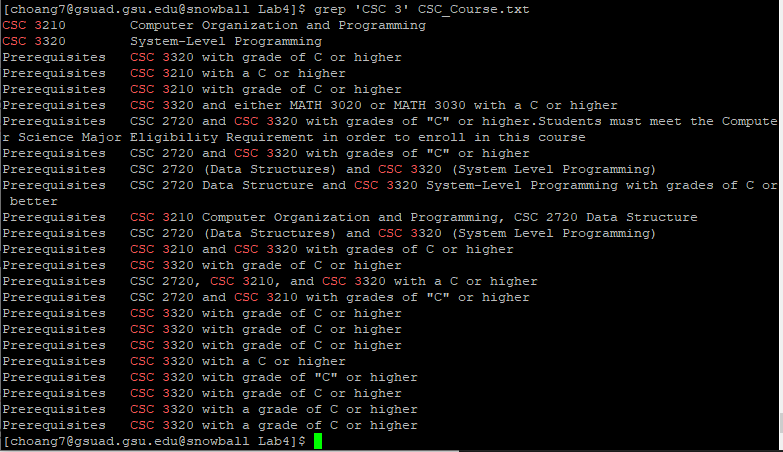
Cuong Hoang

CSC3320

Lab 4 In-lab Assignment

Part 1:

4)



This command outputs the lines containing the string “CSC 3” in the file ‘CSC\_Course.txt’.

5)

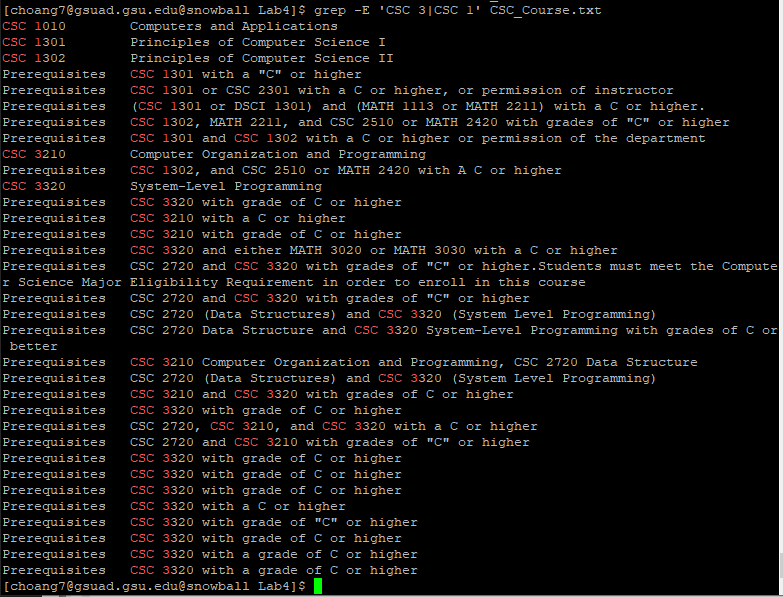


N/A. This command does not print out any output. It’s because the symbol ‘|’ alone is not recognized in basic regular expression command ‘grep <pattern> <file>’, otherwise, we need to insert a backslash escape for basic regular expression to recognize it as a meta-character. To make this print out the output, we can modify the command as the following:

grep ‘CSC 3\|CSC 1’ CSC\_Course.txt

This will output the lines containing either the string “CSC 3” or “CSC 1” in the file ‘CSC\_Course.txt’.

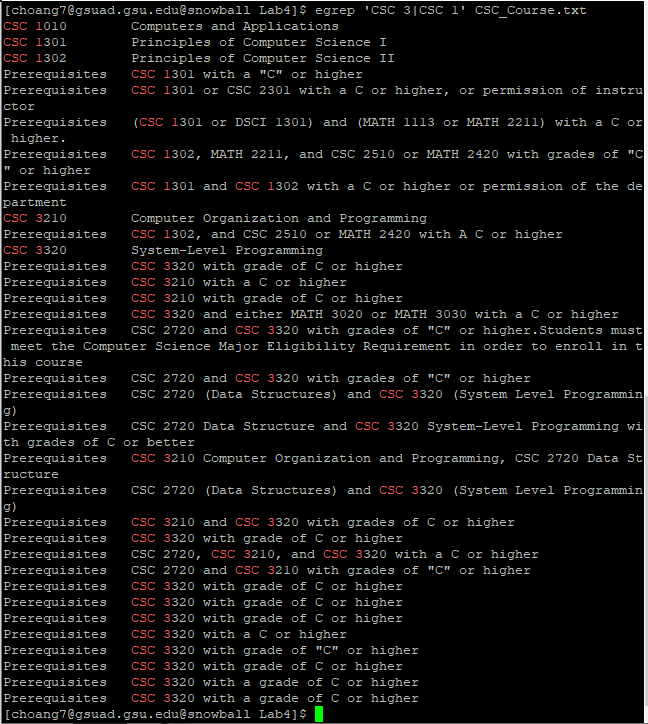
6)



The ‘-E’ in this grep command indicates that the pattern we need to file as an Extended Regular Expression (ERE), and ERE does recognize ‘|’ as a special character indicating the matched string will the preceding element or the following element of the ‘|’ character.

This command output the lines containing either the string “CSC 3” or “CSC 1” in the file ‘CSC\_Course.txt’.

7)



egrep uses extended regular expression set for pattern (the same as grep -E), so the special character here does not lose its meaning. Therefore, this command output all the lines containing either the string “CSC 3” or “CSC 1” in the file ‘CSC\_Course.txt’.

8)



N/A.

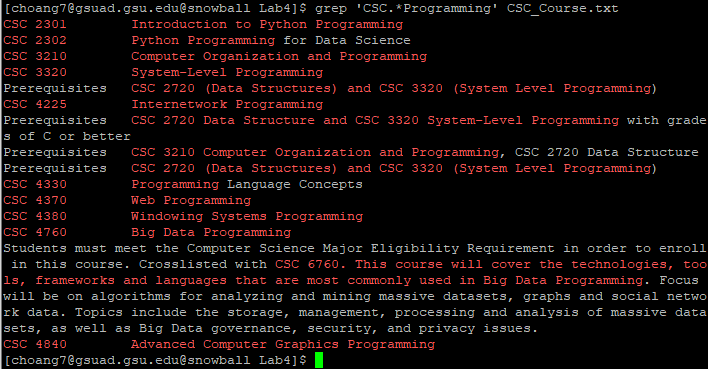
fgrep is used to search fixed-character strings in a file instead of regular expression. This command does not output anything because the file does not contain the string “3.000 Credit hours”.

9)



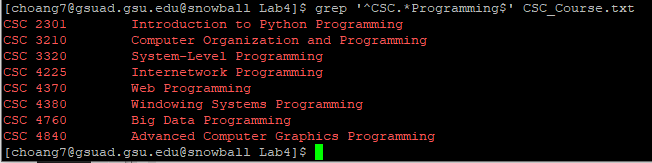
N/A. This command doesn’t output anything simply for the same reason as stated in question 8, but we can also see that an option for fgrep is included in here, -x, which means it will print only lines matched entirely.

10)



This command indicates that the matched pattern in the file ‘CSC\_Course.txt’ must start with ‘CSC’ and end with ‘Programming’, and it picks up any characters except a line break in between.

11)



This command output all the lines that must start with ‘CSC’ and the end of that line must be ‘Programming’.

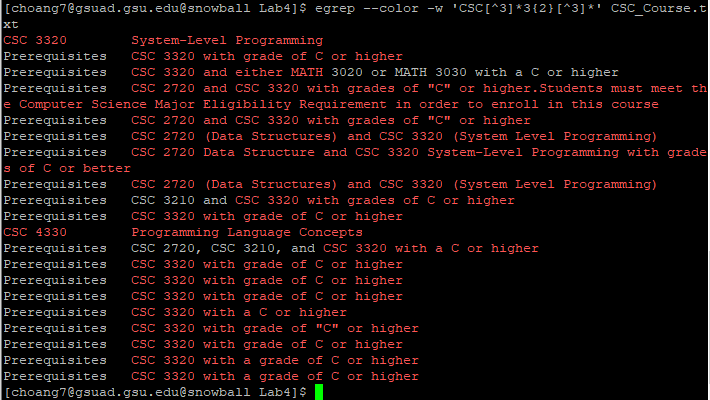
12)



N/A

The command has the option ‘--color', which is used to modify the color-highlight of the matched string in the output. The matched string for this command should start with ‘CSC’ and end with ‘33’, and there should be no ‘3’ character in between, e.g., CSC1233.

13)



egrep indicates the looking for pattern is written in extended regular expression.

‘--color' is used to modify the color-highlight option in the output.

The ‘-w’ option makes the command matches only pattern that are presented in the whole word, instead of the pattern that is found as a substring.

The matched pattern should start with ‘CSC’. After that, the following character can be any other than ‘3’ or none. Next it must be a ‘33’ and end with any character excepts for ‘3’.

This command output all the line containing the pattern with option mentioned above.

14)



N/A

grep indicates the pattern is in basic regular expression, which means it doesn’t recognize ‘+’ as a special character.

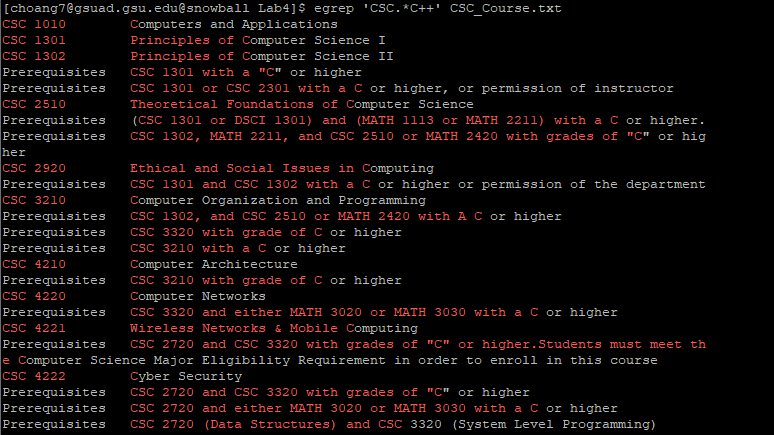
15)



N/A

This command uses ‘egrep’ meaning it uses extended regular expression, there for it will treat ‘\+\+’ as literally. Since there are no pattern that end with ‘C\+\+’ in the file CSC\_Course.txt, it doesn’t output anything.

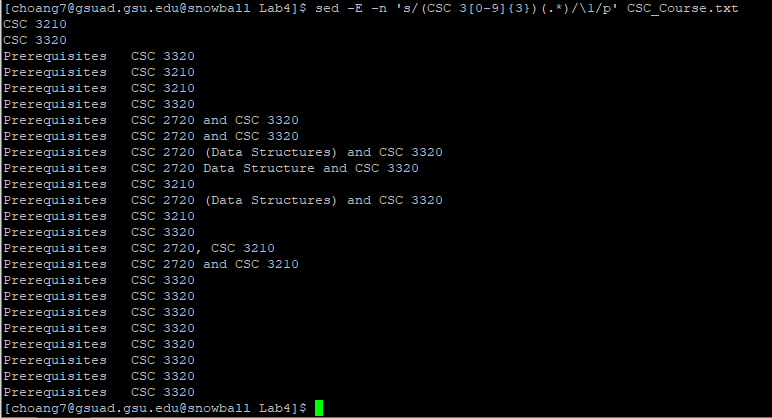
16)



The command outputs all the lines containing the matched string that start with ‘CSC’ and end with a ‘C’ with one or more occurrence as it treat ‘+’ as a quantifier (special character).

Optional Part:

1)



sed performs an action on all lines that match a particular condition.

-E (option): use extended regular expressions rather than basic regular expressions.

-n (option): suppress the automatic printing of pattern space.

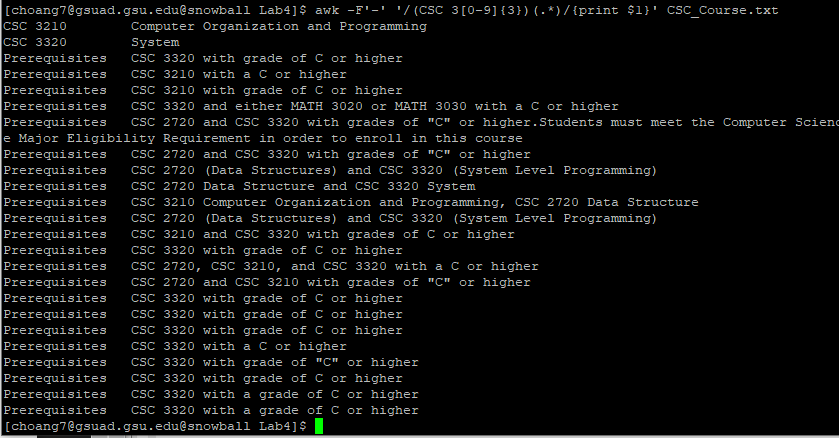
Capturing Group 1 is string that starts with ‘CSC 3’ and ends with any 3 digits.

Capturing Group 2 is any character excepts a line break with infinite occurrence or none.

s/ (action): substitute group 1 with itself (group 1 – ‘\1’).

/p (action): print all lines containing the pattern above in the file ‘CSC\_Course.txt’.

2.



-F’-‘ (option): uses ‘-‘ as the field separator

$1: refer to the first field.

printf $1: print out the first field, each field is separated by ‘-‘, in the file ‘CSC\_Course.txt’.

3.



N/A

The option is the same as in the command in question 1.

Capturing Group 1: string that starts with CSC and end with any 4 digits.

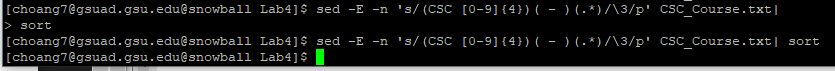
Capturing Group 2: ‘ - '.

Capturing Group 3: Any character excepts line break with infinite occurrence or none.

s/: substitute group 1 with group 3 (‘\3’).

/p: print all lines containing the pattern above in the file ‘CSC\_Course.txt’.

4)



N/A

But the command is the same as explained in question 3 excepts for addition ‘sort’ command, which is used to sort the context of the supposed output in lexicographical order.