Assignment\_1

Section : A

Roll no : 422147

1)grep:

1>grep -c pattern filename~ this prints only a count of the lines that match the

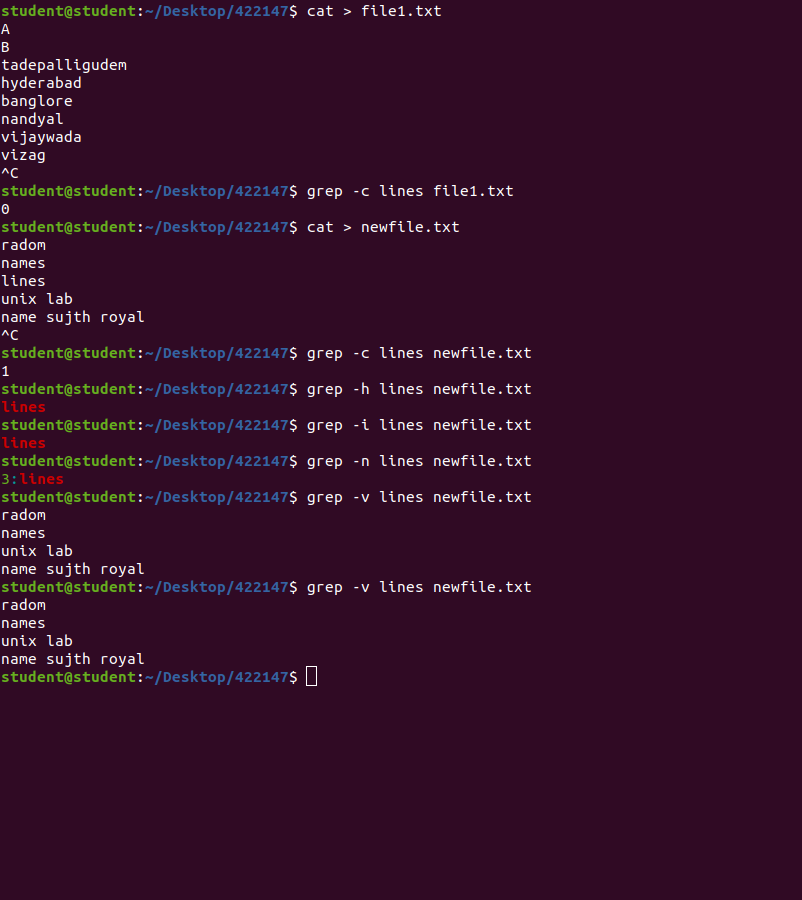
pattern(words,nums,charcs).

2>grep -h pattern filename ~ display the matched lines, but do not display the filenames .

3>grep -i pattern filename ~ ignores case for matching.

4>grep -n pattern filename ~ display the matched lines and their line numbers.

5>grep -v pattern filename ~ this prints out all the lines that do not matches the pattern.



2. uniq command:

1>uniq -c filename~ it tells how many times a line was repeated by displaying a number as a prefix with

the line

2>uniq -d filename ~ it only prints the repeated lines and not the lines which arent repeated

3>uniq -f N - ~ -skip fields(N) : it allows you to skip N fields (a field is a group of characters,delimited by

whitespace) of a line before determining the uniqueness of a line

. 4>uniq -i - ~ -ignore case :By default , comparisons done are case sensitive but with option(i) case

insensitive comparisons can be made

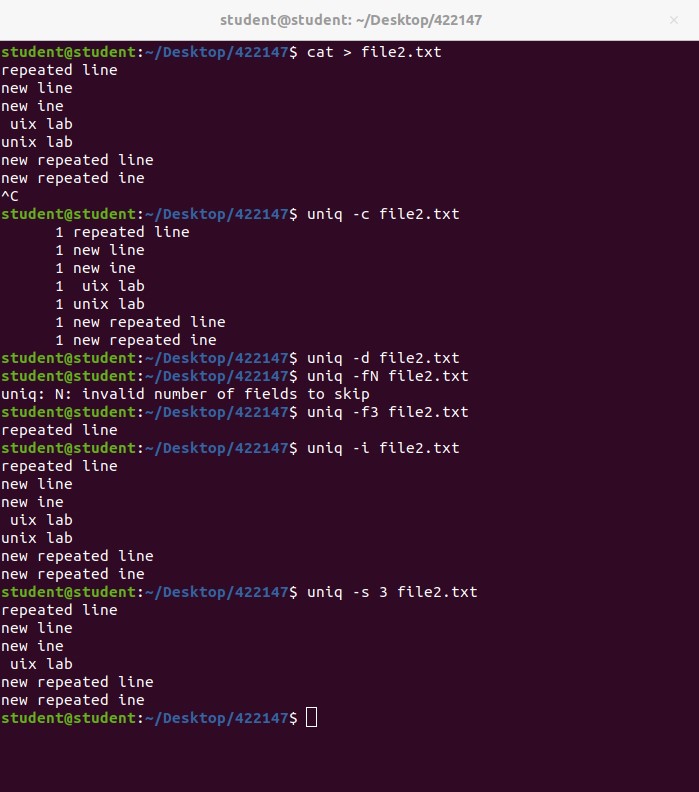
5>uniq -s N - ~ skip chars(N) : it doesnt compare the first N characters of each line while determining

uniqueness.this is like the -f option but it skips the individual characters rather than fields

6>uniq -u ~ -unique : it allows you to print only unique lines

7>uniq -z ~ -zero terminated: it will make a line end with 0 bytes(NULL),instead of a newline

8>uniq -w N ~ check chars(N) ~iit only compares N characters in line

3)tr command

1>cat filename | tr [a-z] [A-Z] ~ converts lower case characters to upper case characters

2>cat filename |tr [:lower:] [:upper:] ~ converts lower case characters to upper case characters

3>echo “sentence” | tr [:space:] ‘\t’ ~ translates white-space characters to tabs

4>$ tr “{}” “()” newfile.txt~ translates braces into parenthesis.

5>echo “sentence” | tr -s “ “~ to squeeze a sequence of repetitive characters using -s option

6>tr -s “ “ <<< “sentence” ~to squeeze a sequence of repetitive characters using -s option but using a

string here

7>echo “sentence” | tr -d W~ to delete specified characters using -d option.

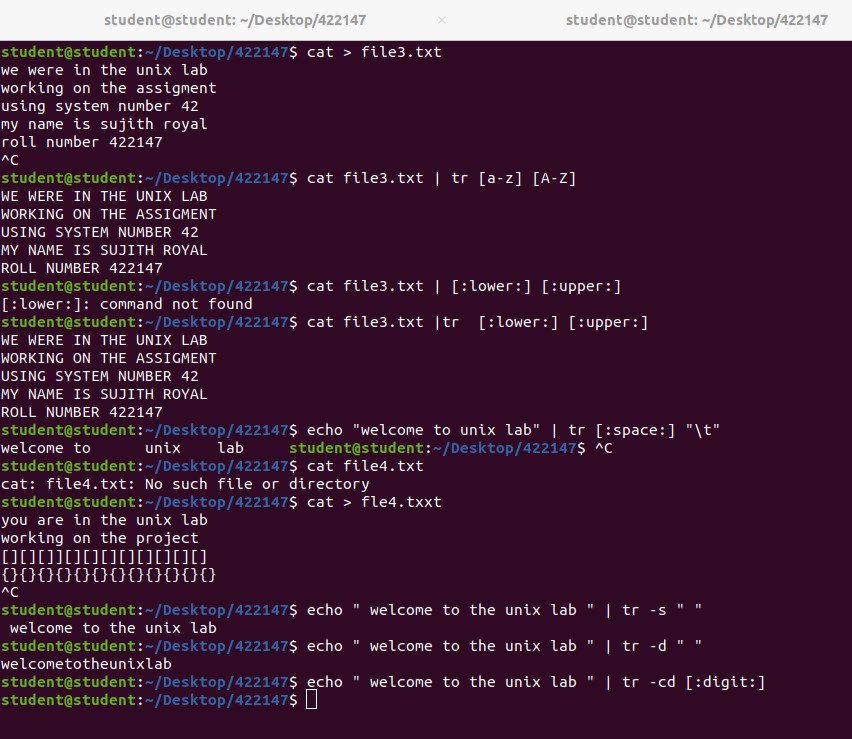
8>tr -d W <<< “sentence” ~ to delete specified characters using -d option using a string here.

9>echo “ sentence containing digits” | tr -d [:digit:]~to remove all the digits from string.

10>tr -d [:digit:] <<< “sentence contaning digits” ~ to remove all the digits from string .

11>echo “sentence containing digits” | tr -cd [:digit:]~complement the sets using -c option .

12>tr -cd [:digit:] <<< “sentence containing digits” ~complement the sets using -c option



4)pr command:

1>pr -k filename ~ we can print the content in k columns .

2>pr -d filename ~ to double paces input,reduces clutter -d option is used.

3>pr -n filename ~ to provide the number lines which helpps in debugging the code -n option is used

.4>pr - -help ~it gives the detail of all options of pr.

5>pr - -version ~ to print the version number of command pr

5>pr - -version ~ to print the version number

5)paste command:

1>paste file1 file2 file 3 ~ without any option paste merges the files in parallel. The paste command

writes corresponding lines from the files with tab as a deliminator on the terminal .

2>paste -d “|” file1 file2 file3 ~ it prints even if any character is specified.

3>paste -d “|,” file1 file2 file3 ~ it prints even multi characters are specified.

4>paste -s file1 file2 file3 AND paste -s -d “:” file1 file2 file3 ~ we can merge the file in sequentially

manner using the -s option. It reads all the lines from a single file and merges all these lines into a single

line with each line separated by tab . and these single lines are separated by newline.

5>cat file1 | paste - - (or) paste - - - < file~ the paste command can also be used to merge N consecutive

lines from file into a single line.here N can be specified by specifying number hyphens (-) after paste.

6>cut -d “ “ -f 1 file1 | file2 file3 ~cut command is used with -f option for cutting out first field of state

and output is piipelined with paste command having one filename and instead of second file name

hyphen is specified (if hyphen is not specified the input from shell is not pasted)

7>cut -d “ “ -f 1 file1 | paste - file2 ~ ordering of pasting can be changed by altering the location of

Hyphen

7. head command:

• Displays the first few lines of a file. Command used: head filename

• Display a specific number of lines: Command used: head -n 10 filename

8.Tail

• Displays the last few lines of a file. Command used: tail filename

• Display a specific number of lines: Command used: tail -n 10 filename

Follow the growth of a file (similar to tail -f): Command used: tail -f filename

9.Sort

• Sort a file alphabetically: Command used: sort filename

• Sort a file numerically: Command used: sort -n filename

• Sort a file in reverse order: Command used: sort -r filename

Sort a file and remove duplicate lines: Command used: sort -u filename

Sort a file based on a specific column (using space as the delimiter): Command used: sort -k filename

• Sort a file in a case-insensitive manner: Command used: sort -f filename

• Sort lines in memory for faster sorting: Command used: sort -S filename

• Sort based on the month abbreviation (e.g., Jan, Feb, Mar): Command used: sort -M filename