KMM COLLEGE OF ARTS AND SCIENCE

THRIKKAKARA, COCHIN–21

RECORD BOOK

KMM COLLEGE OF ARTS AND SCIENCE

THRIKKAKARA, COCHIN–21

RECORD BOOK

Name : STEVE RODRIGUES

Roll No : 16 Reg. No. : 213242210432

Subject : ADVANCE OPERATING SYSTEM LAB USING LINUX

Course : MCA

Semester : 3 Year: 2021-2023

CERTIFICATE

Certified that this is a bonafide record of the practical work done by Mr. **STEVE RODRIGUES** of course **MCA** during academic year **2021-2023** for the subject **ADVANCE OPERATING SYSTEM LAB USING LINUX** in the college.

Staff in Charge	**Head of the Department**
Name:	Name: ASINI K K
Date:	Date:

Internal Examiner	External Examine

INDEX

1. COMMANDS

Date: 06-12-2022

1.1 Installation of Linux Operating system.

Minimum System Requirement for Ubuntu 18.04 LTS (Desktop)

- 2 GB RAM
- Dual Core Processor (2 GH)
- 25 GB free Hard disk space

Procedure to Install Ubuntu 18.04 LTS

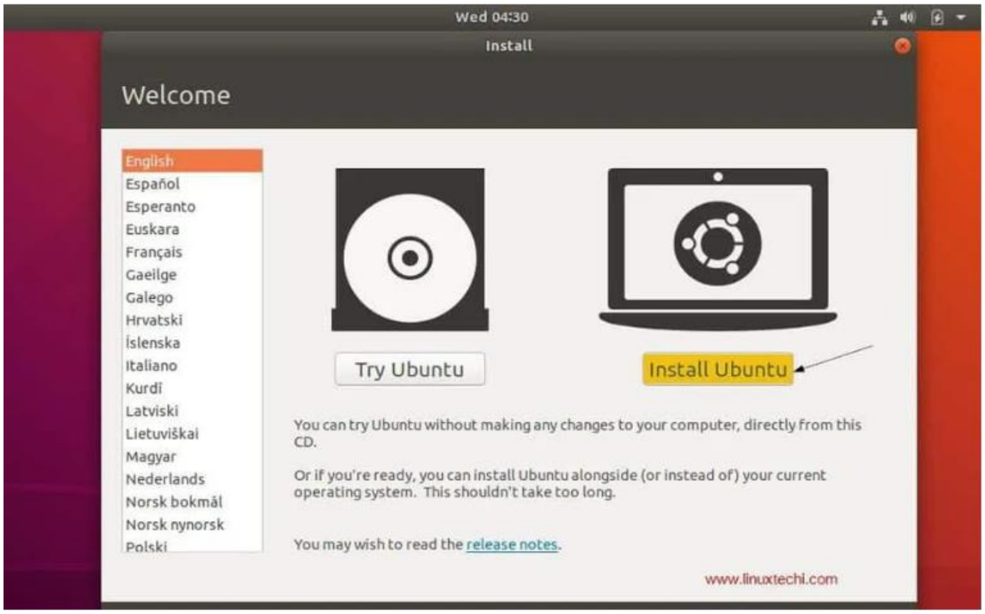
1) Download Ubuntu 18.04 LTS ISO File

Please make sure you have the latest version of Ubuntu 18.04 LTS, If not, please download the ISO file from the link here

https://www.ubuntu.com/download/desktop

2) Boot from USB/DVD or Flash Drive.

Boot your computer with Ubuntu 18.04 installation media like (Burned CD/DVD or flash drive). You can see the following screen presented before you with options including "**Try Ubuntu**" and "**Install Ubuntu**" as shown in the image below,



You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

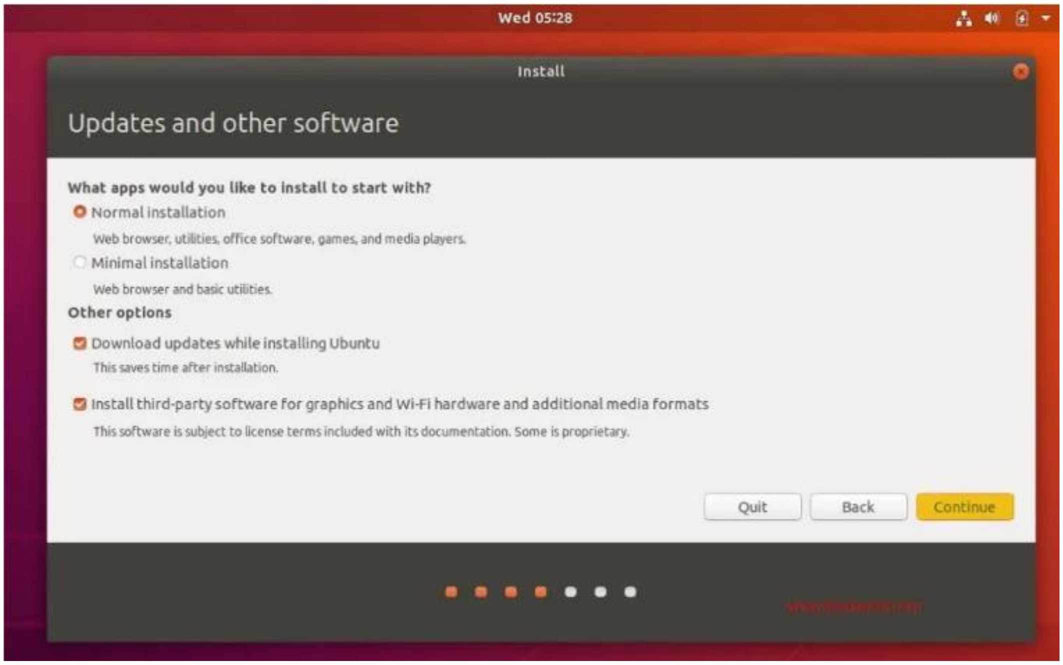
www.linuxtechi.com

3) Choose your Keyboard layout

Choose your favorite keyboard layout and click “Continue”. By default English (US) keyboard is selected and if you want to change, you can change here and click “Continue”,

4) Preparing to Install Ubuntu and other Software

In the next screen, you’ll be provided following beneath options including:



Updates and other software

What apps would you like to install to start with?

● Normal installation

Web browser, utilities, office software, games, and media players.

○ Minimal installation

Web browser and basic utilities.

Other options

☑ Download updates while installing Ubuntu

This saves time after installation.

☑ Install third-party software for graphics and Wi-Fi hardware and additional media formats

This software is subject to license terms included with its documentation. Some is proprietary.

click on “**Continue**” to proceed with installation

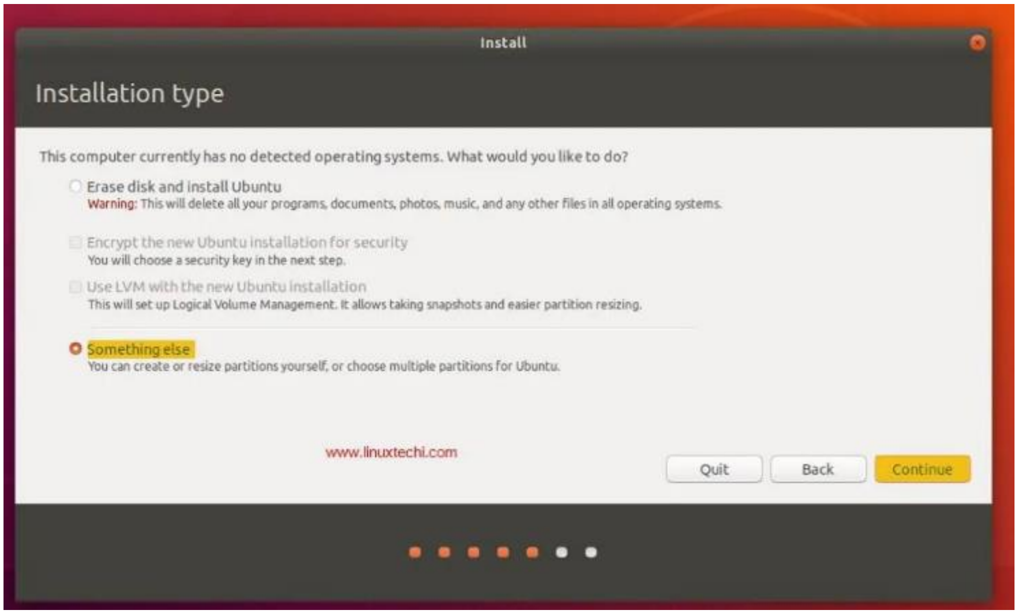
5) Select the appropriate Installation Type

Next the installer presents you with the following installation options including:

- Erase Disk and Install Ubuntu
- Encrypt the new Ubuntu installation for security
- Use LVM with the new Ubuntu installation
- Something Else

We will create our custom partitions on a hard disk of 30 GB and the following partitions are to be created:

- / 30 GB (ext4 file system)
- Swap   8 GB



Install

Installation type

This computer currently has no detected operating systems. What would you like to do?

○ Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.

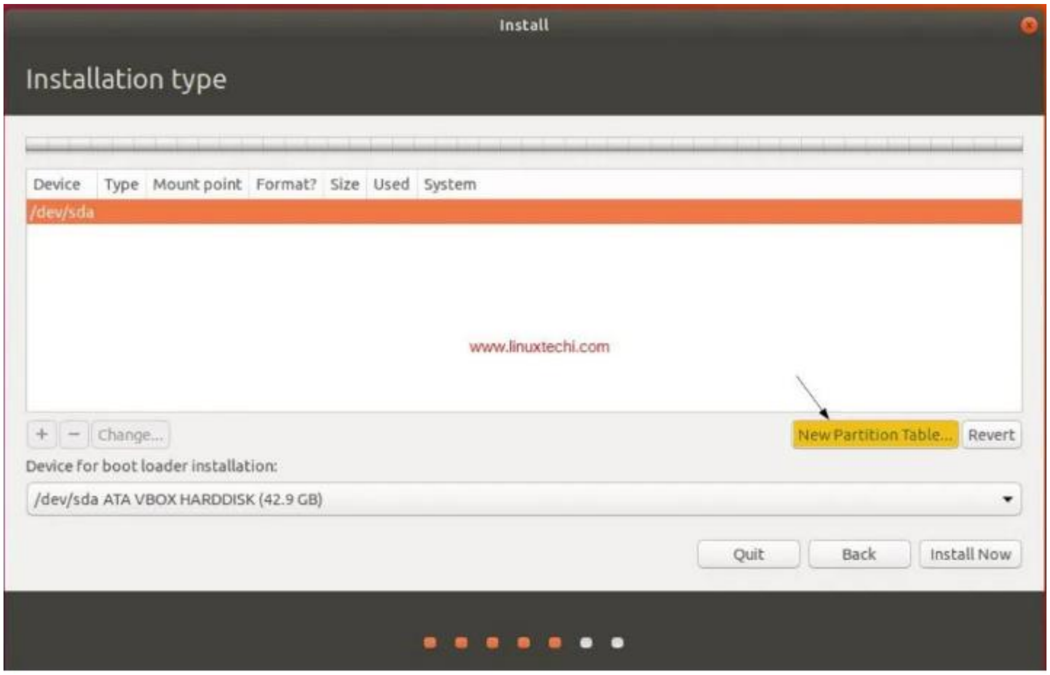
□ Encrypt the new Ubuntu installation for security
You will choose a security key in the next step.

□ Use LVM with the new Ubuntu installation
This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.

● Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Choose “**Something Else**” and Click on continue

Now in order to create your own partitions, click on “**New Partition Table**”



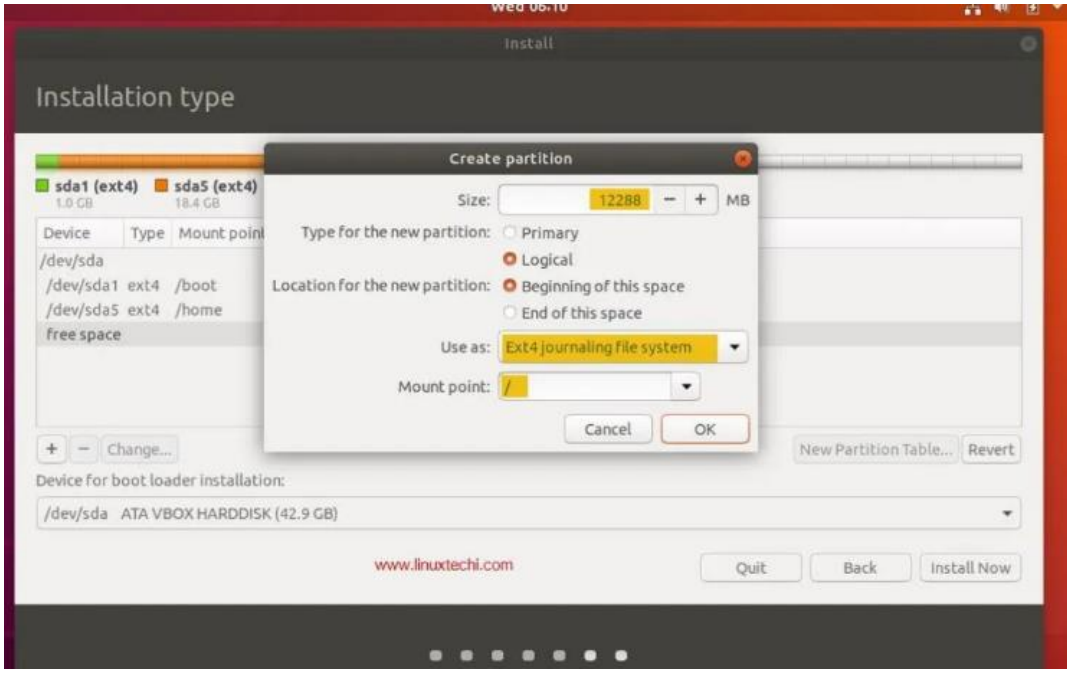
Install

Installation type

Device	Type	Mount point	Format?	Size	Used	System
/dev/sda						

Device for boot loader installation:

/dev/sda ATA VBOX HARDDISK (42.9 GB)



Installation type

🟩 sda1 (ext4) 1.0 GB 🟧 sda5 (ext4) 18.4 GB

Device	Type	Mount point
/dev/sda		
/dev/sda1	ext4	/boot
/dev/sda5	ext4	/home
free space		

Create partition

Size: 12288 MB

Type for the new partition: ○ Primary ● Logical

Location for the new partition: ● Beginning of this space ○ End of this space

Use as: Ext4 journaling file system

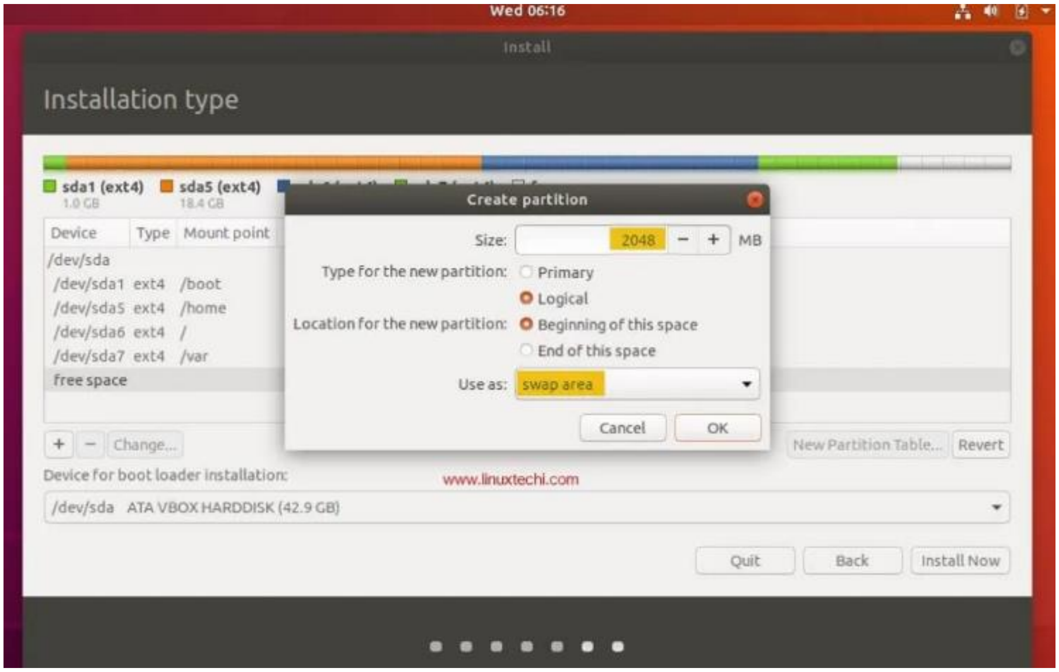
Mount point: /

Device for boot loader installation:

/dev/sda ATA VBOX HARDDISK (42.9 GB)

www.linuxtechi.com

Now create last partition as swap of size 8 GB



Installation type

🟩 sda1 (ext4) 1.0 GB 🟧 sda5 (ext4) 18.4 GB

Device	Type	Mount point
/dev/sda		
/dev/sda1	ext4	/boot
/dev/sda5	ext4	/home
/dev/sda6	ext4	/
/dev/sda7	ext4	/var
free space		

Create partition

Size: 2048 MB

Type for the new partition: ○ Primary ● Logical

Location for the new partition: ● Beginning of this space ○ End of this space

Use as: swap area

Device for boot loader installation:

www.linuxtechi.com

/dev/sda ATA VBOX HARDDISK (42.9 GB)

Click on OK

Once you are done with the partition creation task , then click on “**Install Now**” option to proceed with the installation

6) Select Your Time zone

Choose your favorite time zone and then click on “Continue”

7) Provide your User Credentials

In the next screen you will be prompted to provide your user credentials like user name, passwords.

8) Start Installing Ubuntu 18.04 LTS

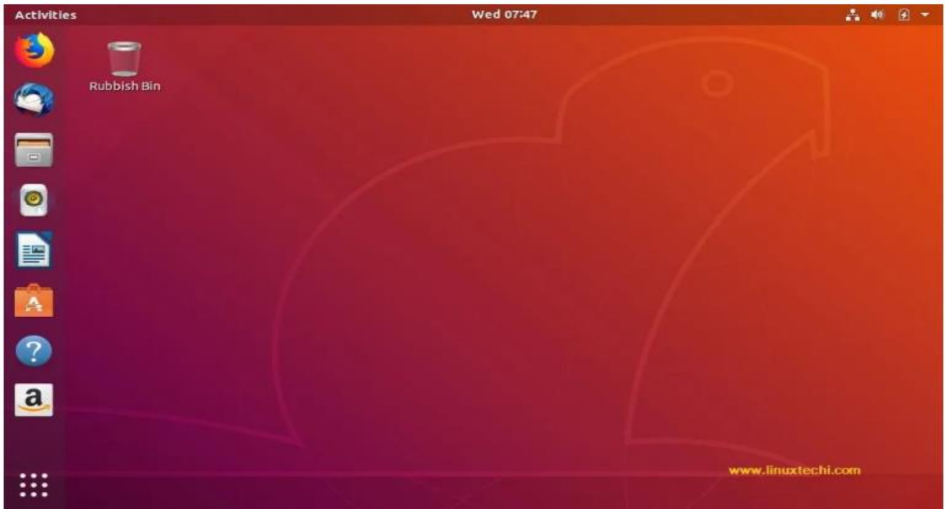
The installation of Ubuntu 18.04 LTS starts now and will take around 5-10 mins depending on the speed of your computer,

9) Restart Your System

Once the installation is completed, remove the USB/DVD from the drive and Click “Restart Now” to restart your system.

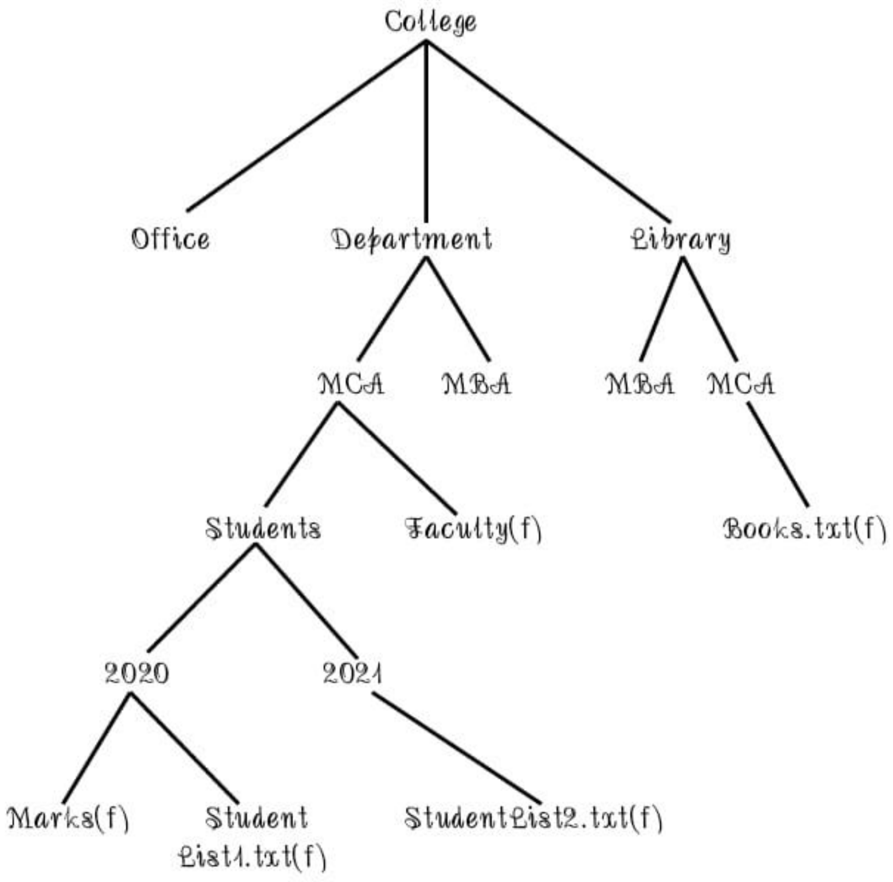
10) Login to Your Ubuntu 18.04 desktop

Once your system has been rebooted after the installation then you will get the beneath login screen, enter the User name and password that you have set during installation



Date: 13-12-2022

1.2 Demonstrate a College Management system in Linux using File and Directory commands.



```
kmm@kmm-H410M-S2-V3:~$ mkdir college

kmm@kmm-H410M-S2-V3:~$ cd College

kmm@kmm-H410M-S2-V3:~/College$ mkdir Office

kmm@kmm-H410M-S2-V3:~/College$ mkdir Department

kmm@kmm-H410M-S2-V3:~/College$ mkdir Library

kmm@kmm-H410M-S2-V3:~/College$ cd Department

kmm@kmm-H410M-S2-V3:~/College/Department$ mkdir MCA

kmm@kmm-H410M-S2-V3:~/College/Department$ mkdir MBA

kmm@kmm-H410M-S2-V3:~/College/Department$ cd MCA

kmm@kmm-H410M-S2-V3:~/College/Department/MCA$ mkdir Students
```

```
kmm@kmm-H410M-S2-V3:~/College/Department/MCA$ cat>Faculty
1. Asini
2. Simna
3. Suny
4. Ambili
5. Vineetha
kmm@kmm-H410M-S2-V3:~/ College/Department/MCA$ cd Students
kmm@kmm-H410M-S2-V3:~/ College/Department/MCA/Students$ mkdir 2020
kmm@kmm-H410M-S2-V3:~/ College/Department/MCA/Students$ mkdir 2021
kmm@kmm-H410M-S2-V3:~/College/Department/MCA/Students$ cd 2020
kmm@kmm-H410M-S2-V3:~/ College/Department/MCA/Students/2020$ cat>Marks
Name            Marks
Akthar          80
Anju            90
Amrutha         85
Ashik           70
Aldrin          75
Hera            97
Krishna         98
Naziya          98
Nishma          85
Subitha         61
Suhai           88
kmm@kmm-H410M-S2-V3:~/College/Department/MCA/Students/2020$ cat>StudentList1.txt
1. Akthar
2. Amrutha
3. Anju Mariya
```

```
4. Ashik

5. Aldrin

6. Hera

7. Krishna

8. Naziya

9. Nishma

10. Subitha

11. Suhail

kmm@kmm-H410M-S2-V3:~/College/Department/MCA/Students/2020$ cd ..

kmm@kmm-H410M-S2-V3:~/College/Department/MCA/Students$ cd 2021

kmm@kmm-H410M-S2-V3:~/College/Department/MCA/Students/2021$ cat>StudentList2.txt

1. Abiram

2. Anurag

3. Akshay

4. Gayathri

5. Greeshma

6. Fabitha

7. Faiza

8. Sanam

9. Jenisha

10. Vandana

11. Sruthilakshmi

12. Nourin

13. Rinsha

14. Steve

15. Vishnu

16. Ragi
```

```
17. Reshma
18. Nikhil
kmm@kmm-H410M-S2-V3:~/College$ cd Library
kmm@kmm-H410M-S2-V3:~/College/Library$ mkdir MBA
kmm@kmm-H410M-S2-V3:~/College/Library$ mkdir MCA
kmm@kmm-H410M-S2-V3:~/College/Library$ cd MCA
kmm@kmm-H410M-S2-V3:~/College/Library/MCA$ cat>Books.txt
Bookname                     Author
Sotware Engineering         Sommerville
Java Script                 Rohit Khurana's
Data Structure              Seymour Lipschutz
Computer Organization       Safwat Zaky
Visual Basic                Greg Perry
C++ Programming Language D.S Malik
Database System             S.K Singh
Operating System            Flynn
Web Design                  Thomas A Powell
Android Programming          Harwani
```

1. Display the file named books one page at a time.

```
kmm@kmm-H410M-S2-V3:~/College/Library/MCA$ less Books.txt
Bookname                                  Author
Software Engineering                     Sommerville
Java Script                               Rohit Khurana's
Windows and Linux Operating System        M.P Singh
Data Structure                            Seymour Lipschutz
Linux Administration                      Wale Sovinka
Linux Operating System                  NIIT
Red Hat Linux                           Bible Christopher Negus
Linux Complete Reference                Petersen
Computer Organization                   Safwat Zaky
Visual Basic                            Greg Perry
C++ Programming Language          D.S Malik
Database System                         S.K Singh
Operating System                        Flynn
Web Design                              Thomas A Powell
Android Programming                     Harwani
(END)
```

2. Display all the lines which contain the word “Linux” in the file book

```
kmm@kmm-H410M-S2-V3:~/College/Library/MCA$ grep "Linux" Books.txt
Windows and Linux Operating System      M.P Singh
Linux Administration                    Wale Sovinka
Linux Operating System                   NIIT
Red Hat Linux Bible                      Christopher Negus
Linux Complete Reference                 Petersen
```

3. Get a count of the number of such lines.

```
kmm@kmm-H410M-S2-V3:~/College/Library/MCA$ grep -c "Linux" Books.txt
5
```

4. Save all the lines having the word “Linux” in another file called “linuxbooks”.

```
mm@kmm-H410M-S2-V3:~/College/Library/MCA$ grep "Linux" books>linuxbooks
kmm@kmm-H410M-S2-V3:~/College/Library/MCA$ ls
books linuxbooks
```

5. Copy the files studentList1 in the directory “office”.

```
kmm@kmm-H410M-S2-V3:~/College/Department/MCA/Students/2020$ cp studentList1
/home/kmm/College/office

kmm@kmm-H410M-S2-V3:~/college/Office$ ls

studentList1
```

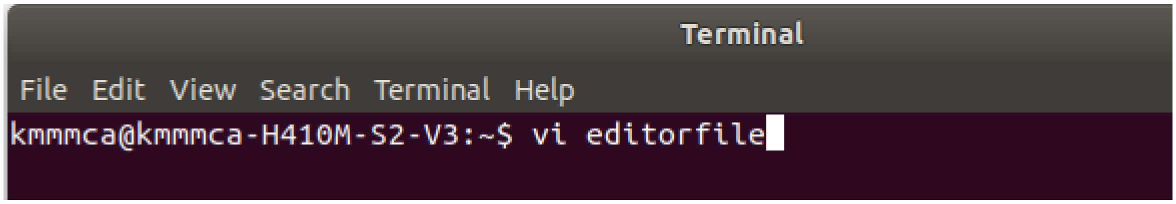
6. Rename the directory named Library as “PGLibrary”.

```
kmm@kmm-H410M-S2-V3:~/College$ mv library PGLibrary

kmm@kmm-H410M-S2-V3:~/College$ ls

Department Office PGLibrary
```

Date: 13-12-2022

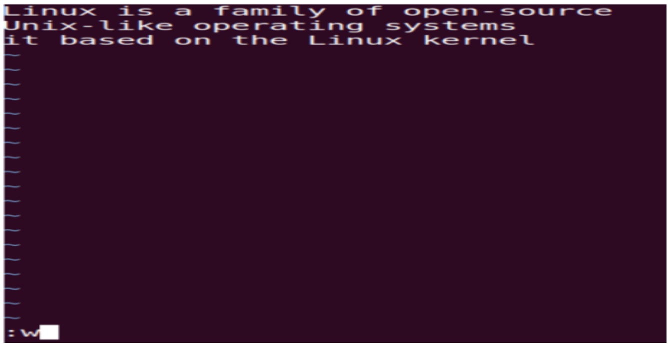
1.3 Familiarize and execute the commands of VI Editor.

Create new vi file



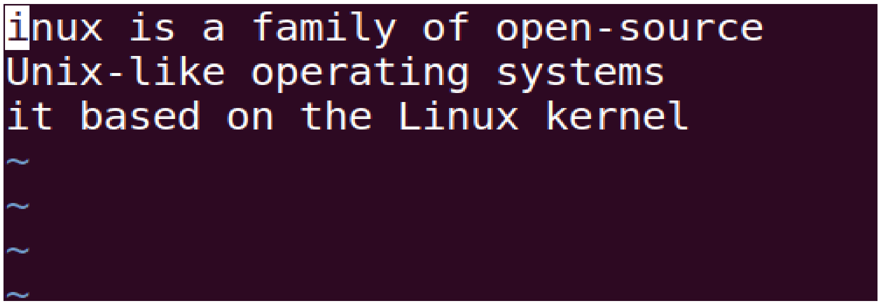
```
kmmmca@kmmmca-H410M-S2-V3:~$ vi editorfile
```

:w – save the file without quitting



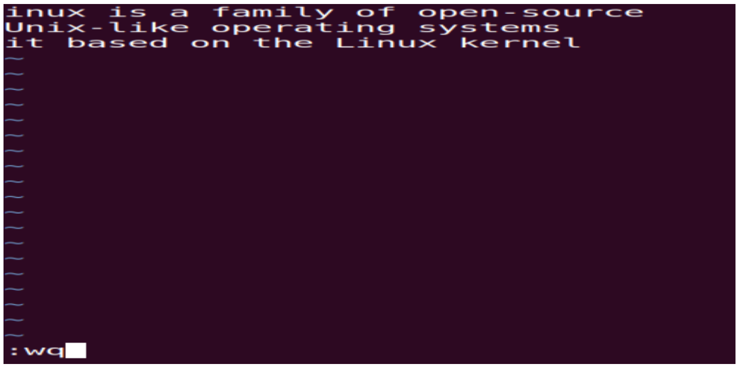
```
Linux is a family of open-source
Unix-like operating systems
it based on the Linux kernel
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:w
```

x - Delete the character at the current key



```
inux is a family of open-source
Unix-like operating systems
it based on the Linux kernel
~
~
~
~
```

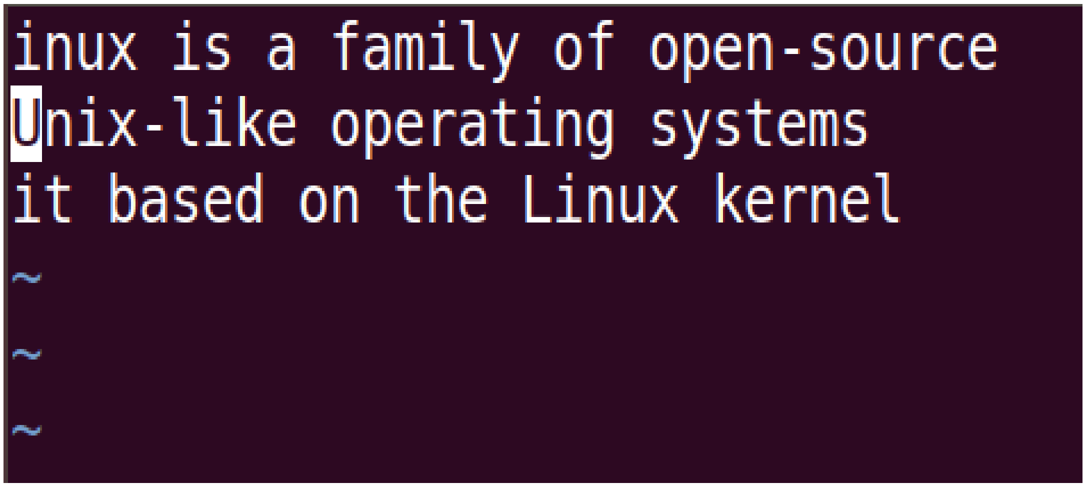
:wq - save the change and quit the window



```
inux is a family of open-source
Unix-like operating systems
it based on the Linux kernel
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

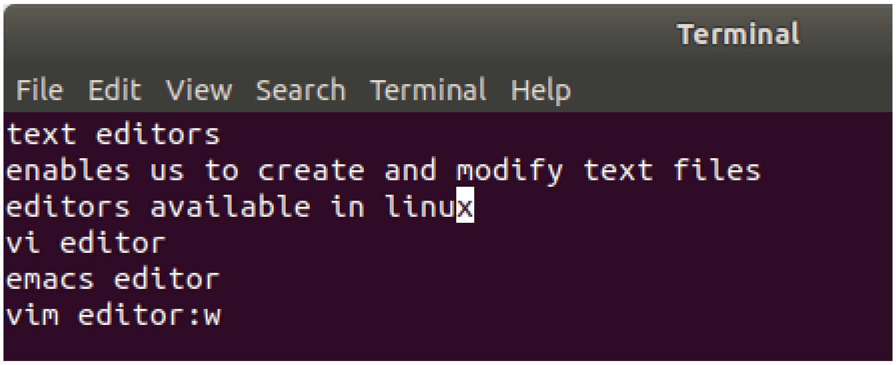
Cursor movement commands in vi

0 (Zero) - Move the cursor to the beginning of the line



```
inux is a family of open-source
Unix-like operating systems
it based on the Linux kernel
~
~
~
```

$ - Move the cursor to the end of the line



```
text editors
enables us to create and modify text files
editors available in linux
vi editor
emacs editor
vim editor:w
```

Date: 19-12-2022

1.4 Practice the Filter commands in Linux with suitable examples.

1. **Head**

   It displays specified number of lines from the beginning of a file by default ten lines.

   ```
   kmmmca@kmmmca-H410M-S2-V3:~$ head books.txt

       To Kill a Mockingbird
       1984
       Harry Potter and the Philosopher's Stone
       The Lord of the Rings
       The Great Gatsby
       Pride and Prejudice
       The Diary Of A Young Girl
       A Passage to India
       A Revenue Stamp
       Death of a City
   ```

2. **tail**

   Displays specified number of lines from the end of a file by default 10 lines

   ```
   kmmmca@kmmmca-H410M-S2-V3:~$ tail books.txt
       A Suitable Boy

       A Tale of Two Cities

       David Copperfield

       Oliver Twist

       Origin of Species

       A Week with Gandhi

       A Woman's Life

       AadheAdhure

       Adventures of Sherlock Holmes

       Adventures of Tom Sawyer
   ```

3. **more**

   Display the content of a large file one screen at a time we cannot scroll up using more command.

   ```
   kmmmca@kmmmca-H410M-S2-V3:~$ more books.txt
       To Kill a Mockingbird

       1984

       Harry Potter and the Philosopher's Stone

       The Lord of the Rings
   ```

```
    The Great Gatsby
    Pride and Prejudice
    The Diary Of A Young Girl
    A Passage to India
    A Revenue Stamp
    Death of a City
    Pinjar
    A Suitable Boy
    A Tale of Two Cities
    David Copperfield
    Oliver Twist
    Origin of Species
    A Week with Gandhi
    A Woman's Life
    AadheAdhure
    Adventures of Sherlock Holmes
    Adventures of Tom Sawyer
```

4. **less**

   It is similar to more command but we can scroll up while viewing the contents. less is faster than more command.

   ```
   kmmmca@kmmmca-H410M-S2-V3:~$ less books.txt
   ```

5. **grep**

   Searches a line for the specified pattern of characters and displays all the lines that contain the pattern

   ```
   kmmmca@kmmmca-H410M-S2-V3:~$ grep 'The' books.txt
       The Lord of the Rings
       The Great Gatsby
       The Diary Of A Young Girl
   ```

6. **sort**

   Sorts the contents of the given file.

   ```
   kmmmca@kmmmca-H410M-S2-V3:~$ sort books.txt
       1984
   ```

```
    AadheAdhure
    Adventures of Sherlock Holmes
    Adventures of Tom Sawyer
    A Passage to India
    A Revenue Stamp
    A Suitable Boy
    A Tale of Two Cities
    A Week with Gandhi
    A Woman's Life
    David Copperfield
    Death of a City
    Harry Potter and the Philosopher's Stone
    Oliver Twist
    Origin of Species
    Pinjar
    Pride and Prejudice
    The Diary Of A Young Girl
    The Great Gatsby
    The Lord of the Rings
    To Kill a Mockingbird
```

7. **wc**

wc - print newline, word, and byte counts for each file

```
kmmmca@kmmmca-H410M-S2-V3:~$ cat country.txt
    india
    korea
    Japan

kmmmca@kmmmca-H410M-S2-V3:~$ wc-l country.txt
    3country.txt
```

8. **nl**

it numbers the line.

```
kmmmca@kmmmca-H410M-S2-V3:~$ nl -s "." country.txt
```

1. india
2. korea
3. Japan

9. pg

Displays contents of text files one page at a time

10. tr

It translate or delete characters.

```
kmmmca@kmmmca-H410M-S2-V3:~$ echo “welcome” | tr e a
walcoma

kmmmca@kmmmca-H410M-S2-V3:~$ echo “Welocome” | tr [:lower:] [:upper:]
WELOCOME

kmmmca@kmmmca-H410M-S2-V3:~$ echo “WELCOME” | tr [:upper:] [:lower:]
welcome

kmmmca@kmmmca-H410M-S2-V3:~$ echo “welcome” | tr [:lower:] 1
1111111

kmmmca@kmmmca-H410M-S2-V3:~$ echo "welcome3” | tr [:digit:] s
welcomes

kmmmca@kmmmca-H410M-S2-V3:~$ tr A-Z a-z
WELCOME
welcome

kmmmca@kmmmca-H410M-S2-V3:~$ tr a-z A-Z
welcome to linux
WELCOME TO LINUX

kmmmca@kmmmca-H410M-S2-V3:~$ echo “welcome to linux” | tr [:blank:] x
welcomextoxlinux

kmmmca@kmmmca-H410M-S2-V3:~$ echo “welcomea” | tr -d ‘a’
welcome

kmmmca@kmmmca-H410M-S2-V3:~$ echo “we lcome” | tr -d [:blank:]
welcome

kmmmca@kmmmca-H410M-S2-V3:~$ echo “we lcome 1234” | tr -cd [:digit:]
1234

kmmmca@kmmmca-H410M-S2-V3:~$ cat hello.txt | tr [:lower:] [:upper:]
WELCOME
```

```
TO
LINUX
```

11. tee

It is used to redirect the standard input and write it into standard output to file.

```
kmmmca@kmmmca-H410M-S2-V3:~$ ls -l|tr -s " "|cut -d " " -f9,5|tee file5
    41 a1
    10 a2
    5 a3
    35 a5
    0 cat
    4096 Desktop
    4096 Documents
    4096 Downloads
    114 employee.dat
    69 employee.net
    8980 examples. desktop
    0 file1
    23 file2
    25 file3
    36 file4
    0 file5
    19 file7
    0 file9
    4096 mcas3
    4096 Music
    4096 Pictures
    4096 Public
    4096 s3mca
    4096 Templates
    4096 Videos

kmmmca@kmmmca-H410M-S2-V3:~$ cat file5
    41 a1
    10 a2
    5 a3
    35 a5
    0 cat
    4096 Desktop
    4096 Documents
    4096 Downloads
    114 employee.dat
    69 employee.net
    8980 examples. desktop
    0 file1
    23 file2
    25 file3
    36 file4
    0 file5
    19 file7
    0 file9
    4096 mcas3
    4096 Music
    4096 Pictures
    4096 Public
    4096 s3mca
    4096 Templates
    4096 eos
```

12. cut

It remove sections from each line of files.

```
kmmmca@kmmmca-H410M-S2-V3:~$ cat>state.txt
    Andhra Pradesh
    Arunachal Pradesh
    Assam
    Bihar

    Chhattisgarh
```

```
kmmmca@kmmmca-H410M-S2-V3:~$ cut -c 1-7 state.txt
    Andhra
    Arunach
    Assam
    Bihar
    Chhatti
```

```
kmmmca@kmmmca-H410M-S2-V3:~$ cut -d “ ”-f 1 state.txt
    Andhra
    Arunachal
    Assam
    Bihar
    Chhattisgarh
```

13. paste

it merge lines of files vertically.

```
kmmmca@kmmmca-H410M-S2-V3:~$ paste country  state

    indiaAndhra Pradesh
    koreaArunachal Pradesh
    japan    Assam
    Bihar
    Chhattisgarh
```

14. sed

stream editor for filtering and transforming text.

```
kmmmca@kmmmca-H410M-S2-V3:~$ sed 's/a/y/' state

    yndhryPrydesh
    yrunychylPrydesh
    yssym
    Bihyr
    Chhyttisgyrh

kmmmca@kmmmca-H410M-S2-V3:~$ sed 'a\==new line inserted' state

Andhra Pradesh

==new line inserted

          Arunachal Pradesh

==new line inserted

          Assam

==new line inserted

           Bihar

==new line inserted

         Chhattisgarh

==new line inserted
```

15. awk

it is a programmable filter.

```
kmmmca@kmmmca-H410M-S2-V3:~$ cat>file1

    1005 yasin computer cs

    1002 abdulla zoology zoo

    1001 ibrahim computer cs

    1004 abdulla botany bot

kmmmca@kmmmca-H410M-S2-V3:~$ awk '/cs/{print}' file1

    1005 yasin computer cs

    1001 ibrahim computer cs

kmmmca@kmmmca-H410M-S2-V3:~$ awk 'NR==2{print}' file1

    1002 abdulla zoology zoo

kmmmca@kmmmca-H410M-S2-V3:~$ awk '{print $2,$4}' file1
```

```
    yasin cs
    abdulla zoo
    ibrahim cs
    abdulla bot
kmmmca@kmmmca-H410M-S2-V3:~$ awk '/cs/{print $2,$4}' file1
    yasin cs
    ibrahim cs
kmmmca@kmmmca-H410M-S2-V3:~$ awk 'NR==2{print $2,$4}' file1
    abdulla zoo
kmmmca@kmmmca-H410M-S2-V3:~$ awk 'NR==2{print NR $2,$4}' file1
    2abdulla zoo
```

Date: 05-01-2023

1.5 How to execute Redirection and pipes in Linux.

1. **Redirection**
   Redirection is the ability of the linux operating system that allows us to change the standard input and standard output when executing a command on the terminal.

```
kmm@kmm-H410M-S2-V3:~$ cat < test1
    Linux Programming
    Python
    PHP
    Java
    C
    DBMS

kmm@kmm-H410M-S2-V3:~$ cat test1 > test2
kmm@kmm-H410M-S2-V3:~$ cat test2
    Linux Programming
    Python
    PHP
    Java
    C
    DBMS

kmm@kmm-H410M-S2-V3:~$ cat test1 >> test2
kmm@kmm-H410M-S2-V3:~$ cat test2
Linux Programming
    Python
    PHP
    Java
    C
    DBMS
    Linux Programming
    Python
    PHP
    Java
    C
    DBMS
```

2. Pipes

The pipe command lets you sends the output of one command to another.

```
kmm@kmm-H410M-S2-V3:~$ ls -l | tr -s " " | cut -d " " -f 5,9 | more
   12 a1
   9 a3
   479 arithemetic.sh
   4096 college
   4096 d1
   4096 d2
   4096 d3
   4096 Desktop
   4096 dirnew
   4096 Documents
   4096 Downloads
   40 e1
   31 e2
   449 emp.sh
   0 err
   108 f1
   52 f2
   327 factorial.sh
   19 file1
   19 file2
   10 file7
   13 file8
   16 file9
   45 file.gz
   687 fileop.sh
   --More—
```

Date: 05-01-2023

1.6 Implement File Administration Commands.

1. **Touch**

   Create empty files or change file timestamps.

```
kmm@kmm-H410M-S2-V3:~$ touch f3
kmm@kmm-H410M-S2-V3:~$ ls -l f3
    -rw-rw-r-- 1 kmmkmm 0 Feb 15 11:30 f3
```

2. **cat**

   Concatenate files and print on the standard output.

```
kmm@kmm-H410M-S2-V3:~$ cat state.txt
Andhra Pradesh
    Arunachal Pradesh
    Assam
    Bihar
    Chhattisgarh

    kmm@kmm-H410M-S2-V3:~$ cat f4 f3
    Linux Program
    Hello
    Welcome
```

3. **cp**

   Copy files and directories.

```
kmm@kmm-H410M-S2-V3:~$ cp f3 f4
kmm@kmm-H410M-S2-V3:~$ cat f4
    Hello
    Welcome

kmm@kmm-H410M-S2-V3:~$ cp -r d1 d2
kmm@kmm-H410M-S2-V3:~$ cd d2
kmm@kmm-H410M-S2-V3:~/d2$ ls
    d1
```

4. rm

remove files or directory.

```
kmm@kmm-H410M-S2-V3:~$ ls
a1 al d2 Documents f1 f3 Music Public Videos
   a3 d1 Desktop Downloads f2 f4 Pictures Templates

kmm@kmm-H410M-S2-V3:~$ rm f3
kmm@kmm-H410M-S2-V3:~$ ls
   a1 a3 al d1 d2 Desktop Documents Downloads f1 f2 f4 Music Pictures Public Templates Videos

kmm@kmm-H410M-S2-V3:~$ rm f4 a3
kmm@kmm-H410M-S2-V3:~$ ls
   a1 al d1 d2 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
```

5. mv

move (rename) files.

```
kmm@kmm-H410M-S2-V3:~$ ls
   a1 d1 d2 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
kmm@kmm-H410M-S2-V3:~$ mv a1 d1
kmm@kmm-H410M-S2-V3:~$ cd d1
kmm@kmm-H410M-S2-V3:~/d1$ ls
   a1 file1
kmm@kmm-H410M-S2-V3:~$ mv d2 d1
kmm@kmm-H410M-S2-V3:~$ cd d1
kmm@kmm-H410M-S2-V3:~/d1$ ls
   a1 d2 file1
```

6. ls

list directory contents.

```
kmm@kmm-H410M-S2-V3:~$ ls
   d1 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
```

7. mkdir

make directories.

```
kmm@kmm-H410M-S2-V3:~$ ls
   d1 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
```

```
kmm@kmm-H410M-S2-V3:~$ mkdir d2
kmm@kmm-H410M-S2-V3:~$ ls
    d1 d2 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
```

8. find

search for files in a directory hierarchy.

```
kmm@kmm-H410M-S2-V3:~$ find . -name a1
    ./d1/a1
    ./a1
kmm@kmm-H410M-S2-V3:~$ find . -type f -name a1 -exec cat {} \;
    Linux
    Program
    hi
kmm@kmm-H410M-S2-V3:~$ find . -type f -name a1 -ok cat {} \;
    < cat ... ./d1/a1 > ? y
    Linux
    Program
    < cat ... ./a1 > ? n
```

Wild card characters

1. *** wild card**

   shell interprets * wild card as a string of none,one or more characters.

   ```
   kmm@kmm-H410M-S2-V3:~$ ls
   a1 a2 d1 d2 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
   kmm@kmm-H410M-S2-V3:~$ ls a*
   a1 a2
   ```

2. **? wild card**

   ```
   kmm@kmm-H410M-S2-V3:~$ ls
       a1 a2 d1 d2 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
   kmm@kmm-H410M-S2-V3:~$ ls f?
       f1 f2
   ```

3. **[] wild card**

   matches one of the specified set of characters.

   ```
   kmm@kmm-H410M-S2-V3:~$ ls
       a1 a2 d1 d2 Desktop Documents Downloads f1 f2 Music Pictures Public Templates Videos
   kmm@kmm-H410M-S2-V3:~$ ls f[13]
       f1
   ```

Date: 10-01-2023

1.7 Execute the Disk related commands in Linux.

1. **df(disk free)**

   It is used to display information related to file systems about totalspace and available space.

```
kmmmca@kmmmca-H410M-S2-V3:~$ df
    Filesystem 1K-blocks Used Available Use% Mounted on
    udev 1902816 0 1902816 0% /dev
    tmpfs 386448 1596 384852 1% /run
    /dev/sda7 66761128 3839612 59487176 7% /
    tmpfs 1932224 0 1932224 0% /dev/shm
    tmpfs 5120 4 5116 1% /run/lock
    tmpfs 1932224 0 1932224 0% /sys/fs/cgroup
    /dev/loop2 1664 1664 0 100% /snap/gnome-calculator/154
    /dev/loop0 3456 3456 0 100% /snap/gnome-system-monitor/36
    /dev/loop5 143488 143488 0 100% /snap/gnome-3-26-1604/59
    /dev/loop4 88704 88704 0 100% /snap/core/4486
    /dev/loop1 21504 21504 0 100% /snap/gnome-logs/25
    /dev/loop3 12544 12544 0 100% /snap/gnome-characters/69
    /dev/sda2 97280 28988 68292 30% /boot/efi
    tmpfs 386444 36 386408 1% /run/user/1000
    kmmmca@kmmmca-H410M-S2-V3:~$ df -m
    Filesystem 1M-blocks Used Available Use% Mounted on
    udev 1859 0 1859 0% /dev
    tmpfs 378 2 376 1% /run
    /dev/sda7 65197 3750 58093 7% /
    tmpfs 1887 0 1887 0% /dev/shm
    tmpfs 5 1 5 1% /run/lock
    tmpfs 1887 0 1887 0% /sys/fs/cgroup
    /dev/loop2 2 2 0 100% /snap/gnome-calculator/154
    /dev/loop0 4 4 0 100% /snap/gnome-system-monitor/36
    /dev/loop5 141 141 0 100% /snap/gnome-3-26-1604/59
    /dev/loop4 87 87 0 100% /snap/core/4486
    /dev/loop1 21 21 0 100% /snap/gnome-logs/25
    /dev/loop3 13 13 0 100% /snap/gnome-characters/69
    /dev/sda2 95 29 67 30% /boot/efi
    tmpfs 378 1 378 1% /run/user/1000
```

2. **du (disk usage)**

   It is used to estimate file space usage.

```
kmmmca@kmmmca-H410M-S2-V3:~$ du
    4       ./.local/share/applications
    4       ./.local/share/nautilus/scripts
    8       ./.local/share/nautilus
    12      ./.local/share/keyrings
    80      ./.local/share/zeitgeist/fts.index
    388     ./.local/share/zeitgeist
    32      ./.local/share/gnome-software
    4       ./.local/share/sounds
    4       ./.local/share/icc
    4       ./.local/share/evolution/addressbook/trash
    4       ./.local/share/evolution/addressbook/system/photos
    92      ./.local/share/evolution/addressbook/system

kmmmca@kmmmca-H410M-S2-V3:~$ du -m
    1       ./.local/share/applications
    1       ./.local/share/nautilus/scripts
    1       ./.local/share/nautilus
    1       ./.local/share/keyrings
    1       ./.local/share/zeitgeist/fts.index
    1       ./.local/share/zeitgeist
    1       ./.local/share/gnome-software

kmmmca@kmmmca-H410M-S2-V3:~$ du -b
    4096  ./.local/share/applications
    4096  ./.local/share/nautilus/scripts
    8192  ./.local/share/nautilus
    4408  ./.local/share/keyrings
```

3. ulimit

It is used to see, set, or limit the resource usageof the current user.

```
    kmmmca@kmmmca-H410M-S2-V3:~$ ulimit
    unlimited
    mmmca@kmmmca-H410M-S2-V3:~$ ulimit -t
    unlimited
    kmmmca@kmmmca-H410M-S2-V3:~$ ulimit -u
    14593
```

Date: 10-01-2023

1.8.1 Illustrate the Communication commands in Linux.

1. **Ping**

   Ping is used to check whether a network is available and if a host is reachable.

```
kmm@kmm-H410M-S2-V3:~$ ping google.com
PING google.com (142.250.183.238) 56(84) bytes of data.
64 bytes from maa05s23-in-f14.1e100.net (142.250.183.238): icmp_seq=1 ttl=118 time=16.7 ms
64 bytes from maa05s23-in-f14.1e100.net (142.250.183.238): icmp_seq=2 ttl=118 time=16.0 ms
64 bytes from maa05s23-in-f14.1e100.net (142.250.183.238): icmp_seq=3 ttl=118 time=16.1 ms
64 bytes from maa05s23-in-f14.1e100.net (142.250.183.238): icmp_seq=4 ttl=118 time=16.2 ms
64 bytes from maa05s23-in-f14.1e100.net (142.250.183.238): icmp_seq=5 ttl=118 time=16.3 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 15.975/16.286/16.717/0.254 ms
```

2. **Traceroute**

   Trace route command is used to determine the path between two connections.

```
kmm@kmm-H410M-S2-V3:~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  130.230.88.202.asianet.co.in (202.88.230.130)  18.619 ms  18.622 ms  18.521 ms
 5  77.252.88.202.asianet.co.in (202.88.252.77)  17.616 ms  18.663 ms  18.563 ms
 6  * * *
 7  dns.google (8.8.8.8)  16.868 ms  16.850 ms  16.891 ms
```

3. **Finger**

   Finger command is a user information lookup command which gives details of all the user logged in.

```
kmm@kmm-H410M-S2-V3:~$ finger
Login     Name       Tty      Idle  Login Time   Office     Office Phone
kmm       kmm        tty2       50  Jun 30 14:25 (tty2)
kmm@kmm-H410M-S2-V3:~$ finger kmm
Login: kmm                      Name: kmm
```

```
Directory: /home/kmm                    Shell: /bin/bash
On since Thu Jun 30 14:25 (IST) on tty2 from tty2
   51 minutes 23 seconds idle
No mail.
No Plan.
```

4. **mesg**

   mesg command is used in Linux to control whether other users can send messages to us.

   ```
   kmm@kmm-H410M-S2-V3:~$ mesg
    y
   ```

5. **talk**

   We can use talk utility to talk or chat with another user on the network.

6. **write**

   To send a message on another users terminals.

7. **wall**

   To send messages to all the users connected on the Linux server.

2. SHELL SCRIPTS

Program N0:-1 **Date: 12-01-2023**

AIM

Write a script to read any 2 floating values and find the sum, difference, quotient, and remainder.

SOURCE CODE

```
#!/bin/bash
echo "MENU"
echo "1.sum"
echo "2.Difference"
echo "3.Quotient"
echo "4.Remainder"
echo "enter 1st number"
read n1
echo "enter second number"
read n2
echo "enter your choice"
read c
case $c in
1) res=`expr "scale=2;$n1+$n2"|bc`
echo "$n1 + $n2=$res"
;;
2) res=`expr "scale=2;$n1-$n2"|bc`
echo "$n1 - $n2=$res"
;;
3) res=`expr "scale=2;$n1/$n2"|bc`
echo "$n1 / $n2=$res"
;;
4) res=`expr "scale=2;$n1%$n2"|bc`
echo "$n1 % $n2=$res"
;;
esac
```

OUTPUT

```
kmmmca@kmmmca-H410M-S2-V3:~$ bash cal.sh
MENU
1.sum
2.Difference
3.Quotient
4.Remainder
enter 1st number
1
enter second number
3.5
enter your choice
1
1 + 3.5=4.5
```

```
kmmmca@kmmmca-H410M-S2-V3:~$ bash cal.sh
MENU
1.sum
2.Difference
3.Quotient
4.Remainder
enter 1st number
2
enter second number
1.5
enter your choice
2
2 – 1.5=.5
```

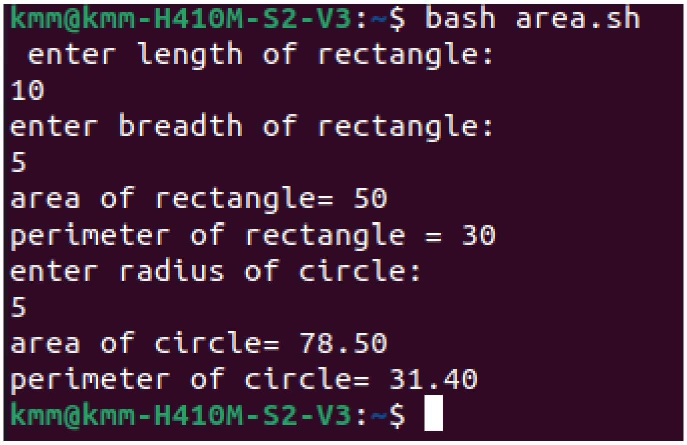
Program N0:-2 **Date: 12-01-2023**

AIM

Write a script to read the length and breadth of a rectangle and radius of a circle and calculate the area and perimeter of the rectangle and area and circumference of the circle.

```
#!/bin/bash
echo " enter length of rectangle:"
read l
echo "enter breadth of rectangle:"
read b
area=$(( $l * $b))
echo "area of rectangle= $area"
peri=$(( 2 * ($l + $b)))
echo "perimeter of rectangle = $peri"
echo "enter radius of circle:"
read r
area=$(echo " 3.14 * $r * $r" | bc)
echo "area of circle= $area"
per=$(echo  "2 * 3.14 * $r" | bc)
echo "perimeter of circle= $per"
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ bash area.sh
 enter length of rectangle:
10
enter breadth of rectangle:
5
area of rectangle= 50
perimeter of rectangle = 30
enter radius of circle:
5
area of circle= 78.50
perimeter of circle= 31.40
kmm@kmm-H410M-S2-V3:~$
```

Program N0:-3 **Date: 17-01-2023**

AIM

Write a shell program to find
a) Sum of digits of a number
b) Reverse of the number
c) Determine whether the given number is a palindrome or not.

SOURCE CODE

```
#!/bin/bash
echo "enter a number"
readnum
tm=$num
while [ $num -ne 0 ]
do
 a=$((num%10))
r=$((((r*10)+a))
s=$((s+a))
num=$((num/10))
done
echo "sum of number is $s"
echo "Reverse of number is $r"
if [ $tm -eq $r ]
then
echo "$tm is palindrome"
else
echo "$tm is not palindrome"
fi
```

OUTPUT

```
kmmmca@kmmmca-H410M-S2-V3:~  $ bash palin.sh
enter a number
134
sum of number is 8
Reverse of number is 431
134 is not palindrome
```

Program N0:-4 **Date: 17-01-2023**

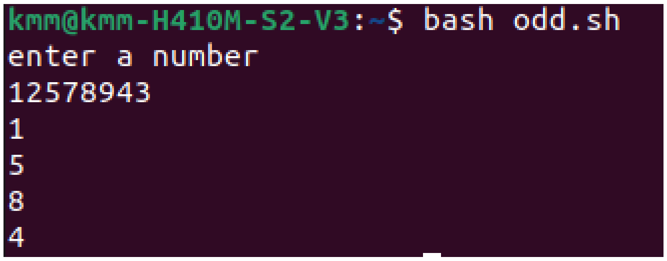
AIM

Write a shell script to display the digits which are in odd positions in a given integer.

SOURCE CODE

```
#!/bin/bash
echo "enter a number"
read n
l=${#n}
i=1
while [ $i -le $l ]
do
d=$(echo $n | cut -c $i )
echo $d
i=$(( $i + 2))
done
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ bash odd.sh
enter a number
12578943
1
5
8
4
```

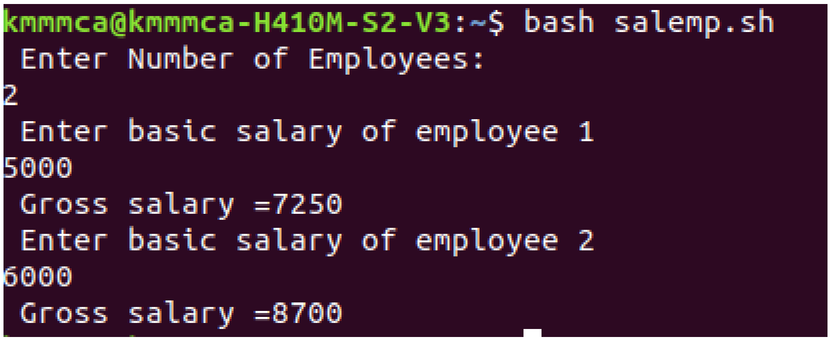
Program N0:-5 **Date: 24-01-2023**

AIM

Write a script to read the basic salary of n employees and calculate the gross salary

```
#!/bin/bash
echo " enter number of employees:"
read n

for ((i=1;i<=$n;i++))
do
echo " enter basic salary of employee $i"
read bs
da=$(((bs*30)/100))
hra=$(((bs*5)/100))
ta=$(((bs*10)/100))
gross=$(($bs + $da + $hra + $ta))
echo "gross salary = $gross"
done
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash salemp.sh
 Enter Number of Employees:
2
 Enter basic salary of employee 1
5000
 Gross salary =7250
 Enter basic salary of employee 2
6000
 Gross salary =8700
```

Program N0:-6 **Date: 02-02-2023**

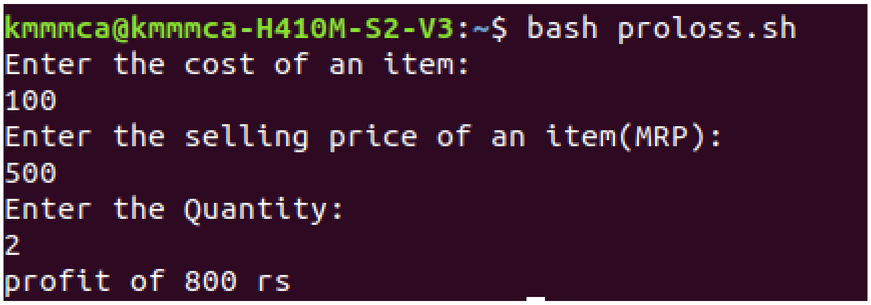
AIM

Write a script to read the cost and selling price of an item and to decide how much loss or profit has incurred by the seller.

SOURCE CODE

```
#!/bin/bash
echo "Enter the cost of an item:"
read cp
echo "Enter the selling price of an item(MRP):"
read sp
echo "Enter the Quantity:"
read q
if [ $cp -eq $sp ]
then
echo "no profit"
else
p=$(((sp-cp)*q))
echo "profit of $p rs"
fi
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash proloss.sh
Enter the cost of an item:
100
Enter the selling price of an item(MRP):
500
Enter the Quantity:
2
profit of 800 rs
```

Program N0:-7 **Date: 02-02-2023**

AIM

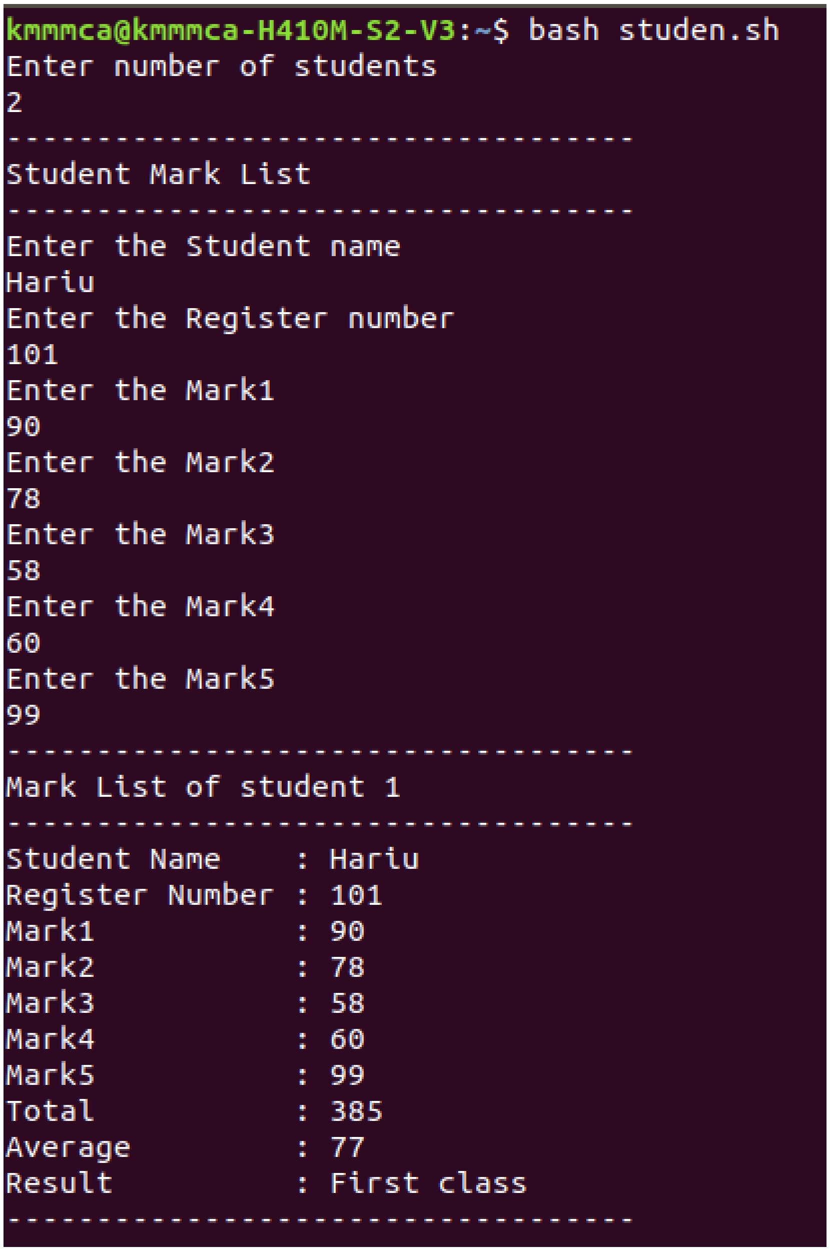
Write a script to read 5 marks of n students. Find the total and return distinction if the total percentage >= 80. [Distinction] if total % is >=60 and <80 [first class].if total % is >=50 and <60 [second class] else print failed [<50].

SOURCE CODE

```
#!/bin/bash
echo "Enter number of students"
read n
for((i=1;i<=n;i++))
do
echo ----------------------------------
echo "Student Mark List"
echo ----------------------------------
echo Enter the Student name
read name
echo Enter the Register number
read rno
echo Enter the Mark1
read m1
echo Enter the Mark2
read m2
echo Enter the Mark3
read m3
echo Enter the Mark4
read m4
echo Enter the Mark5
read m5
tot=$(expr $m1 + $m2 + $m3 + $m4 + $m5)
avg=$(expr $tot / 5)
echo ----------------------------------
echo "Mark List of student $i"
echo ----------------------------------
echo "Student Name    : $name"
echo "Register Number : $rno"
echo "Mark1           : $m1"
echo "Mark2           : $m2"
echo "Mark3           : $m3"
echo "Mark4           : $m4"
echo "Mark5           : $m5"
echo "Total           : $tot"
echo "Average         : $avg"
if [ $avg -ge 80 ]
then
        echo "Result          : Disinction"
elif [ $avg -ge 60 ]
then
        echo "Result          : First class"
```

```
elif [ $avg -ge 50 ]
then
        echo "Result          : Second class"
else
echo "Result                  : Fail"
fi
echo ----------------------------------
done
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash studen.sh
Enter number of students
2
-----------------------------------
Student Mark List
-----------------------------------
Enter the Student name
Hariu
Enter the Register number
101
Enter the Mark1
90
Enter the Mark2
78
Enter the Mark3
58
Enter the Mark4
60
Enter the Mark5
99
-----------------------------------
Mark List of student 1
-----------------------------------
Student Name    : Hariu
Register Number : 101
Mark1           : 90
Mark2           : 78
Mark3           : 58
Mark4           : 60
Mark5           : 99
Total           : 385
Average         : 77
Result          : First class
-----------------------------------
```

Program N0:-8 **Date: 07-02-2023**

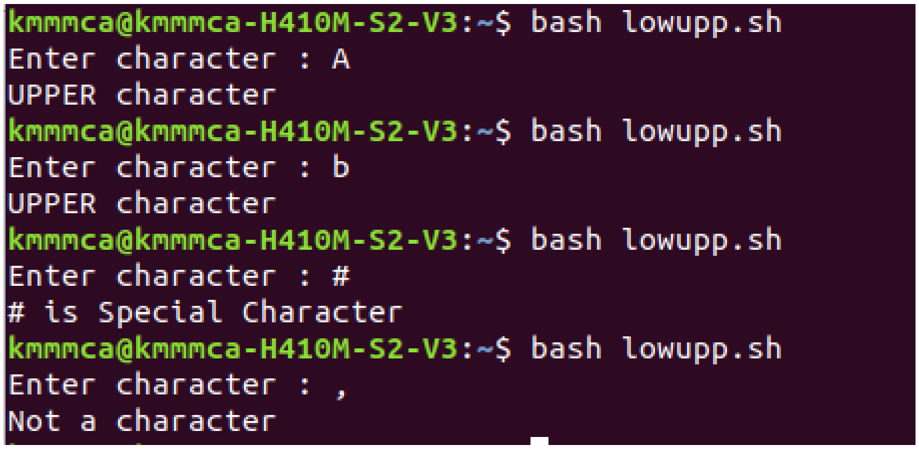
AIM

Write a script to read a character and to display if it is lowercase, uppercase, digit or special character or not a character.

SOURCE CODE

```
#!/bin/bash
c=""
echo -n "Enter character : "
read c
if [[ $c == [A-Z] ]];
then
    echo "UPPER character"
elif  [[ "$c" == [a-z] ]]
then
    echo "lower character"
elif [[ "$c" == [0-9] ]]
then
echo "$c is Digit"
elif [[ "$c" == [\!@#\$%^\&*()_+.] ]]
then
echo "$c is Special Character"
else
 echo "Not a character"
fi
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash lowupp.sh
Enter character : A
UPPER character
kmmmca@kmmmca-H410M-S2-V3:~$ bash lowupp.sh
Enter character : b
UPPER character
kmmmca@kmmmca-H410M-S2-V3:~$ bash lowupp.sh
Enter character : #
# is Special Character
kmmmca@kmmmca-H410M-S2-V3:~$ bash lowupp.sh
Enter character : ,
Not a character
```

Program N0:-9 **Date: 07-02-2023**

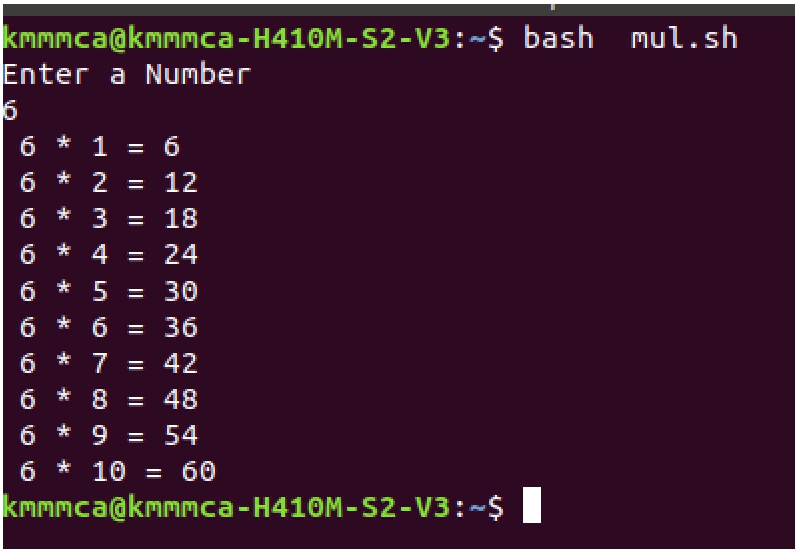
AIM

Write a script to prepare a multiplication table of a given number to any order

SOURCE CODE

```
echo "Enter a Number"
read n
i=1

while [ $i -le 10 ]
do
   echo " $n * $i = $(( n * i ))"
   i=$(( i + 1 ))
done
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash  mul.sh
Enter a Number
6
 6 * 1 = 6
 6 * 2 = 12
 6 * 3 = 18
 6 * 4 = 24
 6 * 5 = 30
 6 * 6 = 36
 6 * 7 = 42
 6 * 8 = 48
 6 * 9 = 54
 6 * 10 = 60
kmmmca@kmmmca-H410M-S2-V3:~$ 
```

Program N0:-10 **Date: 07-02-2023**

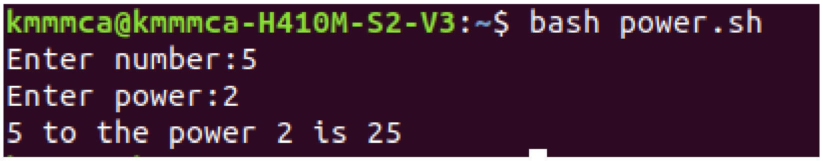
AIM

Write a script to find the value of one number raised to the power

SOURCE CODE

```
#!/bin/bash
power()
{
num=$1
pow=$2
count=1
result=1
if((pow==0))
then
result=1
fi
if ((num==0))
then
result=0
fi
if((num>=1&&pow>=1))
then
while((count<=pow))
do
result=$((result*num))
count=$((count+ 1))
done
fi
echo "$1 to the power $2 is $result"
}
read -p "Enter number:" num
read -p "Enter power:" pow
power $num $pow
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash power.sh
Enter number:5
Enter power:2
5 to the power 2 is 25
```

Program N0:-11 **Date: 13-02-2023**

AIM

Write a script to print all prime numbers from 1 to n.

SOURCE CODE

```
#!/bin/bash
echo "enter range"
read limit
echo"prime numbers upto $limit are"
i=2
while [ $i -le $limit ]
do
 f=1
j=2
while [ $j -lt $i ]
do
rem=$(($i % $j))
if [ $rem -eq 0 ]
then
f=0
break
fi
j=$(($j+1))
done
if [ $f -eq 1 ]
then
echo "$i"
fi
i=$(($i+1))
done
```

OUTPUT

```
kmmmca@kmmmca-H410M-S2-V3:~  $ bash primelimit.sh
enter limit
30
prime numbers are:
2
3
5
7
11
13
17
19
23
29
```

Program N0:-12 **Date: 13-02-2023**

AIM

Write a script to generate all combinations of a, b and c.

SOURCE CODE

```
#!bin/bash
        a=a
        b=b
        c=c
for((i=0;i<3;i++))
do
for((j=1;j<=2;j=j+1))
do
echo "$a$b$c"
        t=$b
        b=$c
        c=$t
done
        t=$a
        a=$b
        b=$c
        c=$t
done
```

OUTPUT

```
kmmmca@kmmmca-H410M-S2-V3:~$ bash combo
abc
acb
bca
bac
cab
cba
```

Program N0:-13 **Date: 13-02-2023**

AIM

Write a shell script to sum up the following series(1/1! + 2/2! + 3/3! + ….)

SOURCE CODE

```
#!bin/bash
echo "Enter the Range "
read n
fact=1
s=0
for((i=1;i<=n;i++))
do
fact=$((i*fact))
ser=`echo -e "scale=4 \n $i/$fact" |bc`
s=`echo -e " scale=4 \n $ser+$s" |bc`
done
echo "Sum is $s"
```

OUTPUT

```
kmm@kmm-H410M-S2-V3:~ $ vi factseries.sh
kmm@kmm-H410M-S2-V3:~$ bash factseries.sh
Enter the Range
15
Sum is 2.7179
```

Program N0:-14 **Date: 16-02-2023**

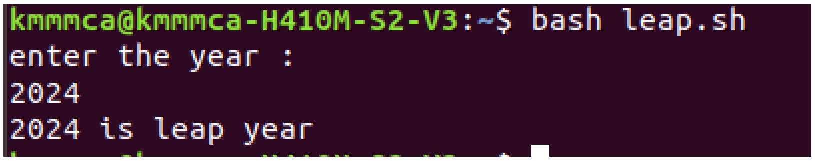
AIM

Write a script to read a year and to decide whether it is a leap year or not. If no year is supplied then the current year is assumed

SOURCE CODE

```
#!/bin/bash
echo "enter the year :"
read y
a=`expr $y % 4`
b=`expr $y % 100`
c=`expr $y % 400`
if [ $a -eq 0 -a $b -ne 0 -o $c -eq 0 ]
then
echo "$y is leap year"
else
echo "$y is not leap year"
fi
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash leap.sh
enter the year :
2024
2024 is leap year
```

Program N0:-15 **Date: 16-02-2023**

AIM

Shell script to perform operations like display, list, make directory and copy, rename, delete, edit file.

SOURCE CODE

```
#!/bin/bash
echo "MENU"
echo "1.Display"
echo "2.Make directory"
echo "3.List file"
echo "4.Copy"
echo "5.Rename"
echo "6.Delete"
echo "7.Edit"
echo "enter your choice"
read c
case $c in
1) echo "enter file name"
read file
cat $file
echo "file displayed"
;;
2) echo "enter directory name"
read nm
mkdir $nm
echo "$nm created"
;;
3)
echo "enter file name"
read file
ls -l $file
echo "file listed"
;;
4)
echo "enter file name for copy"
read c
cp $file $c
echo "file copied"
;;
5)
echo "enter the file name for rename"
read g
mv $file $g
echo "file name changed to $g"
;;
6) echo "enter file name for delete"
read file
rm $file
echo "$file deleted"
```

```
;;
7)
echo "enter file name for edit"
read f
echo "Enter text : "
read txt
echo $txt>>$f
echo "file edited"
;;
*)  echo "invalid choice"
;;
esac
```

OUTPUT

```
kmmmca@kmmmca-H410M-S2-V3:~ $ vi fileop.sh
kmmmca@kmmmca-H410M-S2-V3:~ $ bash fileop.sh
MENU
1.Display
2.Make directory
3.List file
4.Copy
5.Rename
6.Delete
7.Edit
enter your choice
1
enter file name
a2
1002    anu      comp   cs
1005    anju     bot    bt
1010    ammu     chem   ch
file displayed
kmmmca@kmmmca-H410M-S2-V3:~ $ bash fileop.sh
MENU
1.Display
2.Make directory
3.List file
4.Copy
5.Rename
6.Delete
7.Edit
enter your choice
2
enter directory name
l2
l2 created
kmmmca@kmmmca-H410M-S2-V3:~ $ bash fileop.sh
```

```
MENU
1.Display
2.Make directory
3.List file
4.Copy
5.Rename
6.Delete
7.Edit
enter your choice
3
enter file name
a2
-rw-r--r-- 1 kmmmcakmmmca 53 Feb 16 19:50 a2
file listed
```

Program N0:-16 **Date: 21-02-2023**

AIM

Write a menu driven program to display the following options.

Contents of /etc/passwd
List of output of ‘who’
Present working directory
Exit

SOURCE CODE

```
#!/bin/bash
echo&quot;MENU&quot;
echo&quot;1. Contents of /etc/passwd&quot;
echo&quot;2. List of output of who&quot;
echo&quot;3. Present working directory&quot;
echo&quot;4. Exit&quot;
echo&quot;Enter your choice&quot;
read c
case $c in
1)
cat /etc/passwd
;;
2)
who
echo&quot;Whodisplayed&quot;
;;
3)
pwd
echo&quot;Current working directory&quot;
;;
4)
exit
;;
*)
echo&quot;InavlidOption&quot;
;;
esac
```

OUTPUT

```
kmm@kmm-H410M-S2-V3:~ $ vi commands.sh
kmm@kmm-H410M-S2-V3:~ $ bash commands.sh
MENU
1. Contents of /etc/passwd
2. List of output of who
3. Present working directory
4. Exit
Enter your choice
1
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
Enter your choice
2
kmm tty2 2022-05-11 18:45 (tty2)
Who displayed
kmm@kmm-H410M-S2-V3:~  $ bash commands.sh
MENU
1. Contents of etc/passwd
2. List of output of who
3. Present working directory
4. Exit
Enter your choice
3
/home/kmm
Current working directory
```

Program N0:-17 **Date:21-02-2023**

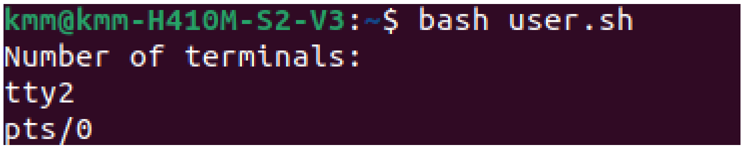
AIM

Write a shell script to find how many terminals this user logged in.

SOURCE CODE

```
echo "Number of terminals: "
ps -u|grep "^kmmmca"|tr -s ""|cut -d "" -f 7|uniq
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ bash user.sh
Number of terminals:
tty2
pts/0
```

Program N0:-18 **Date: 23-02-2023**

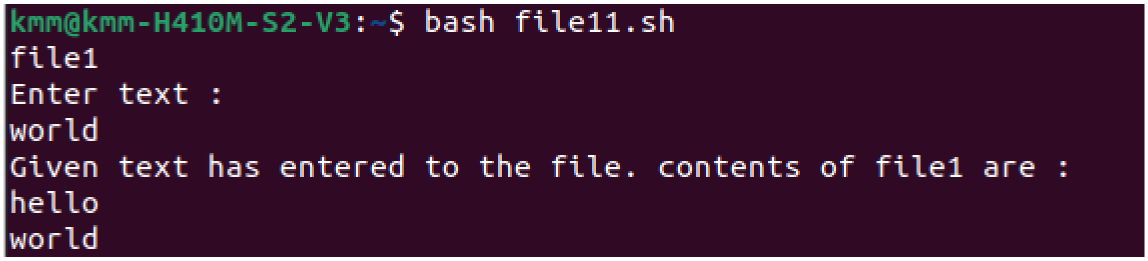
AIM

Write a script to accept a filename while running the script and check it has the write permission, ifyes prompt the user to enter a text and append the text to the given filename.

SOURCE CODE

```
read file1
if [ -w $file1 ]
then
echo "Enter text : "
read txt
echo $txt>>$file1
echo "Given text has entered to the file. contents of $file1 are : "
cat $file1
else
echo "User has no write permission"
fi
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ bash file11.sh
file1
Enter text :
world
Given text has entered to the file. contents of file1 are :
hello
world
```

Program N0:-19 **Date: 07-03-2023**

AIM

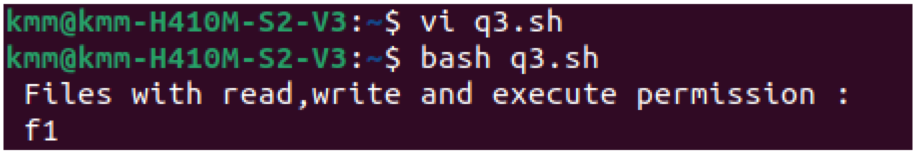
Write a shell script which displays a list of all files in the current directory to which you have read,write& execute permissions.

SOURCE CODE

```
#!bin/bash
echo " Files with read,write and execute permission : "
for f in `ls -l | grep "^-rwx" | tr -s " " | cut -d " " -f9 `
do
echo " $f"
done
```

OUTPUT

```
//file creation(read,write and execution)
kmm@kmm-H410M-S2-V3:~$ nano f1
kmm@kmm-H410M-S2-V3:~$ chmod 700 f1
```



```
kmm@kmm-H410M-S2-V3:~$ vi q3.sh
kmm@kmm-H410M-S2-V3:~$ bash q3.sh
 Files with read,write and execute permission :
 f1
```

Program N0:-20 **Date: 07-03-2023**

AIM

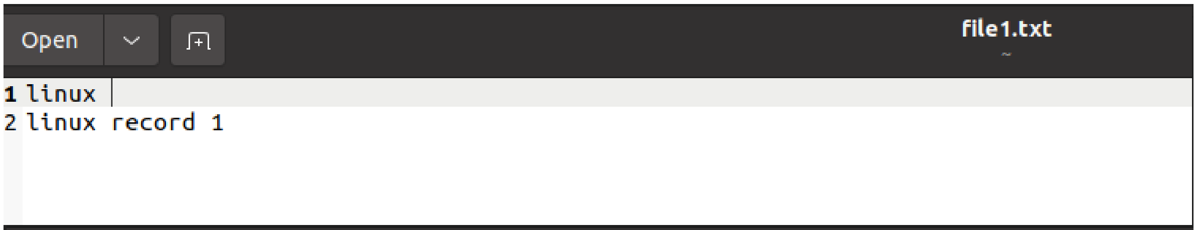
Write a shell script which receives two file names as arguments. It should check whether the two file's contents are the same or not .If they are the same, delete the second file.

SOURCE CODE

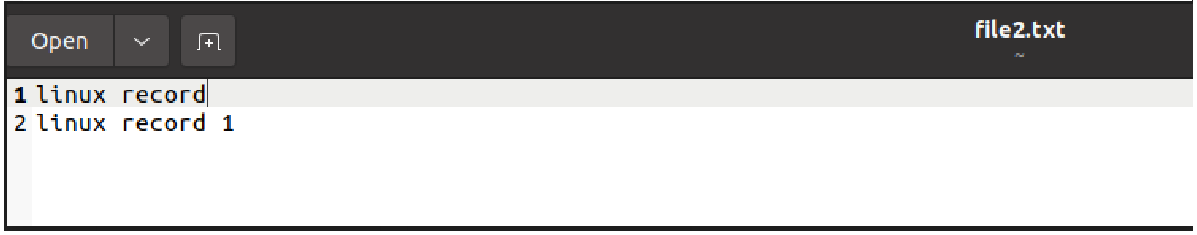
```
#!bin/bash
r=`cmp $1 $2`
if [ -z $r ]
then
echo "Two Files Contents are same"
rm $2
echo "Second file is removed"
else
echo "Two File  contents are not same.."
fi
```

OUTPUT

```
kmm@kmm:~/s4mca$ bash 13.sh myfile file1
Two Files Contents are same
Second file is removed
kmm@kmm:~/s4mca$ bash 13.sh myfileyourfile
Two File contents are not same..
```



```
linux
linux record 1
```



```
linux record
linux record 1
```

Program N0:-21 **Date: 09-03-2023**

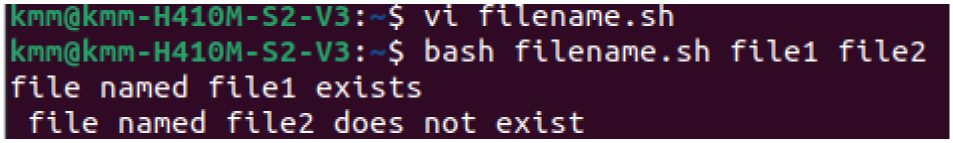
AIM

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

SOURCE CODE

```
#!bin/bash
forarg in $*
do
if [ -e $arg ]
then
echo   "file named $arg exists "
else
echo " file named $arg does not exist "
fi
done
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ vi filename.sh
kmm@kmm-H410M-S2-V3:~$ bash filename.sh file1 file2
file named file1 exists
 file named file2 does not exist
```

Program N0:-22 **Date: 09-03-2023**

AIM

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

SOURCE CODE

```
#!/bin/bash
clear
while [ true ]
do
echo "1. create database "
echo "2. view database "
echo "3. insert a record "
echo "4. delete record "
echo "5. exit"
echo "Enter your choice "
read d
case $d in
1)
touch stud
echo "Database Created"
;;
2)
echo "Show Student Database"
cat stud
;;
3)
echo "Enter rollnumber of student "
readrn
echo "Enter name of student "
read nm
echo "Enter marks Linux "
readsk
echo " enter marks Python "
readeg
echo " enter marks Cyber Informatics "
readhn
record="$rn $nm $sk $eg $hn "
echo $record>>stud
;;
4) echo " enter roll number "
readrn
w=`grep -c ^$rn stud`
if [ $w -eq 0 ]
then
echo "record for the given roll number does not exist "
else
grep -v "^$rn " stud >tmp
cptmp stud
```

```
echo "deletion completed "
fi
;;
5)
exit;;
* ) echo "enter right choice"
esac
done
```

OUTPUT

```
kmm@kmm-H410M-S2-V3:~ $ vi studentdata.sh
kmm@kmm-H410M-S2-V3:~ $ bash studentdata.sh

1. create database
2. view database
3. insert a record
4. delete record
5. exit
Enter your choice
1
Database Created
1. create database
2. view database
3. insert a record
4. delete record
5. exit
Enter your choice
3
Enter rollnumber of student
1
Enter name of student
Arya
Enter marks Linux
89
enter marks Python
78
enter marks Cyber Informatics
98
1. create database
2. view database
3. insert a record
4. delete record
5. exit
Enter your choice
3
Enter rollnumber of student
2
Enter name of student
```

```
Krishna
Enter marks Linux
88
enter marks Python
98
enter marks Cyber Informatics
99
1. create database
2. view database
3. insert a record
4. delete record
5. exit
Enter your choice
2
Show Student Database
1 Arya 89 78 98
2 Krishna 88 98 99
1. create database
2. view database
3. insert a record
4. delete record
5. exit
Enter your choice
4
enter roll number
2
deletion completed
1. create database
2. view database
3. insert a record
4. delete record
5. exit
Enter your choice
2
Show Student Database
1 Arya 89 78 98
1. create database
2. view database
3. insert a record
4. delete record
5. exit
Enter your choice
5
```

Program N0:-23 **Date: 09-03-2023**

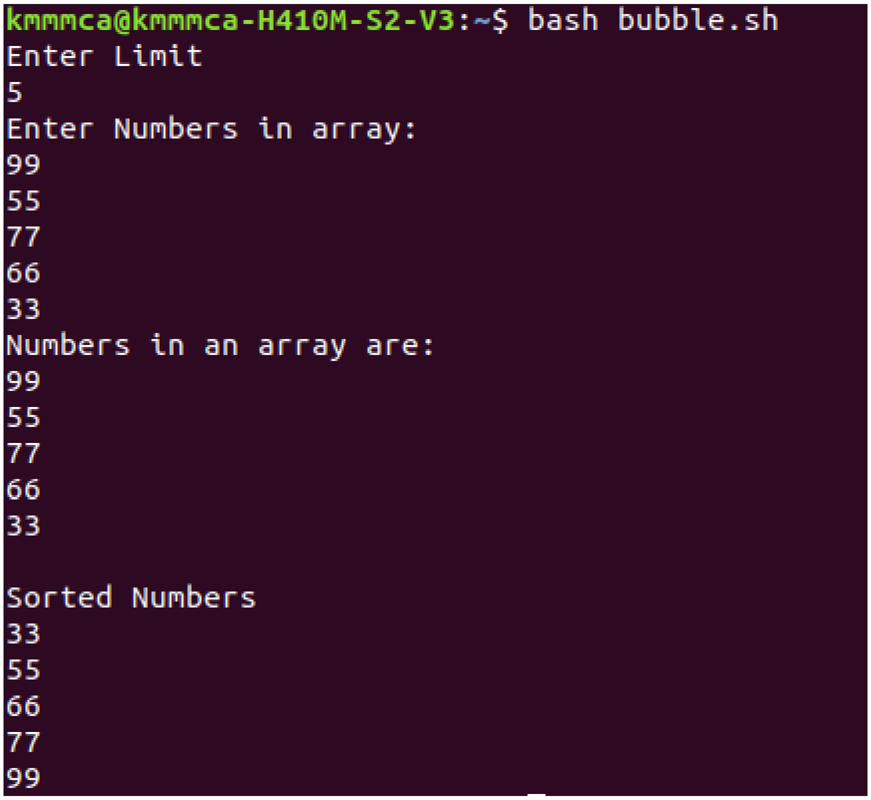
AIM

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

SOURCE CODE

```
#!/bin/bash
echo "Enter Limit"
read n
echo "Enter Numbers in array:"
for (( i = 0; i < $n; i++ ))
do
read arr[$i]
done
echo "Numbers in an array are:"
for (( i = 0; i < $n; i++ ))
do
echo ${arr[$i]}
done
for (( i = 0; i < $n ; i++ ))
do
for (( j = $i; j < $n; j++ ))
do
if [ ${arr[$i]} -gt ${arr[$j]}  ];
then
t=${arr[$i]}
arr[$i]=${arr[$j]}
arr[$j]=$t
fi
done
done
echo -e "\nSorted Numbers "
for (( i=0; i < $n; i++ ))
do
echo ${arr[$i]}
done
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash bubble.sh
Enter Limit
5
Enter Numbers in array:
99
55
77
66
33
Numbers in an array are:
99
55
77
66
33

Sorted Numbers
33
55
66
77
99
```

Program N0:-24 **Date: 09-03-2023**

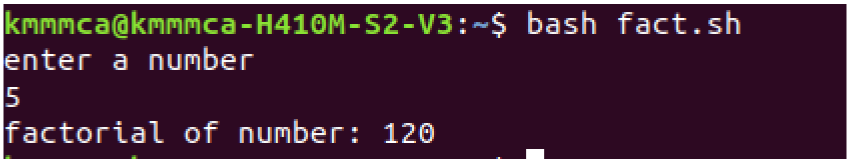
AIM

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

SOURCE CODE

```
#!/bin/bash
factorial()
{
num=$1
fact=1
if [ $num == 0 ]
then
echo "Factorial of number: 1"
elif [ $num == 1 ]
then
echo "Factorial of number 1"
else
for (( i=1; i<=num; i++ ))
do
fact=$((fact*i))
done
echo "factorial of number: $fact "
fi
}
echo "enter a number"
read num
factorial $num
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash fact.sh
enter a number
5
factorial of number: 120
```

Program N0:-25 **Date: 14-03-2023**

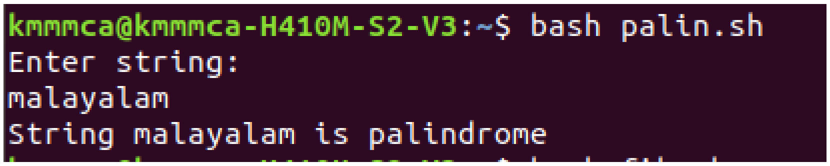
AIM

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

SOURCE CODE

```
#!/bin/bash
strpal()
{
len=${#str}
while [ $len -ne 0 ]
do
rev=$rev`echo $str|cut -c $len`
len=$(($len-1))
done
if [ $str = $rev ]
then
echo "String $str is palindrome"
else
echo "$str is not palindrome"
fi
}
echo "Enter string:"
read str
strpal $str
```

OUTPUT



```
kmmmca@kmmmca-H410M-S2-V3:~$ bash palin.sh
Enter string:
malayalam
String malayalam is palindrome
```

Program N0:-26 **Date: 14-03-2023**

AIM

Write a script to rename all c files to cpp files.

SOURCE CODE

```
#! /bin/bash
for x in `ls *\.c`
do
mv $x ${x}pp
done
echo " renamed files"
```

OUTPUT

```
kmmmca@kmmmca-H410M-S2-V3:~ $ ls
a1  a2  a3.c  combo  ctocpp  f  file1.c  file2.c  fsame  special
kmmmca@kmmmca-H410M-S2-V3:~ $ bash ctocpp
" renamed files"
kmmmca@kmmmca-H410M-S2-V3:~ $ ls
a1  a2  a3.cpp  combo  ctocpp  f  file1.cpp  file2.cpp  fsame  special
```

Program N0:-27 **Date: 14-03-2023**

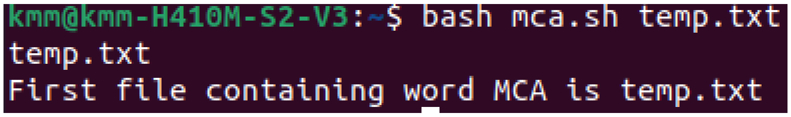
AIM

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

SOURCE CODE

```
#!bin/bash
for i in $*
do
w=`grep -c "MCA" $i`
echo "$i"
if [ $w -gt 0 ]
then
echo "First file containing word MCA is $1"
exit
fi
done
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ bash mca.sh temp.txt
temp.txt
First file containing word MCA is temp.txt
```

Program N0:-28 **Date: 14-03-2023**

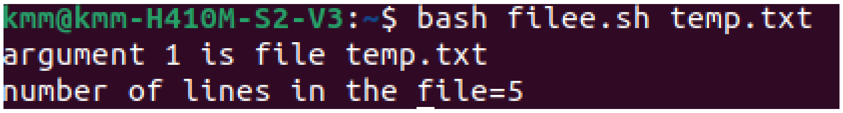
AIM

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 a 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.

SOURCE CODE

```
#!bin/bash
i=1
for arg in $*
do
if [ -f $arg ]
then
echo "argument $i is file $arg "
echo "number of lines in the file=`wc -l $arg | cut -d ' ' -f 1`"
else
echo "argument $i is directory $arg"
fi
((i=i+1))
done
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ bash filee.sh temp.txt
argument 1 is file temp.txt
number of lines in the file=5
```

Program N0:-29 **Date: 18-03-2023**

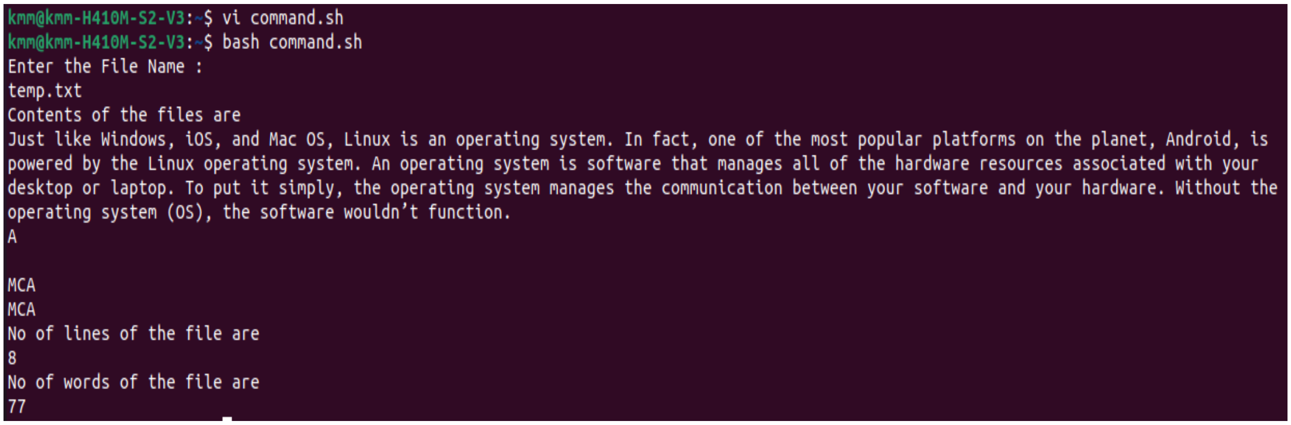
AIM

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.

SOURCE CODE

```
#!bin/bash
if [ $# -eq 0 ]
then
echo "Enter the File Name :"
read f1
else
f1=$1
fi
echo "Contents of the files are "
cat $f1
echo "No of lines of the file are "
wc -l $f1|cut -d " " -f1
echo "No of words of the file are"
wc -w $f1|cut -d" " -f1
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ vi command.sh
kmm@kmm-H410M-S2-V3:~$ bash command.sh
Enter the File Name :
temp.txt
Contents of the files are
Just like Windows, iOS, and Mac OS, Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is
powered by the Linux operating system. An operating system is software that manages all of the hardware resources associated with your
desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the
operating system (OS), the software wouldn’t function.
A

MCA
MCA
No of lines of the file are
8
No of words of the file are
77
```

Program N0:-30 **Date: 18-03-2023**

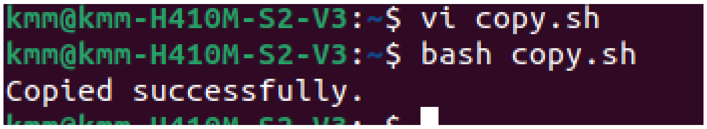
AIM

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.

SOURCE CODE

```
#!bin/bash
r=$#
if [ $((r%2)) -ne 0 ]
then
echo "Odd number of files .."
else
for((i=0;i<$r;i=i+2))
do
f1=$1
f2=$2
cp $f1 $f2
shift 2
done
echo "Copied successfully."
Fi
```

OUTPUT



```
kmm@kmm-H410M-S2-V3:~$ vi copy.sh
kmm@kmm-H410M-S2-V3:~$ bash copy.sh
Copied successfully.
kmm@kmm-H410M-S2-V3:~$
```

Program N0:-31 **Date: 18-03-2023**

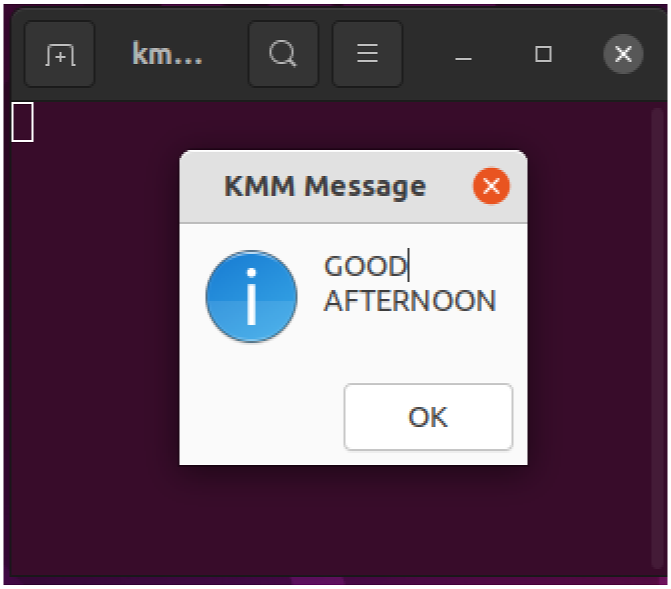
AIM

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.

SOURCE CODE

```
#!bin/bash
clear
k=`date +"%k"`
if [ $k -gt 0 -a $k -lt 12 ]
then
zenity --info --text="GOOD MORNING" --title="KMM Message"
elif [ $k -gt 12 -a $k -lt 16 ]
then
zenity --info --text="GOOD AFTERNOON" --title="KMM Message"
elif [ $k -gt 16 -a $k -lt 18 ]
then
zenity --info --text="GOOD EVENING" --title="KMM Message"
else
zenity --info --text="GOOD NIGHT" --title="KMM Message"
fi
```

OUTPUT



GOOD AFTERNOON