VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES

VIVEKANANDA SCHOOL OF INFORMATION TECHNOLOGY

VIPS

योगः कर्मसु कौशलम्

IN PURSUIT OF PERFECTION

BACHELOR OF COMPUTER APPLICATION
Practical- LINUX – OS LAB File
BCA-371

Guru Gobind Singh Indraprastha University
Sector - 16C Dwarka, Delhi – 110078

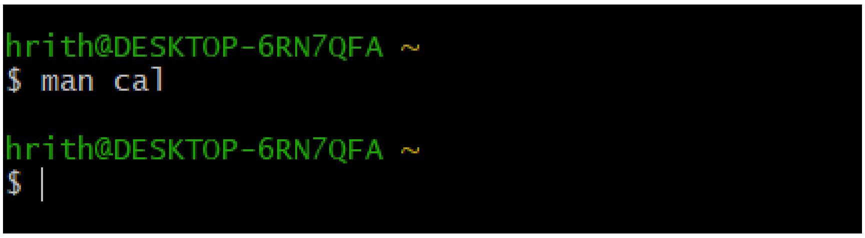
SUBMITTED TO:
Ms. Megha Bansal
Assistant Professor
VSIT

SUBMITTED BY:
Tushar Arora
00529802021
BCA 5EA

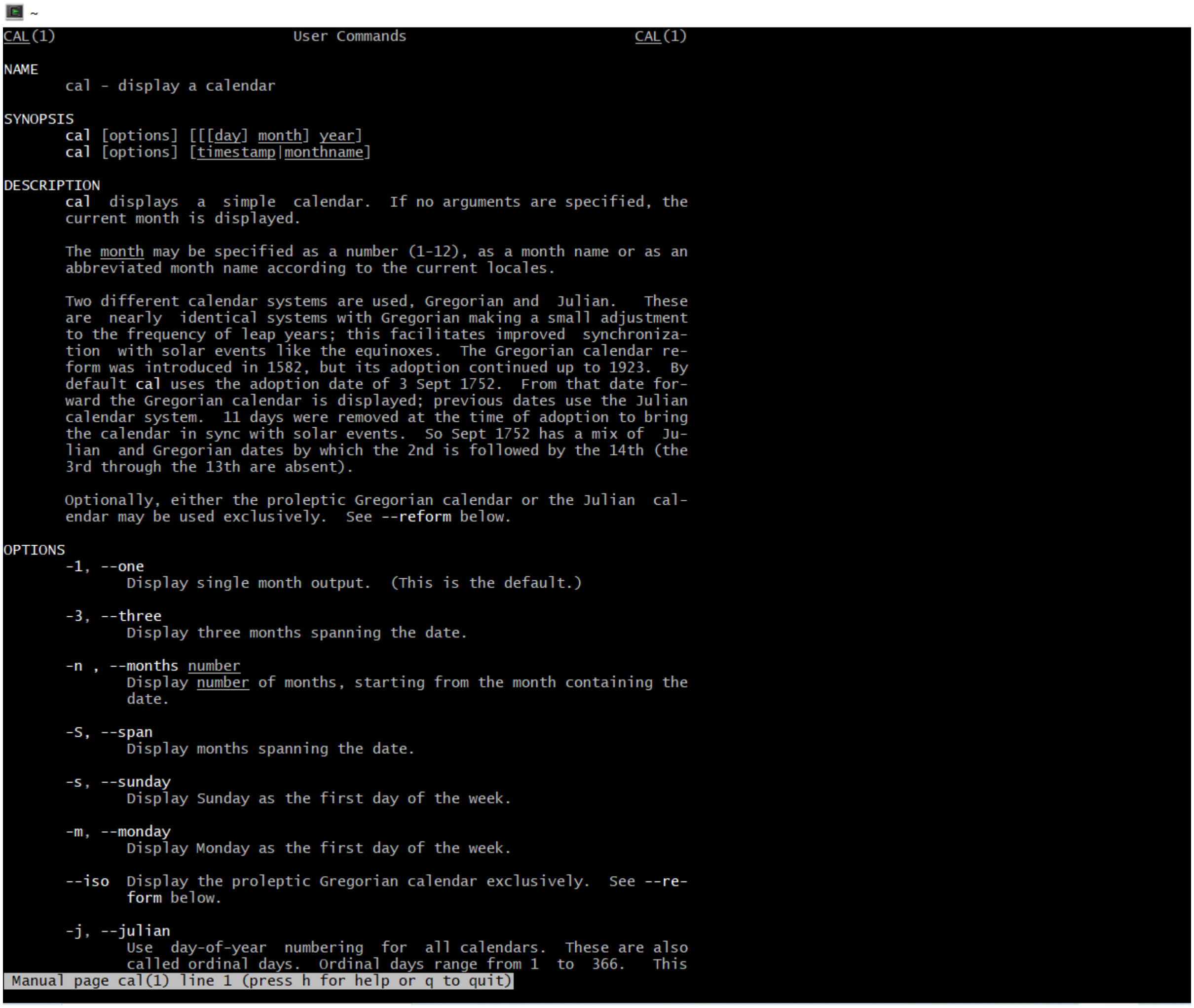
INDEX

Ques. To understand help commands like: -man, info, help, apropos, whatis.

1. Man - The man command is used to format and display the man pages.

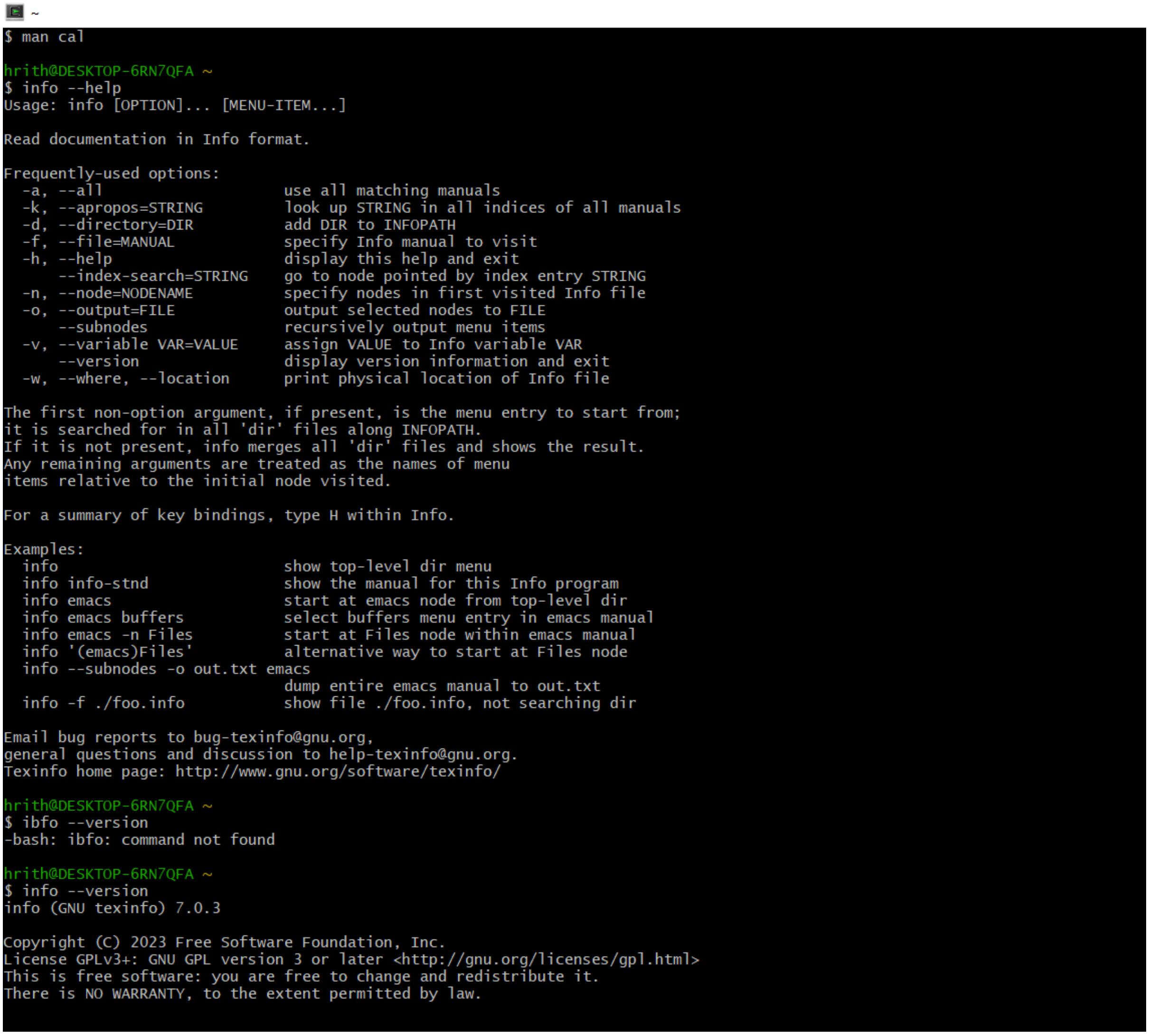


```
hrith@DESKTOP-6RN7QFA ~
$ man cal

hrith@DESKTOP-6RN7QFA ~
$ |
```

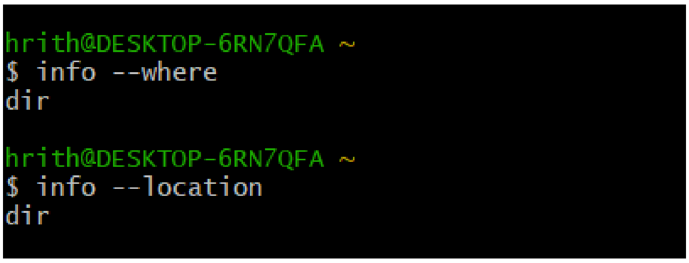


```
~
CAL(1)                      User Commands                      CAL(1)

NAME
       cal - display a calendar

SYNOPSIS
       cal [options] [[[day] month] year]
       cal [options] [timestamp|monthname]

DESCRIPTION
       cal  displays  a  simple  calendar.  If no arguments are specified, the
       current month is displayed.

       The month may be specified as a number (1-12), as a month name or as an
       abbreviated month name according to the current locales.

       Two different calendar systems are used, Gregorian and  Julian.   These
       are  nearly  identical systems with Gregorian making a small adjustment
       to the frequency of leap years; this facilitates improved  synchroniza-
       tion  with solar events like the equinoxes.  The Gregorian calendar re-
       form was introduced in 1582, but its adoption continued up to 1923.  By
       default cal uses the adoption date of 3 Sept 1752.  From that date for-
       ward the Gregorian calendar is displayed; previous dates use the Julian
       calendar system.  11 days were removed at the time of adoption to bring
       the calendar in sync with solar events.  So Sept 1752 has a mix of  Ju-
       lian  and Gregorian dates by which the 2nd is followed by the 14th (the
       3rd through the 13th are absent).

       Optionally, either the proleptic Gregorian calendar or the Julian  cal-
       endar may be used exclusively.  See --reform below.

OPTIONS
       -1, --one
              Display single month output.  (This is the default.)

       -3, --three
              Display three months spanning the date.

       -n , --months number
              Display number of months, starting from the month containing the
              date.

       -S, --span
              Display months spanning the date.

       -s, --sunday
              Display Sunday as the first day of the week.

       -m, --monday
              Display Monday as the first day of the week.

       --iso  Display the proleptic Gregorian calendar exclusively.  See --re-
              form below.

       -j, --julian
              Use  day-of-year  numbering  for  all calendars.  These are also
              called ordinal days.  Ordinal days range from 1  to  366.   This
 Manual page cal(1) line 1 (press h for help or q to quit)
```

2. Info – info reads documentation in the info format.

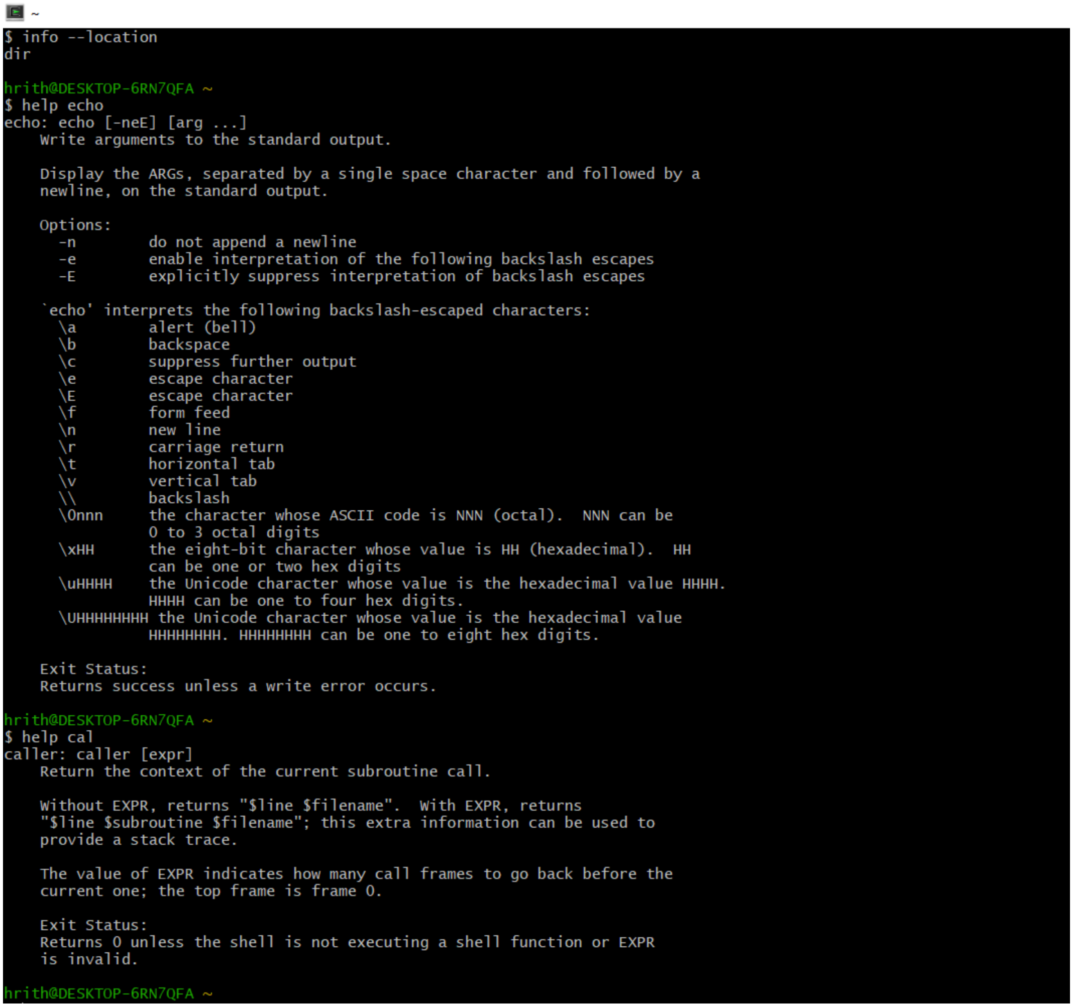


```
~
$ man cal

hrith@DESKTOP-6RN7QFA ~
$ info --help
Usage: info [OPTION]... [MENU-ITEM...]

Read documentation in Info format.

Frequently-used options:
  -a, --all                    use all matching manuals
  -k, --apropos=STRING         look up STRING in all indices of all manuals
  -d, --directory=DIR          add DIR to INFOPATH
  -f, --file=MANUAL            specify Info manual to visit
  -h, --help                   display this help and exit
      --index-search=STRING    go to node pointed by index entry STRING
  -n, --node=NODENAME          specify nodes in first visited Info file
  -o, --output=FILE            output selected nodes to FILE
      --subnodes               recursively output menu items
  -v, --variable VAR=VALUE     assign VALUE to Info variable VAR
      --version                display version information and exit
  -w, --where, --location      print physical location of Info file

The first non-option argument, if present, is the menu entry to start from;
it is searched for in all 'dir' files along INFOPATH.
If it is not present, info merges all 'dir' files and shows the result.
Any remaining arguments are treated as the names of menu
items relative to the initial node visited.

For a summary of key bindings, type H within Info.

Examples:
  info                         show top-level dir menu
  info info-stnd               show the manual for this Info program
  info emacs                   start at emacs node from top-level dir
  info emacs buffers           select buffers menu entry in emacs manual
  info emacs -n Files          start at Files node within emacs manual
  info '(emacs)Files'          alternative way to start at Files node
  info --subnodes -o out.txt emacs
                               dump entire emacs manual to out.txt
  info -f ./foo.info           show file ./foo.info, not searching dir

Email bug reports to bug-texinfo@gnu.org,
general questions and discussion to help-texinfo@gnu.org.
Texinfo home page: http://www.gnu.org/software/texinfo/

hrith@DESKTOP-6RN7QFA ~
$ ibfo --version
-bash: ibfo: command not found

hrith@DESKTOP-6RN7QFA ~
$ info --version
info (GNU texinfo) 7.0.3

Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://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.
```



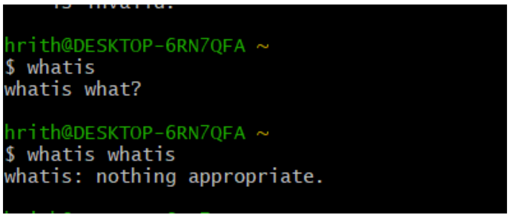
```
hrith@DESKTOP-6RN7QFA ~
$ info --where
dir

hrith@DESKTOP-6RN7QFA ~
$ info --location
dir
```

3. help – Display information about built-in commands.



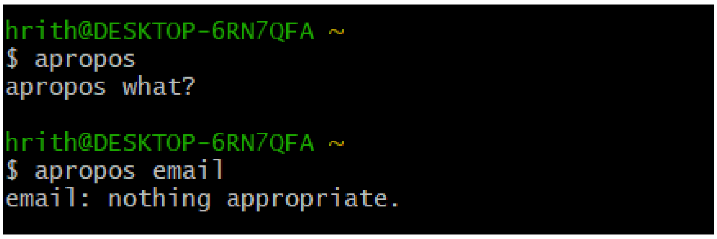
```
~
$ info --location
dir

hrith@DESKTOP-6RN7QFA ~
$ help echo
echo: echo [-neE] [arg ...]
    Write arguments to the standard output.

    Display the ARGs, separated by a single space character and followed by a
    newline, on the standard output.

    Options:
      -n        do not append a newline
      -e        enable interpretation of the following backslash escapes
      -E        explicitly suppress interpretation of backslash escapes

    `echo' interprets the following backslash-escaped characters:
      \a        alert (bell)
      \b        backspace
      \c        suppress further output
      \e        escape character
      \E        escape character
      \f        form feed
      \n        new line
      \r        carriage return
      \t        horizontal tab
      \v        vertical tab
      \\        backslash
      \0nnn     the character whose ASCII code is NNN (octal).  NNN can be
                0 to 3 octal digits
      \xHH      the eight-bit character whose value is HH (hexadecimal).  HH
                can be one or two hex digits
      \uHHHH    the Unicode character whose value is the hexadecimal value HHHH.
                HHHH can be one to four hex digits.
      \UHHHHHHHH the Unicode character whose value is the hexadecimal value
                HHHHHHHH. HHHHHHHH can be one to eight hex digits.

    Exit Status:
    Returns success unless a write error occurs.

hrith@DESKTOP-6RN7QFA ~
$ help cal
caller: caller [expr]
    Return the context of the current subroutine call.

    Without EXPR, returns "$line $filename".  With EXPR, returns
    "$line $subroutine $filename"; this extra information can be used to
    provide a stack trace.

    The value of EXPR indicates how many call frames to go back before the
    current one; the top frame is frame 0.

    Exit Status:
    Returns 0 unless the shell is not executing a shell function or EXPR
    is invalid.

hrith@DESKTOP-6RN7QFA ~
```

4. whatis - whatis command in Linux is used to get a one-line manual page description



```
hrith@DESKTOP-6RN7QFA ~
$ whatis
whatis what?

hrith@DESKTOP-6RN7QFA ~
$ whatis whatis
whatis: nothing appropriate.
```

5. apropos - The apropos command shows the manual sections that contain any of the keywords.

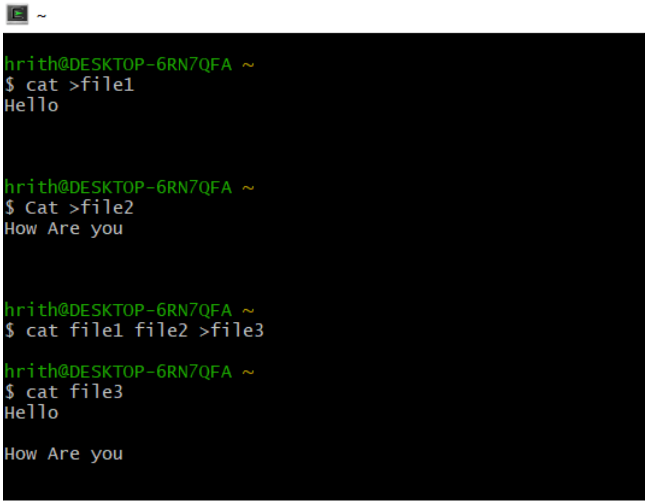


```
hrith@DESKTOP-6RN7QFA ~
$ apropos
apropos what?

hrith@DESKTOP-6RN7QFA ~
$ apropos email
email: nothing appropriate.
```

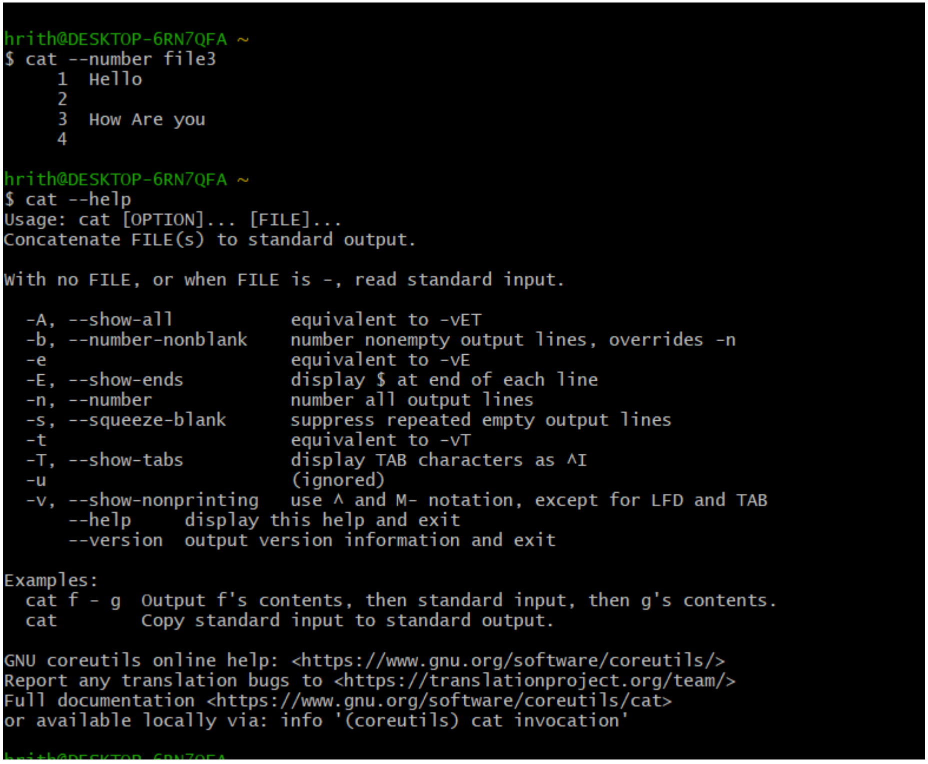
Ques. To understand basic directory navigation commands like cat, cd, mv, cp, rm, mkdir, rmdir, file, pwd command.

1. cat - Concatenate files

The command **cat** is a multi-purpose utility and is mostly used with text files.

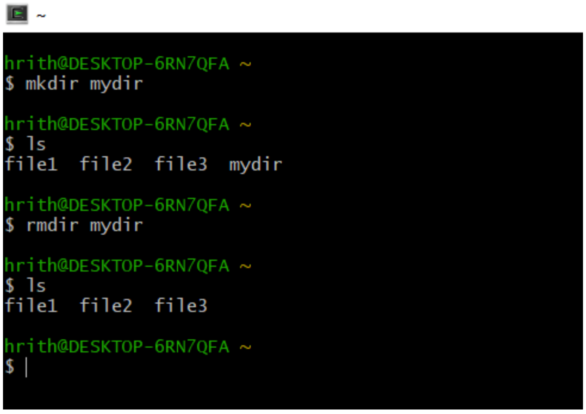


```
hrith@DESKTOP-6RN7QFA ~
$ cat >file1
Hello


hrith@DESKTOP-6RN7QFA ~
$ Cat >file2
How Are you


hrith@DESKTOP-6RN7QFA ~
$ cat file1 file2 >file3

hrith@DESKTOP-6RN7QFA ~
$ cat file3
Hello

How Are you
```



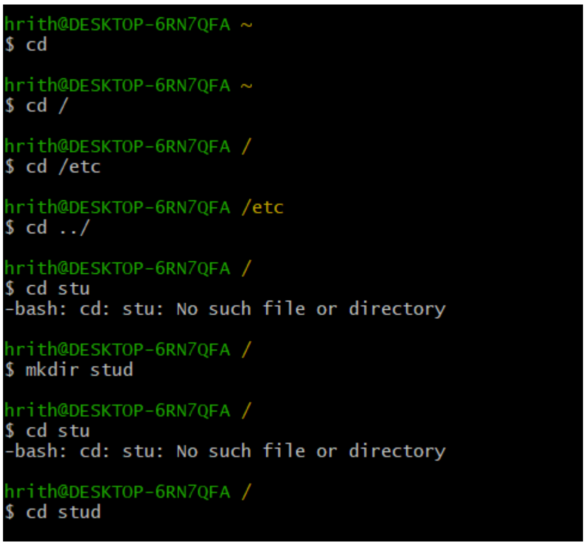
```
hrith@DESKTOP-6RN7QFA ~
$ cat --number file3
     1  Hello
     2
     3  How Are you
     4

hrith@DESKTOP-6RN7QFA ~
$ cat --help
Usage: cat [OPTION]... [FILE]...
Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

  -A, --show-all           equivalent to -vET
  -b, --number-nonblank    number nonempty output lines, overrides -n
  -e                       equivalent to -vE
  -E, --show-ends          display $ at end of each line
  -n, --number             number all output lines
  -s, --squeeze-blank      suppress repeated empty output lines
  -t                       equivalent to -vT
  -T, --show-tabs          display TAB characters as ^I
  -u                       (ignored)
  -v, --show-nonprinting   use ^ and M- notation, except for LFD and TAB
      --help     display this help and exit
      --version  output version information and exit

Examples:
  cat f - g  Output f's contents, then standard input, then g's contents.
  cat        Copy standard input to standard output.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/cat>
or available locally via: info '(coreutils) cat invocation'

hrith@DESKTOP-6RN7QFA
```

2. mkdir, rmdir - Create/Delete directories



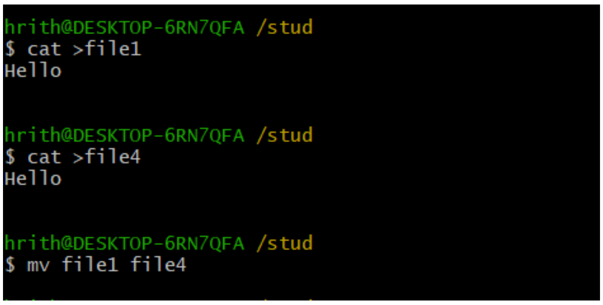
```
hrith@DESKTOP-6RN7QFA ~
$ mkdir mydir

hrith@DESKTOP-6RN7QFA ~
$ ls
file1  file2  file3  mydir

hrith@DESKTOP-6RN7QFA ~
$ rmdir mydir

hrith@DESKTOP-6RN7QFA ~
$ ls
file1  file2  file3

hrith@DESKTOP-6RN7QFA ~
$ |
```

3. cd - Switch to another directory



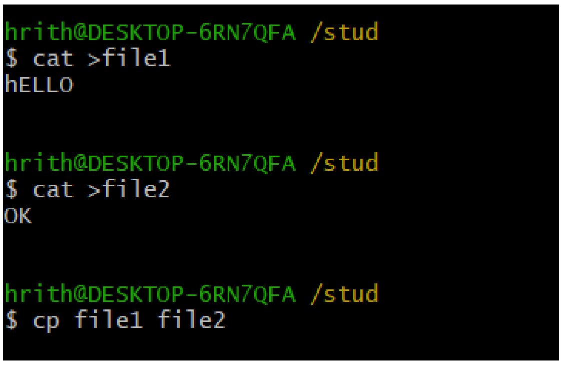
```
hrith@DESKTOP-6RN7QFA ~
$ cd

hrith@DESKTOP-6RN7QFA ~
$ cd /

hrith@DESKTOP-6RN7QFA /
$ cd /etc

hrith@DESKTOP-6RN7QFA /etc
$ cd ../

hrith@DESKTOP-6RN7QFA /
$ cd stu
-bash: cd: stu: No such file or directory

hrith@DESKTOP-6RN7QFA /
$ mkdir stud

hrith@DESKTOP-6RN7QFA /
$ cd stu
-bash: cd: stu: No such file or directory

hrith@DESKTOP-6RN7QFA /
$ cd stud
```

4. mv - The command **mv** is used to rename a file



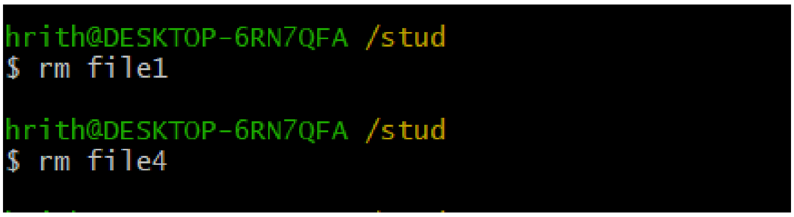
```
hrith@DESKTOP-6RN7QFA /stud
$ cat >file1
Hello


hrith@DESKTOP-6RN7QFA /stud
$ cat >file4
Hello


hrith@DESKTOP-6RN7QFA /stud
$ mv file1 file4
```

5. cp - To copy a file, the command **cp** is used



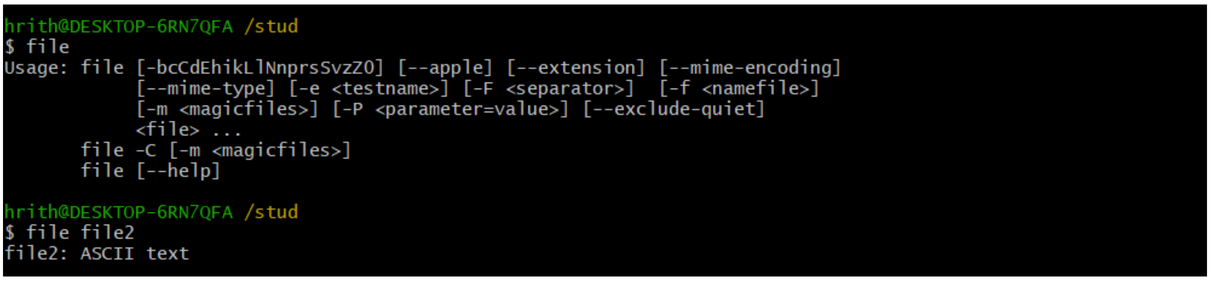
```
hrith@DESKTOP-6RN7QFA /stud
$ cat >file1
hELLO


hrith@DESKTOP-6RN7QFA /stud
$ cat >file2
OK


hrith@DESKTOP-6RN7QFA /stud
$ cp file1 file2
```

6. rm - Delete files



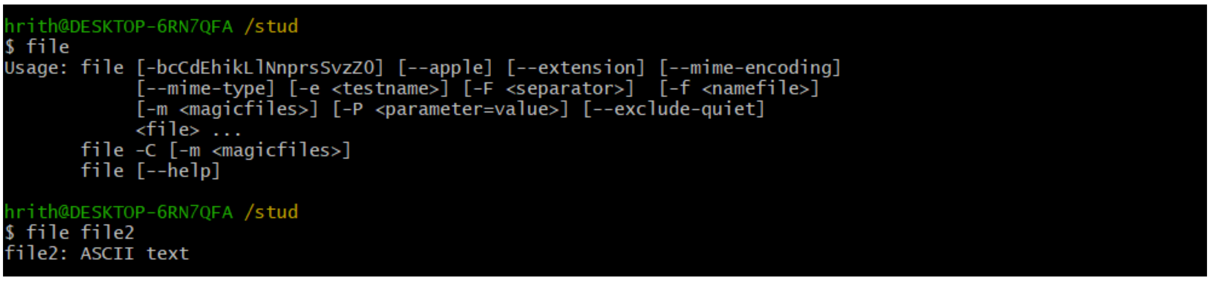
```
hrith@DESKTOP-6RN7QFA /stud
$ rm file1

hrith@DESKTOP-6RN7QFA /stud
$ rm file4
```

7. file



```
hrith@DESKTOP-6RN7QFA /stud
$ file
Usage: file [-bcCdEhikLlNnprsSvzZ0] [--apple] [--extension] [--mime-encoding]
            [--mime-type] [-e <testname>] [-F <separator>]  [-f <namefile>]
            [-m <magicfiles>] [-P <parameter=value>] [--exclude-quiet]
            <file> ...
       file -C [-m <magicfiles>]
       file [--help]

hrith@DESKTOP-6RN7QFA /stud
$ file file2
file2: ASCII text
```

8. pwd - Display name of current directory

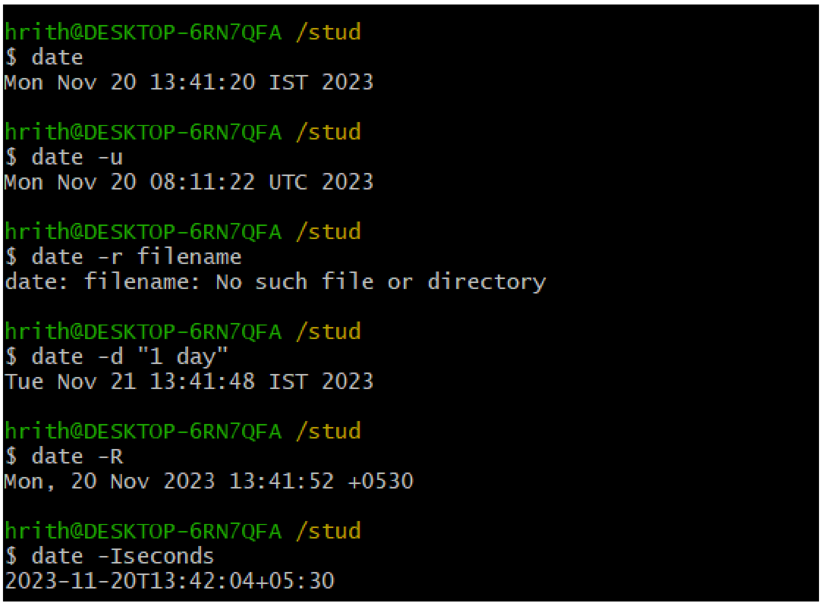


```
hrith@DESKTOP-6RN7QFA /stud
$ file
Usage: file [-bcCdEhikLlNnprsSvzZ0] [--apple] [--extension] [--mime-encoding]
            [--mime-type] [-e <testname>] [-F <separator>]  [-f <namefile>]
            [-m <magicfiles>] [-P <parameter=value>] [--exclude-quiet]
            <file> ...
       file -C [-m <magicfiles>]
       file [--help]

hrith@DESKTOP-6RN7QFA /stud
$ file file2
file2: ASCII text
```

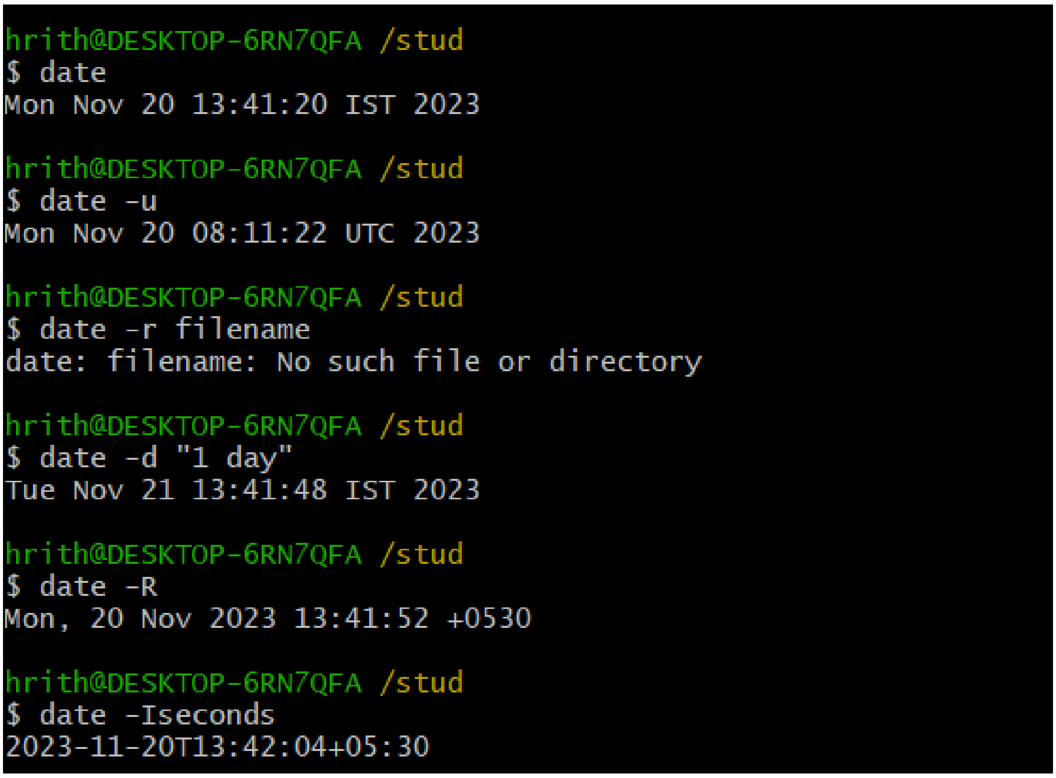
Ques. To understand basic commands like:

date, cal, echo, bc, ls, who, whoami, hostname, uname, tty, aliase.

1. date

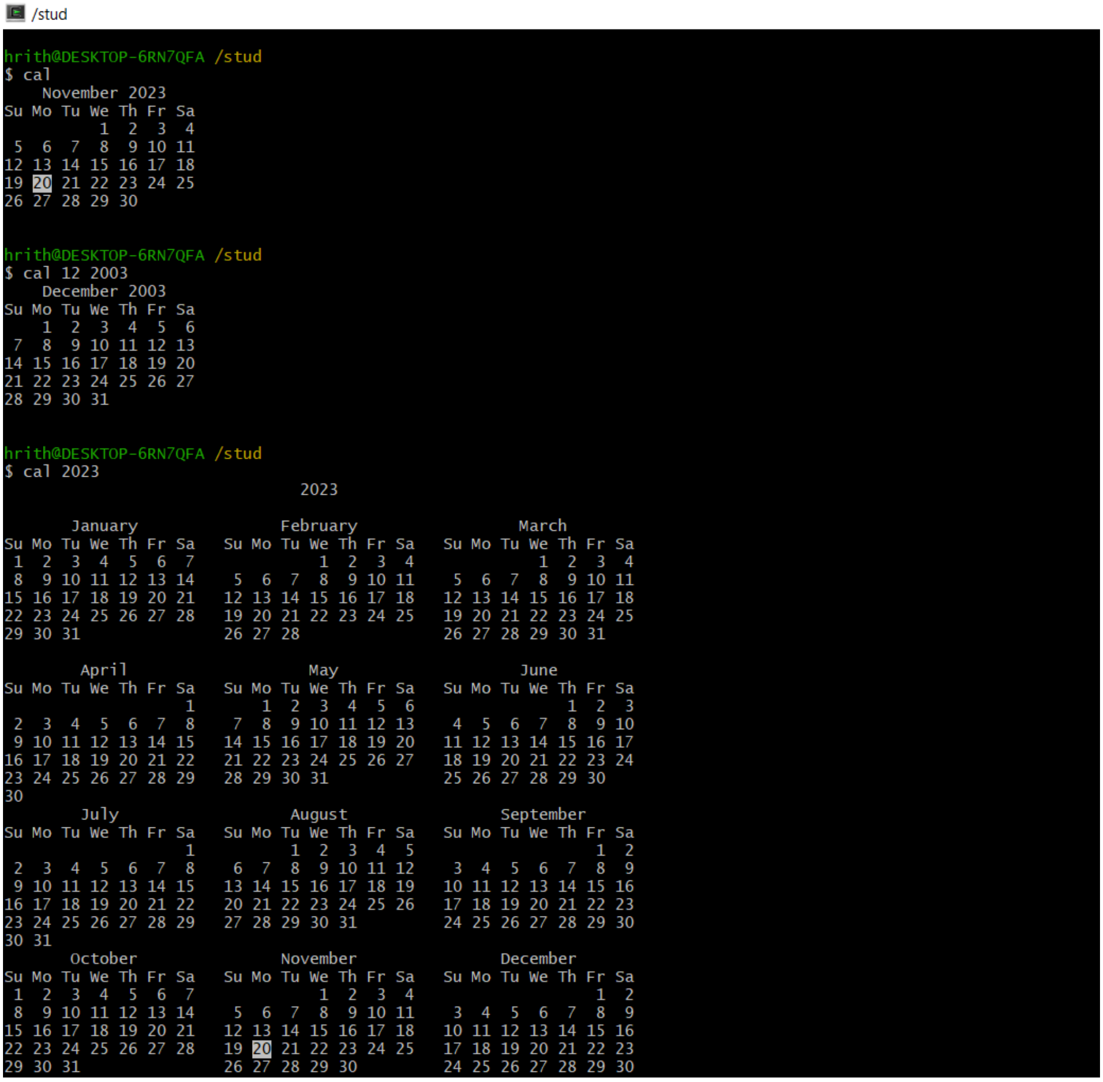


```
hrith@DESKTOP-6RN7QFA /stud
$ date
Mon Nov 20 13:41:20 IST 2023

hrith@DESKTOP-6RN7QFA /stud
$ date -u
Mon Nov 20 08:11:22 UTC 2023

hrith@DESKTOP-6RN7QFA /stud
$ date -r filename
date: filename: No such file or directory

hrith@DESKTOP-6RN7QFA /stud
$ date -d "1 day"
Tue Nov 21 13:41:48 IST 2023

hrith@DESKTOP-6RN7QFA /stud
$ date -R
Mon, 20 Nov 2023 13:41:52 +0530

hrith@DESKTOP-6RN7QFA /stud
$ date -Iseconds
2023-11-20T13:42:04+05:30
```

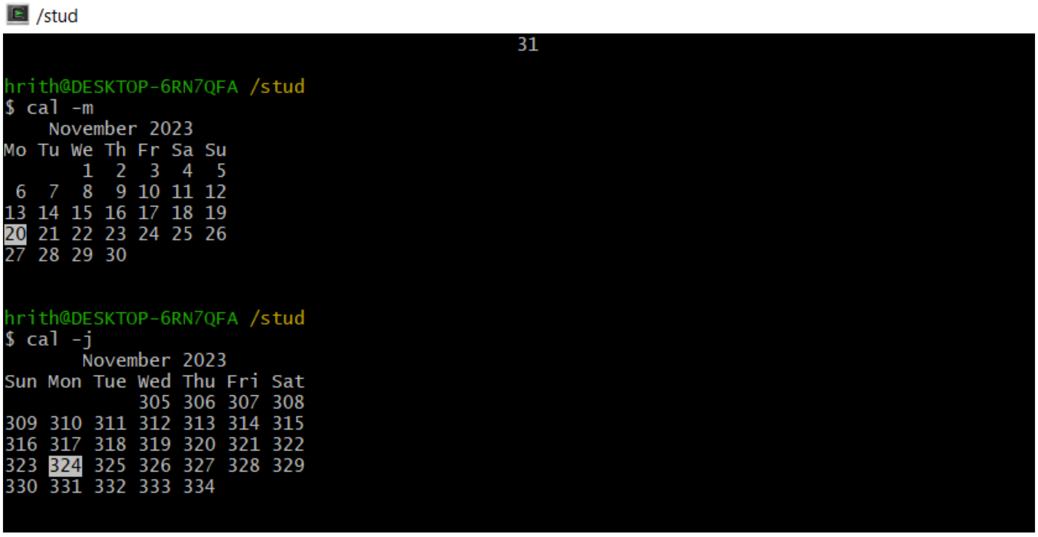


```
hrith@DESKTOP-6RN7QFA /stud
$ date
Mon Nov 20 13:41:20 IST 2023

hrith@DESKTOP-6RN7QFA /stud
$ date -u
Mon Nov 20 08:11:22 UTC 2023

hrith@DESKTOP-6RN7QFA /stud
$ date -r filename
date: filename: No such file or directory

hrith@DESKTOP-6RN7QFA /stud
$ date -d "1 day"
Tue Nov 21 13:41:48 IST 2023

hrith@DESKTOP-6RN7QFA /stud
$ date -R
Mon, 20 Nov 2023 13:41:52 +0530

hrith@DESKTOP-6RN7QFA /stud
$ date -Iseconds
2023-11-20T13:42:04+05:30
```

2. cal

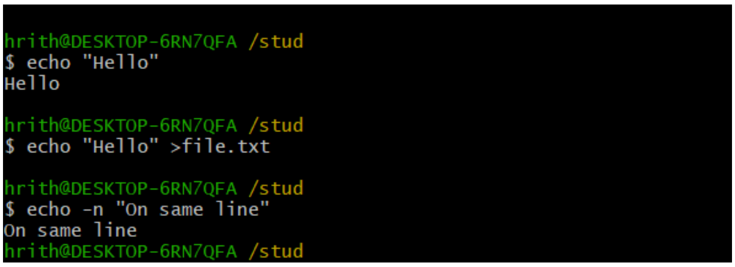


```
hrith@DESKTOP-6RN7QFA /stud
$ cal
    November 2023
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30


hrith@DESKTOP-6RN7QFA /stud
$ cal 12 2003
    December 2003
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31


hrith@DESKTOP-6RN7QFA /stud
$ cal 2023
                            2023

       January               February               March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7            1  2  3  4            1  2  3  4
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   5  6  7  8  9 10 11
15 16 17 18 19 20 21  12 13 14 15 16 17 18  12 13 14 15 16 17 18
22 23 24 25 26 27 28  19 20 21 22 23 24 25  19 20 21 22 23 24 25
29 30 31              26 27 28              26 27 28 29 30 31

        April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1      1  2  3  4  5  6               1  2  3
 2  3  4  5  6  7  8   7  8  9 10 11 12 13   4  5  6  7  8  9 10
 9 10 11 12 13 14 15  14 15 16 17 18 19 20  11 12 13 14 15 16 17
16 17 18 19 20 21 22  21 22 23 24 25 26 27  18 19 20 21 22 23 24
23 24 25 26 27 28 29  28 29 30 31           25 26 27 28 29 30
30
        July                 August                September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1         1  2  3  4  5                  1  2
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   3  4  5  6  7  8  9
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  10 11 12 13 14 15 16
16 17 18 19 20 21 22  20 21 22 23 24 25 26  17 18 19 20 21 22 23
23 24 25 26 27 28 29  27 28 29 30 31        24 25 26 27 28 29 30
30 31
       October               November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7            1  2  3  4                  1  2
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21  12 13 14 15 16 17 18  10 11 12 13 14 15 16
22 23 24 25 26 27 28  19 20 21 22 23 24 25  17 18 19 20 21 22 23
29 30 31              26 27 28 29 30        24 25 26 27 28 29 30
```



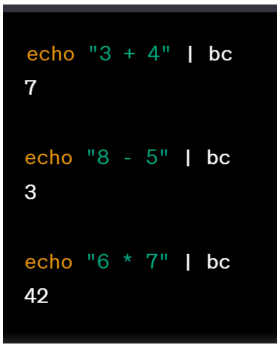
```
                                                  31

hrith@DESKTOP-6RN7QFA /stud
$ cal -m
    November 2023
Mo Tu We Th Fr Sa Su
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30


hrith@DESKTOP-6RN7QFA /stud
$ cal -j
       November 2023
Sun Mon Tue Wed Thu Fri Sat
            305 306 307 308
309 310 311 312 313 314 315
316 317 318 319 320 321 322
323 324 325 326 327 328 329
330 331 332 333 334
```

3. echo



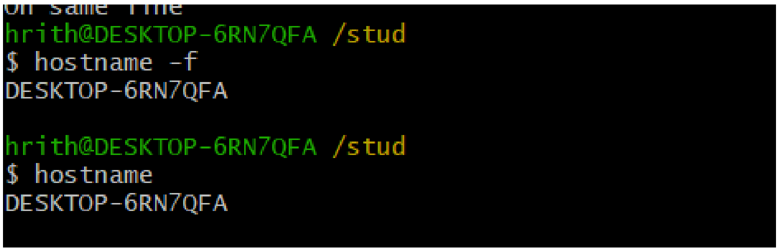
```
hrith@DESKTOP-6RN7QFA /stud
$ echo "Hello"
Hello

hrith@DESKTOP-6RN7QFA /stud
$ echo "Hello" >file.txt

hrith@DESKTOP-6RN7QFA /stud
$ echo -n "On same line"
On same line
hrith@DESKTOP-6RN7QFA /stud
```

4. bc



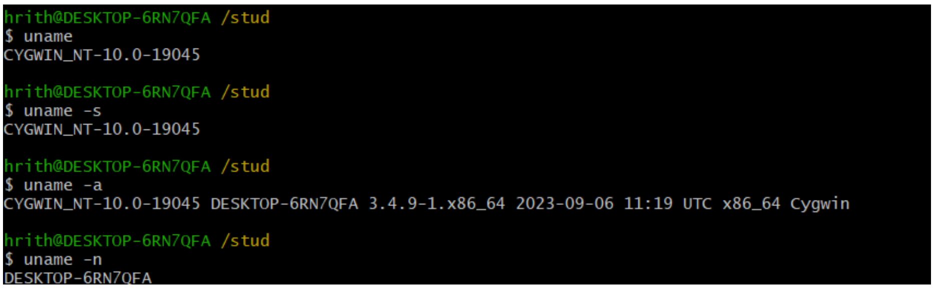
```
echo "3 + 4" | bc
7

echo "8 - 5" | bc
3

echo "6 * 7" | bc
42
```

5. hostname



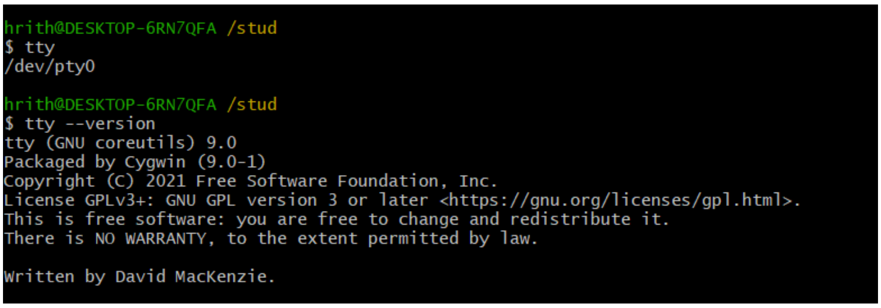
```
On same line
hrith@DESKTOP-6RN7QFA /stud
$ hostname -f
DESKTOP-6RN7QFA

hrith@DESKTOP-6RN7QFA /stud
$ hostname
DESKTOP-6RN7QFA
```

6. uname



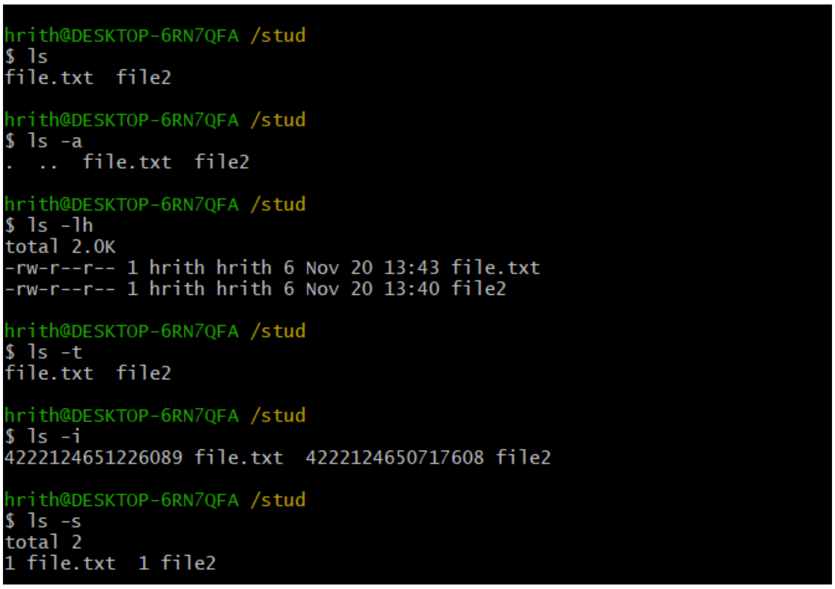
```
hrith@DESKTOP-6RN7QFA /stud
$ uname
CYGWIN_NT-10.0-19045

hrith@DESKTOP-6RN7QFA /stud
$ uname -s
CYGWIN_NT-10.0-19045

hrith@DESKTOP-6RN7QFA /stud
$ uname -a
CYGWIN_NT-10.0-19045 DESKTOP-6RN7QFA 3.4.9-1.x86_64 2023-09-06 11:19 UTC x86_64 Cygwin

hrith@DESKTOP-6RN7QFA /stud
$ uname -n
DESKTOP-6RN7QFA
```

7. tty



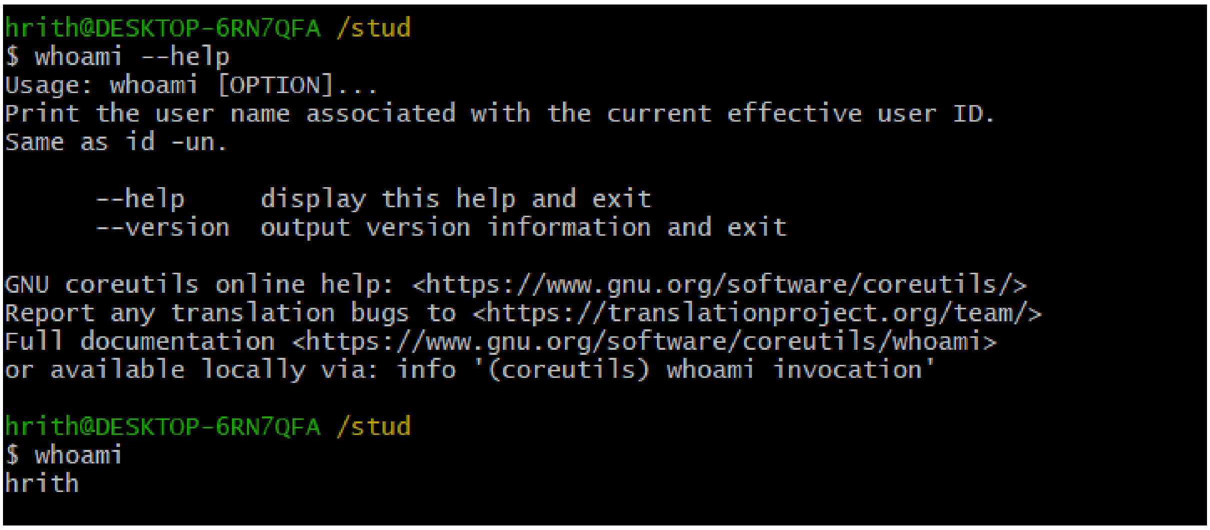
```
hrith@DESKTOP-6RN7QFA /stud
$ tty
/dev/pty0

hrith@DESKTOP-6RN7QFA /stud
$ tty --version
tty (GNU coreutils) 9.0
Packaged by Cygwin (9.0-1)
Copyright (C) 2021 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 David MacKenzie.
```

8. ls



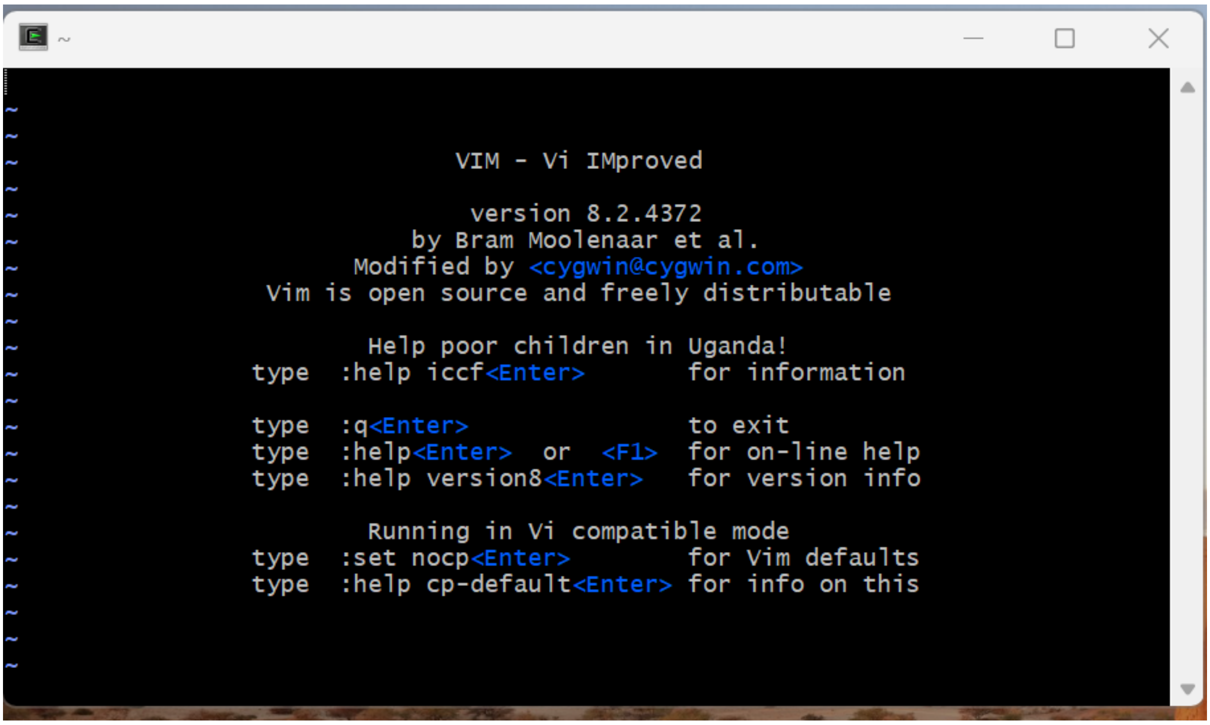
```
hrith@DESKTOP-6RN7QFA /stud
$ ls
file.txt  file2

hrith@DESKTOP-6RN7QFA /stud
$ ls -a
.  ..  file.txt  file2

hrith@DESKTOP-6RN7QFA /stud
$ ls -lh
total 2.0K
-rw-r--r-- 1 hrith hrith 6 Nov 20 13:43 file.txt
-rw-r--r-- 1 hrith hrith 6 Nov 20 13:40 file2

hrith@DESKTOP-6RN7QFA /stud
$ ls -t
file.txt  file2

hrith@DESKTOP-6RN7QFA /stud
$ ls -i
4222124651226089 file.txt  4222124650717608 file2

hrith@DESKTOP-6RN7QFA /stud
$ ls -s
total 2
1 file.txt  1 file2
```

9. whoami

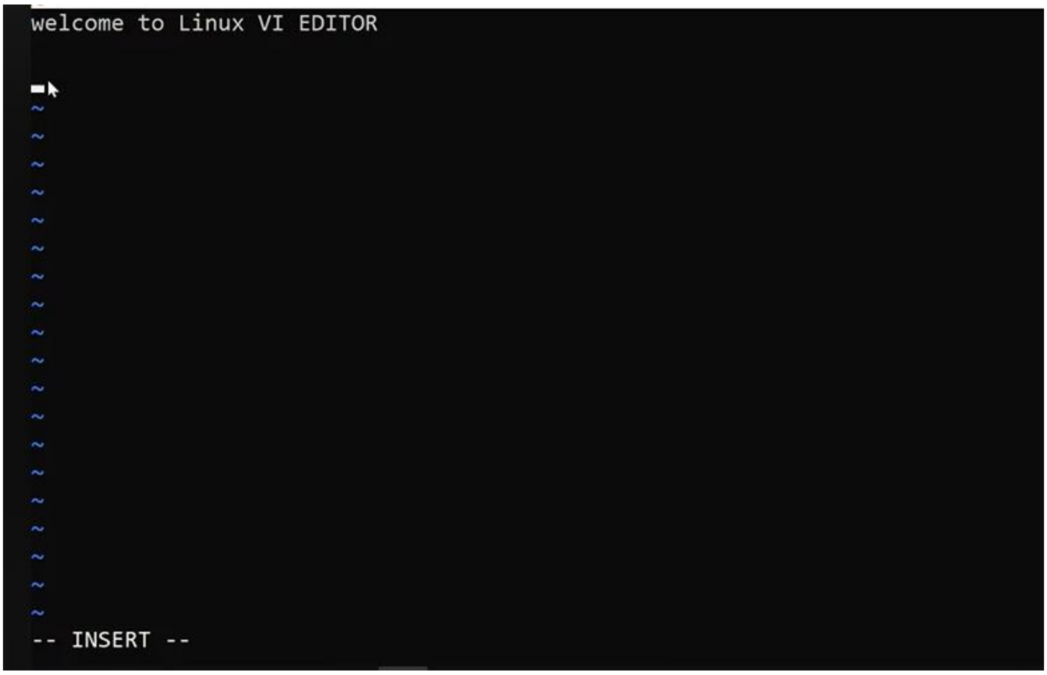


```
hrith@DESKTOP-6RN7QFA /stud
$ whoami --help
Usage: whoami [OPTION]...
Print the user name associated with the current effective user ID.
Same as id -un.

      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/whoami>
or available locally via: info '(coreutils) whoami invocation'

hrith@DESKTOP-6RN7QFA /stud
$ whoami
hrith
```

Ques. To understand vi basics, Three modes of vi Editor, how to write, save, execute a shell script in vi editor.



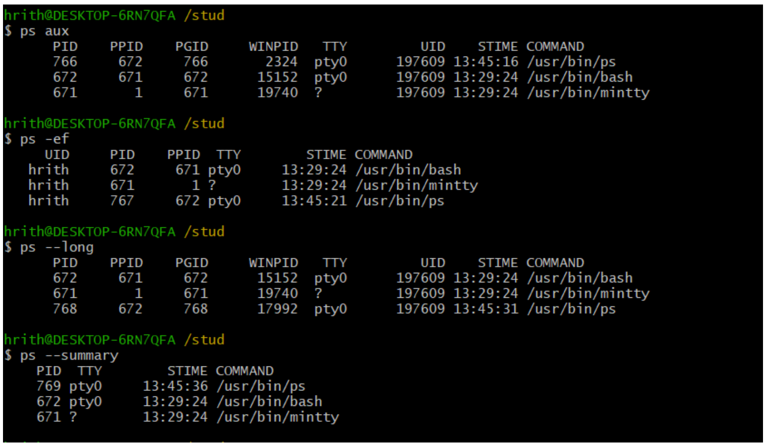
```
~
~
~                    VIM - Vi IMproved
~
~                     version 8.2.4372
~                 by Bram Moolenaar et al.
~            Modified by <cygwin@cygwin.com>
~      Vim is open source and freely distributable
~
~              Help poor children in Uganda!
~     type  :help iccf<Enter>       for information
~
~     type  :q<Enter>               to exit
~     type  :help<Enter>  or  <F1>  for on-line help
~     type  :help version8<Enter>   for version info
~
~              Running in Vi compatible mode
~     type  :set nocp<Enter>        for Vim defaults
~     type  :help cp-default<Enter> for info on this
~
~
~
```



```
welcome to Linux VI EDITOR

▬
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

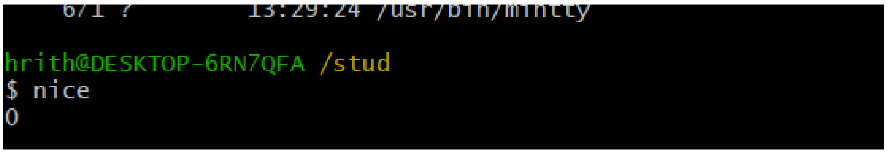
Ques. To understand process related commands like: - ps, top, nice, renice in Linux.

1. ps - The ps command writes the status of active processes



```
hrith@DESKTOP-6RN7QFA /stud
$ ps aux
      PID    PPID    PGID     WINPID   TTY         UID    STIME COMMAND
      766     672     766       2324  pty0      197609 13:45:16 /usr/bin/ps
      672     671     672      15152  pty0      197609 13:29:24 /usr/bin/bash
      671       1     671      19740  ?         197609 13:29:24 /usr/bin/mintty

hrith@DESKTOP-6RN7QFA /stud
$ ps -ef
     UID     PID    PPID  TTY        STIME COMMAND
   hrith     672     671 pty0     13:29:24 /usr/bin/bash
   hrith     671       1 ?        13:29:24 /usr/bin/mintty
   hrith     767     672 pty0     13:45:21 /usr/bin/ps

hrith@DESKTOP-6RN7QFA /stud
$ ps --long
      PID    PPID    PGID     WINPID   TTY         UID    STIME COMMAND
      672     671     672      15152  pty0      197609 13:29:24 /usr/bin/bash
      671       1     671      19740  ?         197609 13:29:24 /usr/bin/mintty
      768     672     768      17992  pty0      197609 13:45:31 /usr/bin/ps

hrith@DESKTOP-6RN7QFA /stud
$ ps --summary
    PID  TTY        STIME COMMAND
    769 pty0     13:45:36 /usr/bin/ps
    672 pty0     13:29:24 /usr/bin/bash
    671 ?        13:29:24 /usr/bin/mintty
```

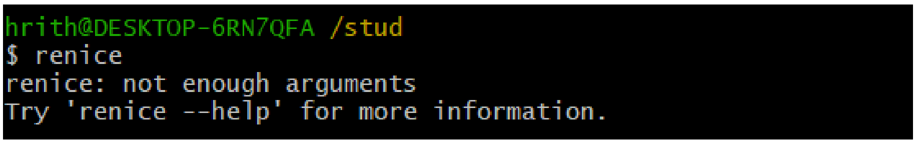
2. nice - The nice command lets you run a command at a priority lower than the command's normal priority



```
    671 ?        13:29:24 /usr/bin/mintty

hrith@DESKTOP-6RN7QFA /stud
$ nice
0
```

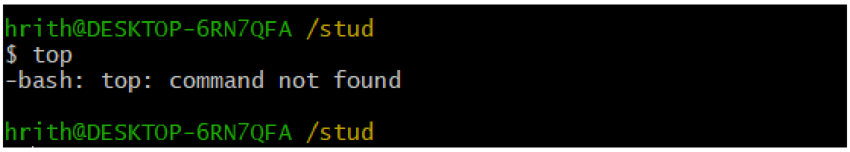
3. renice –

The **renice command** alters the nice value of one or more running processes.



```
hrith@DESKTOP-6RN7QFA /stud
$ renice
renice: not enough arguments
Try 'renice --help' for more information.
```

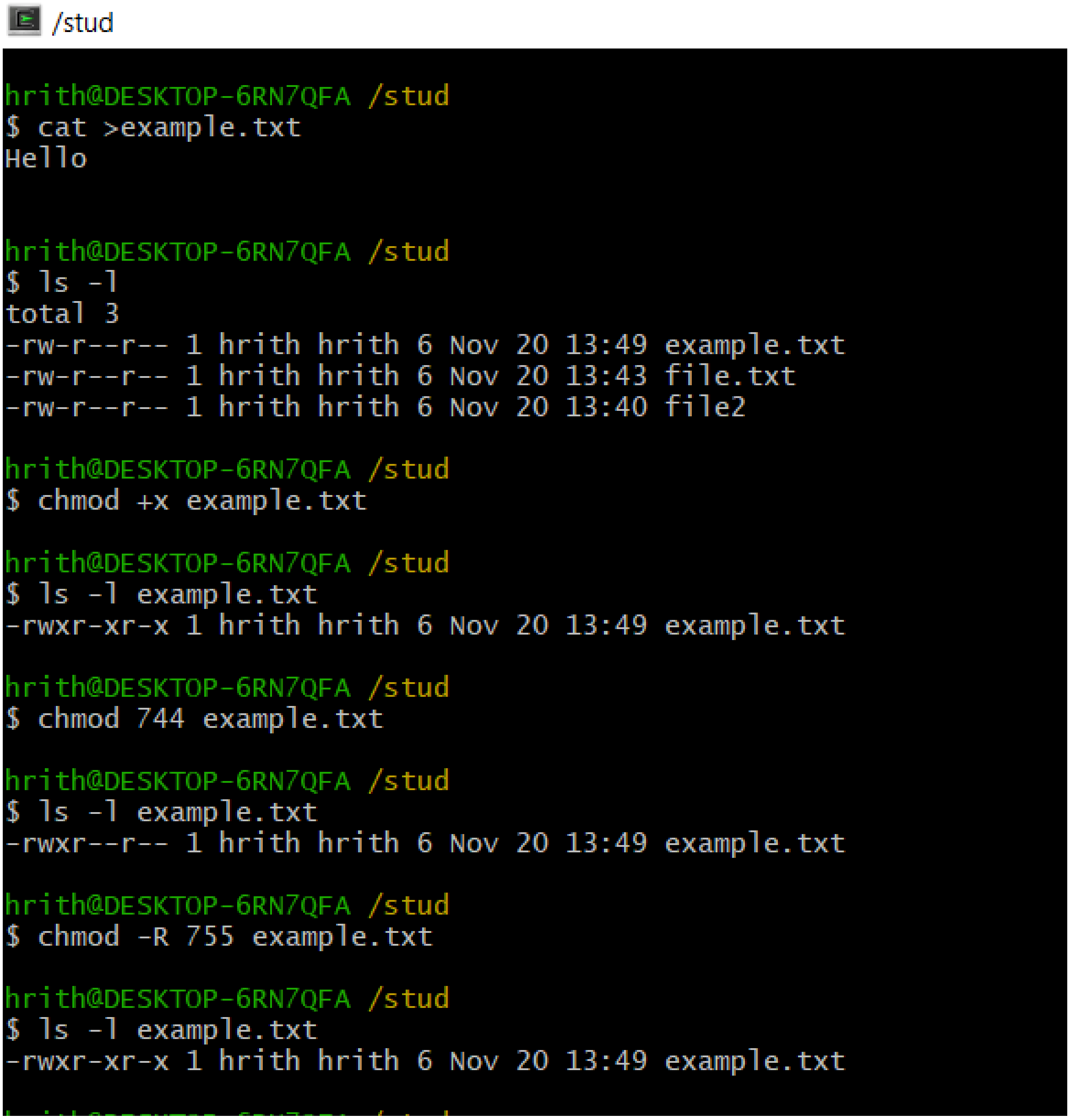
4. top - The top (table of processes) command shows a real-time view of running processes in Linux and displays kernel-managed tasks.



```
hrith@DESKTOP-6RN7QFA /stud
$ top
-bash: top: command not found

hrith@DESKTOP-6RN7QFA /stud
```

Ques. To understand how to examine and change File permissions.

Output –

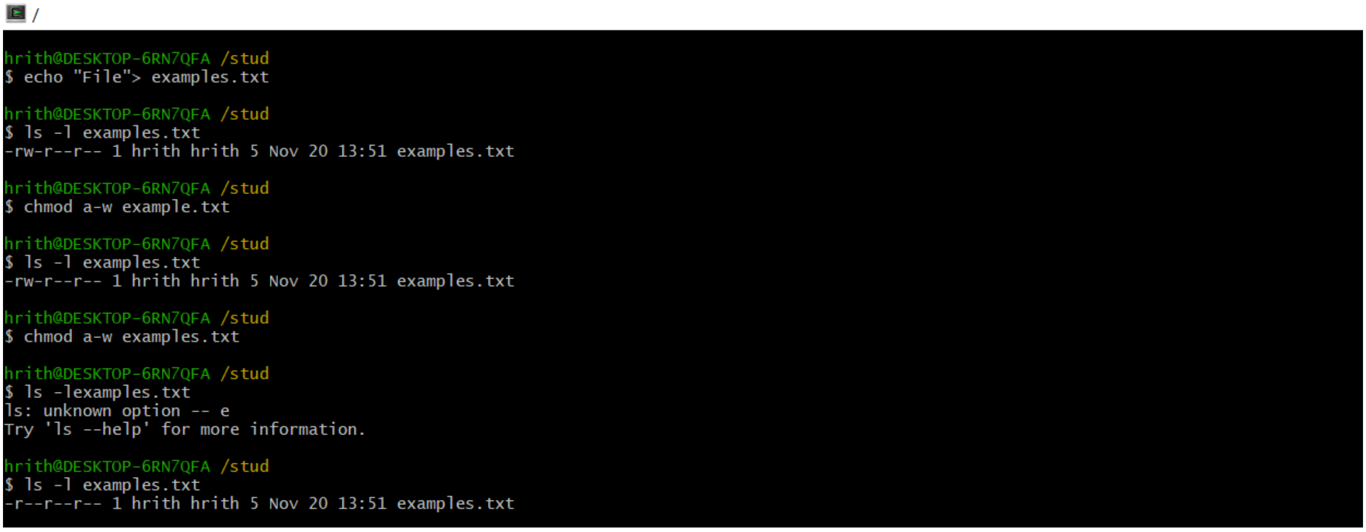


/stud

```
hrith@DESKTOP-6RN7QFA /stud
$ cat >example.txt
Hello


hrith@DESKTOP-6RN7QFA /stud
$ ls -l
total 3
-rw-r--r-- 1 hrith hrith 6 Nov 20 13:49 example.txt
-rw-r--r-- 1 hrith hrith 6 Nov 20 13:43 file.txt
-rw-r--r-- 1 hrith hrith 6 Nov 20 13:40 file2

hrith@DESKTOP-6RN7QFA /stud
$ chmod +x example.txt

hrith@DESKTOP-6RN7QFA /stud
$ ls -l example.txt
-rwxr-xr-x 1 hrith hrith 6 Nov 20 13:49 example.txt

hrith@DESKTOP-6RN7QFA /stud
$ chmod 744 example.txt

hrith@DESKTOP-6RN7QFA /stud
$ ls -l example.txt
-rwxr--r-- 1 hrith hrith 6 Nov 20 13:49 example.txt

hrith@DESKTOP-6RN7QFA /stud
$ chmod -R 755 example.txt

hrith@DESKTOP-6RN7QFA /stud
$ ls -l example.txt
-rwxr-xr-x 1 hrith hrith 6 Nov 20 13:49 example.txt
```

Ques. Set a file to be read-only with the chmod command. Interpret the file CO4 Bachelor of Computer Applications permissions displayed by the ls -l command.

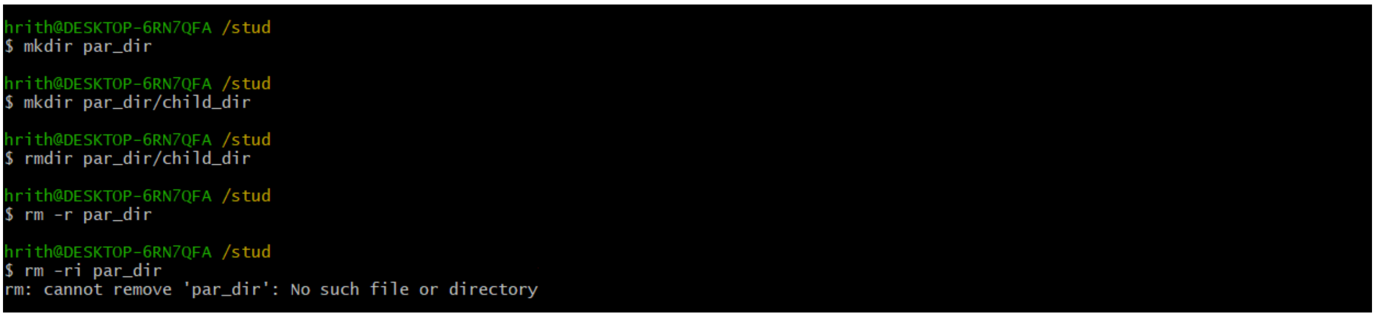
Output -



```
/
hrith@DESKTOP-6RN7QFA /stud
$ echo "File"> examples.txt

hrith@DESKTOP-6RN7QFA /stud
$ ls -l examples.txt
-rw-r--r-- 1 hrith hrith 5 Nov 20 13:51 examples.txt

hrith@DESKTOP-6RN7QFA /stud
$ chmod a-w example.txt

hrith@DESKTOP-6RN7QFA /stud
$ ls -l examples.txt
-rw-r--r-- 1 hrith hrith 5 Nov 20 13:51 examples.txt

hrith@DESKTOP-6RN7QFA /stud
$ chmod a-w examples.txt

hrith@DESKTOP-6RN7QFA /stud
$ ls -lexamples.txt
ls: unknown option -- e
Try 'ls --help' for more information.

hrith@DESKTOP-6RN7QFA /stud
$ ls -l examples.txt
-r--r--r-- 1 hrith hrith 5 Nov 20 13:51 examples.txt
```

Ques. Delete one or more directories with the rmdir command. See what happens if the directory is not empty. Experiment (carefully!) with the rm -r command to delete a directory and its content.

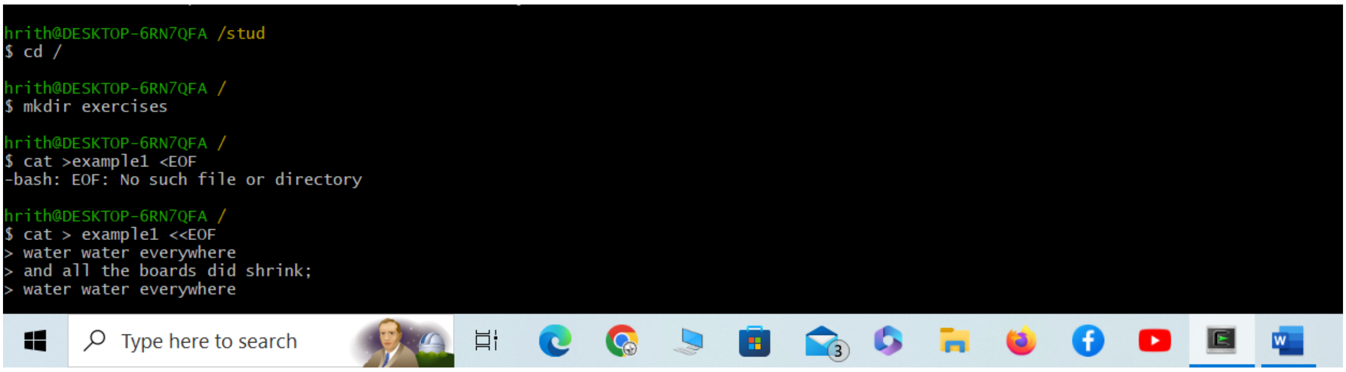
Output –



```
hrith@DESKTOP-6RN7QFA /stud
$ mkdir par_dir

hrith@DESKTOP-6RN7QFA /stud
$ mkdir par_dir/child_dir

hrith@DESKTOP-6RN7QFA /stud
$ rmdir par_dir/child_dir

hrith@DESKTOP-6RN7QFA /stud
$ rm -r par_dir

hrith@DESKTOP-6RN7QFA /stud
$ rm -ri par_dir
rm: cannot remove 'par_dir': No such file or directory
```

Ques. Change your directory to the directory exercises. Create a file in that directory, named the file as example1 using the cat command containing the following text: water, water everywhere and all the boards did shrink; water, water everywhere, no drop to drink.

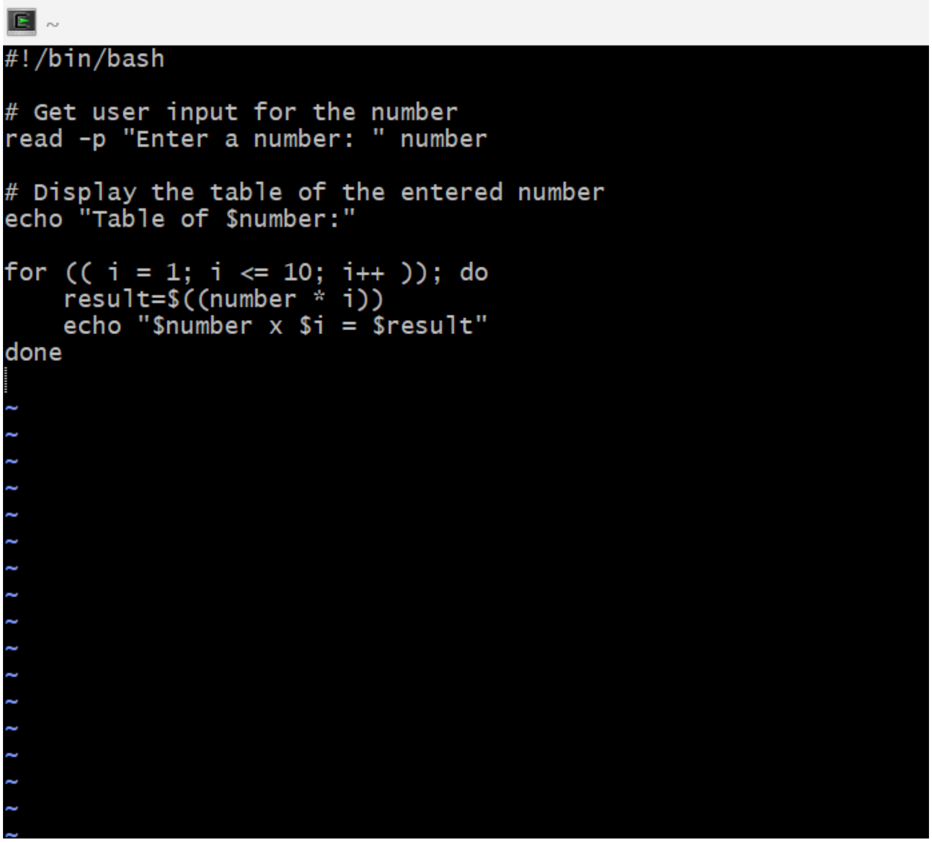
Output –



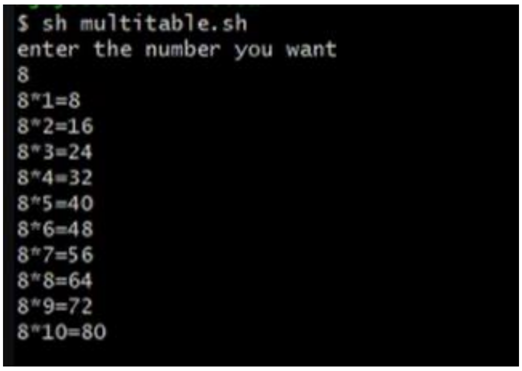
```
hrith@DESKTOP-6RN7QFA /stud
$ cd /

hrith@DESKTOP-6RN7QFA /
$ mkdir exercises

hrith@DESKTOP-6RN7QFA /
$ cat >example1 <EOF
-bash: EOF: No such file or directory

hrith@DESKTOP-6RN7QFA /
$ cat > example1 <<EOF
> water water everywhere
> and all the boards did shrink;
> water water everywhere
```

Ques. Write basic shell script to display the table of a number.

Output –



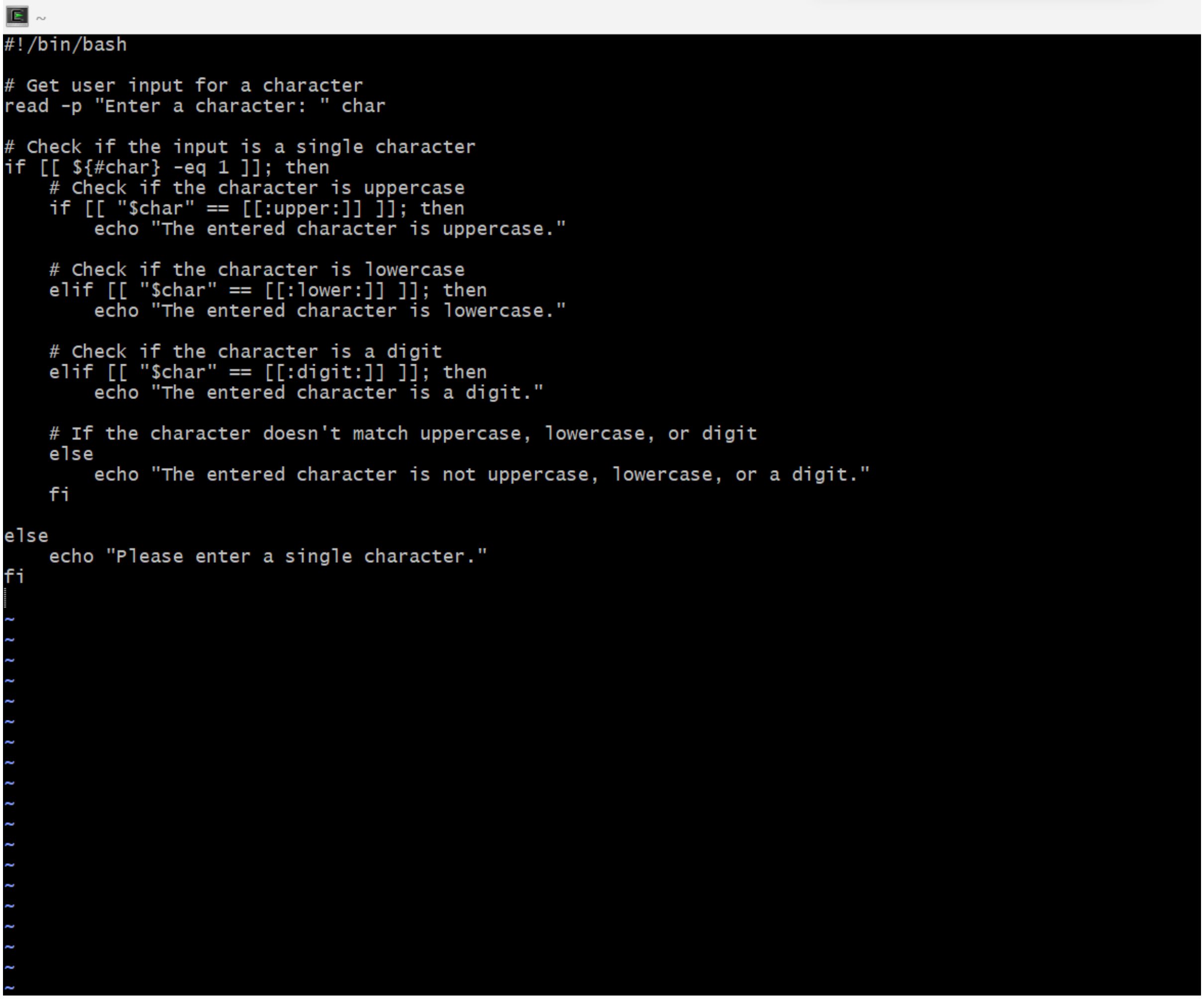
```
#!/bin/bash

# Get user input for the number
read -p "Enter a number: " number

# Display the table of the entered number
echo "Table of $number:"

for (( i = 1; i <= 10; i++ )); do
    result=$((number * i))
    echo "$number x $i = $result"
done

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```



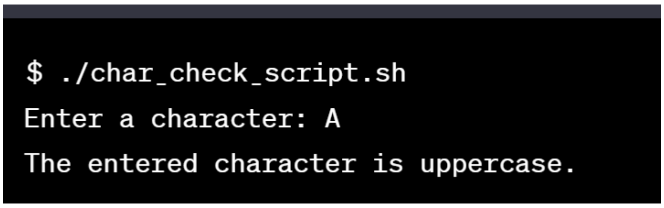
```
$ sh multitable.sh
enter the number you want
8
8*1=8
8*2=16
8*3=24
8*4=32
8*5=40
8*6=48
8*7=56
8*8=64
8*9=72
8*10=80
```

Ques. Write basic shell script to input a character from user and then check whether it is uppercase, lowercase or digit.

Output –

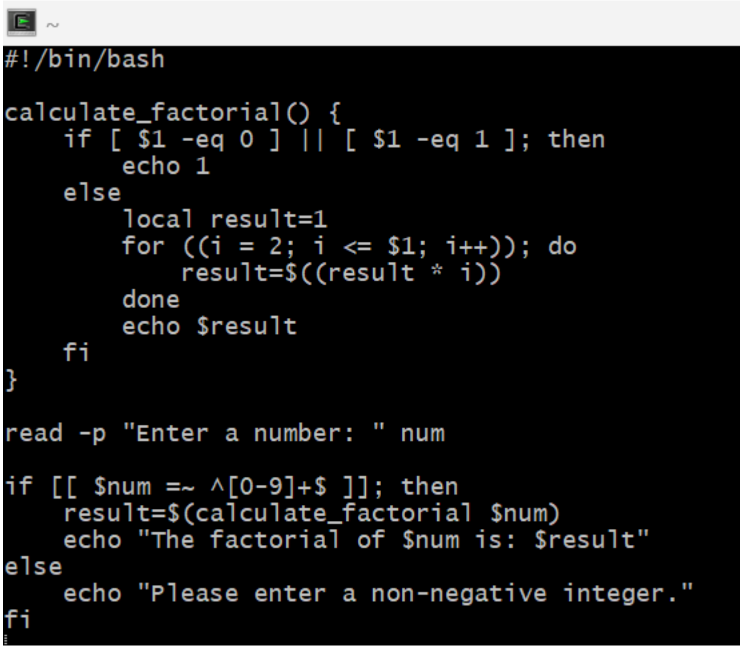


```
#!/bin/bash

# Get user input for a character
read -p "Enter a character: " char

# Check if the input is a single character
if [[ ${#char} -eq 1 ]]; then
    # Check if the character is uppercase
    if [[ "$char" == [[:upper:]] ]]; then
        echo "The entered character is uppercase."

    # Check if the character is lowercase
    elif [[ "$char" == [[:lower:]] ]]; then
        echo "The entered character is lowercase."

    # Check if the character is a digit
    elif [[ "$char" == [[:digit:]] ]]; then
        echo "The entered character is a digit."

    # If the character doesn't match uppercase, lowercase, or digit
    else
        echo "The entered character is not uppercase, lowercase, or a digit."
    fi

else
    echo "Please enter a single character."
fi
```

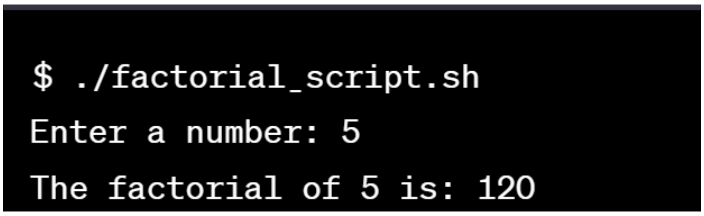


```
$ ./char_check_script.sh
Enter a character: A
The entered character is uppercase.
```

Ques. Write basic shell script to calculate factorial of a number.

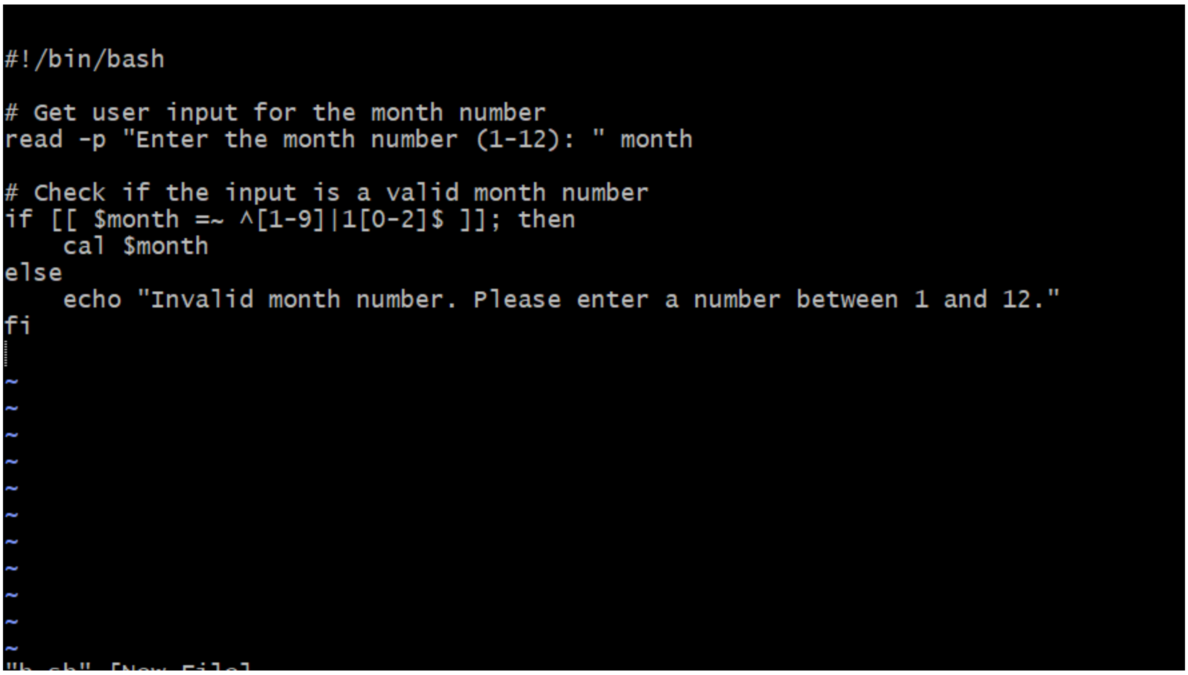


```
#!/bin/bash

calculate_factorial() {
    if [ $1 -eq 0 ] || [ $1 -eq 1 ]; then
        echo 1
    else
        local result=1
        for ((i = 2; i <= $1; i++)); do
            result=$((result * i))
        done
        echo $result
    fi
}

read -p "Enter a number: " num

if [[ $num =~ ^[0-9]+$ ]]; then
    result=$(calculate_factorial $num)
    echo "The factorial of $num is: $result"
else
    echo "Please enter a non-negative integer."
fi
```

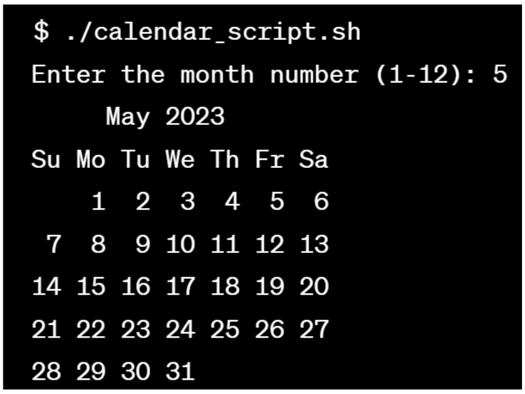


```
$ ./factorial_script.sh
Enter a number: 5
The factorial of 5 is: 120
```

Ques. Write basic shell script to input the month number and generate corresponding calendar.

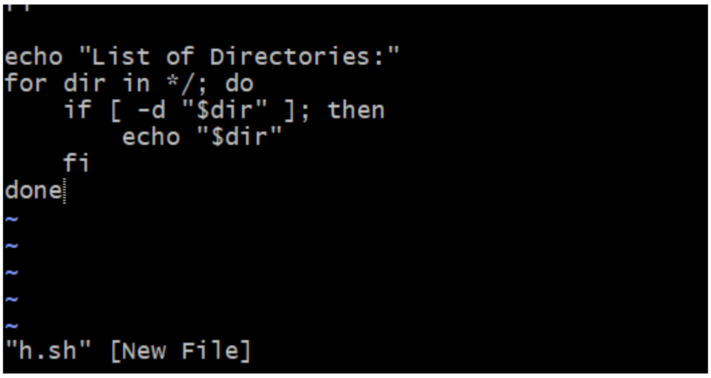


```
#!/bin/bash

# Get user input for the month number
read -p "Enter the month number (1-12): " month

# Check if the input is a valid month number
if [[ $month =~ ^[1-9]|1[0-2]$ ]]; then
    cal $month
else
    echo "Invalid month number. Please enter a number between 1 and 12."
fi

~
~
~
~
~
~
~
~
~
~
~
"b.sh" [New File]
```

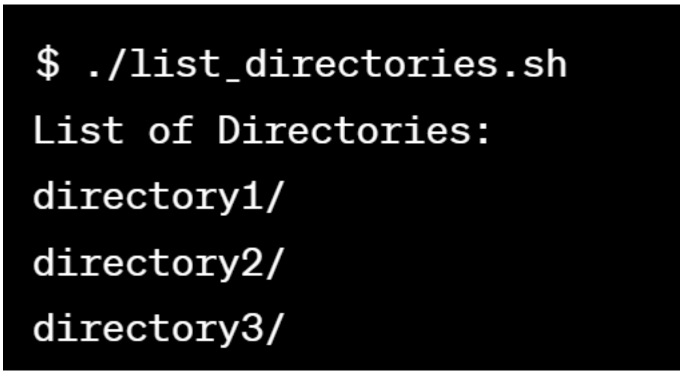


```
$ ./calendar_script.sh
Enter the month number (1-12): 5
     May 2023
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

Ques. Write basic shell script to list all directories.

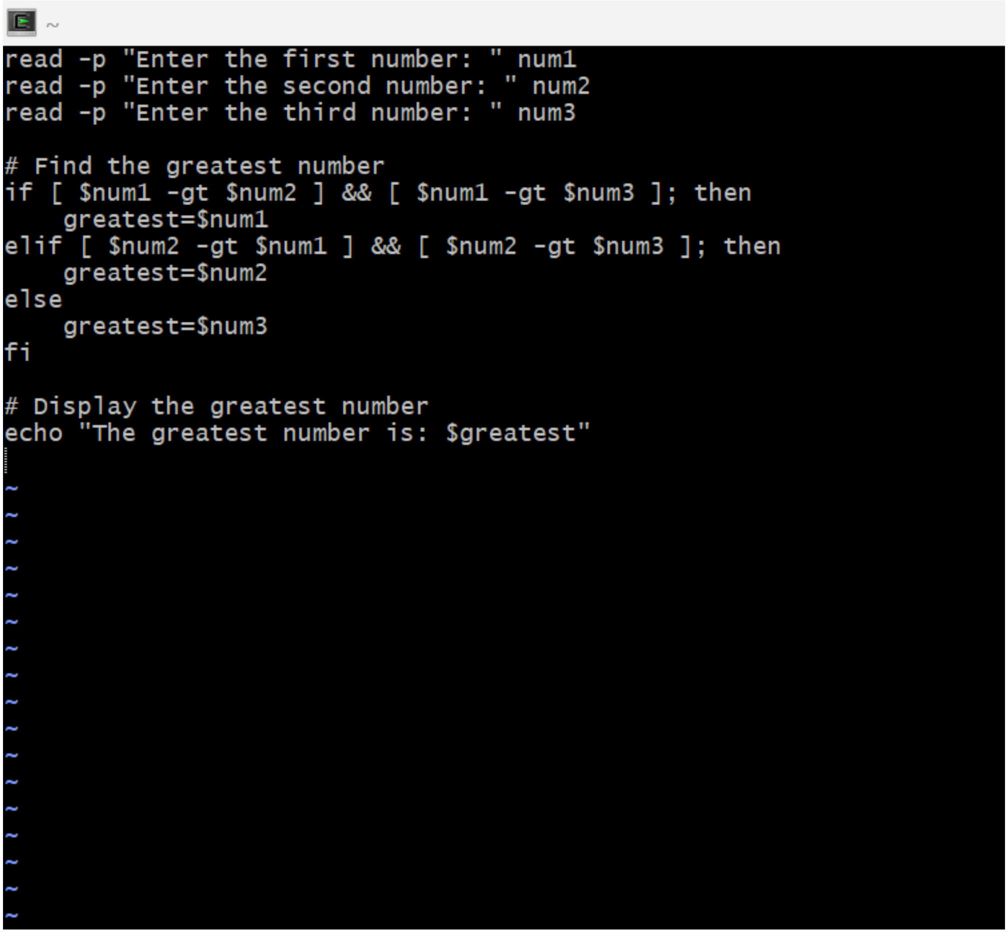


```
echo "List of Directories:"
for dir in */; do
    if [ -d "$dir" ]; then
        echo "$dir"
    fi
done
~
~
~
~
~
"h.sh" [New File]
```

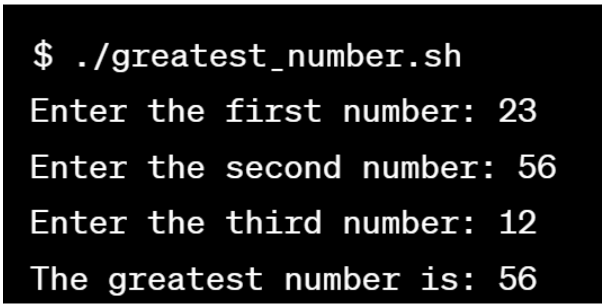


```
$ ./list_directories.sh
List of Directories:
directory1/
directory2/
directory3/
```

Ques. Write basic shell script to display greatest of three numbers.

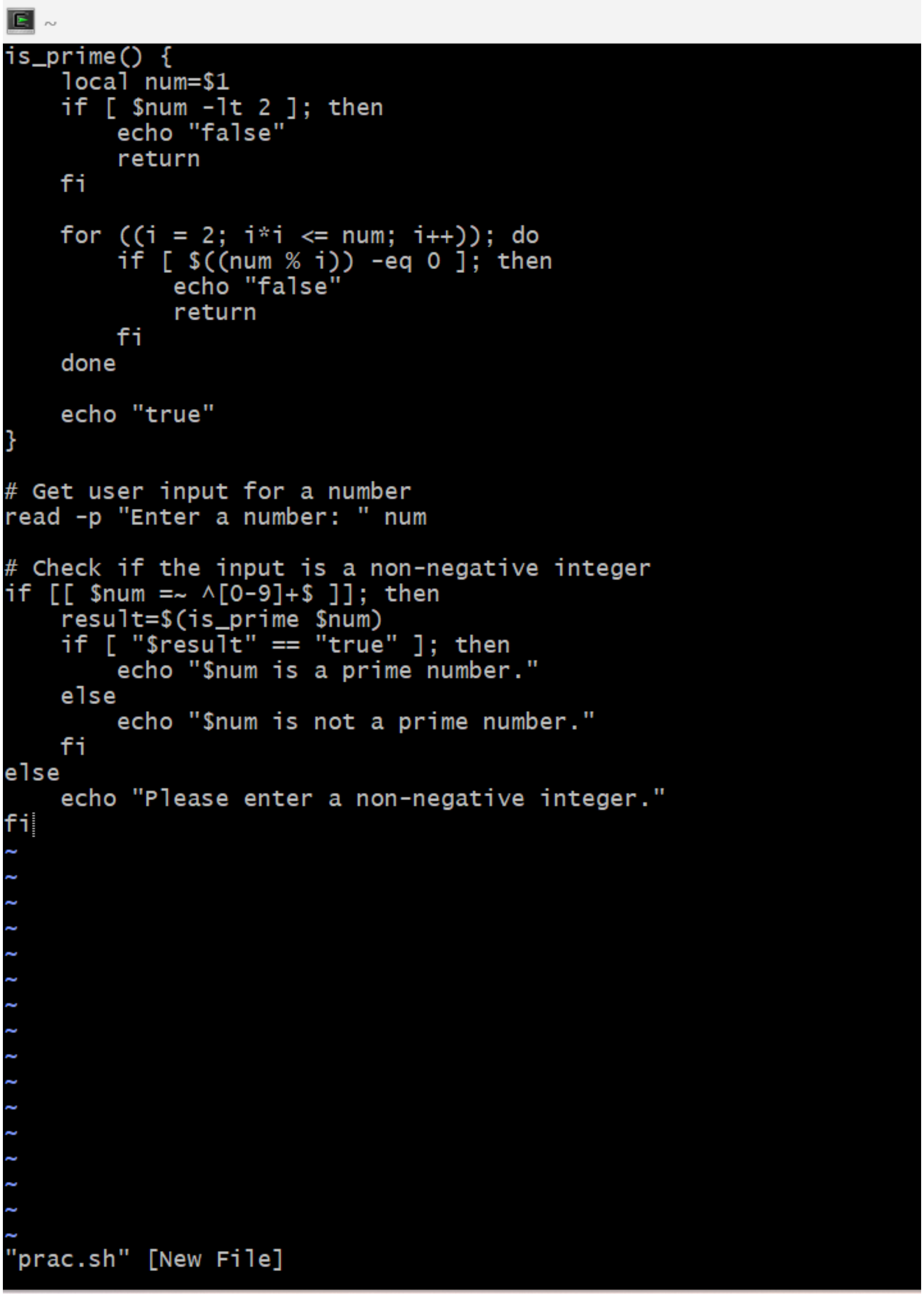


```
read -p "Enter the first number: " num1
read -p "Enter the second number: " num2
read -p "Enter the third number: " num3

# Find the greatest number
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]; then
    greatest=$num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]; then
    greatest=$num2
else
    greatest=$num3
fi

# Display the greatest number
echo "The greatest number is: $greatest"

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

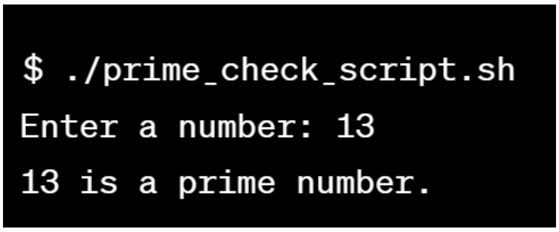


```
$ ./greatest_number.sh
Enter the first number: 23
Enter the second number: 56
Enter the third number: 12
The greatest number is: 56
```

Ques. Write basic shell script to check whether the number entered by user is prime or not.



```
is_prime() {
    local num=$1
    if [ $num -lt 2 ]; then
        echo "false"
        return
    fi

    for ((i = 2; i*i <= num; i++)); do
        if [ $((num % i)) -eq 0 ]; then
            echo "false"
            return
        fi
    done

    echo "true"
}

# Get user input for a number
read -p "Enter a number: " num

# Check if the input is a non-negative integer
if [[ $num =~ ^[0-9]+$ ]]; then
    result=$(is_prime $num)
    if [ "$result" == "true" ]; then
        echo "$num is a prime number."
    else
        echo "$num is not a prime number."
    fi
else
    echo "Please enter a non-negative integer."
fi
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"prac.sh" [New File]
```



```
$ ./prime_check_script.sh
Enter a number: 13
13 is a prime number.
```