:23]
welcome to the lab session2 of linux.
labex:Desktop/ $ cat file4 [1:11:00]
welcome to the lab session2 of linux.
labex:Desktop/ $ ls [1:11:11]
file1 file3 gedit.desktop netsurf-gtk.desktop xfce4-terminal.desktop
file2.txt file4 gvim.desktop sublime_text.desktop
labex:Desktop/ $

```



```

labex:~/ $ ls [1:16:37]
Code Desktop
labex:~/ $ cat > file1 [1:17:22]
welcome to web programming
This a new platform for online session.
labex:~/ $ cat > file2 [1:18:34]
welcome to web programming
This a new platform for online session.
lab session2 deals with file system.
labex:~/ $ cat file2 [1:19:25]
welcome to web programming
This a new platform for online session.
lab session2 deals with file system.
labex:~/ $ cat file1 [1:19:36]
welcome to web programming
This a new platform for online session.
labex:~/ $ cp file1 file2 [1:19:41]
labex:~/ $ ls [1:19:58]
Code Desktop file1 file2
labex:~/ $ cp file1 [1:20:01]
cp: missing destination file operand after 'file1'
Try 'cp --help' for more information.
labex:~/ $ cat file1 [1:20:14]
welcome to web programming
This a new platform for online session.
labex:~/ $ cat file2 [1:20:36]
welcome to web programming
This a new platform for online session.
labex:~/ $ cat > file3 [1:20:55]

```

```

All are file commands
labex:~/ $ cp file4 file3 [1:23:21]
labex:~/ $ cat file4 [1:23:50]
All are file commands
labex:~/ $ cat file3 [1:23:55]
All are file commands
labex:~/ $ ls [1:24:18]
Code Desktop file1 file2 file3 file4
labex:~/ $ locate file1 [1:25:45]
zsh: command not found: locate
labex:~/ $ locate file1.txt [1:26:02]
zsh: command not found: locate
labex:~/ $ pwd [1:26:08]
/home/labex
labex:~/ $ ls [1:26:43]
Code Desktop file1 file2 file3 file4
labex:~/ $ mkdir Projects [1:28:08]
labex:~/ $ ls [1:30:32]
Code Desktop Projects file1 file2 file3 file4
labex:~/ $ mkdir -P Projects/Linux/Basics [1:30:34]
mkdir: invalid option -- 'P'
Try 'mkdir --help' for more information.
labex:~/ $ mkdir -p Projects/Linux/Basics [1:31:21]
labex:~/ $ ls [1:31:34]
Code Desktop Projects file1 file2 file3 file4
labex:~/ $ mkdir -p Projects/Linux [1:31:39]
labex:~/ $ ls [1:32:57]
Code Desktop Projects file1 file2 file3 file4
labex:~/ $ ls Projects [1:33:00]
Linux
labex:~/ $ ls Projects/Linux [1:33:44]
Basics
labex:~/ $ █ [1:34:04]

```

```

labex:~/ $ ls [1:36:34]
Code Desktop Projects file1 file2 file3 file4
labex:~/ $ ls Projects/Linux [1:36:39]
Basics
labex:~/ $ rmdir Linux [1:37:14]
rmdir: failed to remove 'Linux': No such file or directory
labex:~/ $ rmdir Code [1:37:48]
labex:~/ $ ls [1:38:01]
Desktop Projects file1 file2 file3 file4
labex:~/ $ rm file1 [1:38:09]
labex:~/ $ ls [1:38:53]
Desktop Projects file2 file3 file4
labex:~/ $ █ [1:38:56]

```

Result:

Thus the above commands are executed successfully.

EX.NO: 2

NAME:

DATE:

REG NO

WORK WITH FILES AND DIRECTORY COMMANDS

Aim

To demonstrate files and directory commands to work in the Linux environment

Commands

```
$ mkdir indra
$ cd indra
$ pwd
$ ls
sample.txt
t1.txt
t3.txt
t4.txt
t5.txt
$ rmdir indra
$ rm -i foo.txt rm
remove regular file
$ ls harish.txt sam.txt sample.txt
sathu.txt
$ rm -i sam.txt rm: remove regular file
'sam.txt'? y
$ ls
harish.txt sample.txt sathu.txt
$ rm -rf sample.txt
$ ls
foo.txt sample.txt t1.txt t3.txt t4.txt t5.txt
mkdir mystudent
$ nano indra.txt
```



Press Ctrl + x → Press Y → Enter

\$ cat indra.txt

\$ touch f1.txt

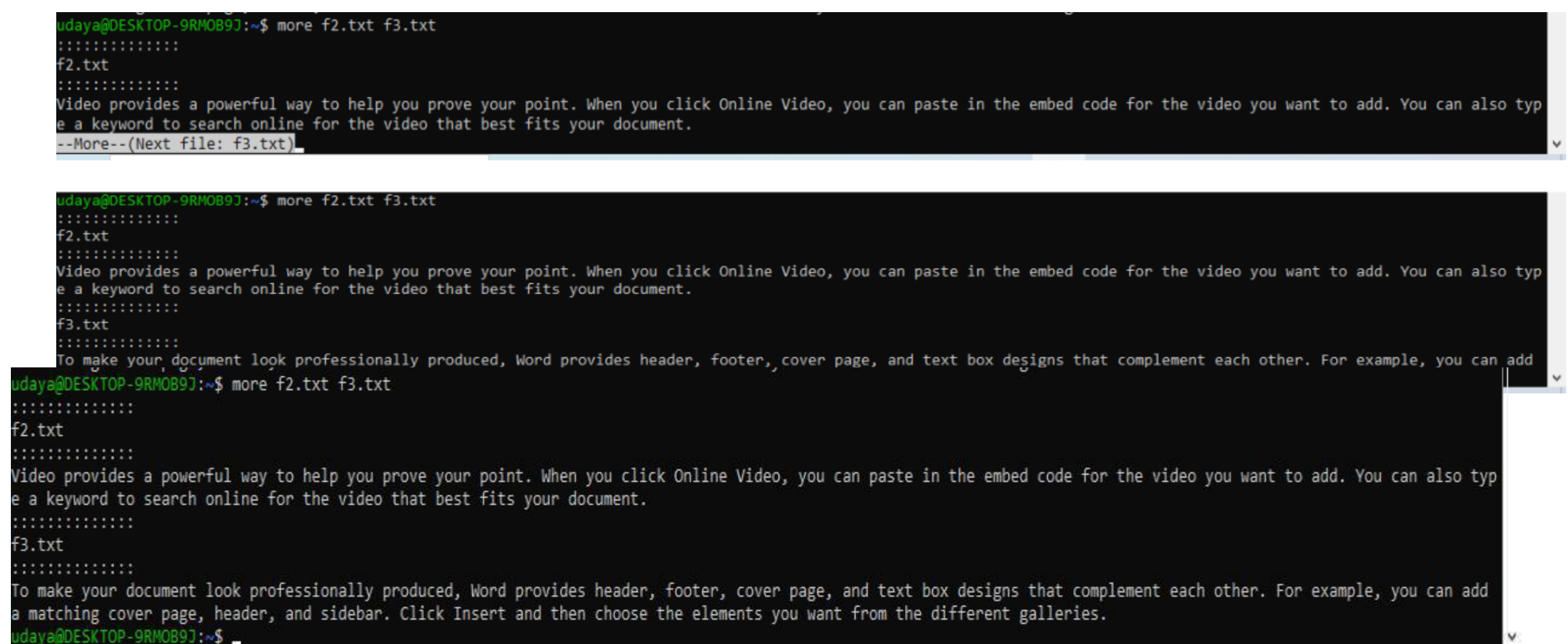
\$ cat >f1.txt

Welcome to Linux

\$ cat f1.txt

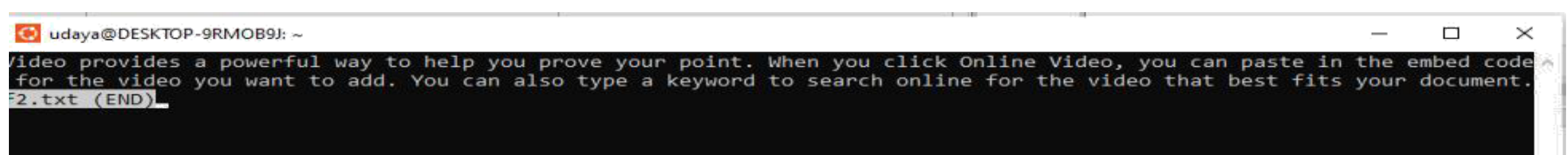
Welcome to Linux

\$ more f2.txt f3.txt



The first screenshot shows the terminal prompt 'udaya@DESKTOP-9RMOB9J:~\$' followed by the command 'more f2.txt f3.txt'. The output shows the first part of f2.txt, which contains a paragraph about online video, and then the prompt '--More--(Next file: f3.txt)'. The second screenshot shows the same terminal window with the 'more' command still running, displaying the second part of f2.txt. The third screenshot shows the terminal window with the 'more' command still running, displaying the first part of f3.txt, which contains a paragraph about Microsoft Word.

\$ less f2.txt



The screenshot shows a terminal window titled 'udaya@DESKTOP-9RMOB9J: ~'. The output of the 'less' command shows the first part of f2.txt, which contains a paragraph about online video, and then the prompt 'f2.txt (END)'. The terminal window has standard window controls (minimize, maximize, close) in the top right corner.

Press q to exit from the less console

\$ sudo apt install lpr

\$ lpr -j f2.txt


```
usage: lpr [-cdfghlmnpqrstv] [-#num] [-1234 font] [-C class] [-i
[numcols]] [-J job] [-Pprinter] [-T title] [-U user] [-wnum]
[name ...]
```

```
$ touch f5.txt
$ cp f2.txt f3.txt
$ cat f3.txt
Video provides a powerful way to help you prove your point. When you click Online Video,
you can paste in the embed code for the video you want to add. You can also type a
keyword to search online for the video that best fits your document.
$ mv f2.txt f5.txt
$ cat f5.txt
```

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.

```
$ cat f2.txt
```

```
cat: f2.txt: No such file or directory
```

```
$ ln --version
```

```
ln (GNU coreutils) 8.30
```

```
Copyright (C) 2018 Free Software Foundation, Inc.
```

```
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
```

```
This is free software: you are free to change and redistribute it.
```

```
There is NO WARRANTY, to the extent permitted by law.
```

```
Written by Mike Parker and David MacKenzie.
```

```
$ ln -v f3.txt f7.txt 'f7.txt' => 'f3.txt'
```

```
$ chmod o+w *.txt
```

```
$ ls -l
```

```
total 0
```

```
-rw-rw-rw- 1 udaya udaya 17 Jul 19 21:04 f1.txt
```

```
-rw-rw-rw- 2 udaya udaya 241 Jul 19 21:32 f3.txt
```

```
-rw-rw-rw- 1 udaya udaya 241 Jul 19 21:09 f5.txt
```

Result : The above commands executed and verified successfully

EX.NO: 3

NAME:

DATE:

REG NO:

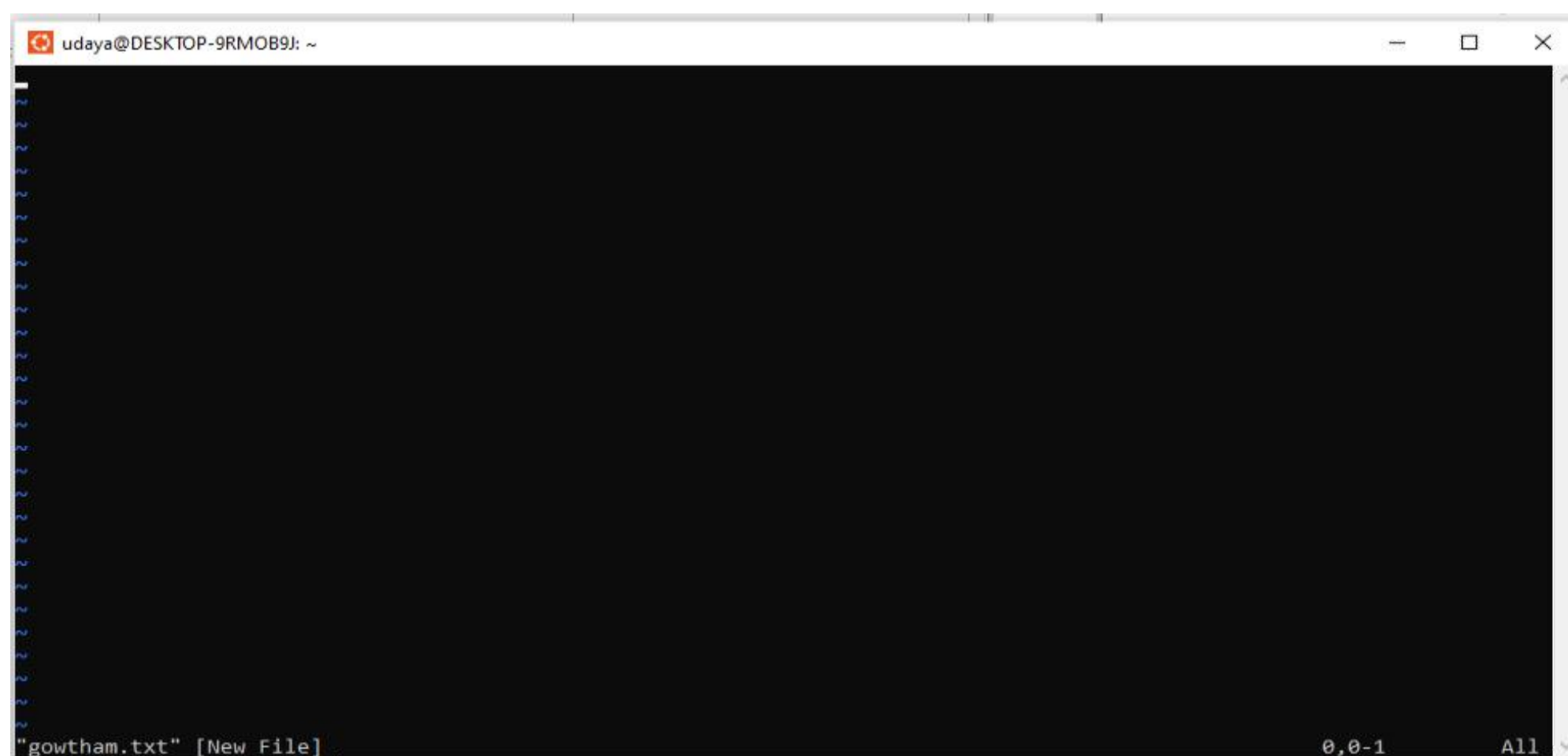
WORKING WITH FILE COMMANDS, CREATING AND MODIFYING FILES USING VI EDITOR

Aim

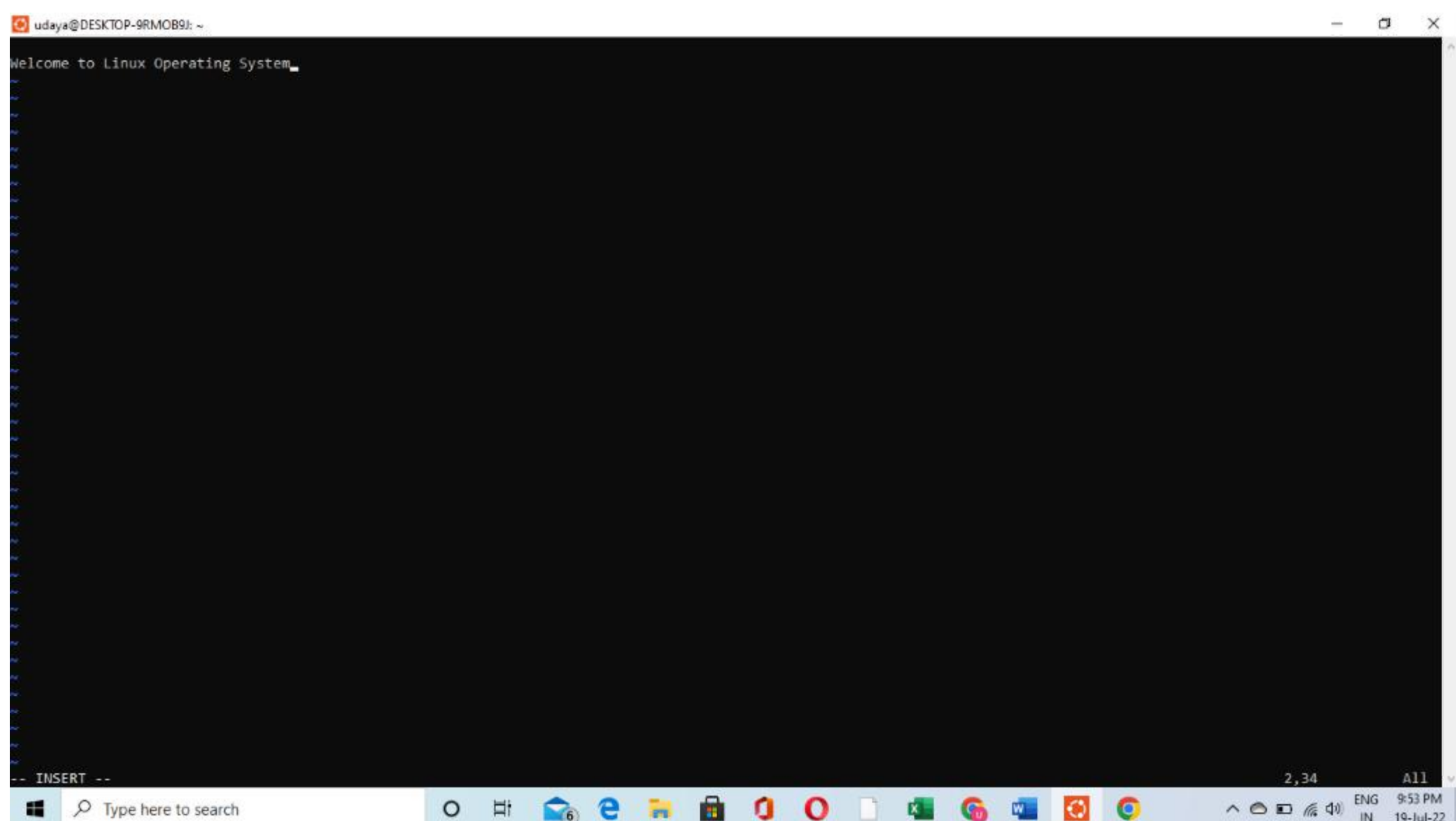
To demonstrate file commands, Creating and modifying files using Vi Editor to work in the Linux environment

Commands

\$ vi gowtham.txt



For Insert press i key in the keyboard

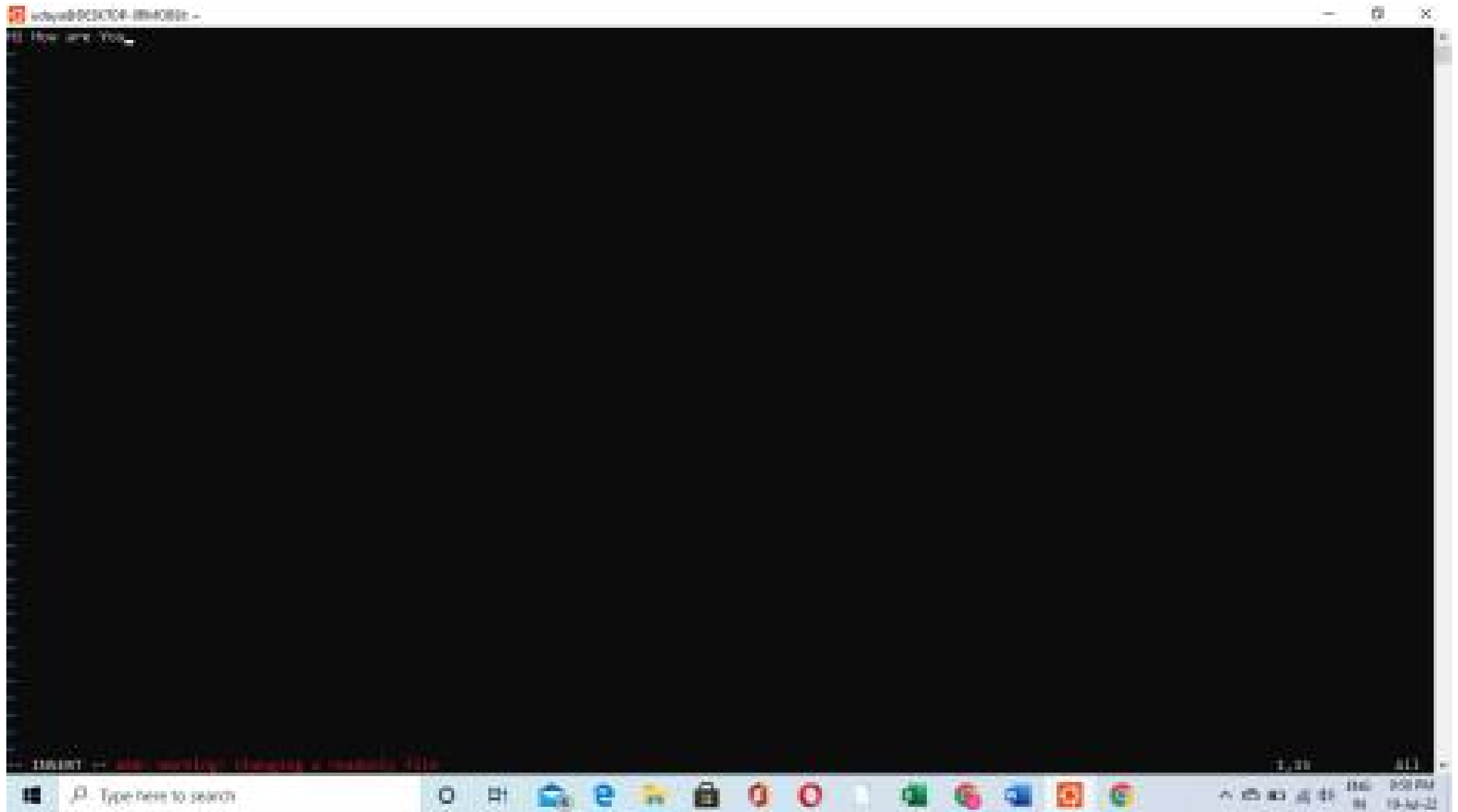


To save and Quit press esc key and press :wq and press Enter

```
$ cat gowtham.txt
```

Welcome to Linux Operating System

```
$ vi -R harish.txt
```



To save and Quit press esc key and press :wq!
and press Enter

```
$ cat harish.txt Hi How are You
```

Result : The above commands executed and verified successfully

EX.NO: 4(a)

NAME:

DATE:

REG NO:

PERFORM ARITHMETIC OPERATORS

Aim:

Write a PHP program to implement a simple calculator.

Program:

```
<?php
echo "Simple Calculator<br>";
echo "*****<br>";
$x=30;
$y=15;
echo "  <pre> x:= $x <br>  <pre> y:= $y <br><br><br>";
$z=$x+$y;
echo "<pre> Addition of x and y numbers (x+y):= $z <br><br>";
$z=$x-$y;
echo "<pre> Subtraction of x and y numbers (x-y):= $z <br><br>";
$z=$x*$y;
echo "<pre> Multiplication of x and y numbers (x*y):= $z <br><br>";
$z=$x/$y;
echo "<pre> Division of x and y numbers (x/y):= $z <br><br>";
?>
```


Output:**Simple Calculator**

x:= 30

y:= 15

Addition of x and y numbers (x+y):= 45

Subtraction of x and y numbers (x-y):= 15

Multiplication of x and y numbers (x*y):= 450

Division of x and y numbers (x/y):= 2

Result:

Thus the program has been executed successfully.

EX.NO: 4(b)

NAME:

DATE:

REG NO:

DISPLAY BIO-DATA

Aim:

Write a PHP program to prepare a Bio-Data.

Program:

```
<?php
print "Bio-Data <br> ";
print "***** <pre> ";
print "Name : Mr. A. Kenward Samuel <pre>";
print "Date of Birth: 07-04-2006 <pre>";
print "Class: III BSC A Section <pre>";
print "Father Name: Mr. Solomon <pre>";
print "Mother Name: Mrs. Malar <pre>";
print "Father Occupation: Manager <pre>";
print "Father Contact No: 9484434532 <pre>";
print "Address: 1st Street, Royapuram, Chennai";
?>
```

Output:

Bio-Data

Name : Mr. A. Kenward Samuel
Date of Birth: 07-04-2006
Class: III BSC A Section
Father Name: Mr. Solomon
Mother Name: Mrs. Malar
Father Occupation: Manager
Father Contact No: 9444434532
Address: 1st Street, Royapuram, Chennai

Result:

Thus the program has been executed successfully.

EX.NO: 4(c)

NAME:

DATE:

REG NO:

FIND AREA OF CIRCLE

Aim:

To develop a program to find area a of circle.

Program:

```
<html>
<body>
<form action = "AOC.php" method = "get">
Enter radius : <input type = "numeric" name = "radius">
<input type = "submit" value = "submit">
</form>
<?php
@$radius=$_GET["radius"];
$a = 3.14*$radius*$radius;
echo "Area of Circle".$a;
?>
</body>
</html>
```

Output:

5

Area of Circle : 78.5

Result :

Thus the program is executed successfully.

EX.NO: 4(d)

NAME:

DATE:

REG NO:

FIND AREA OF SQUARE

Aim:

Write a PHP program to find Area and Perimeter of a Square.

Program:

```
<html>
<head>
<title>PHP Program To find Area and Perimeter of a Square</title>
</head>
<body>
<form method="post">
<input type="text" name="num1" value="" placeholder="Enter length of square side"/>
<input type="submit" name="submit" value="Submit"/>
</form>
<?php
if(isset($_POST['submit']))
{
$a = $_POST['num1'];
$area = $a * $a; //Equation for Area of a square
$perimeter = 4 * $a; //Equation for Perimeter of a square
echo "Area of square $a is = $area ";
echo "<br>Perimeter of square $a is = $perimeter";
return 0;
}
?>
</body>
</html>
```

Output:

//To find Area and Perimeter of a Square

Submit

Area of square 20 is = 400

Perimeter of square 20 is = 80

Result:

Thus the program has been executed successfully.

EX.NO: 4(e)

NAME:

DATE:

REG NO:

FIND SUM OF N NUMBERS

Aim:

To find the sum of numbers

Program :

```
<?php
```

```
$s=0;
```

```
$n=10;
```

```
For ($i=1; $i<=$n; $i++)
```

```
{
```

```
$s=$s+$i;
```

```
}
```

```
Echo"$s";
```

```
?>
```

Output:

55

Result :

Thus the program is completed successfully

EX.NO: 4(f)

NAME:

DATE:

REG NO:

FIND SUM OF EVEN NUMBERS

Aim:

To find the sum of even numbers

Program :

```
<?php
$s=0;
$n=10;
For ($i=2; $i<=$n; $i+2)
{
    $s=$s+$i;
}
Echo"$s";
?>
```

Output :

30

Result :

Thus the program is completed successfully

EX.NO: 4(g)

NAME:

DATE:

REG NO:

FIND SUM OF ODD NUMBERS

Aim:

To find the sum of Odd numbers

Program :

```
<?php
$s=0;
$n=20;
For ($i=1; $i<=$n; $i+2)
{
    $s=$s+$i;
}
Echo" The sum of odd numbers is $s";
?>
```

Output :

100

Result :

Thus the program is completed successfully

EX.NO: 5(a)

NAME:

DATE:

REG NO:

FIND GIVEN NUMBER IS ARMSTRONG NUMBER OR NOT

Aim:

Write a PHP program to find whether a given number is Armstrong Number or not.

Program:

```
<html>
<title>Armstrong Number</title>
<head><br><h1>Armstrong Number</h1></br></head>
<body>
  <form method="post">
<br>
    Enter the Number:
      <input type="number" name="number">
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
<?php
if($_POST)
{
  //get the number entered
  $number = $_POST['number'];
  //store entered number in a variable
  $a = $number;
  $sum = 0;
  //run loop till the quotient is 0
  while( $a != 0 )
  {
```

```
$rem = $a % 10; //find reminder
$sum = $sum + ( $rem * $rem * $rem ); //cube the reminder and add it to the sum
variable till the loop ends
$a = $a / 10; //find quotient. if 0 then loop again
}
//if the entered number and $sum value matches then it is an armstrong number
if( $number == $sum )
{
    echo "Yes $number an Armstrong Number";
}else
{
    echo "$number is not an Armstrong Number";
}
}
?>
```

Output:

Armstrong Number

Enter the Number:

158 is not an Armstrong Number

Result:

Thus the program has been executed successfully.

EX.NO: 5(b)

NAME:

DATE:

REG NO:

FIND GIVEN NO. IS PRIME OR NOT

Aim:

Write a program to find if the Given Number is a Prime number

Program :

```
<?php
$c=0;
$n=5;
For ($i=1; $i<=$n; $i++)
{
if ($n%$i==0)
    $i=$c+$i;
}
If ($c==2)
Echo"the given number is a prime number";
else
Echo"the given number is not a prime number";
?>
```

Output :

the given number is not a prime number

Result :

Thus the program is completed successfully

EX.NO: 5(c)

NAME:

DATE:

REG NO:

MULTIPLICATION TABLE

Aim:

Write a PHP program to display Multiplication Table 1 to 10.

Program:

```
<html>
<body>
<title> Multiplication Table </title>
<head> <h1> Multiplication Table 1 to 5 </h1>
<br>
<br>
<table align="left" border="1" cellpadding="3" cellspacing="0">
<?php
for($i=1;$i<=10;$i++)
{
echo "<tr>";
for ($j=1;$j<=5;$j++)
{
echo "<td>$i * $j = ".$i*$j."</td>";
}
echo "</tr>";
}
?>
</table>
</body>
</html>
```

Output

Multiplication Table 1 to 5

1 * 1 = 1	1 * 2 = 2	1 * 3 = 3	1 * 4 = 4	1 * 5 = 5
2 * 1 = 2	2 * 2 = 4	2 * 3 = 6	2 * 4 = 8	2 * 5 = 10
3 * 1 = 3	3 * 2 = 6	3 * 3 = 9	3 * 4 = 12	3 * 5 = 15
4 * 1 = 4	4 * 2 = 8	4 * 3 = 12	4 * 4 = 16	4 * 5 = 20
5 * 1 = 5	5 * 2 = 10	5 * 3 = 15	5 * 4 = 20	5 * 5 = 25
6 * 1 = 6	6 * 2 = 12	6 * 3 = 18	6 * 4 = 24	6 * 5 = 30
7 * 1 = 7	7 * 2 = 14	7 * 3 = 21	7 * 4 = 28	7 * 5 = 35
8 * 1 = 8	8 * 2 = 16	8 * 3 = 24	8 * 4 = 32	8 * 5 = 40
9 * 1 = 9	9 * 2 = 18	9 * 3 = 27	9 * 4 = 36	9 * 5 = 45
10 * 1 = 10	10 * 2 = 20	10 * 3 = 30	10 * 4 = 40	10 * 5 = 50

Result:

Thus the program has been executed successfully.

EX.NO: 6

NAME:

DATE:

REG NO:

EMBEDDING PHP SCRIPT IN HTML

Aim:

To embed a sample student marksheet script in HTML

Program:

```
<html>
<head>
    <title>Student Marksheet Processing</title>
</head>
<form name="student" method="post" action="http://localhost/udaya/stdprocess.php">
<br><br><br>
<body bgcolor="pink">

    <h1><center>Student Marksheet
        Processing</h1></center>

<br><br><br>

<center>
<table border=0 width=400>
<tr><td>Enter Student ID <td><input type="text" name="stdid" value="">
<tr><td>Enter Student Name <td><input type="text" name="sname" value="">
<tr><td>Enter Tamil Mark <td><input type="text" name="tam" value="">
<tr><td>Enter English Mark <td><input type="text" name="eng" value="">
<tr><td>Enter Maths Mark <td><input type="text" name="mat" value="">
<tr><td>Enter Major Mark <td><input type="text" name="maj" value="">
<tr align=center><td colspan=2><input type="submit" value="submit">
</table>
</center>
</form>
</body>
</html>
```


Studentprocess.php

```
<html>

<body bgcolor="yellow">

</body>

</html>

<?php
    echo "Your Rollnumber
    is:".$_POST["stdid"]."<br/>"; echo "Your Name
    is:".$_POST["sname"]."<br/>"; echo "Your Tamil
    Mark is:".$_POST["tam"]."<br/>"; echo "Your
    English Mark is:".$_POST["eng"]."<br/>"; echo "Your
    Maths Mark is:".$_POST["mat"]."<br/>"; echo "Your
    Major Mark is:".$_POST["maj"]."<br/>";

    if($_POST['tam']>35 && $_POST['eng']>35 && $_POST['mat']>35 && $_POST['maj']>35)
    {
        $valid_res="Pass";
    }
    else
    {
        $valid_res="Fail";
    }

    $total=$_POST['tam']+$_POST['eng']+$_POST['mat']+$_POST['maj'];
    $avg=$total/4;
    if($avg>89)
    {
        $grade="O+";
    }
    else if($avg>79)
```

```
{
    $grade="A";
}
else if($avg>69)
{
    $grade="B";
}
else if($avg>49)
{
    $grade="C";
}
else
{
    $grade="D";
}
if($valid_res=="Fail")
    $grade="_____";
echo "Your Total is :".$total."<br/>";
echo "Your Grade is :".$grade."<br/>";
echo "Your Result is ".$valid_res."<br/>";
?>
```

Output

The first screenshot shows a web browser window with the title 'Student Marksheet Processing'. The address bar shows 'localhost/udayachstudent.php'. The page has a pink background and a title 'Student Marksheet Processing' in bold black text. Below the title is a form with the following fields and values:

Field	Value
Enter Student ID	101
Enter Student Name	Udayakumar U
Enter Tamil Mark	85
Enter English Mark	95
Enter Maths Mark	74
Enter Major Mark	65

Below the form is a 'submit' button.

The second screenshot shows the same web browser window after the form has been submitted. The page background is yellow. The output text is as follows:

Your Rollnumber is:101
Your Name is:Udayakumar U
Your Tamil Mark is:85
Your English Mark is:95
Your Maths Mark is:74
Your Major Mark is:65
Your Total is :319
Your Grade is :A
Your Result is :Pass

Result:

Thus the program has been executed successfully.

EX.NO: 7(a)

NAME:

DATE:

REG NO:

IMPLEMENT FUNCTIONS WITH PARAMETERS

AIM:

To create a php program using user defined functions

PROGRAM:

```
//Calling function with in a function or inner function
functionadd($a,$b){
    return $a+$b;
}
function sub($a,$b){
    return $a-$b;
}
function math($first, $second)
{
    $res = add($first, $second)/sub($first, $second);
    return (int)$res;
}
echo math(200,100);
```

OUTPUT:

Results 3

RESULT: Thus the program has been successfully completed & executed

EX.NO: 7(b)

NAME:

DATE:

REG NO:

PREPARE GROSS SALARY OF EMPLOYEE

Aim :

Write a program to Find the Gross Salary of an Employee using functions with parameters

Given.

Basic pay =10,000

HRA – 20% from basic pay

DA – 30% from basic pay

Calculations :

$HRA = \text{Basic pay} * 20/100$

$DA = \text{Basic pay} * 30/100$

$\text{Gross pay} = \text{Basic pay} + HRA + DA$

Program :

```
<?php
```

```
Function gro
```

```
sspay ($emp_id, $emp_name, $bp)
```

```
{
```

```
$HRA = $bp * 20/100;
```

```
$DA = $bp * 30/100;
```

```
$gp = $bp + HRA + DA;
```

```
echo $emp_id;
```

```
echo $emp_name;
```

```
echo $gp;
```

```
}  
$emp_id =101;  
$emp_name = Ravi;  
$bp = 10,000;  
grosspay ($emp_id, $emp_name, $bp);  
?>
```

Output:

101

Ravi

15000

Result :

Thus the program is executed successfully

EX.NO: 7(c)

NAME:

DATE:

REG NO:

GENERATE ELECTRICITY BILL USING FUNCTION

Aim :

Write a program to Prepare Electricity Bill Using Function with Parameters

Given :

Unit	Amt
0-100	Rs.0
100-150	Rs.5
151-200	RS.10

Calculations :

Unit = current unit – previous unit

Total amt = unit * amt

Program :

```
<?php
Function EBbill($service_no, $cust_name, $cu, $pu)
{
If ($unit>0)&&($unit<=100)
$tot_amt = 0;
else if ($unit>100)&&($unit<=150)
$tot_amt = unit * 5;
else ($unit>150)&&($unit<=200)
$tot_amt = unit * 10;
echo $service_no;
echo $cust_name;
```



```
echo $unit;  
echo $tot_amt;  
}  
$service_no =111;  
$cust_name = "Ram";  
$pu = 500;  
$cu = 300;  
EBbill($service_no, $cust_name, $cu, $pu);  
?>
```

Output :

111

Ram

200

2000

Result :

Thus the program is executed successfully

EX.NO: 7(d)

NAME:

DATE:

REG NO:

FIND GIVEN NUMBER IS PRIME OR NOT USING FUNCTIONS

Aim:

A program to find a given number is prime or not using functions.

Program:

```
<?php
function check_prime($num)
{
    if ($num == 1)
        return 0;
    for ($i = 2; $i <= $num/2; $i++)
    {
        if ($num % $i == 0)
            return 0;
    }
    return 1;
}
$num = 47;
$flag_val = check_prime($num);
if ($flag_val == 1)
    echo "It is a prime number";
else
    echo "It is a non-prime number"
?>
```

Output :

It is a prime number

Result:

Thus the program has been executed successfully

EX.NO: 8

NAME:

DATE:

REG NO:

PERFORM VARIOUS STRING OPERATIONS IN PHP

Aim:

To Perform Various String Operations in PHP.

Program:

```
<?php

//To convert the string to Uppercase

$str1="MY name is KANIMOZHI";

$str1=strtoupper($str1);

    echo "Upper:

    $str1". "<br>";

//To convert the string to lowercase

$str2="My name is KANI";

$str2=strtolower($str2);

    echo "Lower:

    $str2". "<br>";

//To convert the string to First char upper

$str3="MY name is PRIYA";

$str3=ucfirst($str3);

echo "Upper first character: $str3". "<br>";

//To convert the string to Title case

$str4="MY name is VEENA";

$str4=ucwords($str4);
```

```
echo "Titlecase: $str4"."<br>";
```

```
//To count no of Words in the string
```

```
$str5="We are learning PHP string function";
```

```
$str5=str_word_count($str5);
```

```
echo "Word Count: $str5"."<br>";
```

```
//To perform string reverse
```

```
$str3=strrev($str3);
```

```
echo "StringReverse: $str3"."<br>";
```

```
//Returns the position of searching string
```

```
$str6=strpos("We are learning PHP Arrays and  
functions","PHP"); echo "Index of searching string:  
$str6"."<br>";
```

```
//Replacing the string
```

```
$str7=str_replace("We","You","We are learning PHP Arrays and  
functions"); echo "String Replace: $str7"."<br>";
```

```
//To Display substring
```

```
echo substr("How are you  
doing!"."<br>",8); echo strlen("I like  
Melodies");
```

```
echo trim("<br>." How are you doing!");
```

```
?>
```


Output:

Upper: MY NAME IS KANIMOZHI

Lower: my name is kani

Upper first character: MY name is PRIYA

Titlecase: MY Name Is VEENA

Word Count: 6

StringReverse: AYIRP si eman YM

Index of searching string: 16

String Replace: You are learning PHP Arrays and functions you doing!

15

How are you doing!

Result:

Thus the program has been executed successfully

EX.NO: 9

NAME:

DATE:

REG NO:

FIND GIVEN STRING IS PALINDROME OR NOT

Aim:

Write a program to check the given String is palindrome or not.

Program:

```
<?php
function checkPalindrome($char)

{
$str = "";

$s = strtolower($char);

$len = strlen($char);
for ($i = ($len - 1); $i >= 0; $i--) {

    $str = $str . $s[$i]; // concatenating the character of each index with the value of
    variable str
}
if ($s == $str) {

    echo "$char is a palindrome string";

} else {
    echo "$char is not a palindrome string";
}
}
$char = readline("Enter the string: ");
    checkPalindrome($char);

?>
```

Output:

Madam is a palindrome string

Temple is not a palindrome string

Result:

Thus the program has been executed successfully.

EX.NO: 10

NAME:

DATE:

REG NO:

SORT THE INDEXED ARRAY ELEMENTS IN ASCENDING ORDER AND DESCENDING ORDER

Aim:

Write a PHP program to sort indexed array in ascending and descending order.

Program:

```
<?php
$scars=array("Honda
City","Audi","Toyota","Benz","Skoda"); sort($cars);
$clength=count($cars);
for($x=0;$x<$clength;$x+
+)
{
    echo
    $cars[$x];
    echo "<br>";
}
?>
```

Output:

Aud
i
Ben
z

Honda
City Skoda
Toyota

Sort Array (Ascending Order), According to Value - asort()

```
<?php
$sal=array("Peter"=>"50000","Ben"=>"45000","John"=>"85000");
asort($sal);
foreach($sal as $x=>$x_value)
```

```
{
    echo "Key=" . $x . ", Value=" .
    $x_value; echo "<br>";
}
?>
```

Output:

```
Key=Ben,
Value=45000
Key=Peter,
Value=50000
Key=John,
Value=85000
```

Sort Array (Ascending Order), According to Key - ksort()

```
<?php
$sal=array("Peter"=>"50000","Ben"=>"45000","John"=>"85000");

ksort($sal);

?>
```

Output:

```
Key=Ben, Value=45000
Key=John, Value=85000
Key=Peter, Value=50000
```

Sort Array (Descending Order), According to Value - arsort()

```
<?php
$sal=array("Peter"=>"50000","Ben"=>"45000","John"=>"85000");
arsort($sal);
foreach($sal as $x=>$x_value)
{
    echo "Key=" . $x . ", Value=" .
    $x_value; echo "<br>";
}
```

```
}  
?>
```

Output:

Key=John, Value=85000
Key=Peter, Value=50000
Key=Ben, Value=45000

Sort Array (Descending Order), According to Value - krsort()

```
<?php  
  
$sal=array("Peter"=>"50000","Ben"=>"45000","John"=>"85000");  
  
krsort($sal);  
  
?>
```

Output:

Key=Peter, Value=50000
Key=John, Value=85000
Key=Ben, Value=45000

Result:

Thus the program has been executed successfully

EX.NO: 11

NAME:

DATE:

REG NO:

Displaying Student's Information using Array of Objects

Aim:

Write a PHP to display students' information using array of objects.

Program:

```
<?php
class Student
{
public
    $name;
    public
    $stid;
    public
    $course;
    public
    $college;
}

$st1 = new Student();
$st1->name = 'Harini';
$st1->$stid = '188';
$st1->$course = 'Bsc CS';
$st1->$college = 'SRMIST';
$st2 = new Student();
$st2->name = 'Indhu';
$st2->$stid = '134';
$st2->$course = 'Bsc CS';
$st2->$college = 'SRMIST';

$stud = array($st1, $st2);
?>

<pre><?php print_r($stud);?> </pre>
```

Output:

Array (

 [0] => Student Object (

 [name] => Harini

 [stid] => 188

 [course] => Bsc CS

 [college] => SRMIST

)

 [1] => Student Object (

 [name] => Indhu

 [stid] => 134

 [course] => Bsc CS

 [college] => SRMIST

)

)

Results:

Thus the program has been executed successfully.

EX.NO: 12

NAME:

DATE:

REG NO:

Implement Intropolation

Date:

Aim:

Write a PHP program to demonstrate interpolation.

Program:

```
<?php

class Rectangle
{
    var $dim1 = 2;
    var $dim2 = 10;

    function Rectangle($dim1,$dim2)
    {
        $this->dim1 = $dim1;
        $this->dim2 = $dim2;
    }

    function area()
    {
        return $this->dim1*$this->dim2;
    }

    function display()
    {
        // any code to display info
```

```
}  
  
}  
  
$S = new Rectangle(4,2);  
  
//get the class varibale i.e properties  
  
$class_properties = get_class_vars("Rectangle");  
  
//get object properties  
  
$object_properties = get_object_vars($S);  
  
//get class methods  
  
$class_methods = get_class_methods("Rectangle");  
  
//get class corresponding to an object  
  
$object_class = get_class($S);  
  
print_r($class_properties);  
  
print_r($object_properties);  
  
print_r($class_methods);  
  
print_r($object_class);  
  
?>
```

Output:

```
Array ( [dim1] => 2 [dim2] => 10 ) Array ( [dim1] => 2 [dim2] => 10 ) Array ( [0] =>  
Rectangle [1] => area [2] => display ) Rectangle
```

Result:

Thus the program has been executed successfully.

EX.NO: 13

NAME:

DATE:

REG NO:

DEMONSTRATE DATABASE CONNECTION WITH MYSQL

AIM: To create a database connection with MySQL

PROGRAM:

```
CREATE TABLE Persons
(
  PersonID int,
  LastName varchar(255),
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255)
);
```

SQL NOT NULL Constraint

```
CREATE TABLE Persons
(
  P_Id int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255)
)
```

SQL UNIQUE Constraint on CREATE TABLE

```
CREATE TABLE Persons
(
  P_Id int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255),
  UNIQUE (P_Id)
)
```

SQL PRIMARY KEY Constraint on CREATE TABLE


```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
PRIMARY KEY (P_Id)
)
```

SQL FOREIGN KEY Constraint

```
CREATE TABLE Orders
(
O_Id int NOT NULL,
OrderNo int NOT NULL,
P_Id int,
PRIMARY KEY (O_Id),
FOREIGN KEY (P_Id) REFERENCES Persons(P_Id)
)
```

SQL CHECK Constraint on CREATE TABLE

```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CHECK (P_Id>0)
)SQL DEFAULT Constraint on CREATE TABLE
```

```
CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255) DEFAULT 'Sandnes'
)
```

OUTPUT:

Table created

RESULT: Thus the program has been successfully completed & executed.

EX.NO: 14

NAME:

DATE:

REG NO:

DEMONSTRATE MYSQL QUERIES

AIM:To insert, update and delete rows using MYSQL

A)INSERTING DATA

The INSERT statement is used to insert data into tables.

We will create a new table, where we will do our examples.

```
mysql> CREATE TABLE Books(Id INTEGER PRIMARY KEY, Title VARCHAR(100),  
-> Author VARCHAR(60));
```

We create a new table Books, with Id, Title and Author columns.

```
mysql> INSERT INTO Books(Id, Title, Author) VALUES(1, 'War and Peace',  
-> 'Leo Tolstoy');
```

This is the classic INSERT SQL statement. We have specified all column names after the table name and all values after the VALUES keyword. We add our first row into the table.

```
mysql> SELECT * FROM Books;  
+---+-----+-----+  
| Id | Title      | Author  |  
+---+-----+-----+  
| 1  | War and Peace | Leo Tolstoy |  
+---+-----+-----+
```

We have inserted our first row into the Books table.

```
mysql> INSERT INTO Books(Title, Author) VALUES ('The Brothers Karamazov',  
-> 'Fyodor Dostoyevsky');
```

We add a new title into the Books table. We have omitted the Id column. The Id column has AUTO_INCREMENT attribute. This means that MySQL will increase the Id column automatically. The value by which the AUTO_INCREMENT column is increased is controlled by auto_increment_increment system variable. By default it is 1.

```
mysql> SELECT * FROM Books;
+----+-----+-----+
| Id | Title          | Author        |
+----+-----+-----+
| 1 | War and Peace  | Leo Tolstoy   |
| 2 | The Brothers Karamazov | Fyodor Dostoyevsky |
+----+-----+-----+
```

Here is what we have in the Books table

```
mysql> INSERT INTO Books VALUES(3, 'Crime and Punishment',
-> 'Fyodor Dostoyevsky');
```

In this SQL statement, we did not specify any column names after the table name. In such a case, we have to supply all values.

```
mysql> REPLACE INTO Books VALUES(3, 'Paradise Lost', 'John Milton');
```

Query OK, 2 rows affected (0.00 sec)

The REPLACE statement is a MySQL extension to the SQL standard. It inserts a new row or replaces the old row if it collides with an existing row. In our table, there is a row with Id=3. So our previous statement replaces it with a new row. There is a message that two rows were affected. One row was deleted and one was inserted.

```
mysql> SELECT * FROM Books WHERE Id=3;
+----+-----+-----+
| Id | Title      | Author    |
+----+-----+-----+
| 3 | Paradise Lost | John Milton |
+----+-----+-----+
```

This is what we have now in the third column.

We can use the INSERT and SELECT statements together in one statement.

```
mysql> CREATE TABLE Books2(Id INTEGER PRIMARY KEY AUTO_INCREMENT,
-> Title VARCHAR(100), Author VARCHAR(60)) type=MEMORY;
```

First, we create a temporary table called Books2 in memory.

```
mysql> INSERT INTO Books2 SELECT * FROM Books;
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
```


Here we insert all data into the Books2 that we select from the Books table.

```
mysql> SELECT * FROM Books2;
+---+-----+-----+
| Id | Title          | Author        |
+---+-----+-----+
| 1 | War and Peace  | Leo Tolstoy   |
| 2 | The Brothers Karamazov | Fyodor Dostoyevsky |
| 3 | Paradise Lost  | John Milton   |
+---+-----+-----+
```

We verify it. All OK.

```
mysql> INSERT INTO Books(Title, Author) VALUES ('The Insulted and Humiliated',
'Fyodor Dostoyevsky'), ('Cousin Bette', 'Honore de Balzac');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

B)UPDATING DATA

The UPDATE statement is used to change the value of columns in selected rows of a table.

```
mysql> SELECT * FROM Books;
+---+-----+-----+
| Id | Title          | Author        |
+---+-----+-----+
| 1 | War and Peace  | Leo Tolstoy   |
| 2 | The Brothers Karamazov | Fyodor Dostoyevsky |
| 3 | Paradise Lost  | John Milton   |
| 4 | The Insulted and Humiliated | Fyodor Dostoyevsky |
| 5 | Cousin Bette   | Honore de Balzac |
+---+-----+-----+
```

We recreate the table Books. These are the rows.

Say we wanted to change 'Leo Tolstoy' to 'Lev Nikolayevich Tolstoy' table. The following statement shows, how to accomplish this.

```
mysql> UPDATE Books SET Author='Lev Nikolayevich Tolstoy'
-> WHERE Id=1;
```

The SQL statement sets the author column to 'Lev Nikolayevich Tolstoy' for the column with Id=1.

```
mysql> SELECT * FROM Books WHERE Id=1;
+---+-----+-----+
| Id | Title      | Author                |
+---+-----+-----+
| 1  | War and Peace | Lev Nikolayevich Tolstoy |
+---+-----+-----+
```

The row is correctly updated.

C)DELETING DATA

In MySQL, we can delete data using the DELETE and TRUNCATE statements. The TRUNCATE statement is a MySQL extension to the SQL specification. First, we are going to delete one row from a table. We will use the Books2 table that we have created previously.

```
mysql> DELETE FROM Books2 WHERE Id=1;
```

We delete a row with Id=1.

```
mysql> SELECT * FROM Books2;
+---+-----+-----+
| Id | Title                | Author                |
+---+-----+-----+
| 2  | The Brothers Karamazov | Fyodor Dostoyevsky |
| 3  | Paradise Lost          | John Milton          |
+---+-----+-----+
```

We verify the data.

```
mysql> DELETE FROM Books2;
mysql> TRUNCATE Books2;
```

These two SQL statements delete all data in the table.

RESULT:Thus the queries has been successfully completed & executed

EX.NO: 15

NAME:

DATE:

REG NO:

DEVELOP SIMPLE DATABASE APPLICATIONS

AIM:To achieve database connectivity in php using MYSQL

Empmain.php

```
<html>
<head>
<title>Employee Management</title>
</head>
<form name="emp" method="post" action="http://localhost/empprocess.php">
<br><br><br>
<h1><center>Employee Management System</h1><br><br><br>
<center>
<table border=1 width=400>
<tr><td>Enter Employee ID <td><input type="text" name="empid" value="">
<tr align =center><td colspan=2><input type="submit" value="submit">
</table>
<br><br>
</center>
</center>
</form>
</body>
</html>
```

Employee Management System

Enter Employee ID	
<input type="submit" value="submit"/>	

Empprocess.php

```
<?php
$con=mysqli_connect("localhost","root","","test");
if(mysqli_connect_errno())
{
    echo "Failed to connect to mysql:".mysqli_connect_errno();
}
$result=mysqli_query($con,"select*from emp where empid=".$_POST["empid"]);
echo "<center>";
echo "<font color=green><h2>Result from Server</h2></font><br><br>";
echo "<table border='1'>
<tr>
<th>Employee ID</th>
<th>Employee Name</th>
<th>Designation</th>
<th>Salary</th>
</tr>";

while($row=mysqli_fetch_array($result))
```

```
{
    echo "<tr>";
    echo "<td>".$row['empid']."</td>";
    echo "<td>".$row['empname']."</td>";
    echo "<td>".$row['desig']."</td>";
    echo "<td>".$row['salary']."</td>";
    echo "</tr>";
}
echo "</table>";
echo "</center>";
mysqli_close($con);
?>
```

Result from Server

Employee ID	Employee Name	Designation	Salary
1001	Raju	Prof	100000

RESULT: Thus the queries has been successfully completed & executed

.