

# ITW1 Mid-Semester Assignment

Name: Tarun Arora

Roll No.: 20075092

Branch: CSE

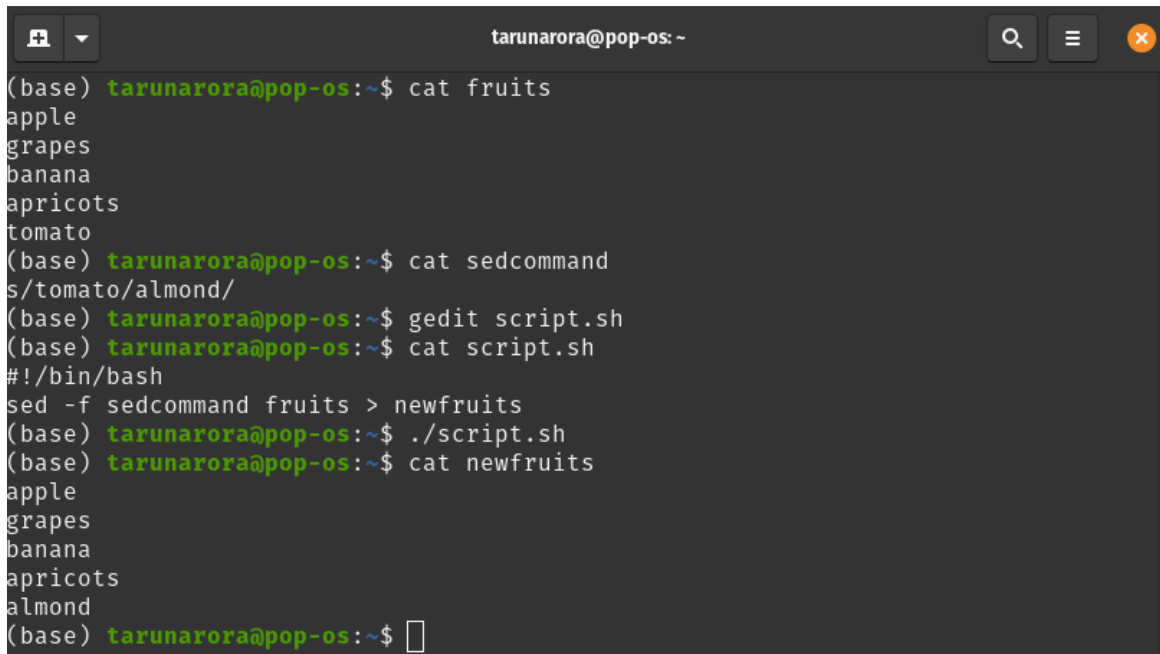
Date: May 17, 2021

**Q1.** Write a shell script using "sed" command to take some command from a file apply that command on content of other file and display output in third file?

**Solution:**

**Code: -**

```
#!/bin/bash  
  
sed -f {file_with_sedcommand} input_file > output_file
```



```
(base) tarunarora@pop-os:~$ cat fruits  
apple  
grapes  
banana  
apricots  
tomato  
(base) tarunarora@pop-os:~$ cat sedcommand  
s/tomato/almond/  
(base) tarunarora@pop-os:~$ gedit script.sh  
(base) tarunarora@pop-os:~$ cat script.sh  
#!/bin/bash  
sed -f sedcommand fruits > newfruits  
(base) tarunarora@pop-os:~$ ./script.sh  
(base) tarunarora@pop-os:~$ cat newfruits  
apple  
grapes  
banana  
apricots  
almond  
(base) tarunarora@pop-os:~$
```

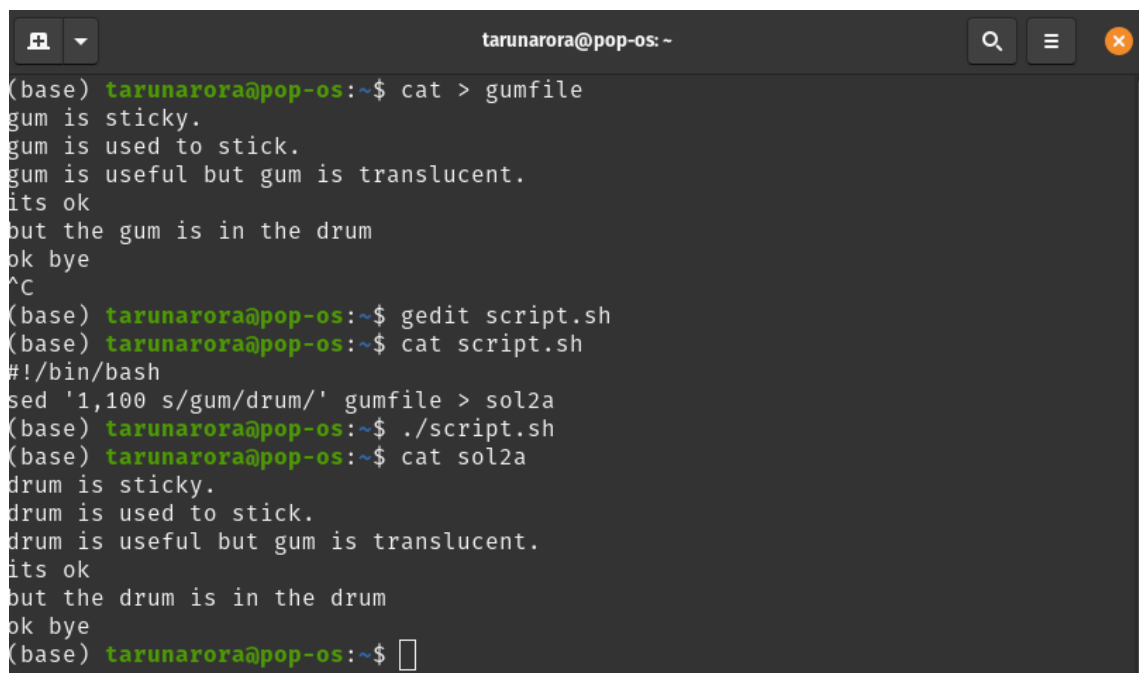
\*\*\*\*\*

- Q2.** Write a shell script using "sed" command to
- (a) replace the word "gum" with "drum" in the first 100 lines of a file,
  - (b) switch the two consecutive words "apple" and "mango" in a file
  - (c) remove the first number on line 5 in file --- and save them in new files.

**Solution:**

**(a) Code: -**

```
#!/bin/bash  
sed '1,100 s/gum/drum/' {input_file} > {output_file}
```



The screenshot shows a terminal window with the following commands and output:

```
(base) tarunarora@pop-os:~$ cat > gumfile  
gum is sticky.  
gum is used to stick.  
gum is useful but gum is translucent.  
its ok  
but the gum is in the drum  
ok bye  
^C  
(base) tarunarora@pop-os:~$ gedit script.sh  
(base) tarunarora@pop-os:~$ cat script.sh  
#!/bin/bash  
sed '1,100 s/gum/drum/' gumfile > sol2a  
(base) tarunarora@pop-os:~$ ./script.sh  
(base) tarunarora@pop-os:~$ cat sol2a  
drum is sticky.  
drum is used to stick.  
drum is useful but gum is translucent.  
its ok  
but the drum is in the drum  
ok bye  
(base) tarunarora@pop-os:~$
```

**(b) Code: -**

```
#!/bin/bash  
sed 's/\(mango\|apple\) \(\(mango\|apple\) \)/\2 \1/' ques2b > newfile
```

```
tarunarora@pop-os: ~
tarunarora@pop-os: ~
tarunarora@pop-os: ~$ cat > ques2b
mango apple
apple mango
apple apple
mango mango
mango & apple
apple & mango
^C
(base) tarunarora@pop-os:~$ gedit script.sh
(base) tarunarora@pop-os:~$ cat script.sh
#!/bin/bash
sed 's/\(mango\|apple\) \(\(mango\|apple\)\)/\2 \1/g' ques2b > newfile
(base) tarunarora@pop-os:~$ ./script.sh
(base) tarunarora@pop-os:~$ cat newfile
apple mango
mango apple
apple apple
mango mango
mango & apple
apple & mango
(base) tarunarora@pop-os:~$
```

(c) Code: -

```
#!/bin/bash

sed '5 s/\([0-9][0-9]*\)//1' ques2c > newfile
```

```
tarunarora@pop-os: ~
tarunarora@pop-os: ~
tarunarora@pop-os: ~$ cat ques2c
i am a file
you can put anything into me
not everything
alright
on 19 this assignment was submitted
ok bye
(base) tarunarora@pop-os:~$ gedit script.sh
(base) tarunarora@pop-os:~$ cat script.sh
#!/bin/bash
sed '5 s/\([0-9][0-9]*\)//1' ques2c > newfile
(base) tarunarora@pop-os:~$ ./script.sh
(base) tarunarora@pop-os:~$ cat newfile
i am a file
you can put anything into me
not everything
alright
on this assignment was submitted
ok bye
(base) tarunarora@pop-os:~$
```

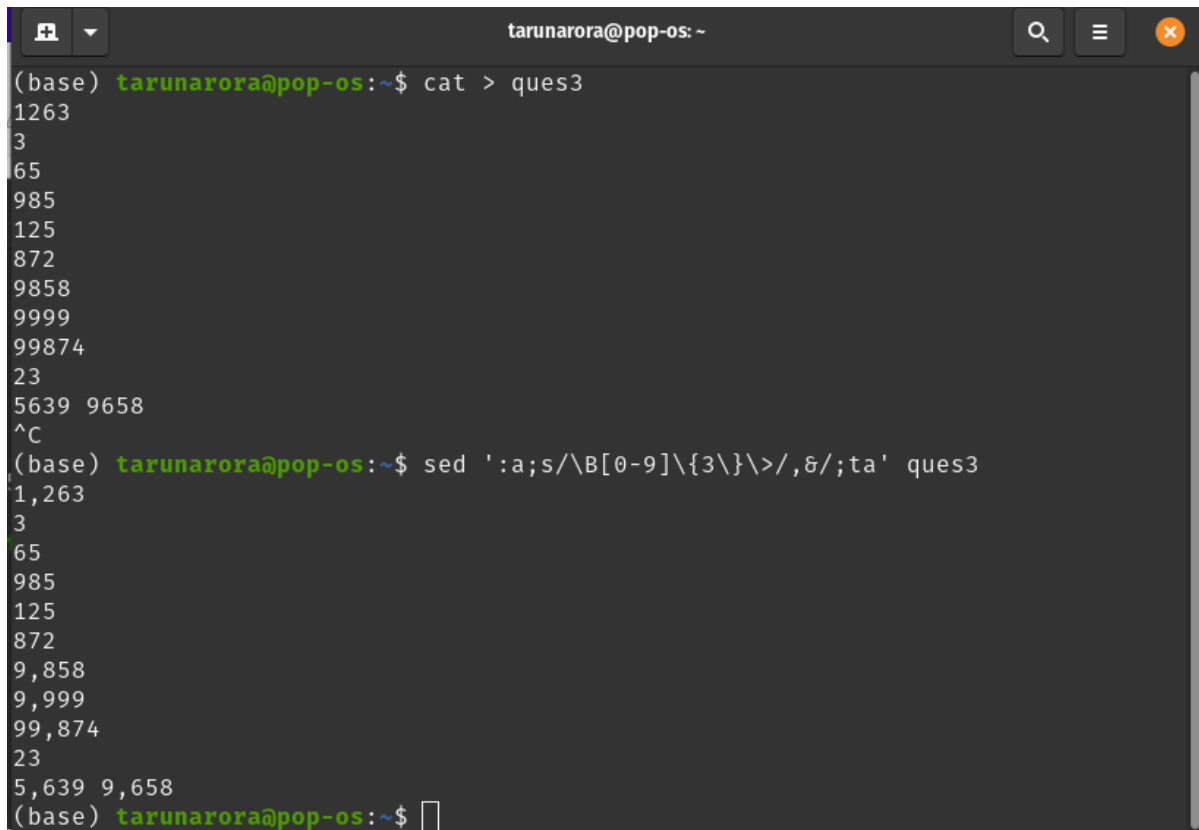
\*\*\*\*\*

**Q3.** Write a sed command to commify the numbers till thousands from a file Number.txt ? The file have numbers in multiple lines.

**Solution:** -

Code: -

```
sed ':a;s/\B[0-9]\{3\}\>/,/&;ta' {filename}
```



```
(base) tarunarora@pop-os:~$ cat > ques3
1263
3
65
985
125
872
9858
9999
99874
23
5639 9658
^C
(base) tarunarora@pop-os:~$ sed ':a;s/\B[0-9]\{3\}\>/,/&;ta' ques3
1,263
3
65
985
125
872
9,858
9,999
99,874
23
5,639 9,658
(base) tarunarora@pop-os:~$
```

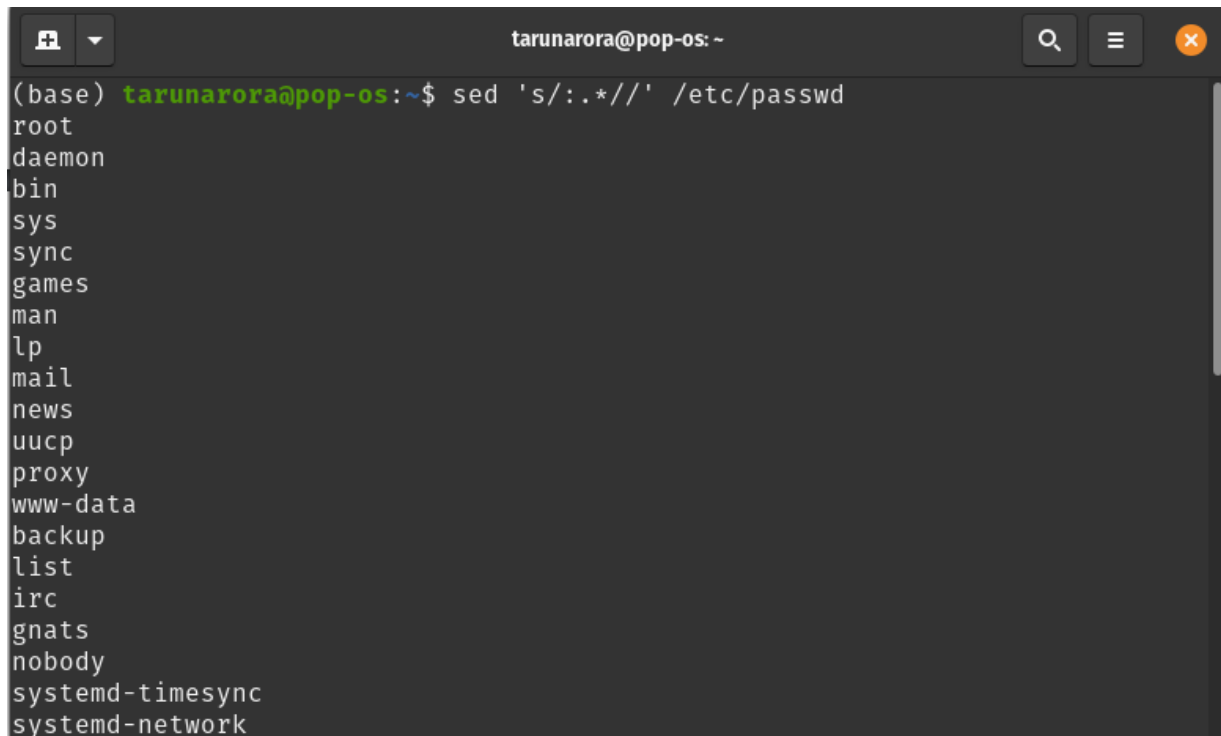
\*\*\*\*\*

**Q4.** Write a sed command to display only the first field from the /etc/passwd file.

**Solution:** -

Code: -

```
sed 's/:.*//' /etc/passwd
```

A terminal window titled 'tarunarora@pop-os: ~' with search, menu, and close buttons in the title bar. The prompt is '(base) tarunarora@pop-os:~\$'. The command 'sed 's/:.\*//' /etc/passwd' has been entered and executed. The output lists the usernames from the /etc/passwd file: root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, systemd-timesync, and systemd-network.

```
(base) tarunarora@pop-os:~$ sed 's/:.*//' /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-timesync
systemd-network
```

\*\*\*\*\*

**Q5.** Write a sed command to delete the first line, last line and all the blank lines from input file. Also write sed command to write first and last line from an input.txt file to the output.txt file respectively.

**Solution:** -

Code: -

'To delete first, last and blank lines'

```
sed '1d;$d;/^$/d' ques5
```



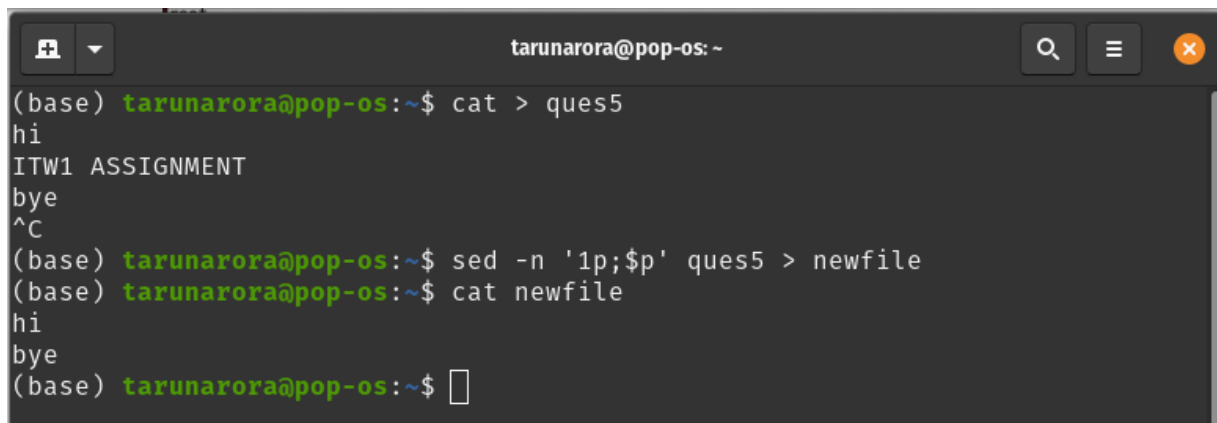
```
(base) tarunarora@pop-os: ~$ cat ques5
hi

This is
an ITW1
assignment

bye
(base) tarunarora@pop-os: ~$ sed '1d;$d;/^$/d' ques5
This is
an ITW1
assignment
(base) tarunarora@pop-os: ~$
```

'To print first and last lines of a file'

```
sed -n '1p;$p' {input_file} > {output_file}
```



```
(base) tarunarora@pop-os: ~$ cat > ques5
hi
ITW1 ASSIGNMENT
bye
^C
(base) tarunarora@pop-os: ~$ sed -n '1p;$p' ques5 > newfile
(base) tarunarora@pop-os: ~$ cat newfile
hi
bye
(base) tarunarora@pop-os: ~$
```

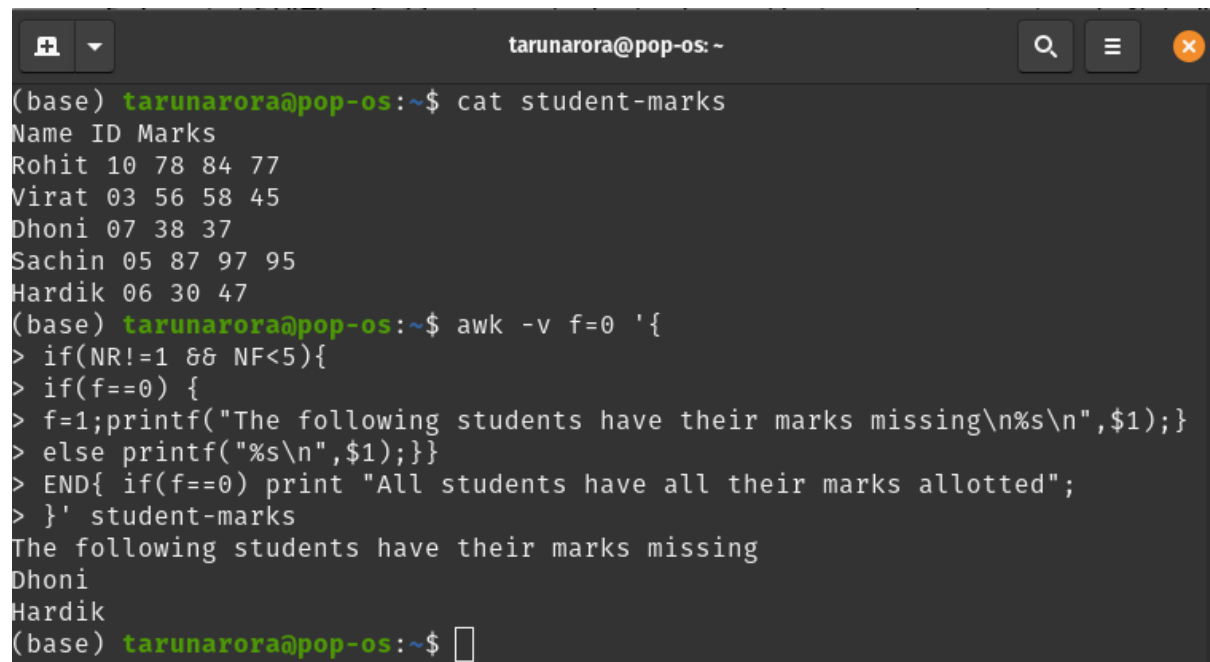
\*\*\*\*\*

**Q6.** Write awk command to check if all marks are there in student-marks file. Display the names of students whose marks are missing.

**Solution: -**

Code: -

```
awk -v f=0 '{
if(NR!=1 && NF<5){
if(f==0) {
f=1;printf("The following students have their marks missing \n%s\n",$1);}
else printf("%s\n",$1);}}
END{ if(f==0) print "All students have all their marks allotted";
}' student-marks
```



```
(base) tarunarora@pop-os:~$ cat student-marks
Name ID Marks
Rohit 10 78 84 77
Virat 03 56 58 45
Dhoni 07 38 37
Sachin 05 87 97 95
Hardik 06 30 47
(base) tarunarora@pop-os:~$ awk -v f=0 '{
> if(NR!=1 && NF<5){
> if(f==0) {
> f=1;printf("The following students have their marks missing\n%s\n",$1);}
> else printf("%s\n",$1);}}
> END{ if(f==0) print "All students have all their marks allotted";
> }' student-marks
The following students have their marks missing
Dhoni
Hardik
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

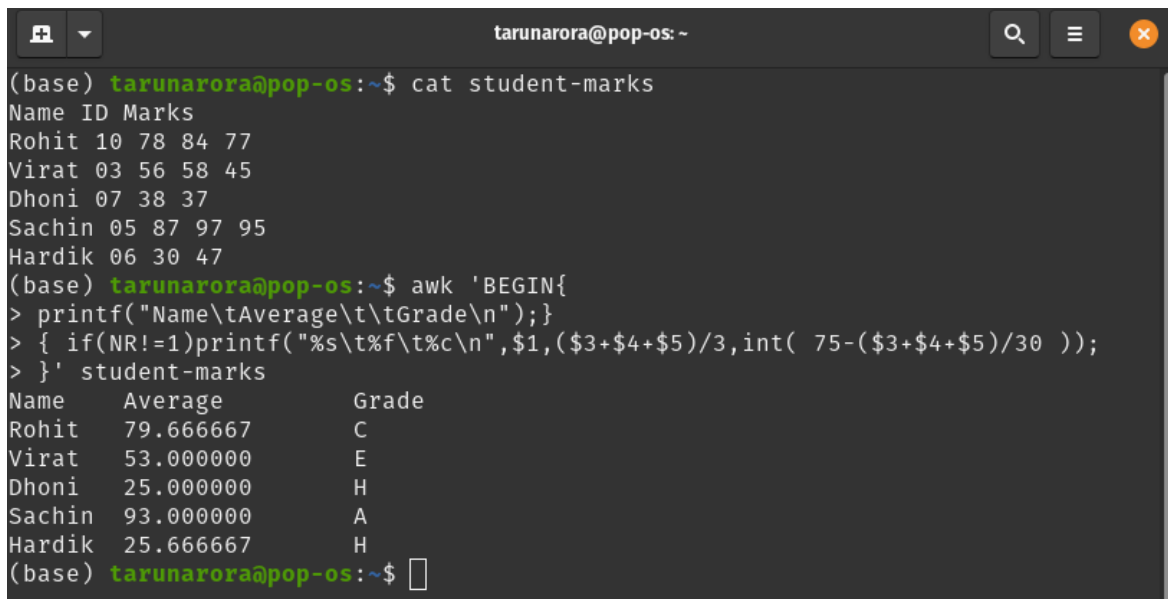
**Q7.** Using awk, find the average marks of students, and grades from student-marks file.

**Solution:** -

Code: -

```
awk 'BEGIN{
printf("Name\tAverage\t\tGrade\n");}

{ if(NR!=1)printf("%s\t%f\t%c\n",$1,($3+$4+$5)/3,int( 75-($3+$4+$5)/30 ));
}' student-marks
```



The terminal window shows the following commands and output:

```
(base) tarunarora@pop-os:~$ cat student-marks
Name ID Marks
Rohit 10 78 84 77
Virat 03 56 58 45
Dhoni 07 38 37
Sachin 05 87 97 95
Hardik 06 30 47
(base) tarunarora@pop-os:~$ awk 'BEGIN{
> printf("Name\tAverage\t\tGrade\n");}
> { if(NR!=1)printf("%s\t%f\t%c\n",$1,($3+$4+$5)/3,int( 75-($3+$4+$5)/30 ));
> }' student-marks
Name      Average      Grade
Rohit     79.666667     C
Virat     53.000000     E
Dhoni     25.000000     H
Sachin    93.000000     A
Hardik    25.666667     H
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*



**Q8.** Write an awk command to concatenate every 3 lines of input file with a comma.

**Solution:** -

Code: -

```
awk '{
    if(NR%3==0) printf("%s\n", $0);
    else printf("%s,", $0);
}' ques8
```

A terminal window titled 'tarunarora@pop-os: ~' with search, menu, and close icons in the top right. The terminal shows the following sequence of commands and output:  
(base) tarunarora@pop-os:~\$ cat ques8  
line1  
line2  
line3  
line4  
line5  
line6  
line7  
line8  
line9  
(base) tarunarora@pop-os:~\$ awk '{  
> if(NR%3==0) printf("%s\n", \$0);  
> else printf("%s,", \$0);  
> }' ques8  
line1,line2,line3  
line4,line5,line6  
line7,line8,line9  
(base) tarunarora@pop-os:~\$   
The prompt is followed by a cursor icon.

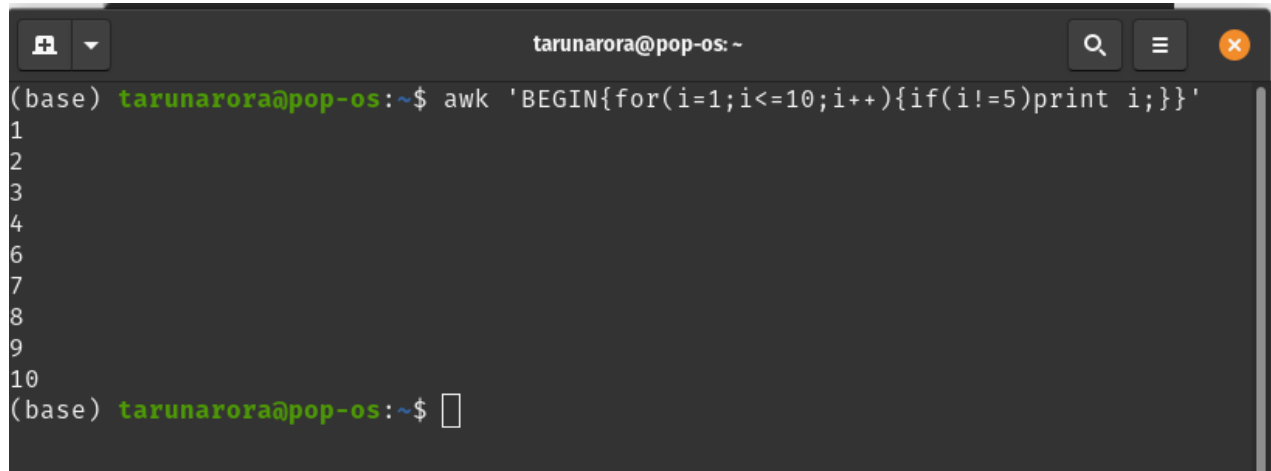
\*\*\*\*\*

**Q9.** Write an awk command to execute the loop except 5th iteration while printing number from 1 to 10.

**Solution:** -

Code: -

```
awk 'BEGIN{for(i=1;i<=10;i++){if(i!=5)print i;}}'
```

A terminal window titled 'tarunarora@pop-os: ~' with search, menu, and close buttons in the title bar. The prompt is '(base) tarunarora@pop-os:~\$'. The command 'awk 'BEGIN{for(i=1;i<=10;i++){if(i!=5)print i;}}'' is entered. The output shows numbers 1 through 10, with the number 5 omitted. The prompt returns to '(base) tarunarora@pop-os:~\$' with a cursor.

```
(base) tarunarora@pop-os:~$ awk 'BEGIN{for(i=1;i<=10;i++){if(i!=5)print i;}}'
1
2
3
4
6
7
8
9
10
(base) tarunarora@pop-os:~$
```

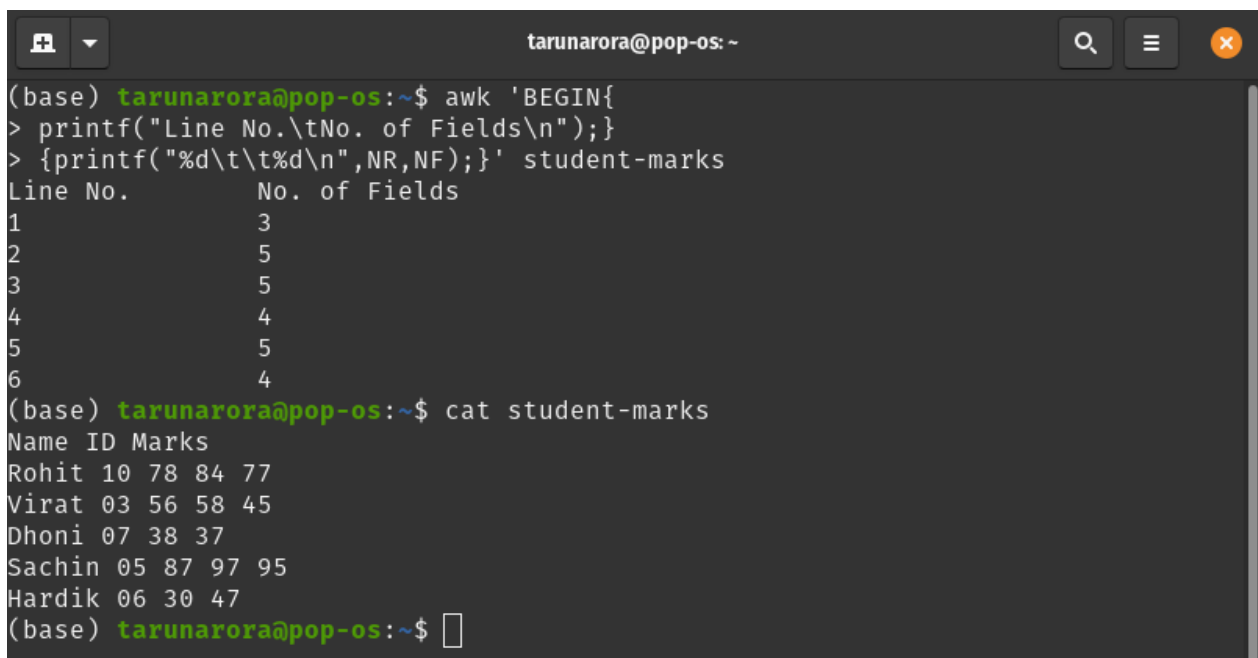
\*\*\*\*\*

**Q10.** Write an awk command to count and display the number of fields in each line of the file student-marks.

**Solution:** -

Code: -

```
awk 'BEGIN{
printf("Line No.\tNo. of Fields\n");}
{printf("%d\t\t%d\n",NR,NF);}' student-marks
```



```
(base) tarunarora@pop-os:~$ awk 'BEGIN{
> printf("Line No.\tNo. of Fields\n");}
> {printf("%d\t\t%d\n",NR,NF);}' student-marks
Line No.      No. of Fields
1             3
2             5
3             5
4             4
5             5
6             4
(base) tarunarora@pop-os:~$ cat student-marks
Name ID Marks
Rohit 10 78 84 77
Virat 03 56 58 45
Dhoni 07 38 37
Sachin 05 87 97 95
Hardik 06 30 47
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

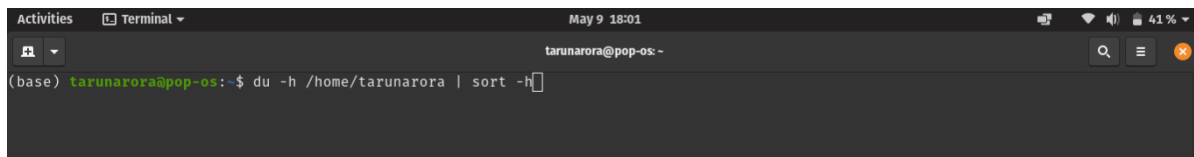
**Q11.** Sort the data that is in human readable format say 1K, 2M, 3G, 2T, where K, M, G, T represents Kilo, Mega, Giga, Tera from the /home/user file.

**Solution:** -

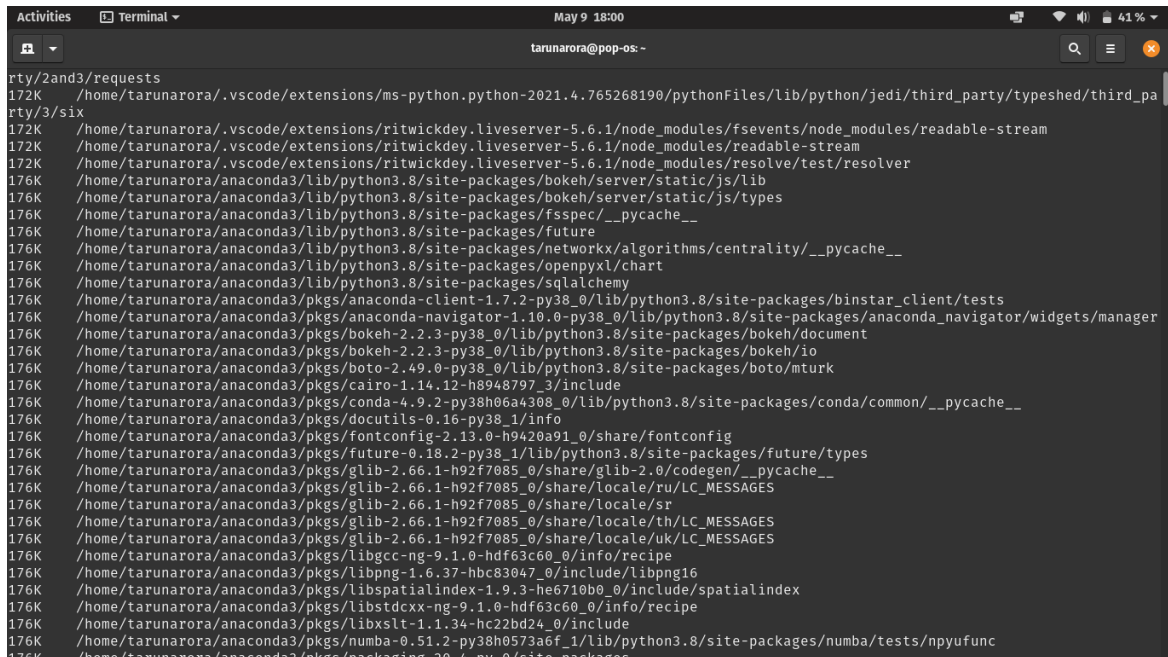
Code: -

```
du -h /home/tarunarora | sort -h
```

(Note: - In this command output is very large and terminal scroll buffer would be exhausted)



A terminal window titled 'Terminal' with a timestamp of 'May 9 18:01'. The prompt is 'tarunarora@pop-os: ~'. The command '(base) tarunarora@pop-os:~\$ du -h /home/tarunarora | sort -h' is entered and the cursor is at the end of the line.



A terminal window titled 'Terminal' with a timestamp of 'May 9 18:00'. The prompt is 'tarunarora@pop-os: ~'. The output of the command 'du -h /home/tarunarora | sort -h' is displayed, showing a list of files and their sizes in human-readable format, sorted by size. The output is truncated on the right side of the terminal window.

```
rty/2and3/requests
172K /home/tarunarora/.vscode/extensions/ms-python.python-2021.4.765268190/pythonFiles/lib/python/jedi/third_party/typeshed/third_pa
rty/3/six
172K /home/tarunarora/.vscode/extensions/ritwickdey.liveserver-5.6.1/node_modules/fsevents/node_modules/readable-stream
172K /home/tarunarora/.vscode/extensions/ritwickdey.liveserver-5.6.1/node_modules/readable-stream
172K /home/tarunarora/.vscode/extensions/ritwickdey.liveserver-5.6.1/node_modules/resolve/test/resolver
176K /home/tarunarora/anaconda3/lib/python3.8/site-packages/bokeh/server/static/js/lib
176K /home/tarunarora/anaconda3/lib/python3.8/site-packages/bokeh/server/static/js/types
176K /home/tarunarora/anaconda3/lib/python3.8/site-packages/fsspec/__pycache__
176K /home/tarunarora/anaconda3/lib/python3.8/site-packages/future
176K /home/tarunarora/anaconda3/lib/python3.8/site-packages/networkx/algorithms/centrality/__pycache__
176K /home/tarunarora/anaconda3/lib/python3.8/site-packages/openpyxl/chart
176K /home/tarunarora/anaconda3/lib/python3.8/site-packages/sqlalchemy
176K /home/tarunarora/anaconda3/pkgs/anaconda-client-1.7.2-py38_0/lib/python3.8/site-packages/binstar_client/tests
176K /home/tarunarora/anaconda3/pkgs/anaconda-navigator-1.10.0-py38_0/lib/python3.8/site-packages/anaconda_navigator/widgets/manager
176K /home/tarunarora/anaconda3/pkgs/bokeh-2.2.3-py38_0/lib/python3.8/site-packages/bokeh/document
176K /home/tarunarora/anaconda3/pkgs/bokeh-2.2.3-py38_0/lib/python3.8/site-packages/bokeh/io
176K /home/tarunarora/anaconda3/pkgs/boto-2.49.0-py38_0/lib/python3.8/site-packages/boto/mturk
176K /home/tarunarora/anaconda3/pkgs/cairo-1.14.12-h8948797_3/include
176K /home/tarunarora/anaconda3/pkgs/conda-4.9.2-py38h06a4308_0/lib/python3.8/site-packages/conda/common/__pycache__
176K /home/tarunarora/anaconda3/pkgs/docutils-0.16-py38_1/info
176K /home/tarunarora/anaconda3/pkgs/fontconfig-2.13.0-h9420a91_0/share/fontconfig
176K /home/tarunarora/anaconda3/pkgs/future-0.18.2-py38_1/lib/python3.8/site-packages/future/types
176K /home/tarunarora/anaconda3/pkgs/glib-2.66.1-h92f7085_0/share/glib-2.0/codegen/__pycache__
176K /home/tarunarora/anaconda3/pkgs/glib-2.66.1-h92f7085_0/share/locale/ru/LC_MESSAGES
176K /home/tarunarora/anaconda3/pkgs/glib-2.66.1-h92f7085_0/share/locale/sr
176K /home/tarunarora/anaconda3/pkgs/glib-2.66.1-h92f7085_0/share/locale/th/LC_MESSAGES
176K /home/tarunarora/anaconda3/pkgs/glib-2.66.1-h92f7085_0/share/locale/uk/LC_MESSAGES
176K /home/tarunarora/anaconda3/pkgs/libgcc-ng-9.1.0-hdf63c60_0/info/recipe
176K /home/tarunarora/anaconda3/pkgs/libpng-1.6.37-hbc83047_0/include/libpng16
176K /home/tarunarora/anaconda3/pkgs/libspatialindex-1.9.3-he6710b0_0/include/spatialindex
176K /home/tarunarora/anaconda3/pkgs/libstdcxx-ng-9.1.0-hdf63c60_0/info/recipe
176K /home/tarunarora/anaconda3/pkgs/libxslt-1.1.34-hc22bd24_0/include
176K /home/tarunarora/anaconda3/pkgs/numba-0.51.2-py38h0573a6f_1/lib/python3.8/site-packages/numba/tests/npufunc
176K /home/tarunarora/anaconda3/pkgs/packaging-20.4-py_0/site-packages
```

```
Activities Terminal May 9 18:00 tarunarora@pop-os: -
1.6G /home/tarunarora/snap
1.6G /home/tarunarora/snap/flutter
1.6G /home/tarunarora/snap/flutter/common
1.6G /home/tarunarora/snap/flutter/common/flutter
2.0G /home/tarunarora/.local/share/flatpak/runtime
2.1G /home/tarunarora/.android
2.1G /home/tarunarora/.android/avd
2.1G /home/tarunarora/.android/avd/Pixel_2_API_28.avd
2.2G /home/tarunarora/.cache
2.2G /home/tarunarora/.local/share/flatpak
2.3G /home/tarunarora/.config
2.7G /home/tarunarora/Android/Sdk/system-images
2.7G /home/tarunarora/Android/Sdk/system-images/android-28
2.7G /home/tarunarora/Android/Sdk/system-images/android-28/google_api_playstore
2.7G /home/tarunarora/Android/Sdk/system-images/android-28/google_api_playstore/x86
2.7G /home/tarunarora/.local/share/Trash/files/Android/Sdk/system-images/android-28
2.7G /home/tarunarora/.local/share/Trash/files/Android/Sdk/system-images/android-28/google_api_playstore
2.7G /home/tarunarora/.local/share/Trash/files/Android/Sdk/system-images/android-28/google_api_playstore/x86
3.1G /home/tarunarora/anaconda3/pkgs
3.2G /home/tarunarora/.local/share/Trash/files/Android/Sdk/system-images/android-30
3.2G /home/tarunarora/.local/share/Trash/files/Android/Sdk/system-images/android-30/google_api
3.2G /home/tarunarora/.local/share/Trash/files/Android/Sdk/system-images/android-30/google_api/x86
3.3G /home/tarunarora/anaconda3
4.3G /home/tarunarora/Android
4.3G /home/tarunarora/Android/Sdk
5.9G /home/tarunarora/.local/share/Trash/files/Android/Sdk/system-images
7.7G /home/tarunarora/.local/share/Trash/files/Android
7.7G /home/tarunarora/.local/share/Trash/files/Android/Sdk
9.0G /home/tarunarora/.local/share/Trash
9.0G /home/tarunarora/.local/share/Trash/files
13G /home/tarunarora/.local
13G /home/tarunarora/.local/share
32G /home/tarunarora
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

**Q12.** In the example ('how;now;brown;cow') convert the semi-colon into a space and display the first, third and fourth fields on the terminal using "cut command".

**Solution: -**

```
echo "how;now;brown;cow" | tr ';' ' ' | cut -d ' ' -f 1,3,4
```

A terminal window with a dark background. The title bar shows 'tarunarora@pop-os: ~'. The prompt is '(base) tarunarora@pop-os:~\$'. The command entered is 'echo "how;now;brown;cow" | tr ';' ' ' | cut -d ' ' -f 1,3,4'. The output is 'how brown cow'. The prompt is now '(base) tarunarora@pop-os:~\$' with a cursor.

```
(base) tarunarora@pop-os:~$ echo "how;now;brown;cow" | tr ';' ' ' | cut -d ' ' -f 1,3,4
how brown cow
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

**Q13.** Using cut command write a command to change the delimiter from input delimiter : (colon) to the output delimiter # (hash) in the field location 1, 6 and 7 from the /etc/passwd file where we have pattern "/bin/bash".

**Solution:** -

Code:-

```
cat /etc/passwd | grep "/bin/bash" | cut -d ':' -f 1,6,7 --output-delimiter='#'
```

A terminal window titled 'tarunarora@pop-os: ~' showing the execution of the command 'cat /etc/passwd | grep "/bin/bash" | cut -d ':' -f 1,6,7 --output-delimiter='#''. The output shows three lines: 'root#/root#/bin/bash', 'tarunarora#/home/tarunarora#/bin/bash', and the prompt '(base) tarunarora@pop-os:~\$' with a cursor.

```
(base) tarunarora@pop-os:~$ cat /etc/passwd | grep "/bin/bash" | cut -d ':' -f 1,6,7 --output-delimiter='#'
root#/root#/bin/bash
tarunarora#/home/tarunarora#/bin/bash
(base) tarunarora@pop-os:~$
```

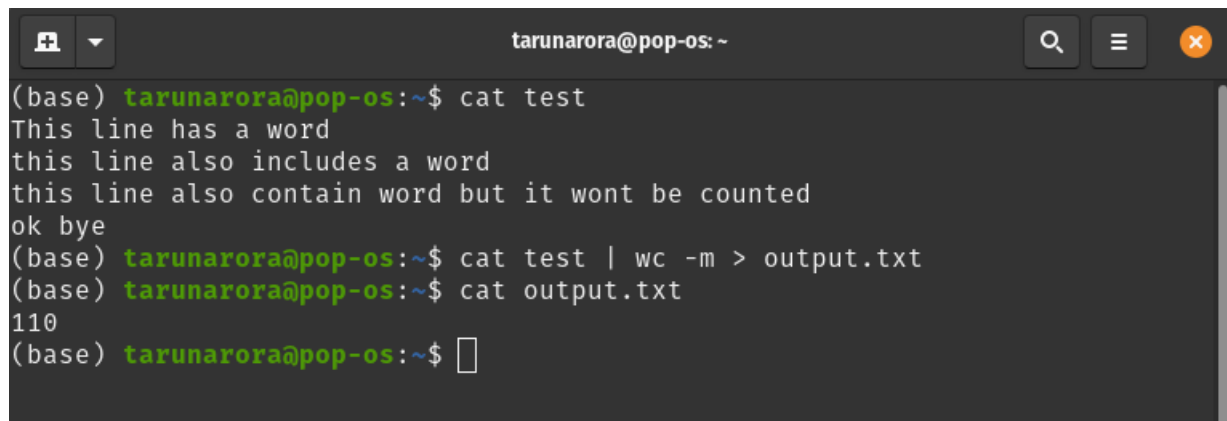
\*\*\*\*\*

**Q14.** Write a command to count number of characters in our file and save the output to new text file at the same time.

**Solution:-**

Code:-

```
cat test | wc -m > output.txt
```

A terminal window titled 'tarunarora@pop-os: ~' with search, menu, and close buttons in the title bar. The terminal shows the following sequence of commands and output:

```
(base) tarunarora@pop-os:~$ cat test
This line has a word
this line also includes a word
this line also contain word but it wont be counted
ok bye
(base) tarunarora@pop-os:~$ cat test | wc -m > output.txt
(base) tarunarora@pop-os:~$ cat output.txt
110
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

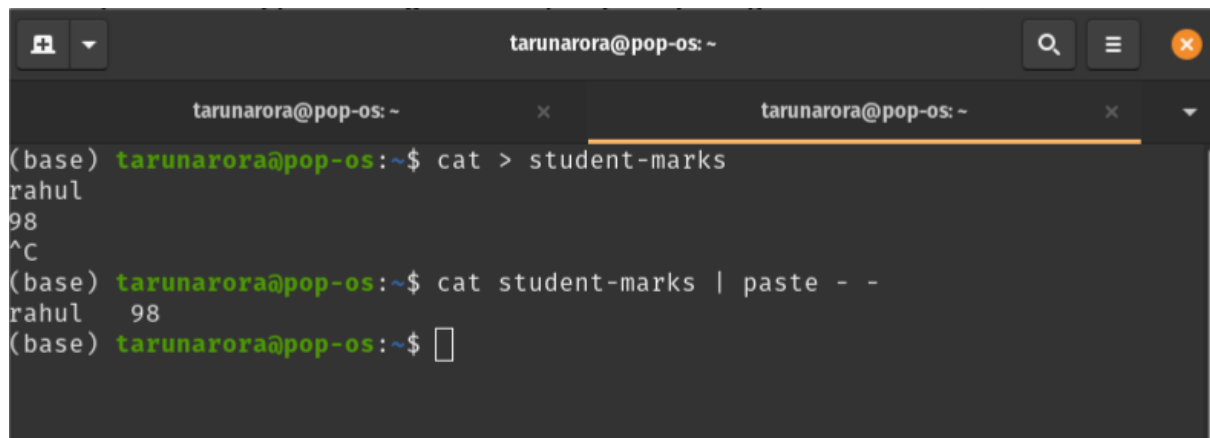


**Q15.** Write a paste command to merge 2 consecutive lines from the file student-marks into a single line.

**Solution:-**

Code: -

```
cat student-marks | paste - -
```

A terminal window titled 'tarunarora@pop-os: ~' with two tabs. The first tab shows the command 'cat > student-marks' being executed, resulting in the output 'rahul' followed by '98' on a new line. The second tab shows the command 'cat student-marks | paste - -' being executed, resulting in the output 'rahul 98' on a single line. The prompt '(base) tarunarora@pop-os:~\$' is visible at the start of each command line.

```
(base) tarunarora@pop-os:~$ cat > student-marks
rahul
98
^C
(base) tarunarora@pop-os:~$ cat student-marks | paste - -
rahul 98
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

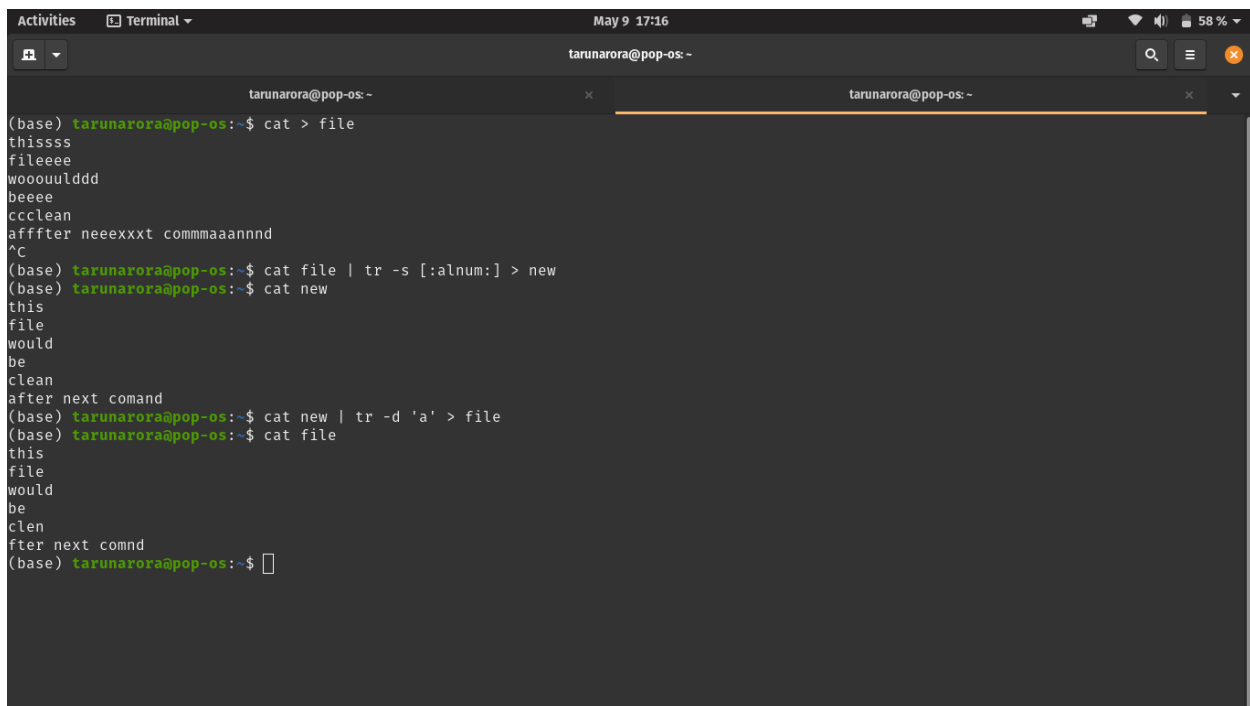
**Q16.** Write a command using `tr` to squeeze the repetition of characters from a file. Also remove the character "a" from the file.

**Solution: -**

Code:-

```
cat {file1} | tr -s [:alnum:] > {file2}
```

```
cat {file1} | tr -d 'a' > {file2}
```



```
Activities Terminal May 9 17:16 58 %
tarunarora@pop-os: -
tarunarora@pop-os: -
tarunarora@pop-os: -
(base) tarunarora@pop-os:~$ cat > file
thissss
fileeee
woooooulddd
beeee
ccclean
afffter neeexxt commmaannnd
^C
(base) tarunarora@pop-os:~$ cat file | tr -s [:alnum:] > new
(base) tarunarora@pop-os:~$ cat new
this
file
would
be
clean
after next comand
(base) tarunarora@pop-os:~$ cat new | tr -d 'a' > file
(base) tarunarora@pop-os:~$ cat file
this
file
would
be
clen
fter next comnd
(base) tarunarora@pop-os:~$
```

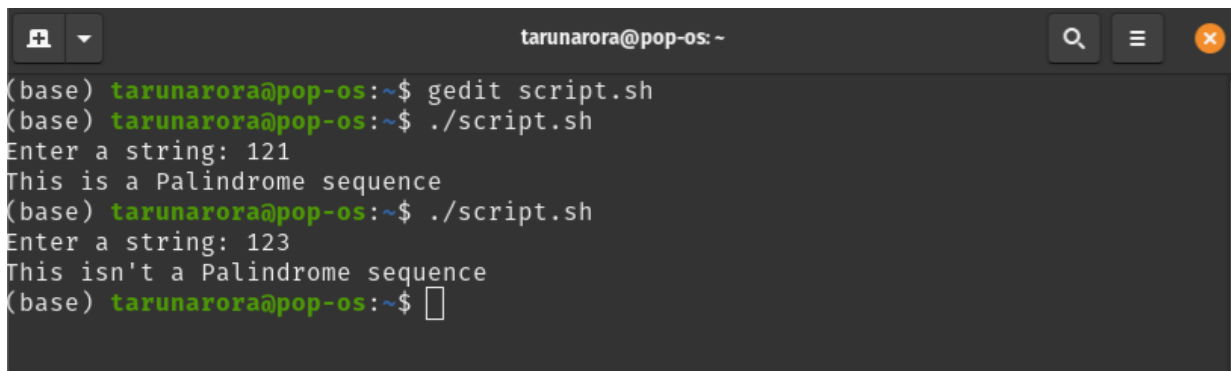
\*\*\*\*\*

**Q17.** Write a shell script which takes input as a string on a terminal and check whether it's palindrome or not a palindrome.

**Solution:** -



```
1#!/bin/bash
2read -p "Enter a string: " string
3if [[ $(rev <<<"$string") == "$string" ]]; then
4echo "This is a Palindrome sequence"
5else echo "This isn't a Palindrome sequence"
6fi
```

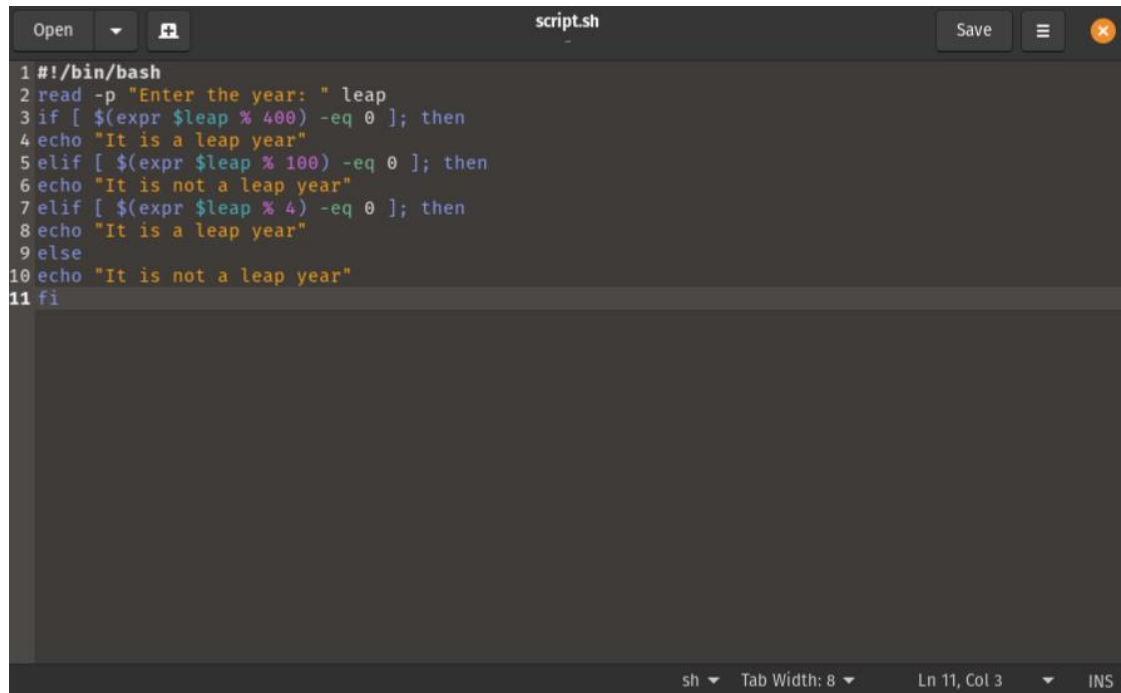


```
(base) tarunarora@pop-os:~$ gedit script.sh
(base) tarunarora@pop-os:~$ ./script.sh
Enter a string: 121
This is a Palindrome sequence
(base) tarunarora@pop-os:~$ ./script.sh
Enter a string: 123
This isn't a Palindrome sequence
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

**Q18.** Write a shell script to check the given year is leap year or not a leap year.

**Solution:** -

A screenshot of a text editor window titled 'script.sh'. The window has a dark background with light-colored text. The code is as follows:

```
1#!/bin/bash
2read -p "Enter the year: " leap
3if [ $(expr $leap % 400) -eq 0 ]; then
4echo "It is a leap year"
5elif [ $(expr $leap % 100) -eq 0 ]; then
6echo "It is not a leap year"
7elif [ $(expr $leap % 4) -eq 0 ]; then
8echo "It is a leap year"
9else
10echo "It is not a leap year"
11fi
```

The status bar at the bottom shows 'sh', 'Tab Width: 8', 'Ln 11, Col 3', and 'INS'.A screenshot of a terminal window titled 'tarunarora@pop-os: ~'. The window has a dark background with light-colored text. The commands and output are as follows:

```
(base) tarunarora@pop-os:~$ gedit script.sh
(base) tarunarora@pop-os:~$ ./script.sh
Enter the year: 1256
It is a leap year
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

**Q19.** Write a shell script to display the list of prime number. It takes input as "How many prime numbers:" from the user. E.g.: How many prime numbers: 4 then it displays 2,3,5,7.

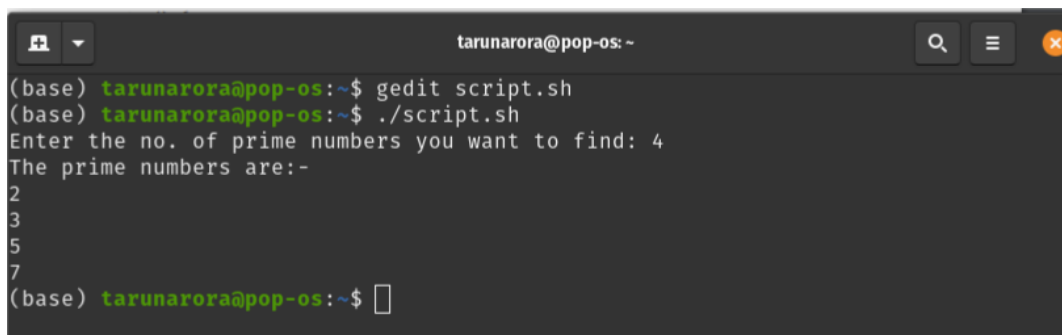
**Solution:** -



```
1 #!/bin/bash
2 flag=0
3 counter=0
4 var=1
5 read -p "Enter the no. of prime numbers you want to find: " count
6
7 prime() {
8   flag=1
9   for ((i = 2; i <= $var / 2; i++)); do
10    ans=$((var % i))
11    if [ $ans -eq 0 ]; then
12      flag=0
13    fi
14  done
15 }
16
17 echo "The prime numbers are:- "
18 while [ $counter -lt $count ]; do
19   var=$((var + 1))
20   prime
21   if [ $flag -eq 1 ]; then
22     echo "$var"
23     counter=$((counter + 1))
24   fi
25 done
```

Loading file "/home/tarunarora/script.sh"...

sh Tab Width: 8 Ln 17, Col 7 INS

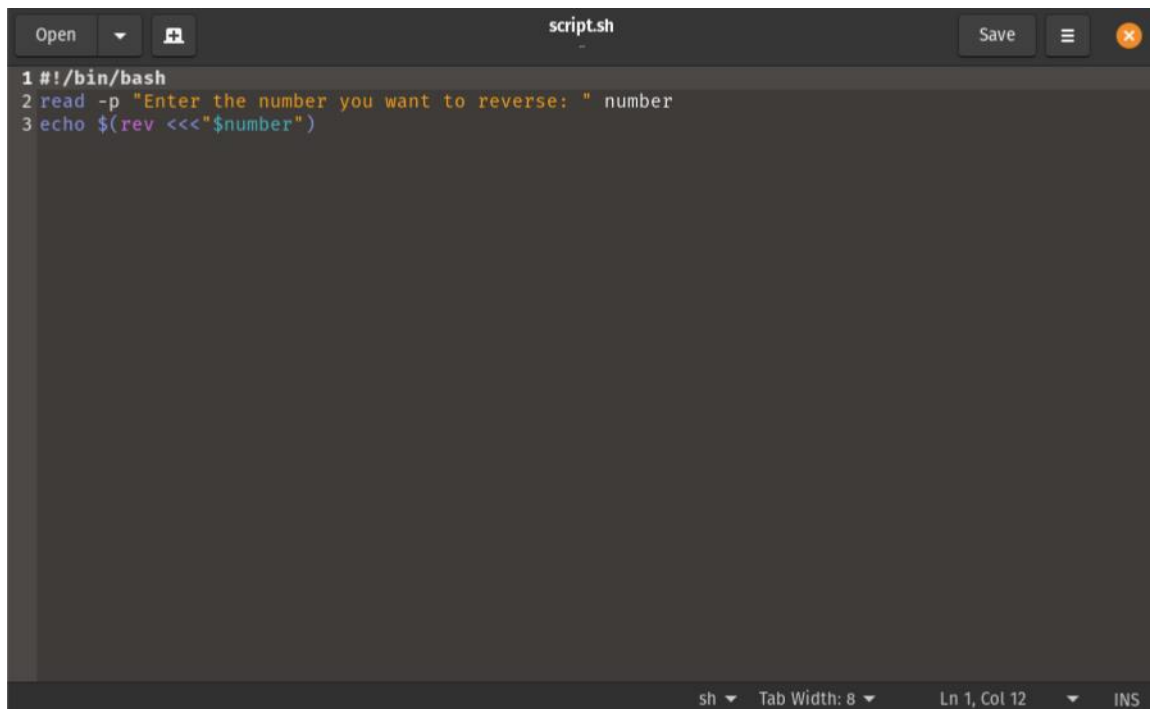


```
(base) tarunarora@pop-os:~$ gedit script.sh
(base) tarunarora@pop-os:~$ ./script.sh
Enter the no. of prime numbers you want to find: 4
The prime numbers are:-
2
3
5
7
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

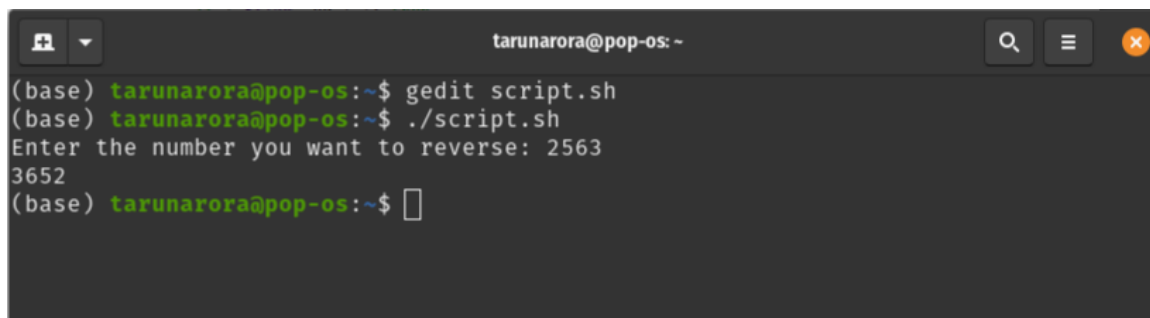
**Q20.** Write a shell script to reverse the input digits.

**Solution:** -



```
1 #!/bin/bash
2 read -p "Enter the number you want to reverse: " number
3 echo $(rev <<<"$number")
```

The screenshot shows a gedit editor window with the title 'script.sh'. The script contains three lines: a shebang line, a 'read' command to prompt for a number, and an 'echo' command using 'rev' to reverse the input. The status bar at the bottom indicates 'sh', 'Tab Width: 8', 'Ln 1, Col 12', and 'INS'.



```
(base) tarunarora@pop-os:~$ gedit script.sh
(base) tarunarora@pop-os:~$ ./script.sh
Enter the number you want to reverse: 2563
3652
(base) tarunarora@pop-os:~$
```

The screenshot shows a terminal window where the script is executed. The user runs 'gedit script.sh' and then './script.sh'. The script prompts for a number, and the user enters '2563'. The output is '3652'. The prompt returns to the shell.

\*\*\*\*\*

**Q21.** Write a script to generate a password of minimum length of 8. It must be alphanumeric, containing at least one upper case and one lower case character.

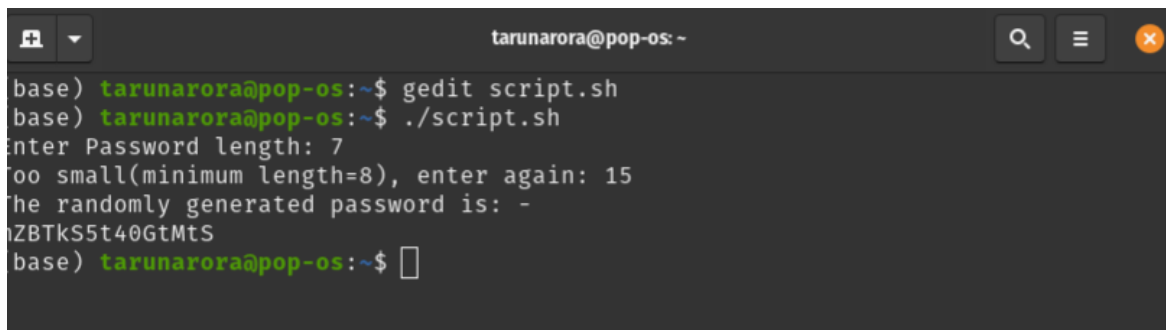
**Solution:** -



```
1#!/bin/bash
2choose() {
3echo -n ${1:RANDOM%${#1}:1}
4}
5echo -n "Enter Password length: "
6while read length; do
7if [ $length -ge 8 ]; then
8break
9else
10echo -n "Too small(minimum length=8), enter again: "
11fi
12done
13length=$((length - 2))
14echo "The randomly generated password is: -"
15choose 'abcdefghijklmnopqrstuvwxyz'
16choose 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
17echo $(tr </dev/urandom -dc _A-Z-a-z-0-9 | head -c$length)
18
```

Saving file "/home/tarunarora/script.sh"...

sh Tab Width: 8 Ln 16, Col 36 INS



```
tarunarora@pop-os: ~
base) tarunarora@pop-os:~$ gedit script.sh
base) tarunarora@pop-os:~$ ./script.sh
Enter Password length: 7
Too small(minimum length=8), enter again: 15
The randomly generated password is: -
nZBTkS5t40GtMtS
base) tarunarora@pop-os:~$
```

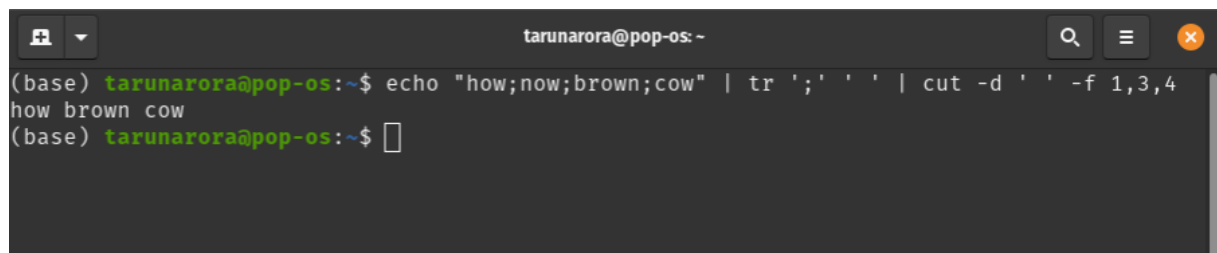
\*\*\*\*\*

**Q22** In the example ('how;now;brown;cow') convert the semi-colon into a space and display the first, third and fourth fields on the terminal using "cut command".

**Solution: -**

Code: -

```
echo "how;now;brown;cow" | tr ';' ' ' | cut -d ' ' -f 1,3,4
```

A terminal window with a dark background. The title bar shows 'tarunarora@pop-os: ~'. The prompt is '(base) tarunarora@pop-os:~\$'. The command entered is 'echo "how;now;brown;cow" | tr ';' ' ' | cut -d ' ' -f 1,3,4'. The output is 'how brown cow'. The prompt is now '(base) tarunarora@pop-os:~\$' with a cursor.

```
(base) tarunarora@pop-os:~$ echo "how;now;brown;cow" | tr ';' ' ' | cut -d ' ' -f 1,3,4
how brown cow
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*



**Q23.** Using cut command write a command to change the delimiter from input delimiter : (colon) to the output delimiter # (hash) in the field location 1, 6 and 7 from the /etc/passwd file where we have pattern "/bin/bash".

**Solution:** -

Code:-

```
cat /etc/passwd | grep "/bin/bash" | cut -d ':' -f 1,6,7 --output-delimiter='#'
```

A terminal window titled 'tarunarora@pop-os: ~' showing the execution of the command 'cat /etc/passwd | grep "/bin/bash" | cut -d ':' -f 1,6,7 --output-delimiter='#''. The output shows three lines: 'root#/root#/bin/bash', 'tarunarora#/home/tarunarora#/bin/bash', and the prompt '(base) tarunarora@pop-os:~\$'.

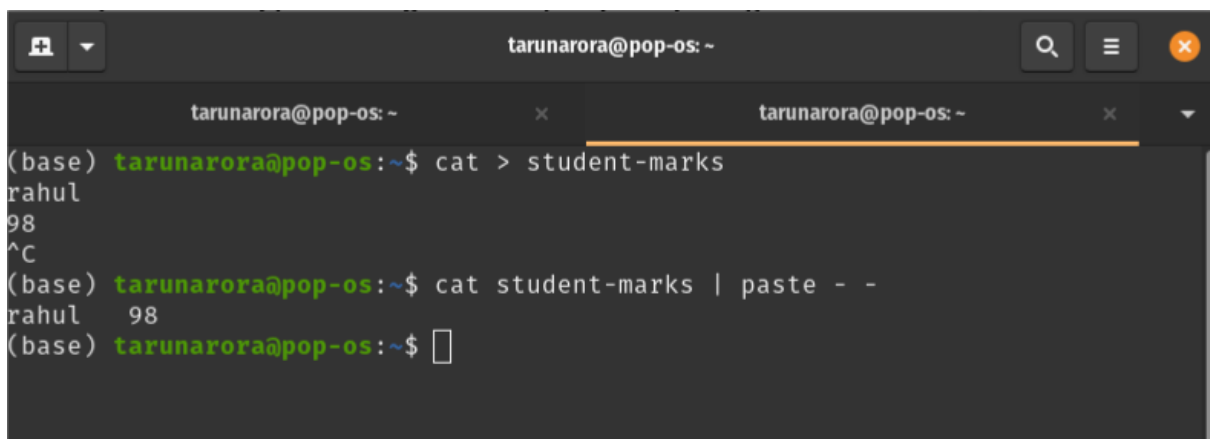
\*\*\*\*\*

**Q24.** Write a paste command to merge 2 consecutive lines from the file student-marks into a single line.

**Solution:-**

Code: -

```
cat student-marks | paste - -
```



The image shows a terminal window with two tabs. The first tab shows the command `cat > student-marks` being executed, which outputs `rahul` and `98`. The second tab shows the command `cat student-marks | paste - -` being executed, which outputs `rahul 98`. The terminal window has a dark background and a light-colored text.

```
(base) tarunarora@pop-os: ~$ cat > student-marks
rahul
98
^C
(base) tarunarora@pop-os: ~$ cat student-marks | paste - -
rahul 98
(base) tarunarora@pop-os: ~$
```

\*\*\*\*\*

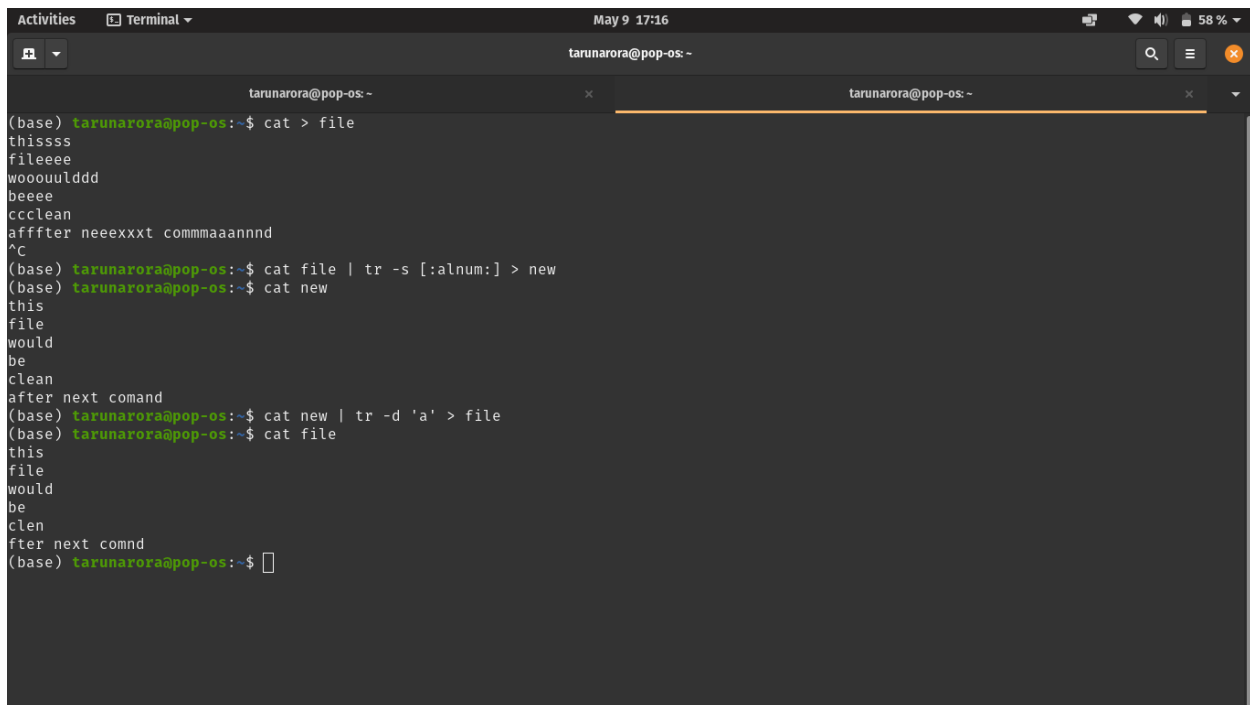
**Q25.** Write a command using `tr` to squeeze the repetition of characters from a file. Also remove the character "a" from the file.

**Solution:** -

Code:-

```
cat {file1} | tr -s [:alnum:] > {file2}
```

```
cat {file1} | tr -d 'a' > {file2}
```



```
Activities Terminal May 9 17:16
tarunarora@pop-os: ~
tarunarora@pop-os: ~$ cat > file
thissss
fileeee
woooooulddd
beeee
ccclean
afffter neeexxt commmaannnd
^C
tarunarora@pop-os: ~$ cat file | tr -s [:alnum:] > new
tarunarora@pop-os: ~$ cat new
this
file
would
be
clean
after next comand
tarunarora@pop-os: ~$ cat new | tr -d 'a' > file
tarunarora@pop-os: ~$ cat file
this
file
would
be
clen
fter next comnd
tarunarora@pop-os: ~$
```

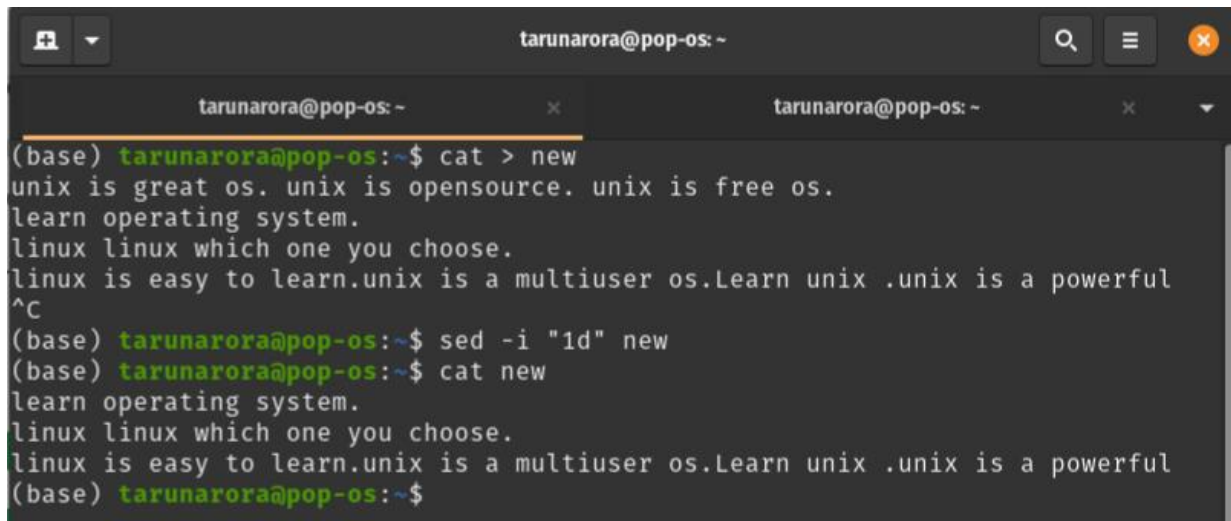
\*\*\*\*\*

**Q26.** How to remove the header from a file?

**Solution:** -

Code: -

```
sed -i "1d" {filename}
```



```
tarunarora@pop-os: ~  
(base) tarunarora@pop-os:~$ cat > new  
unix is great os. unix is opensource. unix is free os.  
learn operating system.  
linux linux which one you choose.  
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful  
^C  
(base) tarunarora@pop-os:~$ sed -i "1d" new  
(base) tarunarora@pop-os:~$ cat new  
learn operating system.  
linux linux which one you choose.  
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful  
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*

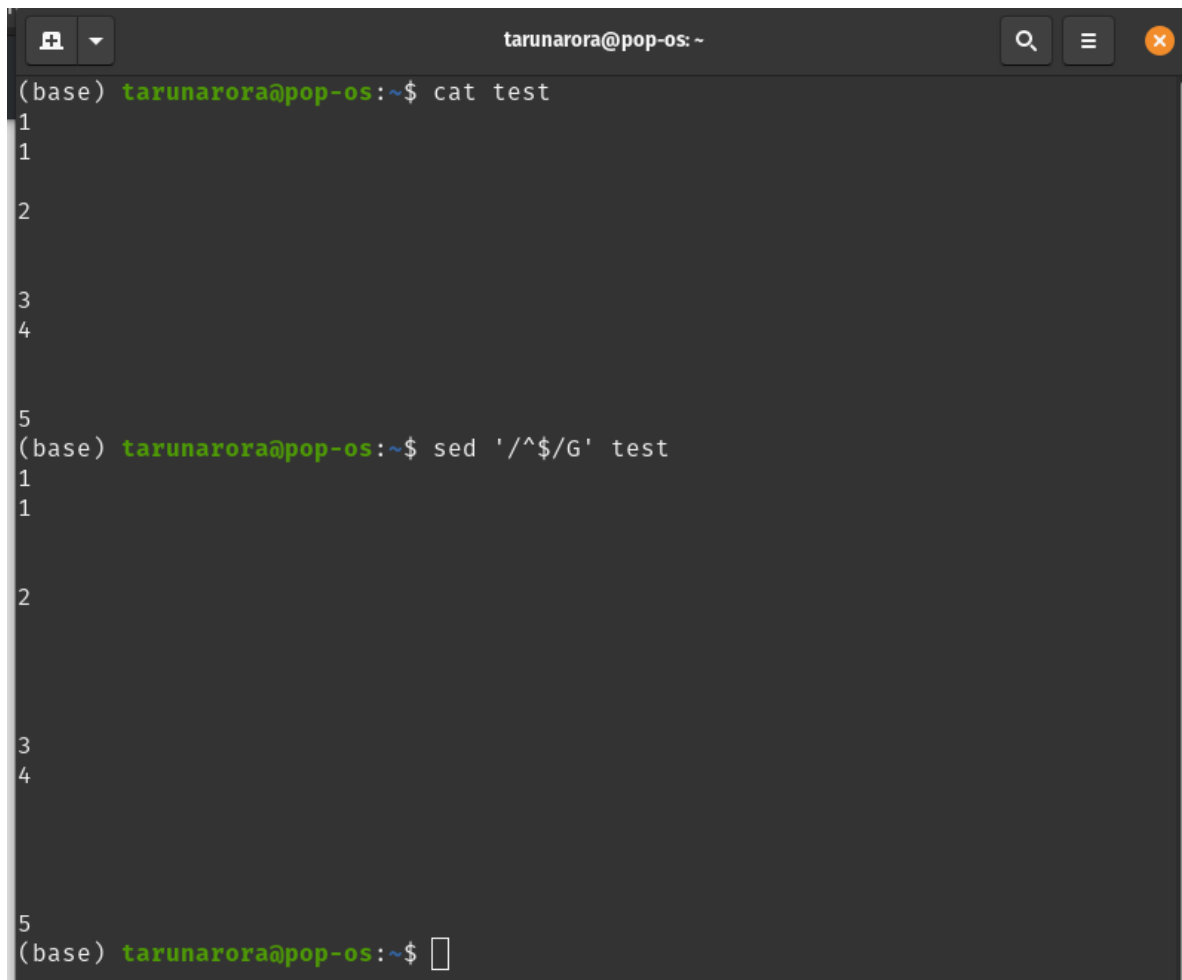
**Q27.** Write a command to duplicate empty lines in a file?  
Also write a command to display all content of a file  
except for some lines.

**Solution:** -

Code: -

```
sed '/^$/G' {input_file}
```

(Note one can use -i option to make changes in that file)



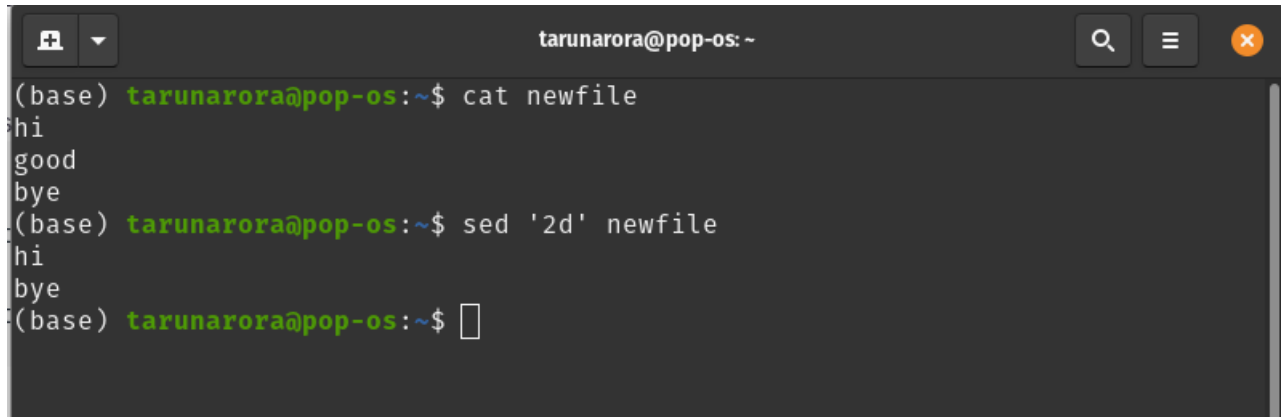
```
tarunarora@pop-os: ~  
(base) tarunarora@pop-os:~$ cat test  
1  
1  
  
2  
  
3  
4  
  
5  
(base) tarunarora@pop-os:~$ sed '/^$/G' test  
1  
1  
  
2  
  
3  
4  
  
5  
(base) tarunarora@pop-os:~$
```

Command to display all content except a few lines: -

`sed 'nd' {file_name}` (To display all lines except nth line)

In case a range of lines is to be deleted: -

`Sed 'n,md' {file_name}` (Lines from nth to mth line will not be displayed)

A terminal window titled 'tarunarora@pop-os: ~' with search, menu, and close buttons in the top right. The terminal shows a sequence of commands and their output. First, the user runs 'cat newfile', which outputs 'hi', 'good', and 'bye'. Then, the user runs 'sed '2d' newfile', which outputs 'hi' and 'bye', as the second line 'good' has been deleted. The prompt '(base) tarunarora@pop-os:~\$' is shown at the end of each command line.

```
(base) tarunarora@pop-os:~$ cat newfile
hi
good
bye
(base) tarunarora@pop-os:~$ sed '2d' newfile
hi
bye
(base) tarunarora@pop-os:~$
```

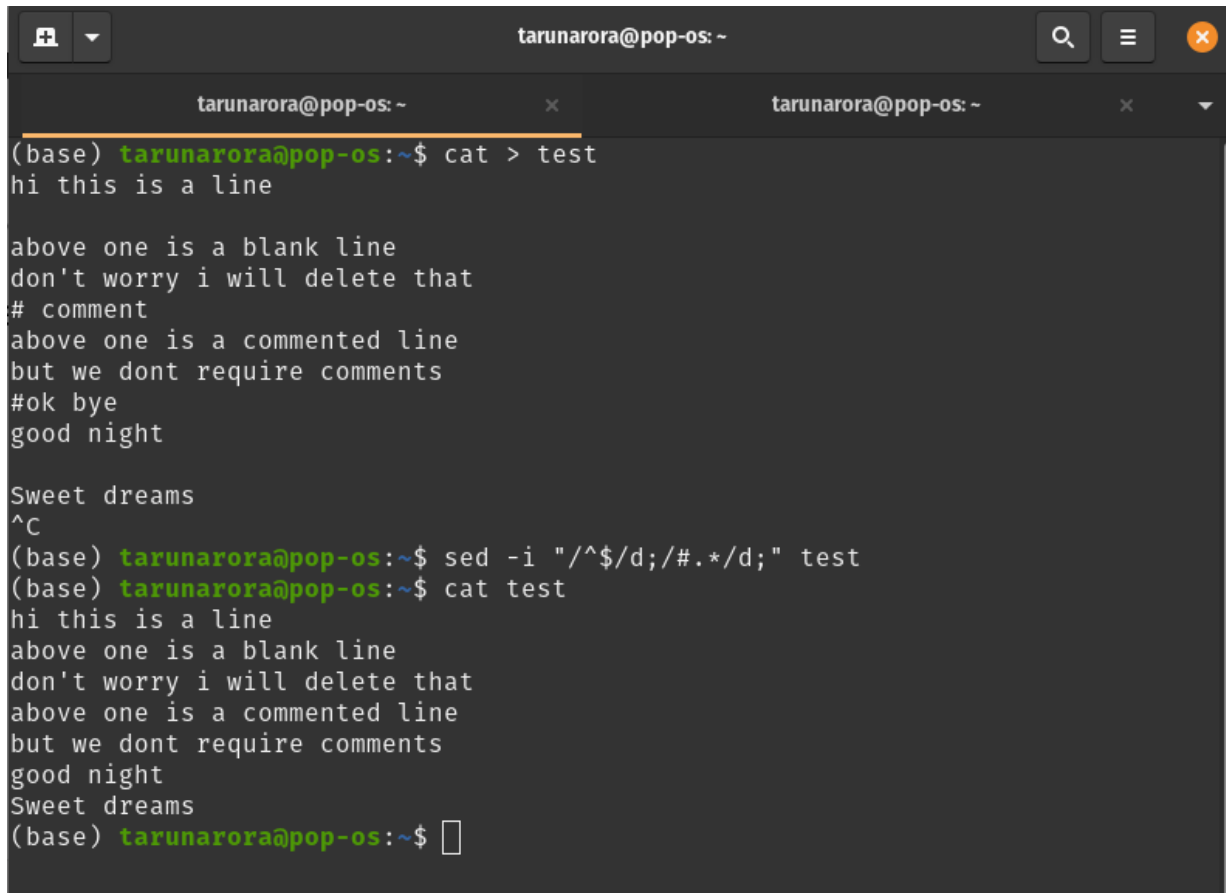
\*\*\*\*\*

**Q28.** Write a sed command to remove all commented and empty lines in a file

**Solution:** -

Code: -

```
sed -i "/^$/d;/#.*d;" {file_name}
```



```
tarunarora@pop-os: ~  
(base) tarunarora@pop-os:~$ cat > test  
hi this is a line  
  
above one is a blank line  
don't worry i will delete that  
# comment  
above one is a commented line  
but we dont require comments  
#ok bye  
good night  
  
Sweet dreams  
^C  
(base) tarunarora@pop-os:~$ sed -i "/^$/d;/#.*d;" test  
(base) tarunarora@pop-os:~$ cat test  
hi this is a line  
above one is a blank line  
don't worry i will delete that  
above one is a commented line  
but we dont require comments  
good night  
Sweet dreams  
(base) tarunarora@pop-os:~$
```

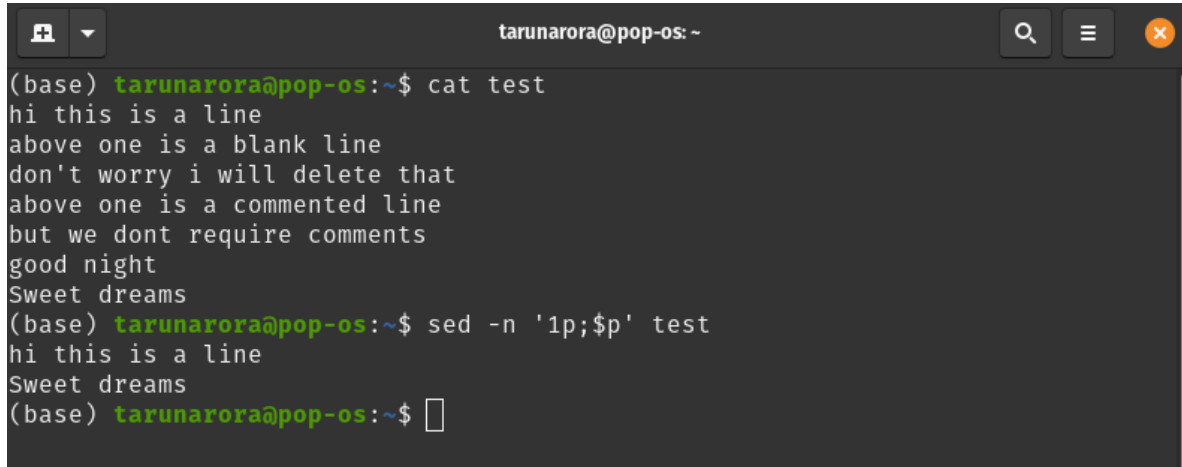
\*\*\*\*\*

**Q29.** How to print first and last line of the file using Sed Command ? Also write a sed command to print the lines that don't contain the word "unix" from a given file?

**Solution:** -

Code: -

```
sed -n '1p;$p' {file_name}
```



```
tarunarora@pop-os: ~  
(base) tarunarora@pop-os:~$ cat test  
hi this is a line  
above one is a blank line  
don't worry i will delete that  
above one is a commented line  
but we dont require comments  
good night  
Sweet dreams  
(base) tarunarora@pop-os:~$ sed -n '1p;$p' test  
hi this is a line  
Sweet dreams  
(base) tarunarora@pop-os:~$
```

Code: -

```
sed '/\unix\b/d' {file_name}
```



```
tarunarora@pop-os: ~  
(base) tarunarora@pop-os:~$ cat test  
unixity  
man sed  
clearly it is unix  
unix  
punixity  
unixik  
(base) tarunarora@pop-os:~$ sed '/\unix\b/d' test  
unixity  
man sed  
punixity  
unixik  
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*



**Q30.** Given a file, replace all occurrence of word "linux" with "unix" from 2th line till end in only those lines that contains word "os".

**Solution:** -

Input: -

linux is great os. unix is opensource. unix is free os.

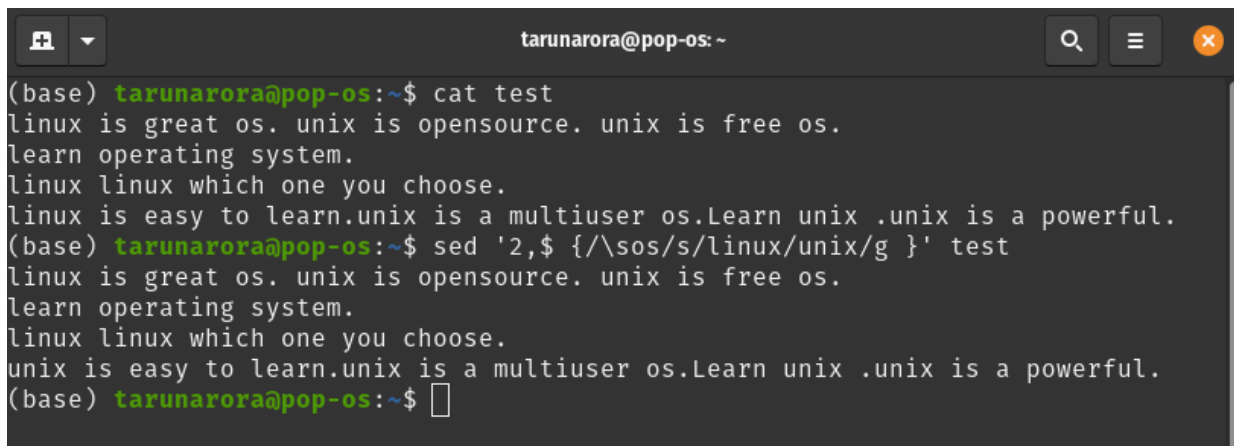
learn operating system.

linux linux which one you choose.

linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

Code: -

```
sed '2,$ {/\sos/s/linux/unix/g }' test
```



```
tarunarora@pop-os: ~  
(base) tarunarora@pop-os:~$ cat test  
linux is great os. unix is opensource. unix is free os.  
learn operating system.  
linux linux which one you choose.  
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.  
(base) tarunarora@pop-os:~$ sed '2,$ {/\sos/s/linux/unix/g }' test  
linux is great os. unix is opensource. unix is free os.  
learn operating system.  
linux linux which one you choose.  
unix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.  
(base) tarunarora@pop-os:~$
```

\*\*\*\*\*EOF\*\*\*\*\*