BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA, BENGALURU - 560064

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

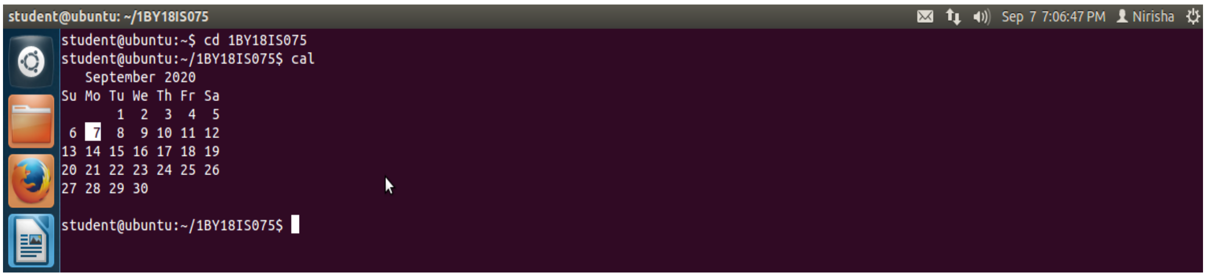
UNIX Programming

ASSIGNMENT- 01

Name	Nirisha B
USN	1BY18IS075
Semester/Section	5th Sem, ‘B’ Sec
Course Code	18CS56
Course Name	UNIX Programming
Faculty	Prof. Mahalakshmi S
Date	08-09-2020

Signature of a Student **Signature of a Faculty**

1. Display the current month(1M)



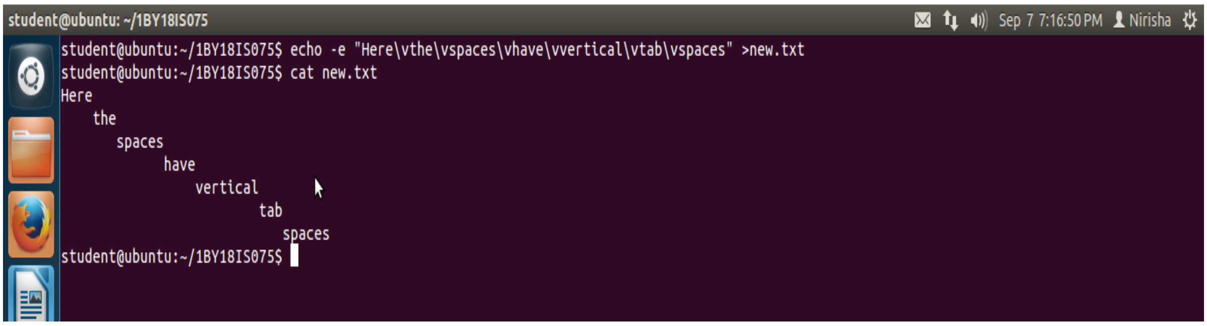
```
student@ubuntu:~$ cd 1BY18IS075
student@ubuntu:~/1BY18IS075$ cal
   September 2020
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30

student@ubuntu:~/1BY18IS075$
```

cal

- This command is used to display the calendar of the current month.
- It also highlights the current date in the calendar.
- **Syntax**: $cal

2. Display the output in the following format using echo command and redirect the output in new.txt file. (2M)

```
Here
        the
                spaces
                        have
                                vertical
                                        tab
                                                spaces.
```



```
student@ubuntu:~/1BY18IS075$ echo -e "Here\vthe\vspaces\vhave\vvertical\vtab\vspaces" >new.txt
student@ubuntu:~/1BY18IS075$ cat new.txt
Here
    the
       spaces
             have
                 vertical
                         tab
                            spaces
student@ubuntu:~/1BY18IS075$
```

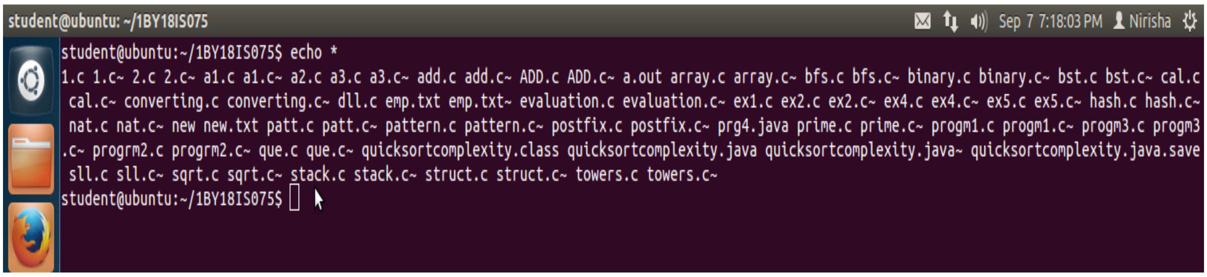
- "\v" is used to obtain vertical tab spaces.
- $cat is the command used to view contents of file.
- $echo is used to print the string present in the inverted comma.
- -e is used to exit immediately if a command exits with a non-zero status. Enables interpretation of backslash escape sequences.

Here the output of the command is stored a file called new.txt. In order to view the file, we use the command "$cat new.txt".

To print the above pattern

Syntax: $echo –e "Here\vthe\vspaces\vhave\vvertical\vtab\vspaces" >new.txt
$cat new.txt

3. Display all files and folders without ls command (Hint: using echo) (1M)



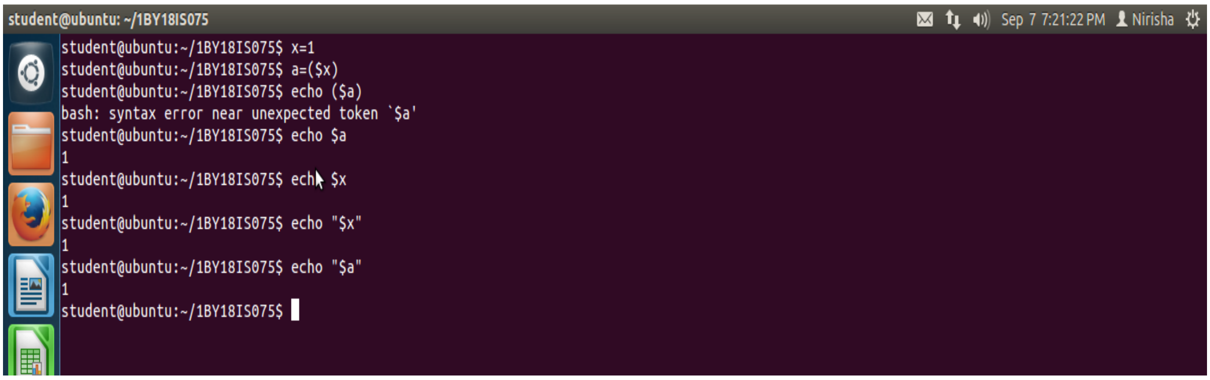
```
student@ubuntu:~/1BY18IS075$ echo *
1.c 1.c~ 2.c 2.c~ a1.c a1.c~ a2.c a3.c a3.c~ add.c add.c~ ADD.c ADD.c~ a.out array.c array.c~ bfs.c bfs.c~ binary.c binary.c~ bst.c bst.c~ cal.c
 cal.c~ converting.c converting.c~ dll.c emp.txt emp.txt~ evaluation.c evaluation.c~ ex1.c ex2.c ex2.c~ ex4.c ex4.c~ ex5.c ex5.c~ hash.c hash.c~
 nat.c nat.c~ new new.txt patt.c patt.c~ pattern.c pattern.c~ postfix.c postfix.c~ prg4.java prime.c prime.c~ progm1.c progm1.c~ progm3.c progm3
.c~ progrm2.c progrm2.c~ que.c que.c~ quicksortcomplexity.class quicksortcomplexity.java quicksortcomplexity.java~ quicksortcomplexity.java.save
 sll.c sll.c~ sqrt.c sqrt.c~ stack.c stack.c~ struct.c struct.c~ towers.c towers.c~
student@ubuntu:~/1BY18IS075$
```

Echo:

- echo command in linux is used to display line of text/string that are passed as an argument.
- echo * displays all the files and folders in the current directory.
- **Syntax:** $echo *

4. Analyse the following code and Why the outputs are different? Debug and show the output . (3 M)

x=1
a=($x)
echo $a #output: 1
echo ($x) #output: -bash: syntax error near unexpected token `$x'



```
student@ubuntu:~/1BY18IS075$ x=1
student@ubuntu:~/1BY18IS075$ a=($x)
student@ubuntu:~/1BY18IS075$ echo ($a)
bash: syntax error near unexpected token `$a'
student@ubuntu:~/1BY18IS075$ echo $a
1
student@ubuntu:~/1BY18IS075$ echo $x
1
student@ubuntu:~/1BY18IS075$ echo "$x"
1
student@ubuntu:~/1BY18IS075$ echo "$a"
1
student@ubuntu:~/1BY18IS075$
```

- In the first line for the variable **x** we are assigning a value 1.
- Next for the another variable **a** we are assigning a value of a variable **x** .
- So, when we echo $a the value stored in variable will display i.e 1.
- When we pass the command $echo ($x), output we get is "bash: syntax error near unexpected token `$x' ".
- This is because ($x) cannot be passed for echo instead '$x' or $x can be passed.
- **Syntax:** $echo $x or
  $echo '$x'

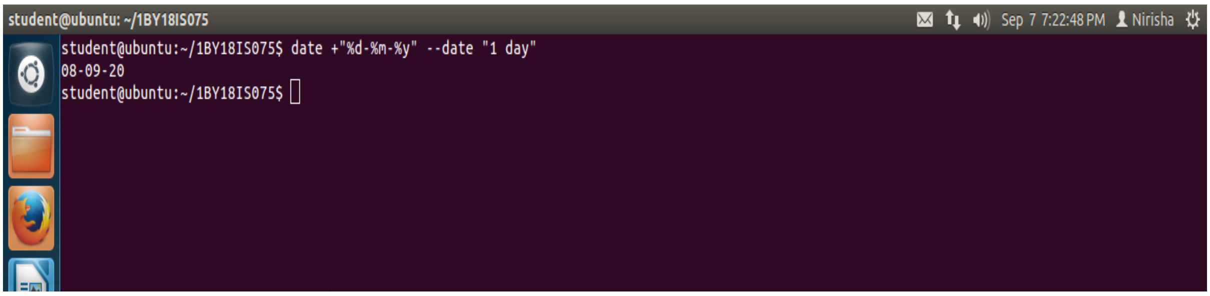
5. Display tomorrow date (1M)

- We can display the date using date command.
- But to display the date other than the current date ,we use the command
  $date - -date "string"

.

Syntax: $date --date “1 day”

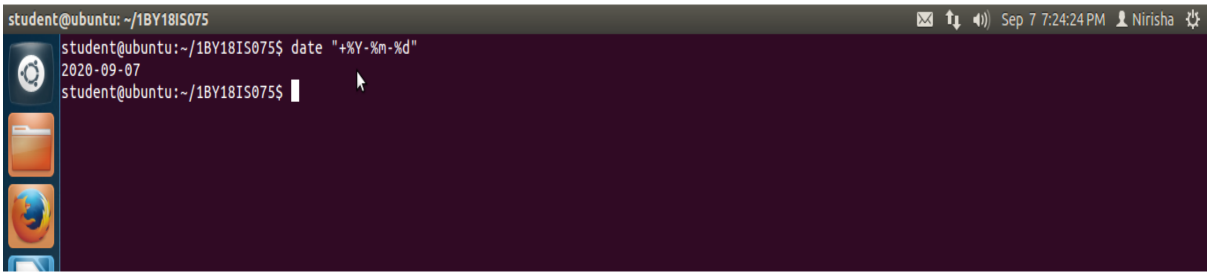
- This command displays the date of tomorrow’s, because the string passed to date is ‘1 day’.
- In order to format the date we use the command.
 $date +”%d-%m-%y” --date “1 day”.



```
student@ubuntu:~/1BY18IS075$ date +"%d-%m-%y" --date "1 day"
08-09-20
student@ubuntu:~/1BY18IS075$ 
```

6. **Show the output for the following commands and explain the use of format specifiers for the same.(3M)**

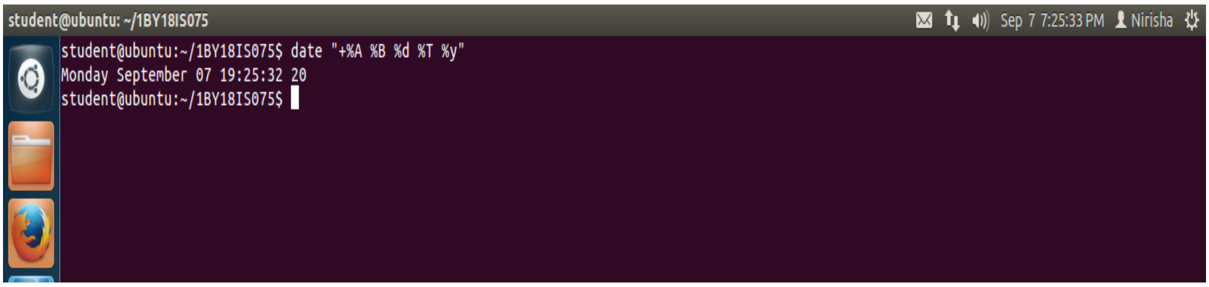
   a. $date "+%Y-%m-%d"



```
student@ubuntu:~/1BY18IS075$ date "+%Y-%m-%d"
2020-09-07
student@ubuntu:~/1BY18IS075$ 
```

%Y - Year as ccyy(4 digits).
%m - Numeric two digit month 00,01,……,12.
%d - Day of the month with two digits leading zeros(01,02,…..,31).

b. $date "+%A %B %d %T %y"

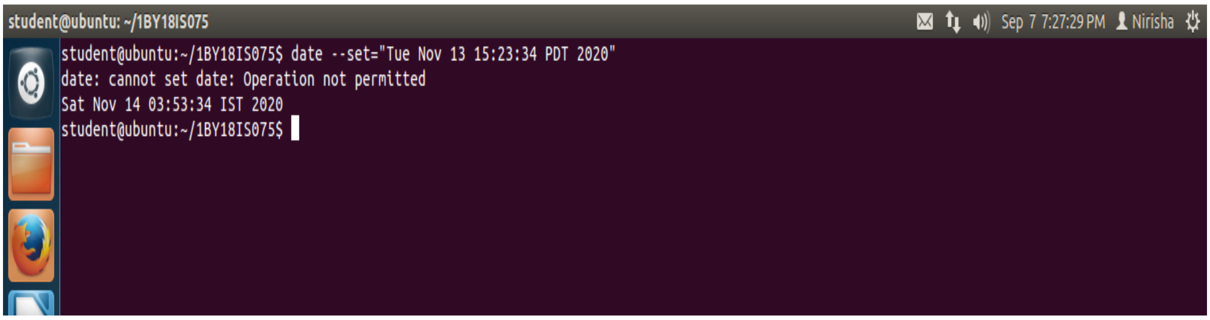


```
student@ubuntu:~/1BY18IS075$ date "+%A %B %d %T %y"
Monday September 07 19:25:32 20
student@ubuntu:~/1BY18IS075$ 
```

%A - Full weekday name such as Monday.
%B - Full month name as September.
%d - Day of the month with two digits leading zeros (01,02,…..,31).

%T - Time in format hour:min:second.
%y - Year as yy(2 digits).

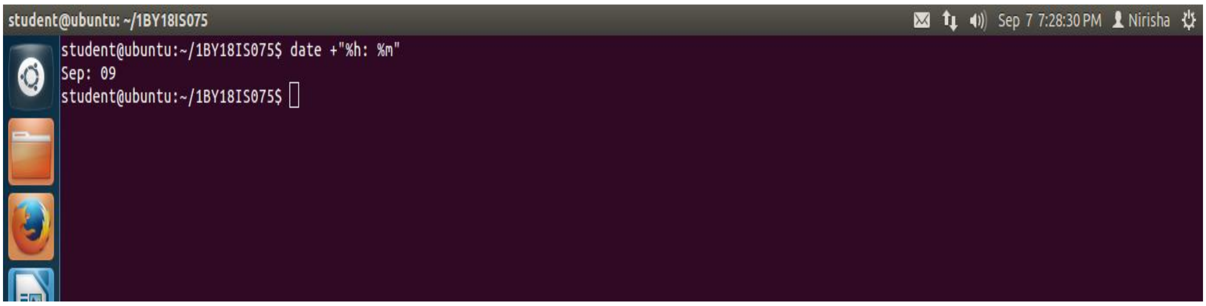
c. $date --set="Tue Nov 13 15:23:34 PDT 2020"



```
student@ubuntu:~/1BY18IS075$ date --set="Tue Nov 13 15:23:34 PDT 2020"
date: cannot set date: Operation not permitted
Sat Nov 14 03:53:34 IST 2020
student@ubuntu:~/1BY18IS075$ 
```

- $date --set is used to set the timing according to the users by specifying it in the inverted comma.
- The date cannot be changed as it is not permitted to do so.
- PDT displays the current time zone.

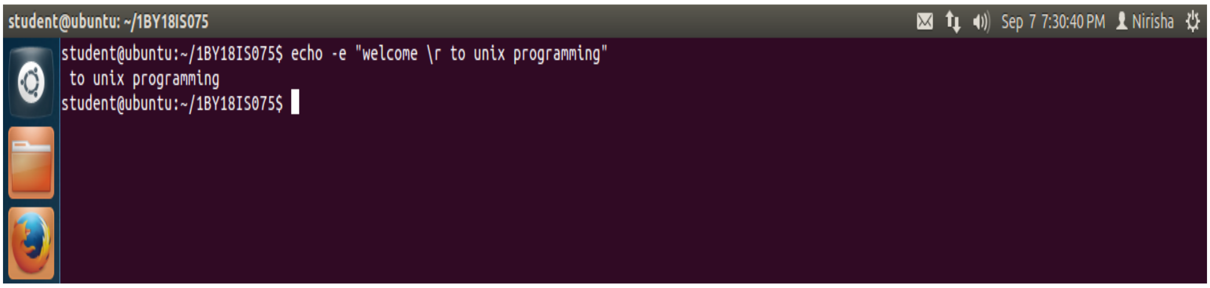
d. $date +”%h: %m”



```
student@ubuntu:~/1BY18IS075$ date +"%h: %m"
Sep: 09
student@ubuntu:~/1BY18IS075$ 
```

%h - Display the name of month as sep.
%m - Numeric two digit month 00,01,……,12.

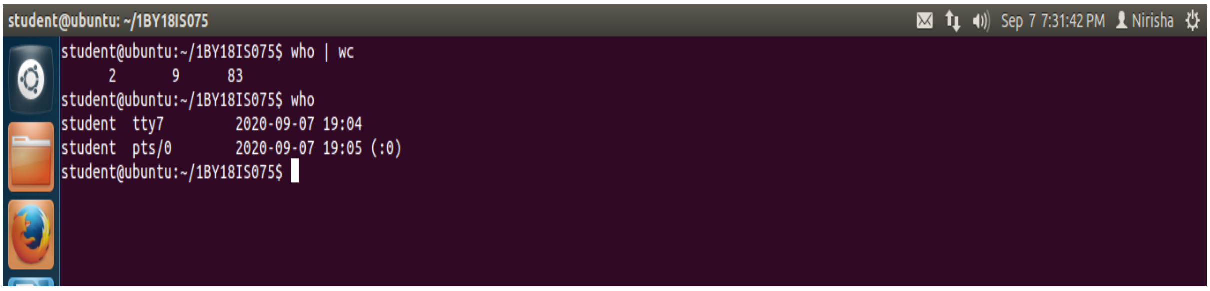
e. echo -e "welcome \r to unix programming”



```
student@ubuntu:~/1BY18IS075$ echo -e "welcome \r to unix programming"
 to unix programming
student@ubuntu:~/1BY18IS075$ 
```

-e - Enable interpretation of backslash escape sequences.
\r - A carriage return used to move the carriage back to the left side of the page.

f. who | wc



```
student@ubuntu:~/1BY18IS075$ who | wc
      2       9      83
student@ubuntu:~/1BY18IS075$ who
student  tty7         2020-09-07 19:04
student  pts/0        2020-09-07 19:05 (:0)
student@ubuntu:~/1BY18IS075$ 
```

| - Pipe.
wc - word count.

wc

- wc is a utility for printing newline, word and byte counts for files.
- It can return the number of lines in a file, the number of characters in a file and the number of words in a file.
- It can also be combine with pipes for general counting operations.

who

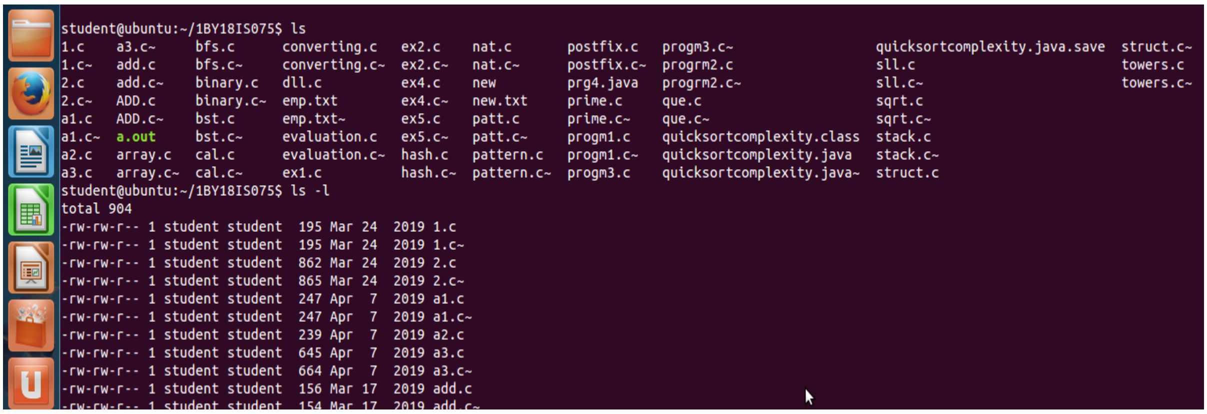
- The who command displays the following information for each user currently logged in to the system if no option is provided :
  Login name of the users
  Terminal line numbers
  Login time of the users in to system
  Remote host name of the user

In this example, 2 represents - there are 2 terminal lines.

9 represents - there are 9 words in the output who.

83 represents – there are 83 characters in the output of who.

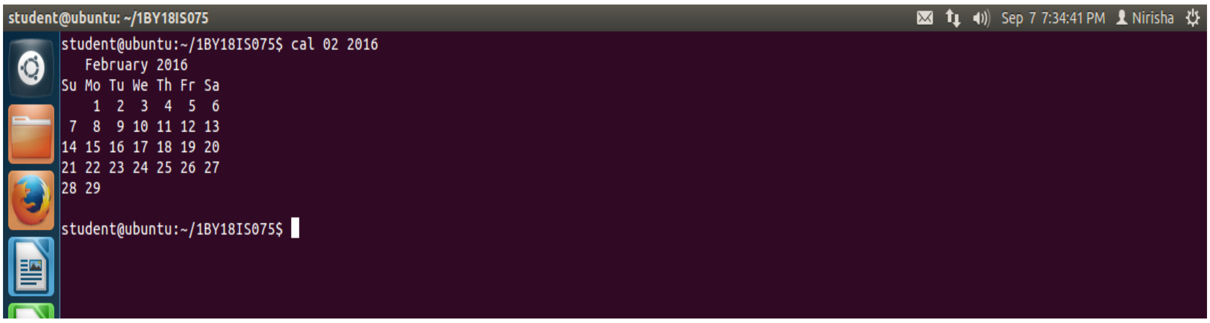
7. Analyse the difference between ls and ls –l (1M)



```
student@ubuntu:~/1BY18IS075$ ls
1.c    a3.c~    bfs.c      converting.c   ex2.c   nat.c       postfix.c   progm3.c~                    quicksortcomplexity.java.save  struct.c~
1.c~   add.c    bfs.c~     converting.c~  ex2.c~  nat.c~      postfix.c~  progrm2.c                    sll.c                          towers.c
2.c    add.c~   binary.c   dll.c          ex4.c   new         prg4.java   progrm2.c~                   sll.c~                         towers.c~
2.c~   ADD.c    binary.c~  emp.txt        ex4.c~  new.txt     prime.c     que.c                        sqrt.c
a1.c   ADD.c~   bst.c      emp.txt~       ex5.c   patt.c      prime.c~    que.c~                       sqrt.c~
a1.c~  a.out    bst.c~     evaluation.c   ex5.c~  patt.c~     progm1.c    quicksortcomplexity.class    stack.c
a2.c   array.c  cal.c      evaluation.c~  hash.c  pattern.c   progm1.c~   quicksortcomplexity.java     stack.c~
a3.c   array.c~ cal.c~     ex1.c          hash.c~ pattern.c~  progm3.c    quicksortcomplexity.java~    struct.c
student@ubuntu:~/1BY18IS075$ ls -l
total 904
-rw-rw-r-- 1 student student  195 Mar 24  2019 1.c
-rw-rw-r-- 1 student student  195 Mar 24  2019 1.c~
-rw-rw-r-- 1 student student  862 Mar 24  2019 2.c
-rw-rw-r-- 1 student student  865 Mar 24  2019 2.c~
-rw-rw-r-- 1 student student  247 Apr  7  2019 a1.c
-rw-rw-r-- 1 student student  247 Apr  7  2019 a1.c~
-rw-rw-r-- 1 student student  239 Apr  7  2019 a2.c
-rw-rw-r-- 1 student student  645 Apr  7  2019 a3.c
-rw-rw-r-- 1 student student  664 Apr  7  2019 a3.c~
-rw-rw-r-- 1 student student  156 Mar 17  2019 add.c
-rw-rw-r-- 1 student student  154 Mar 17  2019 add.c~
```

- $ls is used to display all the files.
- $ls -l will show the entries as a long list along with various attributes like File Permissions, Number of links, Owner, Group, Size, Last modified date and time, File name.

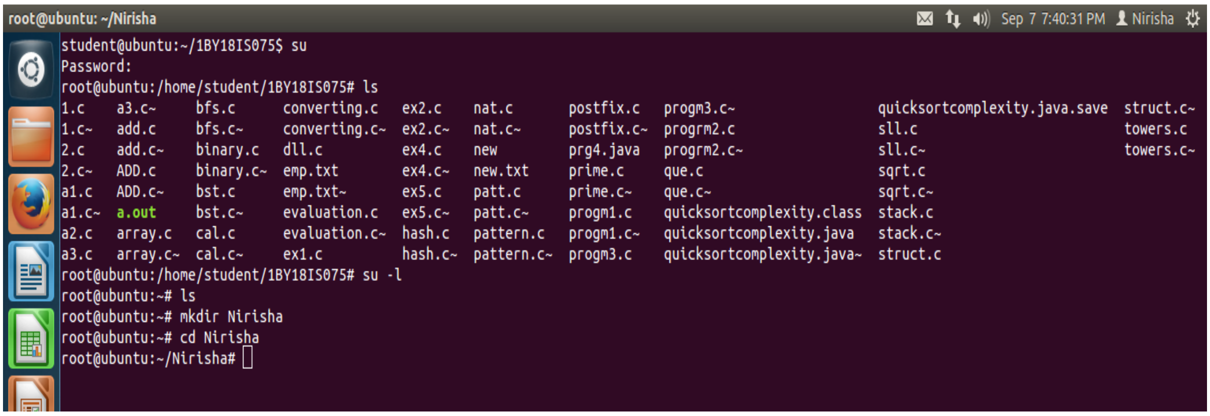
8. Display feb 2016 calendar (1M)



```
student@ubuntu:~/1BY18IS075$ cal 02 2016
   February 2016
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

student@ubuntu:~/1BY18IS075$ 
```

- Calender can be display using **cal** command.
- **Syntax:** $cal options [[month]year]
  Command to display calendar of feb 2016 is $cal 02 2016.

9. Login as root user and change to root's home directory(1M)



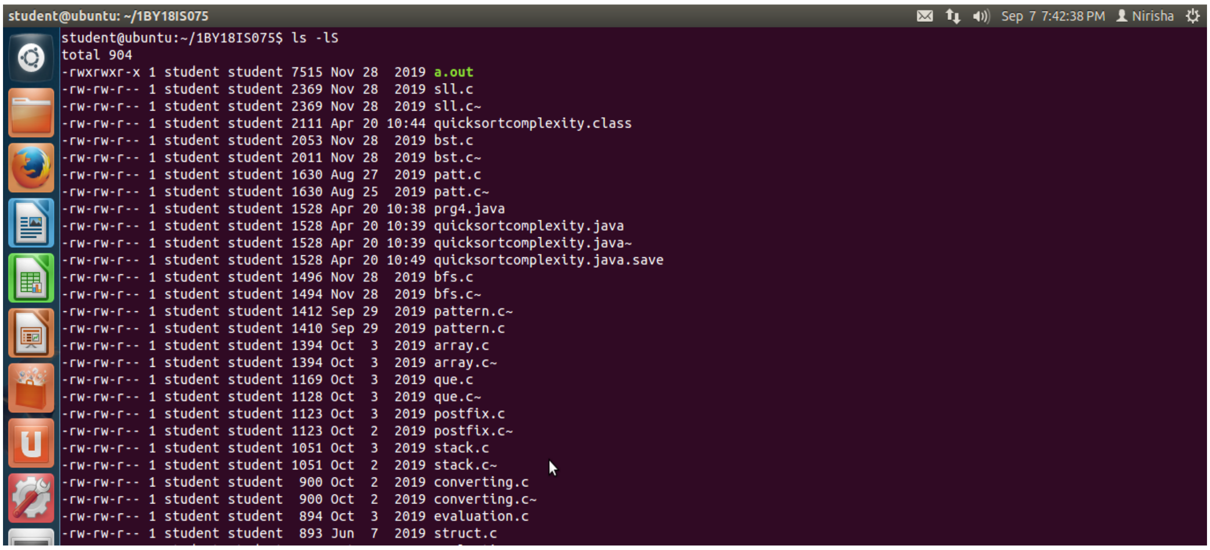
```
student@ubuntu:~/1BY18IS075$ su
Password:
root@ubuntu:/home/student/1BY18IS075# ls
1.c    a3.c~   bfs.c     converting.c   ex2.c   nat.c     postfix.c   progm3.c~                    quicksortcomplexity.java.save  struct.c~
1.c~   add.c   bfs.c~    converting.c~  ex2.c~  nat.c~    postfix.c~  progrm2.c                    sll.c                          towers.c
2.c    add.c~  binary.c  dll.c          ex4.c   new       prg4.java   progrm2.c~                   sll.c~                         towers.c~
2.c~   ADD.c   binary.c~ emp.txt        ex4.c~  new.txt   prime.c     que.c                        sqrt.c
a1.c   ADD.c~  bst.c     emp.txt~       ex5.c   patt.c    prime.c~    que.c~                       sqrt.c~
a1.c~  a.out   bst.c~    evaluation.c   ex5.c~  patt.c~   progm1.c    quicksortcomplexity.class    stack.c
a2.c   array.c cal.c     evaluation.c~  hash.c  pattern.c progm1.c~   quicksortcomplexity.java     stack.c~
a3.c   array.c~ cal.c~   ex1.c          hash.c~ pattern.c~ progm3.c   quicksortcomplexity.java~    struct.c
root@ubuntu:/home/student/1BY18IS075# su -l
root@ubuntu:~# ls
root@ubuntu:~# mkdir Nirisha
root@ubuntu:~# cd Nirisha
root@ubuntu:~/Nirisha# 
```

- To login into root, we use the command $su.
- The root user commands starts with #.
- I have used #ls command to view all the files in the root user of the directory 1BY18IS075.
- #su –l command is used to change to root's home directory.
- Now #ls command displays no files, indicating that there are no files in the home directory.
- To add a directory we can make use of command mikdir and to change the directory wee use cd command.

  Syntax to change to home directory: #su -l

10. List all the files in a particular directory sorted based on file size. (1M)

- To list all the files in particular directory we can use **ls** command.
- To sort a file based on size we use ls –lS.
- **Syntax:** $ls –lS displays all the files in the decending order of the file size.



```
student@ubuntu:~/1BY18IS075$ ls -lS
total 904
-rwxrwxr-x 1 student student 7515 Nov 28  2019 a.out
-rw-rw-r-- 1 student student 2369 Nov 28  2019 sll.c
-rw-rw-r-- 1 student student 2369 Nov 28  2019 sll.c~
-rw-rw-r-- 1 student student 2111 Apr 20 10:44 quicksortcomplexity.class
-rw-rw-r-- 1 student student 2053 Nov 28  2019 bst.c
-rw-rw-r-- 1 student student 2011 Nov 28  2019 bst.c~
-rw-rw-r-- 1 student student 1630 Aug 27  2019 patt.c
-rw-rw-r-- 1 student student 1630 Aug 25  2019 patt.c~
-rw-rw-r-- 1 student student 1528 Apr 20 10:38 prg4.java
-rw-rw-r-- 1 student student 1528 Apr 20 10:39 quicksortcomplexity.java
-rw-rw-r-- 1 student student 1528 Apr 20 10:39 quicksortcomplexity.java~
-rw-rw-r-- 1 student student 1528 Apr 20 10:49 quicksortcomplexity.java.save
-rw-rw-r-- 1 student student 1496 Nov 28  2019 bfs.c
-rw-rw-r-- 1 student student 1494 Nov 28  2019 bfs.c~
-rw-rw-r-- 1 student student 1412 Sep 29  2019 pattern.c~
-rw-rw-r-- 1 student student 1410 Sep 29  2019 pattern.c
-rw-rw-r-- 1 student student 1394 Oct  3  2019 array.c
-rw-rw-r-- 1 student student 1394 Oct  3  2019 array.c~
-rw-rw-r-- 1 student student 1169 Oct  3  2019 que.c
-rw-rw-r-- 1 student student 1128 Oct  3  2019 que.c~
-rw-rw-r-- 1 student student 1123 Oct  3  2019 postfix.c
-rw-rw-r-- 1 student student 1123 Oct  2  2019 postfix.c~
-rw-rw-r-- 1 student student 1051 Oct  3  2019 stack.c
-rw-rw-r-- 1 student student 1051 Oct  2  2019 stack.c~
-rw-rw-r-- 1 student student  900 Oct  2  2019 converting.c
-rw-rw-r-- 1 student student  900 Oct  2  2019 converting.c~
-rw-rw-r-- 1 student student  894 Oct  3  2019 evaluation.c
-rw-rw-r-- 1 student student  893 Jun  7  2019 struct.c
```

Note:

- For the execution of above all the command I have used Ubuntu, which is a Linux-based operating system.
- Shell worked on is bash shell ($).