CSC3320 System Level Programming

Lab Assignment 4 - Part 2 (Out of lab)

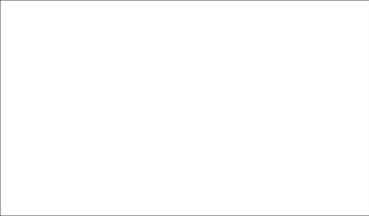
Instructor: Bello Babatunde

Due at 11:59 pm on Wednesday, Sep. 22, 2021

Purpose: Practices on the grep, fgrep, egrep, sed , awk, and sort commands for text processing.

Note: Please follow the instructions below, and write a report by answering the questions and upload the report (named as Lab4\_P2\_FirstNameLastName.pdf or .doc) to the Google Classroom Out of Lab Assignment folder 

Please add the lab assignment NUMBER and your NAME at the top of your file sheet. The following table is from Wikipedia. It shows the eleven highest mountains in Georgia.

Brasstown Bald, (summit),4784,feet,Union County Rabun Bald, (summit),4696,feet,Rabun County 

Dick's Knob, (summit),4620,feet,Rabun County

Hightower Bald, (summit),4568,feet,Towns County

Wolfpen Ridge, (ridge high point),4561,feet,Towns and Union Counties

Blood Mountain, (summit),4458,feet,Union County Tray Mountain, (summit), 4430,feet,Towns County

Grassy Ridge, (ridge high point),4420,feet,Rabun County Slaughter Mountain, (summit),4338,feet,Union County Double Spring Knob, (summit),4280,feet,Rabun County

Coosa Bald, (summit),4280,feet,Union County

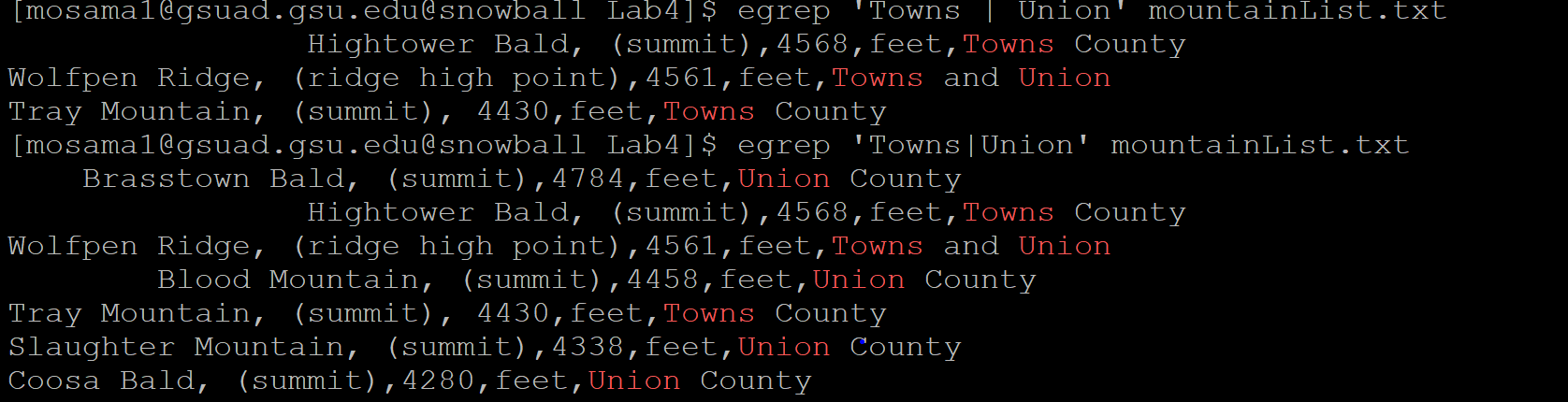
In above table, each line contains 5 fields separated by comma. Open your terminal and connect to snowball server. After that, go to directory Lab4 (cd ~/Lab4) and please download the file " mountainList.txt" by the following

command (internet access required):

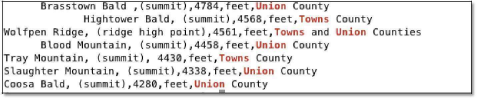
cp /home/bbello1/Public/mountainList.txt mountainList.txt Be sure it succeeds using “ls” to see the file name “mountainList.txt” listed.

1) Use grep to print all lines where the mountains are at Towns or Union

County.

Ans1) couldn’t find the output using grep so used egrep to look for Towns | Union. Used the command $egrep ‘Towns | Union’ mountainList.txt

Sample Output

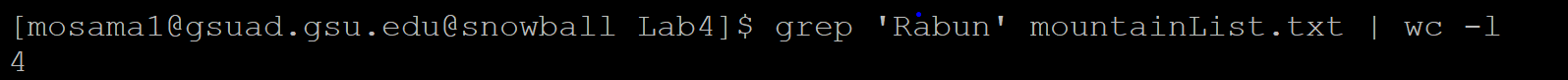


2) Use wc and grep to count the number of mountains located at Rabun

County. Hint: please use

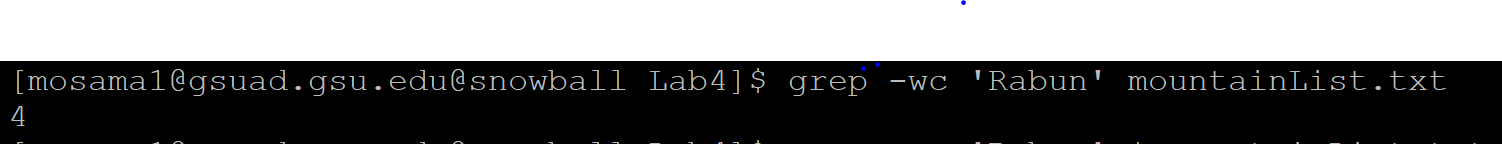
pipe | .

Ans) both commands worked, $grep ‘Rabun’ mountainList.txt | wc -l



Or

$grep -wc ‘Rabun’ mountainList.txt

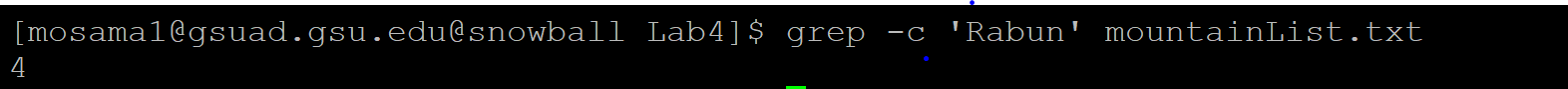


Sample Output

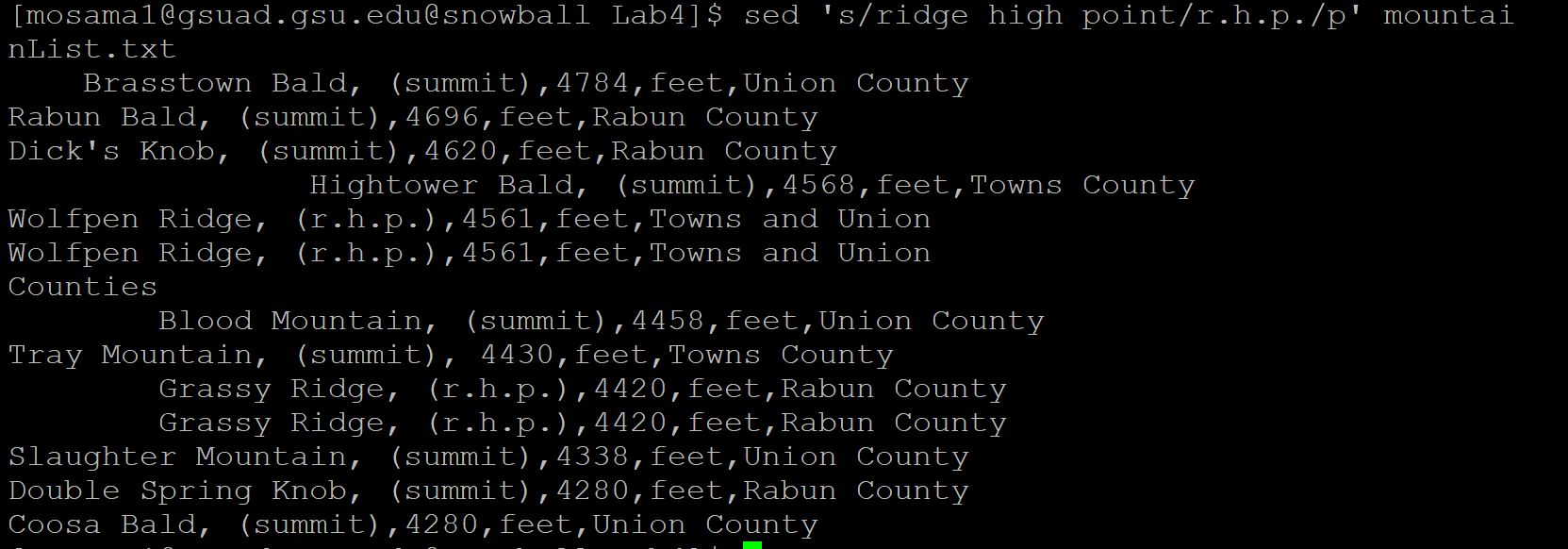
3) Finish task 2) by using only grep.

Hint: open the manual page of grep, and check -c option.

$grep – c ‘Rabun’ mountainList.txt

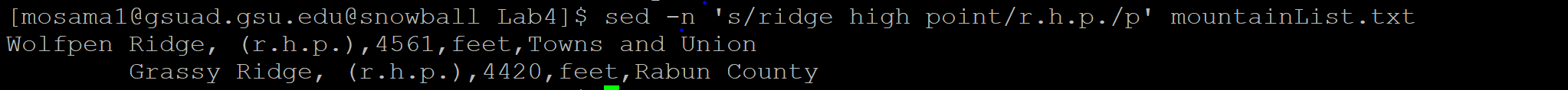


4) A. Type command sed ‘s/ridge high point/r.h.p./p’ mountainList.txt  and execute it. Then attach a screenshot of the output.

$‘s/ridge high point/r.h.p./p mountainList.txt  


B.Type command sed -n ‘s/ridge high point/r.h.p./p’ mountainList.txt and  execute it. Then attach a screenshot of the output.

$sed -n ‘s/ridge high point/r.h.p./p’ mountainList.txt



C. Open the manual page of sed and describe what does –n do in sed?

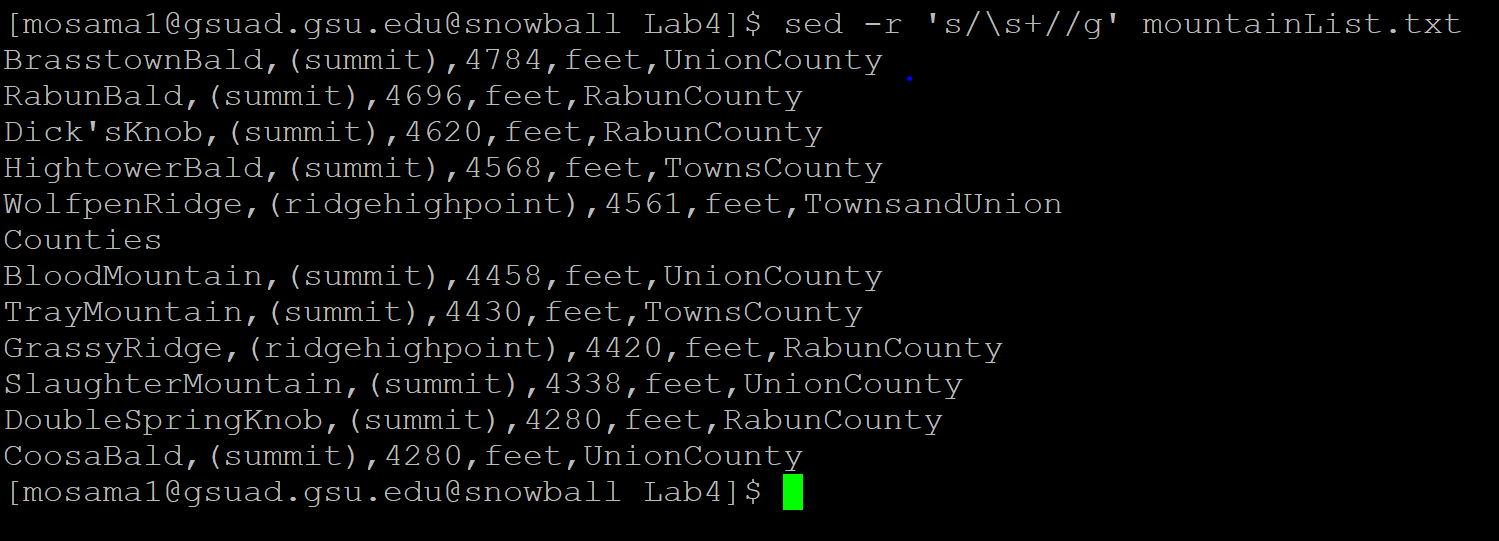
Ans 4C) -n, --quiet, --silent

suppress automatic printing of pattern space  
  
  
D.Describe what does the sed command in (B) do?

Ans 4D) Command B replaces the ridge high point to r.h.p. and the replaced are duplicated by p and by -n we are supressing automatic printing of pattern space and we are only printing the replaced and duplicated ones.

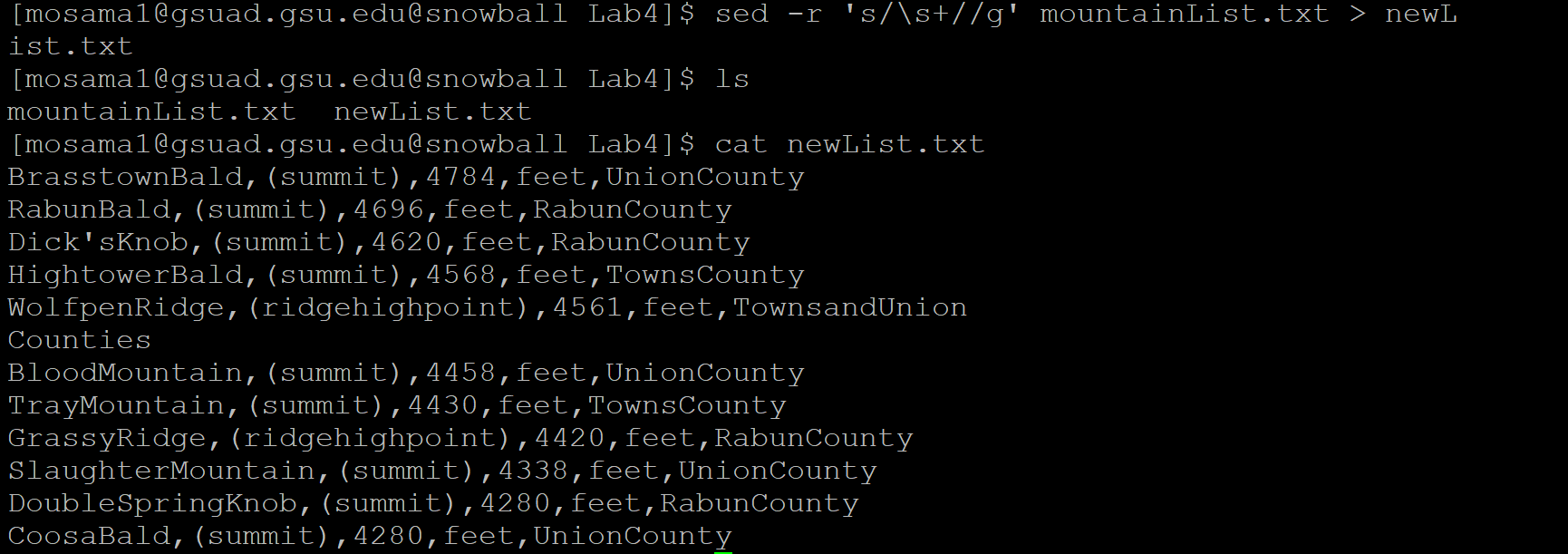
5) Use sed to remove the leading spaces in "mountainList.txt" and print out the processed lines.

$ sed -r ‘s/\s+//g’ mountainList.txt



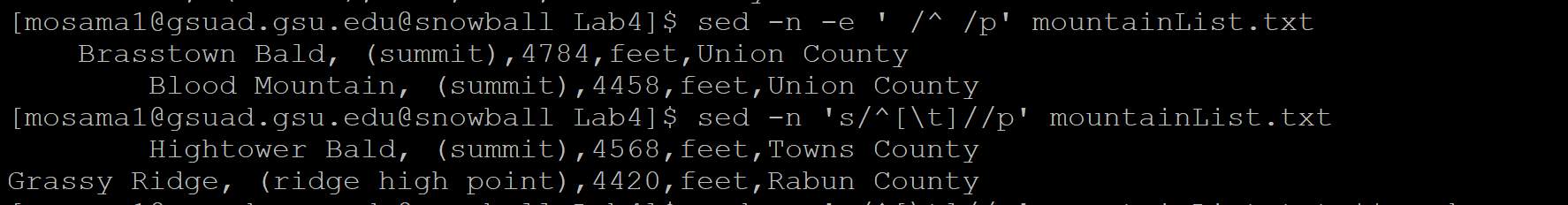
6) Finish task 5) and save the output to file "newList.txt".

$ sed -r ‘s/\s+//g’ mountainList.txt > newList.txt

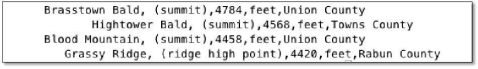


7) Use sed to list the lines beginning with white spaces in "mountainList.txt".

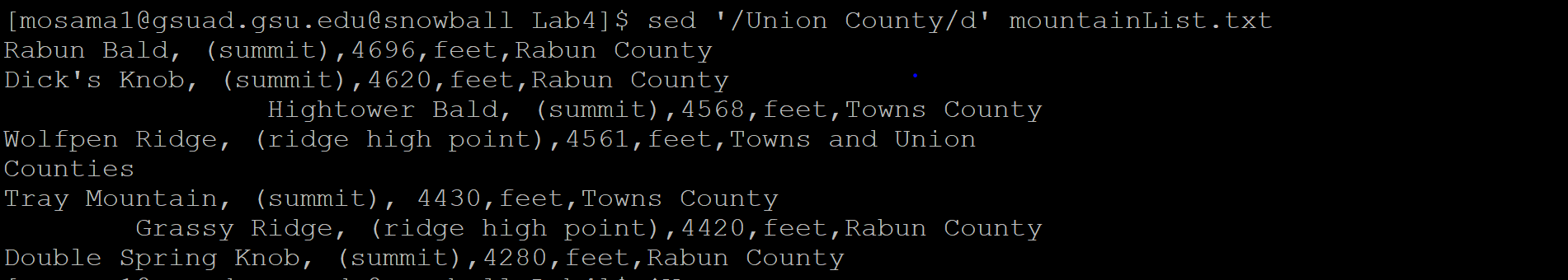
$ sed -n -e ‘/^ /p’ mountainList.txt or sed -n ‘s/^[\t]//p’ mountainList.txt



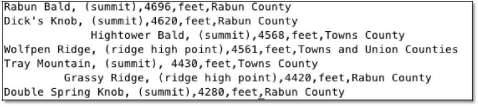
Sample Output



8) Use sed to delete the lines where the mountains are only at Union County in "mountainList.txt".

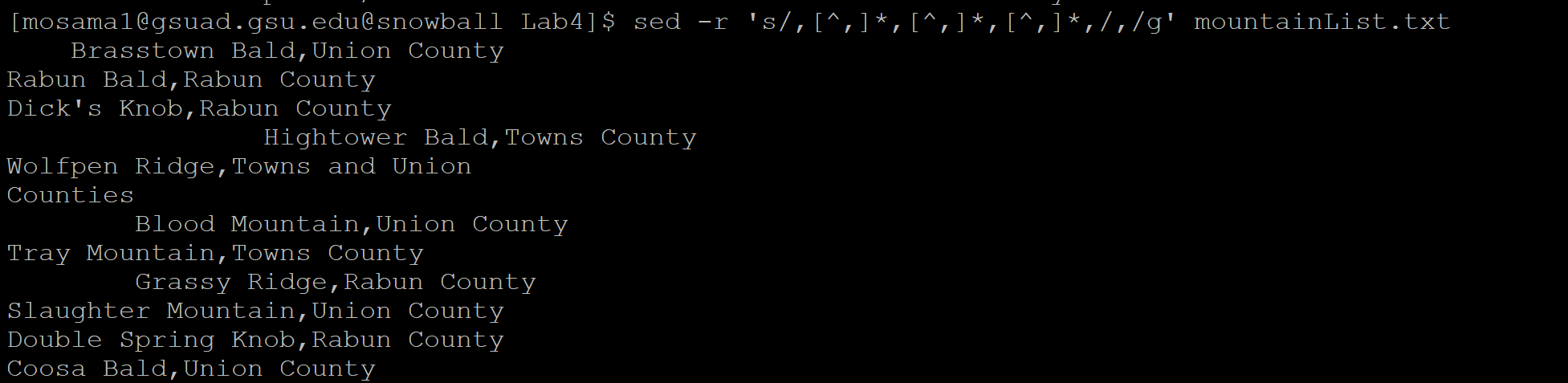
$ Sed ‘/Union County/d’ mountainList.txt

Sample Output

9) Use sed to remove the middle three fields in each line of

"mountainList.txt". Hint: Think about the meaning of regex '[^,]'

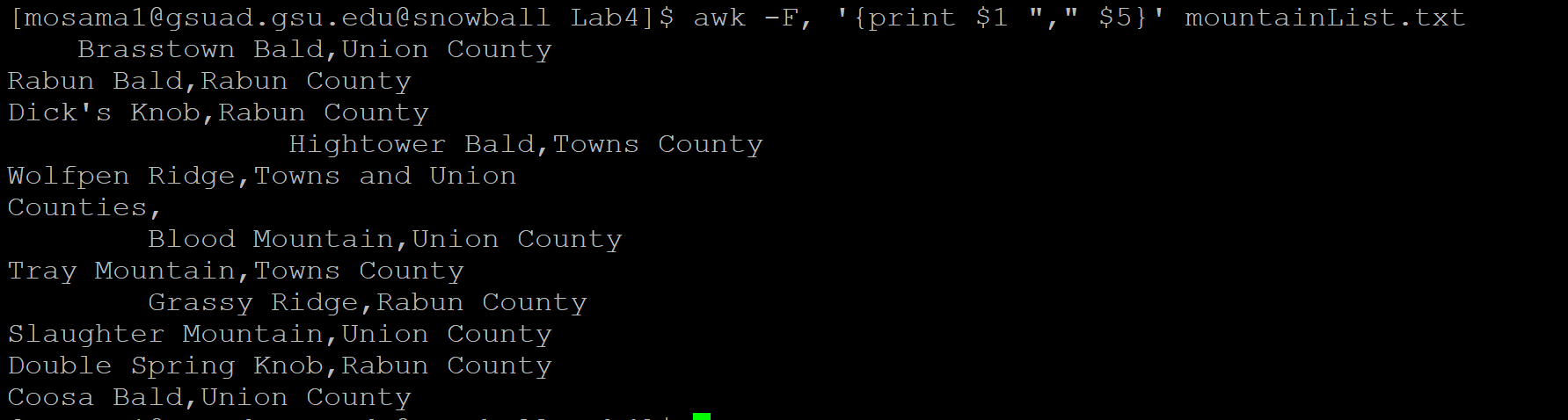
sed -r 's/,([^,]\*){3},/,/g' public/mountainList.txt

$ sed -r 's/,[^,]\*,[^,]\*,[^,]\*,/,/g' mountainList.txt 



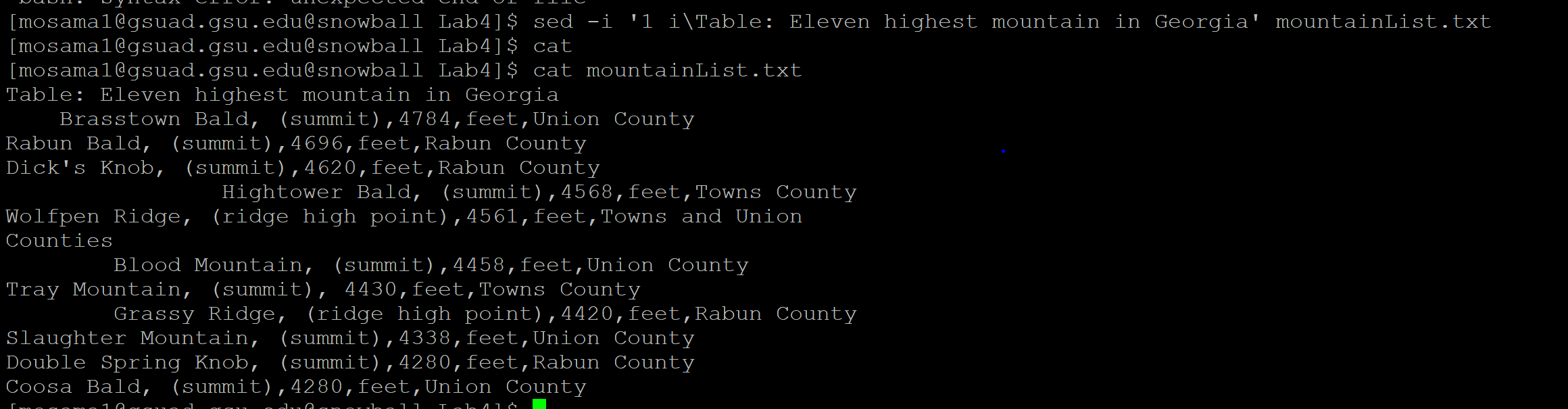
10) Use awk to finish task 9).

awk -F, '{print $1 "," $5}' mountainList.txt



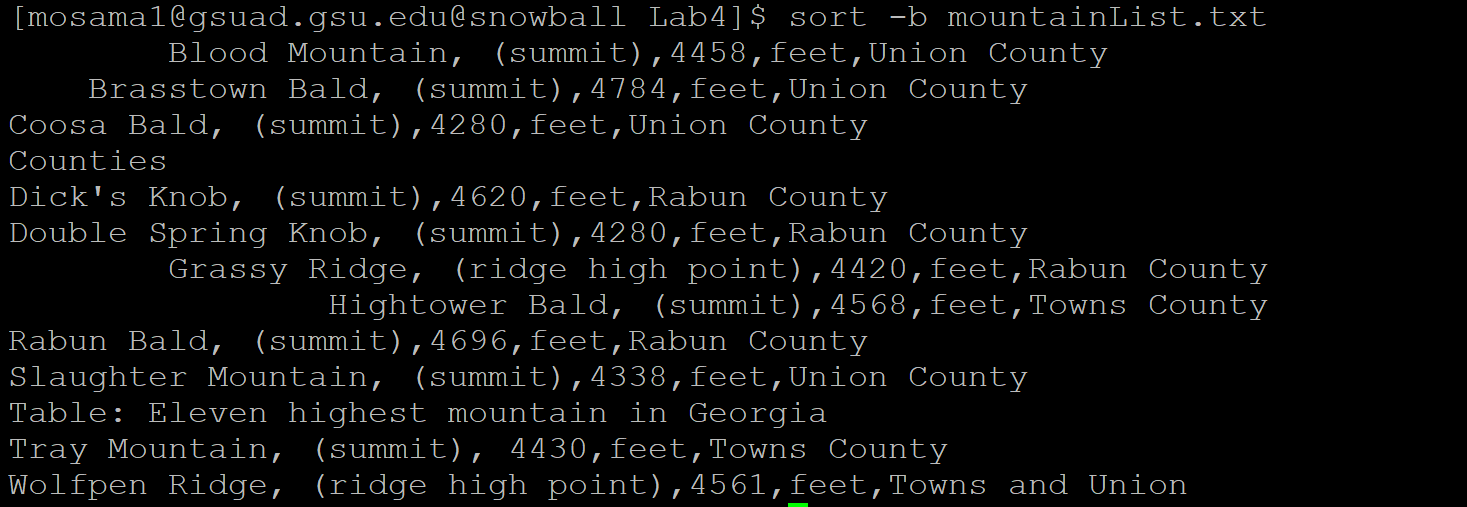
11) Use sed to insert a new line “Table: Eleven highest mountains in Georgia” at the beginning of "mountainList.txt".

sed -i '1 i\Table: Eleven highest mountain in Georgia' mountainList.txt



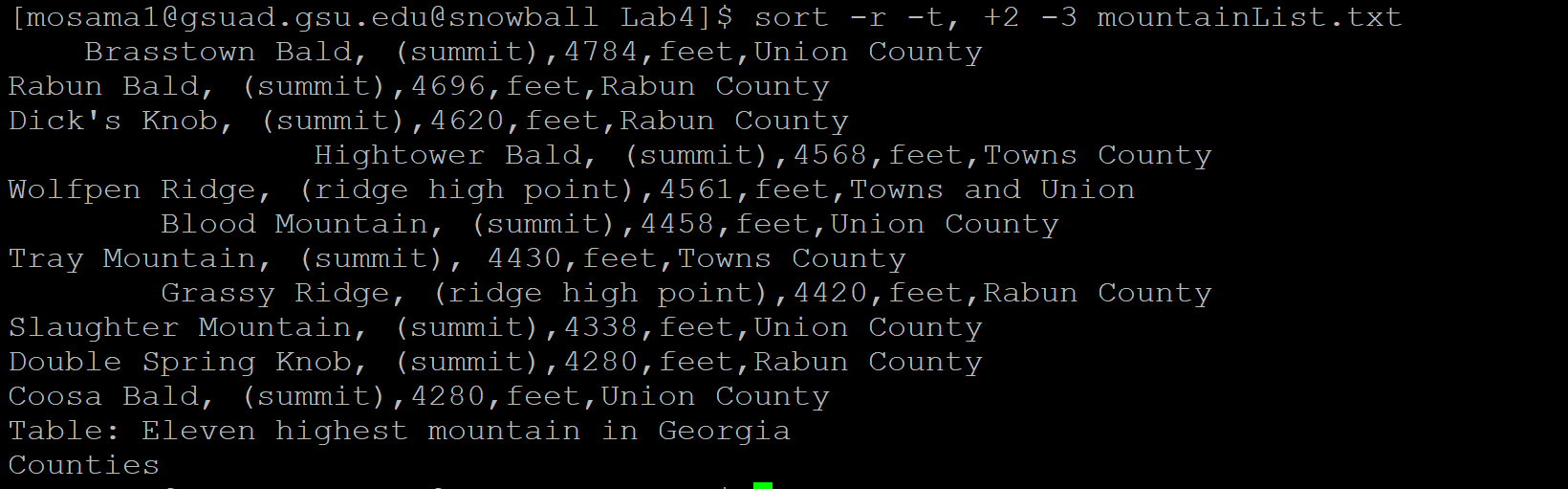
12) Use sort to print out the sorted lines in alphabetical order according to the names of mountains.

$ sort -b mountainList.txt



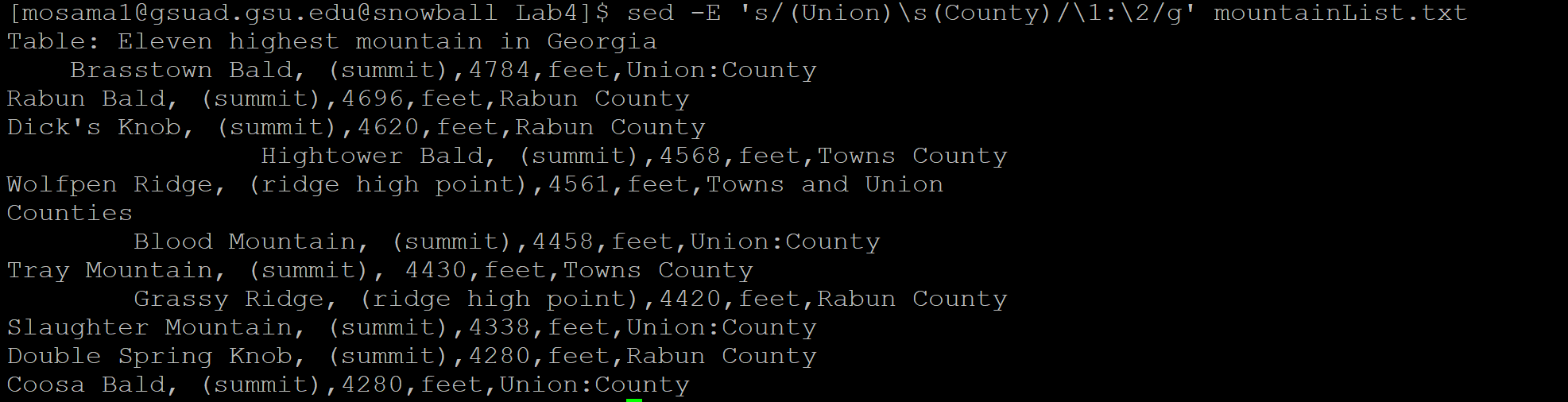
13) Use sort to print out the sorted lines in descending order according to the height of mountains.

$ sort -r -t, +2 -3 mountainList.txt



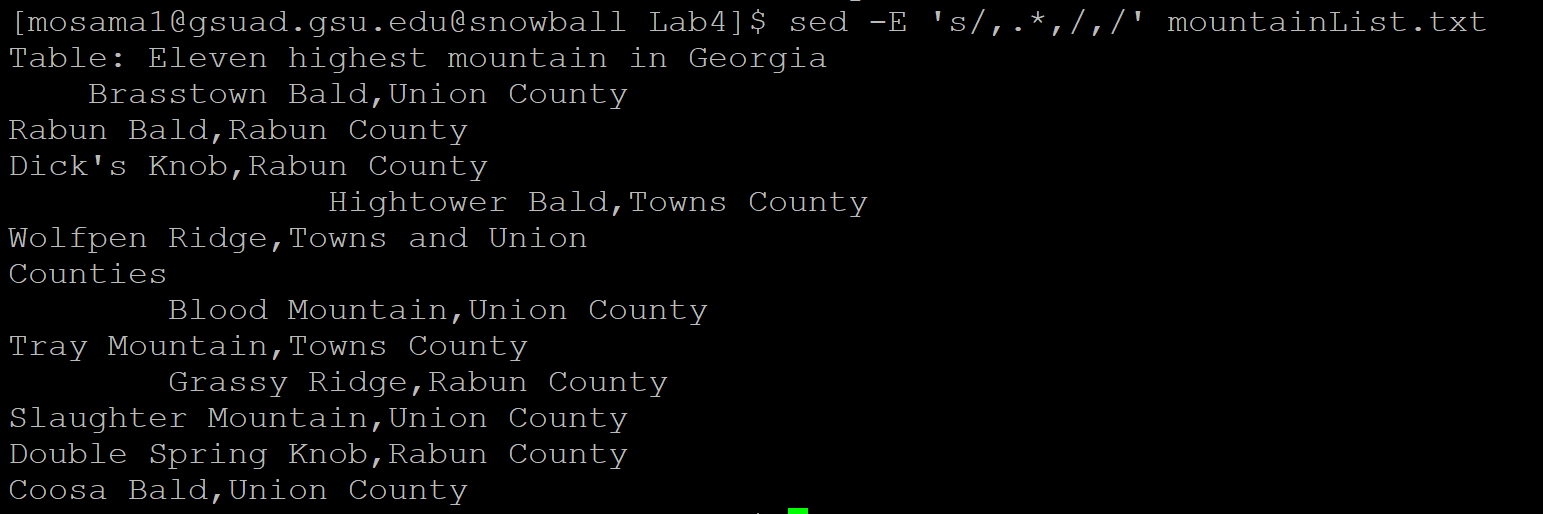
14) “When a pattern groups all or part of its content into a pair of parentheses, it captures that content and stores it temporarily in memory. You can reuse that content if you wish by using a back-reference, in the form:\1 or $1, where \1 or $1 reference the first captured group” (Refer to [1]). For example, the following command add a colon between Union and County sed -E ‘s/(Union)\s(County)/\1:\2/g’ mountainList.txt

Attach a screenshot of the output of the above sed command.



15) Now can you write a command to finish task 9) using sed with backreference?

Ans15) sed -E 's/,.\*,/,/' mountainList.txt



Useful Links:

[1] Introducing Regular Expression - Capturing Groups and Backreferences https://www.safaribooksonline.com/library/view/introducingregular expressions/9781449338879/ch04.html

[2] Drew's grep tutorial http://www.uccs.edu/~ahitchco/grep/

[3] Grep and Regular Expressions!

http://ryanstutorials.net/linuxtutorial/grep.php [4] Web

Scraping with Regular Expressions

https://www.datascraping.co/doc/22/regular-expression