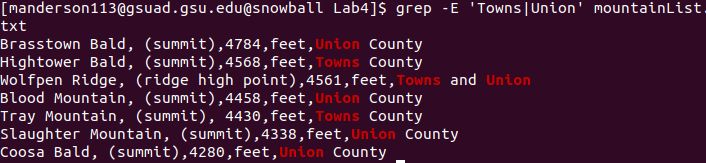
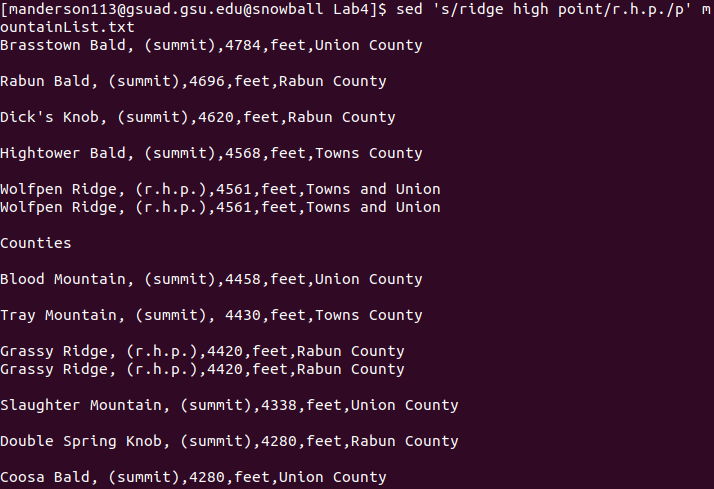
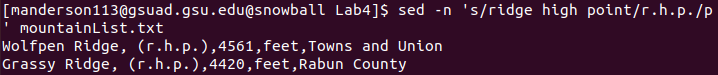
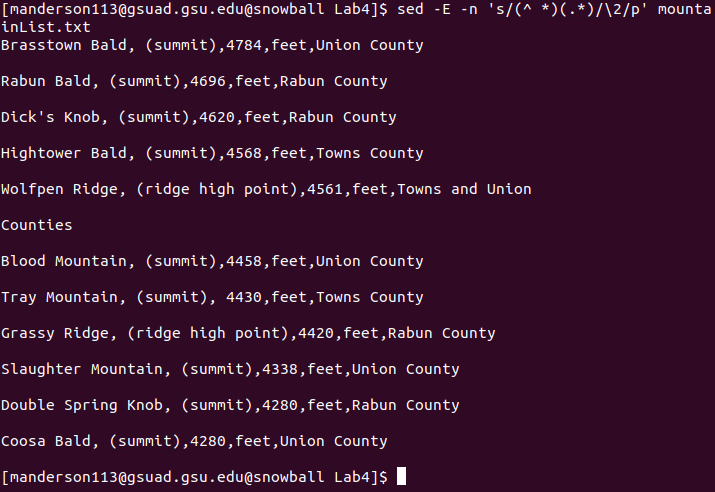
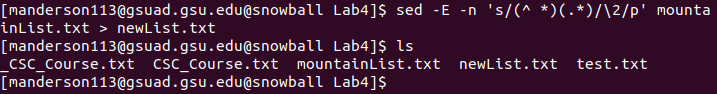
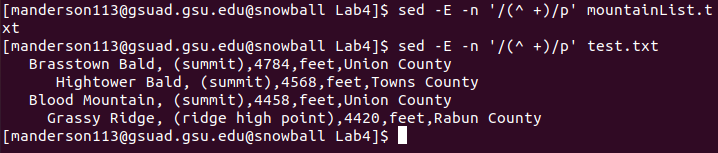
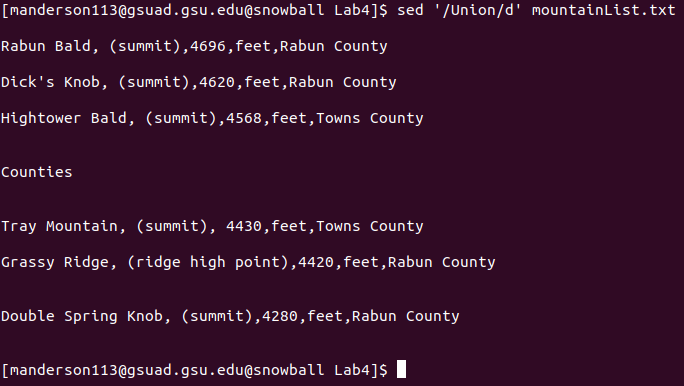
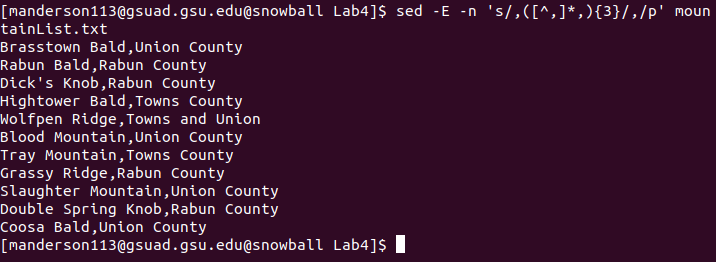
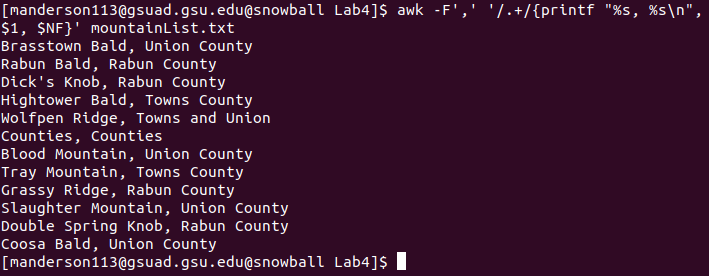
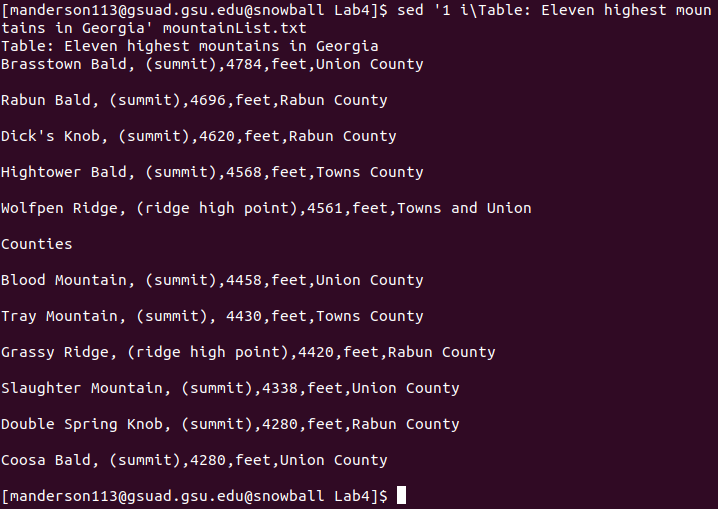
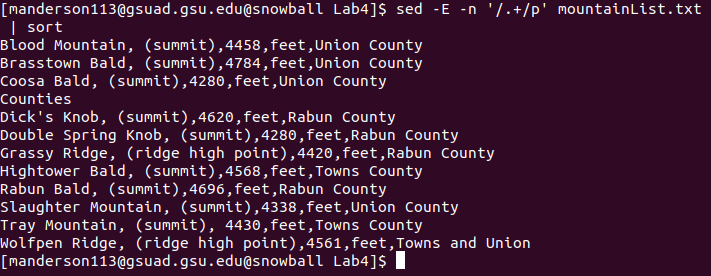
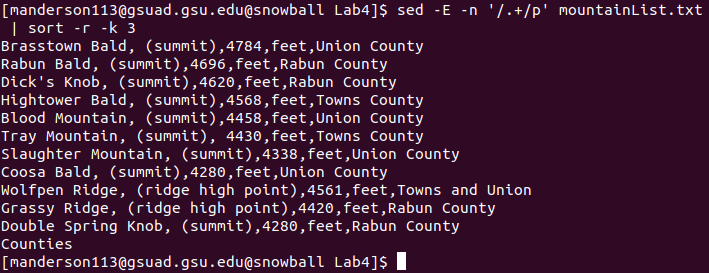
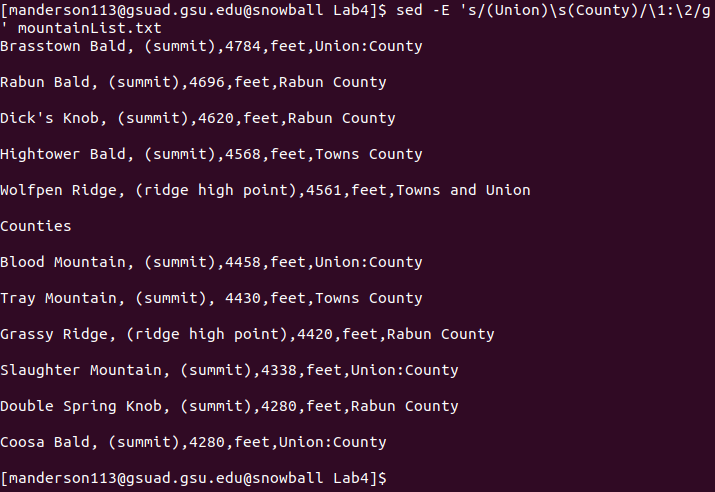
1. egrep ‘Towns|Union’ mountainList.txt
   * 1. 
2. grep ‘Rabun’ mountainList.txt | wc -l
   * 1. 
3. grep -c ‘Rabun’ mountainList.txt
   * 1. 
   1. prints every line, but replaces ‘ridge high point’ with ‘r.h.p.’ and double-prints those lines
      1. 
   2. prints all lines containing ‘ridge high point’ but replacing it with ‘r.h.p.’
      1. 
   3. suppress automatic printing of pattern space
   4. finds all lines that contain ‘ridge high point’, replaces ‘ridge high point’ with ‘r.h.p.’ then prints them, but without printing every line in the file
4. sed -E -n ‘s/(^ \*)(.\*)/\2/p’ mountainList.txt
   * 1. 
5. sed -E -n ‘s/(^ \*)(.\*)/\2/p’ mountainList.txt > newList.txt
   * 1. 
6. sed -E -n ‘/^ +)/p’ mountainList.txt
   * 1. 
     2. I made a copy of mountainList.txt called test.txt that still has the empty lines between entries, but includes the leading spaces assumed to be in the original file that are absent.
7. sed ‘/Union/d’ mountainList.txt
   * 1. 
8. sed -E -n ‘s/,([^,]\*,){3}/,/p’ mountainList.txt
   * 1. 
9. awk -F’,’ ‘/.+/{printf “%s, %s\n”, $1, $NF}’ mountainList.txt
   * 1. 
10. sed ‘1 i\Table: Eleven highest mountains in Georgia’ mountainList.txt
    * 1. 
11. sed -E -n ‘/.+/p’ mountainList.txt | sort
    * 1. 
12. sed -E -n ‘/.+/p’ mountainList.txt | sort -r -k 3
    * 1. 
13. Replaces ‘Union<whitespace>County’ with ‘Union:Country’
    * 1. 
14. sed -E -n ‘s/ ([^,]\*),([^,]\*,){3}(.\*)/\1, \3/p’ mountainList.txt
    * 1. 