sed

- stream oriented, non interactive, text editor
- Original input file is unchanged (unless with -i option); Results sent to standard output (can be redirected to a file)
- Single quotes (' ') are used to delimit the command being executed
- **Delete:**
    - sed ‘ADDRESSd’ filename (sed '2,5!d' fruits)
    - sed ‘/PATTERN/d’ filename (sed '/^$/d' fruits)
- **Substitute:**
    - sed ‘address(es) s/pattern/replacement/[flags]’ filename
    - g - global, p - print contents, I - case insensitive
    - sed 's/wolf/fox/3gI' bigfile - replaces the third and more occurrence of the word in a line
    - sed '3,$ s/wolf/fox/' bigfile - replaces the string from 3rd till end last line. Note the file has blank lines also
    - sed '/^Later/ s/wolf/fox/gI' bigfile - replace in line starting with Later; notice regex patterns are put inside
    - echo "Welcome To The Course CS108" | sed 's/\(\b[A-Z]\)/(\1)/g' - \b does word boundary match,\1 is backreference; Without escaping i.e. \(and \), (and) would be treated as literal characters instead of grouping operators.
- **Print:**
    - sed 'ADDRESSp’ filename
    - sed '/PATTERN/p' filename
    - sed -n '/scream/p' bigfile, sed '1p' bigfile (-n will only print output, otherwise prints each line after applying editing commands)
- **Append:**
    - Places text after the current line in pattern space, [address/pattern]a file
    - sed '2a tomato' fruits
- **Insert:**
    - Places text before the current line in pattern space, [address/pattern]a file
    - sed '$i tomato' fruits
- **Replace:**
    - sed '/mango/c aam ' fruits
    - sed '2c aam ' fruits
- **quit:**
    - causes sed to stop reading new input lines once the line matches the pattern/address, i.e the script terminates.
    - sed '5q' fruits - prints first 5 lines and stops

- **Multiple commands:**
    - Semicolon (sed 's/mango/aam/; s/banana/kela/;' fruits)
    - Precede instruction by -e (sed -e 's/mango/aam/' -e 's/banana/kela/' fruits)