COMP 302 System Programming, Spring 2022
Instructor: Zafer Aydın
Lab Assignment 8

Introduction

In this lab you will explore text processing commands. Submit your solutions to the questions below in a text file (i.e. answers.txt). Then name your file in name_surname.txt format and upload it to Canvas.

Questions

Prepare the following file that contains food (first two fields), cost (third field) and calories (fourth field) in this order. The blanks in the file must be SPACEs not TABs. The number of space characters is not too important as long as the cost and calorie fields are aligned as shown below. LC_ALL local variable must be set to C. You can enter the following commands to set the LC_ALL variable and prepare the file

```
export LC_ALL=C

cat > food.txt

tuna salad       2.25   100

rare hamburger  2.50   500

french fries    0.75   625

tea bag          .50     0

green tea        .50     1

fiji apple      2.00    52

apple pie       1.75  1500

milk shake      3.3    200

cow milk        3       42

potato chips    0.50    80
```

Then press Ctrl-D to exit from `cat`.

1. Provide a single command using `sort` to display the lines of `food.txt` in alphabetical order.

2. Provide a single command using `sort` to display the lines of `food.txt` in reverse alphabetical order.

3. Provide a single command using `sort` and `--key=` option (or `-k` option) to display the lines of `food.txt` sorted by the second word in the name of the food.

4. Provide a single command using `sort` to display the lines of `food.txt` sorted by cost. Use `--ignore-leading-blanks` (or `-b`) option to sort the cost properly.

5. What happens if you do not use the `--ignore-leading-blanks` (or `-b`) option in the previous question? Explain the reason.

6. Provide a single command using `sort` and `--numeric-sort` (or `-n`) option to produce a file called `calo.txt` that holds the lines of `food.txt` ordered by calories so that the food with the most calories appears at the top of the list.

7. Provide a single command using `sort` and `--key=` option(s) to first sort with respect to the cost (as real numbers) and then with respect to the calories so that the calories are sorted with respect to numbers in descending order. Use `-g` option to sort with respect to cost as real numbers.

COMP 302 System Programming, Spring 2022
Instructor: Zafer Aydın
Lab Assignment 8

8. Provide a single-line command that first uses `du` with `-s` option to show the number of bytes occupied by each file and folder under the `/usr/bin/` directory and then sort these using the `sort` command with respect to size in descending order. Hint: Pipe the output of `du` to input of `sort`.

9. Repeat question 8 this time using `ls -l` to display the files and folders under `/usr/bin/` directory and sort these with respect to size using `sort` command together with the `-k` option. Hint: Pipe the output of `ls -l` to input of `sort` and sort with respect to the 5th field of the output of `ls -l`.

10. Use the `cut` command with `-f` option to display the 4th field column of `food.txt` (i.e. the calorie column only). Hint: First use `cat` to send the contents of `food.txt` as input to `tr -s ' '` using pipe and then using another pipe send the output of `tr` as input to your `cut` command. Use `-d ' '` option in your cut command.

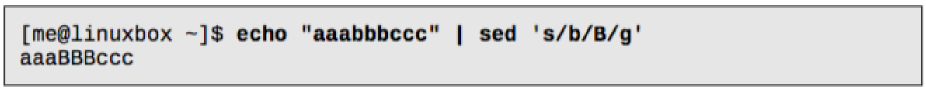
11. Use the `cut` command with `-c` option to display the 1st character of each food name in `food.txt`.

12. Provide a single-line command using `cut` with `-f` option to save the 1st and 2nd field columns into `names.txt`. Use the `-d` option to specify the delimiter as space. Provide a range of fields into `-f` option as `-f 1-2`.

13. Provide a single-line command using `cut` with `-f` option to save the 3rd field column (i.e. the cost column) into `costs.txt`. Use the `-d` option to specify the delimiter as space. Hint: First use `cat` to send the contents of `food.txt` as input to `tr -s ' '` using pipe and then using another pipe send the output of `tr` as input to your `cut` command. Use `-d ' '` option in your cut command.

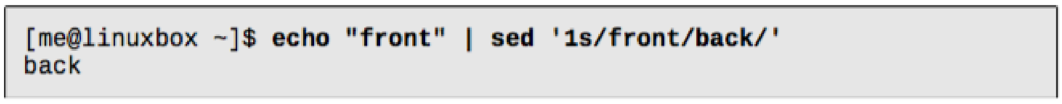
14. Combine the columns in `names.txt` and `costs.txt` using `paste` into a single file called `food_2.txt`, which includes food names and cost information only.

15. Provide a single-line command that uses `sed` and replaces all the food names `tea` with `coffee` in `food.txt` using the `s` option and `g` option. Save the output to `food_3.txt`. Hint: First use `cat` to send the contents of `food.txt` to input of `sed` using pipe. An example `sed` usage for replacing all occurrences of an expression is given below



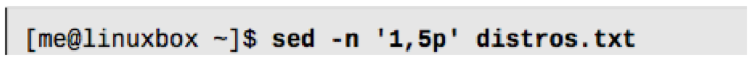
```
[me@linuxbox ~]$ echo "aaabbbccc" | sed 's/b/B/g'
aaaBBBccc
```

16. Provide a single-line command that uses `sed` and replaces the food name `tea` in the 5th line of `food.txt` only with `coffee` using the `s` option. Save the output to `food_4.txt`. Hint: First use `cat` to send the contents of `food.txt` to input of `sed` using pipe. Use the line number in front of the substitute expression as in the following example



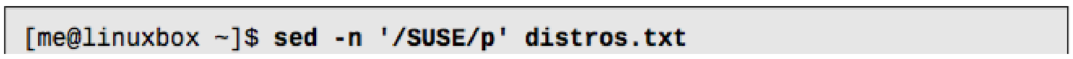
```
[me@linuxbox ~]$ echo "front" | sed '1s/front/back/'
back
```

17. Provide a single-line command that uses `sed` and displays the first 5 lines of `food.txt` using the `p` option and `-n` option. An example usage is given below



```
[me@linuxbox ~]$ sed -n '1,5p' distros.txt
```

18. Provide a single-line command that uses `sed` and displays all the occurrences of `apple` in `food.txt` using the `p` option and `-n` option. An example usage is given below



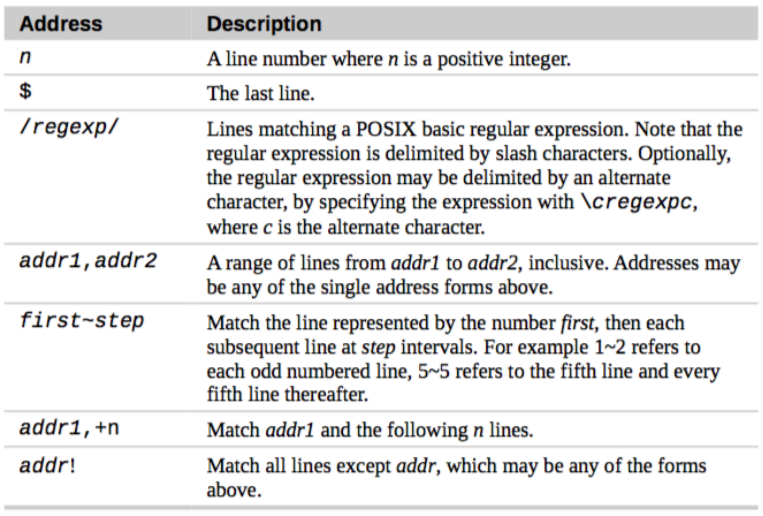
```
[me@linuxbox ~]$ sed -n '/SUSE/p' distros.txt
```

19. What happens if you do not include the `-n` option in question 18? Explain briefly.

COMP 302 System Programming, Spring 2022
Instructor: Zafer Aydın
Lab Assignment 8

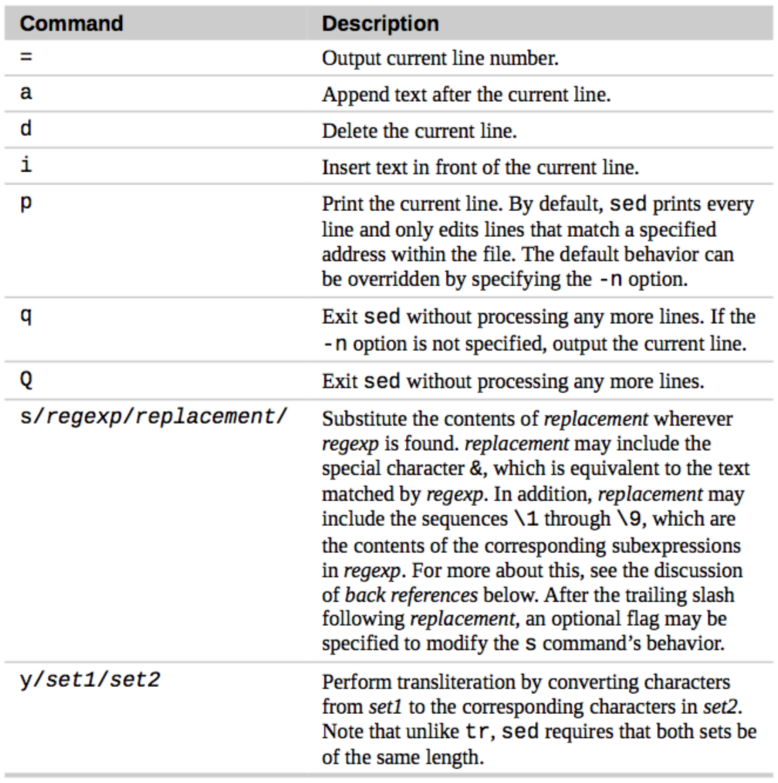
20. Provide a single-line command that uses `sed` and deletes all the lines that contain `tea` in `food.txt` using the `d` option. Save the output to `food_5.txt`

Table 20-7: sed Address Notation



Address	Description
`n`	A line number where *n* is a positive integer.
`$`	The last line.
`/`*`regexp`*`/`	Lines matching a POSIX basic regular expression. Note that the regular expression is delimited by slash characters. Optionally, the regular expression may be delimited by an alternate character, by specifying the expression with `\`*`cregexpc`*, where *c* is the alternate character.
`addr1``,`*`addr2`*	A range of lines from *addr1* to *addr2*, inclusive. Addresses may be any of the single address forms above.
`first``~`*`step`*	Match the line represented by the number *first*, then each subsequent line at *step* intervals. For example 1~2 refers to each odd numbered line, 5~5 refers to the fifth line and every fifth line thereafter.
`addr1``,+n`	Match *addr1* and the following *n* lines.
`addr``!`	Match all lines except *addr*, which may be any of the forms above.

Table 20-8: sed Basic Editing Commands



Command	Description
`=`	Output current line number.
`a`	Append text after the current line.
`d`	Delete the current line.
`i`	Insert text in front of the current line.
`p`	Print the current line. By default, `sed` prints every line and only edits lines that match a specified address within the file. The default behavior can be overridden by specifying the `-n` option.
`q`	Exit `sed` without processing any more lines. If the `-n` option is not specified, output the current line.
`Q`	Exit `sed` without processing any more lines.
`s/`*`regexp`*`/`*`replacement`*`/`	Substitute the contents of *replacement* wherever *regexp* is found. *replacement* may include the special character `&`, which is equivalent to the text matched by *regexp*. In addition, *replacement* may include the sequences `\1` through `\9`, which are the contents of the corresponding subexpressions in *regexp*. For more about this, see the discussion of *back references* below. After the trailing slash following *replacement*, an optional flag may be specified to modify the `s` command's behavior.
`y/`*`set1`*`/`*`set2`*	Perform transliteration by converting characters from *set1* to the corresponding characters in *set2*. Note that unlike `tr`, `sed` requires that both sets be of the same length.