

Homework 5

April 8, 2020

```
[1]: cd /
```

```
[2]: pwd
```

```
/
```

```
[3]: ls -l
```

```
total 4194404
drwxr-xr-x  2 root root      4096 Apr  8 19:31 bin
drwxr-xr-x  4 root root      4096 Apr  8 19:32 boot
drwxr-xr-x  2 root root      4096 Mar  6 20:32 cdrom
drwxr-xr-x 20 root root     4000 Mar 24 22:10 dev
drwxr-xr-x 109 root root     4096 Apr  8 19:32 etc
drwxr-xr-x 33 root root     4096 Apr  8 19:35 home
lrwxrwxrwx  1 root root        33 Apr  7 06:52 initrd.img ->
boot/initrd.img-4.15.0-96-generic
lrwxrwxrwx  1 root root        33 Apr  7 06:52 initrd.img.old ->
boot/initrd.img-4.15.0-91-generic
drwxr-xr-x 22 root root      4096 Mar  6 20:55 lib
drwxr-xr-x  2 root root      4096 Feb  3 18:22 lib64
drwx----- 2 root root    16384 Mar  6 20:31 lost+found
drwxr-xr-x  2 root root      4096 Feb  3 18:22 media
drwxr-xr-x  2 root root      4096 Mar 10 11:55 mkdir
drwxr-xr-x  2 root root      4096 Feb  3 18:22 mnt
drwxr-xr-x  3 root root      4096 Mar 17 17:06 opt
dr-xr-xr-x 177 root root        0 Mar 24 22:09 proc
drwx-----  5 root root      4096 Mar 12 21:39 root
drwxr-xr-x 27 root root       980 Apr  8 19:35 run
drwxr-xr-x  2 root root    12288 Apr  8 19:31 sbin
drwxr-xr-x  4 root root      4096 Mar 17 15:24 snap
drwxr-xr-x  3 root root      4096 Mar 17 17:21 srv
-rw-----  1 root root 4294967296 Mar  6 20:33 swap.img
dr-xr-xr-x 13 root root        0 Apr  1 19:32 sys
drwxrwxrwt 28 root root      4096 Apr  8 19:36 tmp
drwxr-xr-x 10 root root      4096 Feb  3 18:22 usr
drwxr-xr-x 13 root root      4096 Mar 13 11:04 var
lrwxrwxrwx  1 root root        30 Apr  7 06:52 vmlinuz ->
```

```
boot/vmlinuz-4.15.0-96-generic
lrwxrwxrwx 1 root root 30 Apr 7 06:52 vmlinuz.old ->
boot/vmlinuz-4.15.0-91-generic
```

```
[4]: ls
```

```
bin      etc          lib          mkdir  root  srv        usr
boot     home         lib64        mnt    run   swap.img  var
cdrom    initrd.img   lost+found   opt    sbin  sys       vmlinuz
dev      initrd.img.old media        proc   snap  tmp       vmlinuz.old
```

```
[5]: cd
```

```
[6]: pwd
```

```
/home/jupyter-official-nanakai
```

```
[7]: ls
```

```
Leila  nanakai  new_test  shared  Stepik  test  'Unisat HW'
```

```
[8]: cd Leila
```

```
[9]: ls
```

```
Leila_resume      'Leila_USK_01_file(2ns and 3rd assignment).ipynb'
Leila_Resume      Untitled.ipynb
Leila_resume.txt
```

```
[10]: touch leila123.png
```

```
[11]: touch leila123.txt
```

```
[12]: ls -l
```

```
total 20
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Apr 8 20:16
leila123.png
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Apr 8 20:17
leila123.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Mar 27 00:13
Leila_resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 13 Mar 27 00:14
Leila_Resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 631 Mar 27 00:10
Leila_resume.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 4412 Mar 27 00:18
'Leila_USK_01_file(2ns and 3rd assignment).ipynb'
```

```
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 1408 Mar 27 00:11
Untitled.ipynb
```

```
[13]: ls -l leila123.txt
```

```
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Apr 8 20:17
leila123.txt
```

```
[14]: mkdir leila123
```

```
[15]: ls -l leila123
```

```
total 0
```

```
[16]: ls -l
```

```
total 24
drwxr-xr-x 2 jupyter-official-nanakai jupyter-official-nanakai 4096 Apr 8 20:18
leila123
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Apr 8 20:16
leila123.png
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Apr 8 20:17
leila123.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Mar 27 00:13
Leila_resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 13 Mar 27 00:14
Leila_Resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 631 Mar 27 00:10
Leila_resume.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 4412 Mar 27 00:18
'Leila_USK_01_file(2ns and 3rd assignment).ipynb'
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 1408 Mar 27 00:11
Untitled.ipynb
```

```
[17]: chmod 077 leila123
```

```
[18]: ls -l
```

```
total 24
d---rwxrwx 2 jupyter-official-nanakai jupyter-official-nanakai 4096 Apr 8 20:18
leila123
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Apr 8 20:16
leila123.png
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Apr 8 20:17
leila123.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 0 Mar 27 00:13
Leila_resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 13 Mar 27 00:14
```

```

Leila_Resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 631 Mar 27 00:10
Leila_resume.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 4412 Mar 27 00:18
'Leila_USK_01_file(2ns and 3rd assignment).ipynb'
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 1408 Mar 27 00:11
Untitled.ipynb

```

```
[19]: chmod 400 leila123
```

```
[20]: ls -l leila123
```

```
total 0
```

```
[21]: ls -l
```

```

total 24
dr----- 2 jupyter-official-nanakai jupyter-official-nanakai 4096 Apr  8 20:18
leila123
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai    0 Apr  8 20:16
leila123.png
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai    0 Apr  8 20:17
leila123.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai    0 Mar 27 00:13
Leila_resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai  13 Mar 27 00:14
Leila_Resume
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 631 Mar 27 00:10
Leila_resume.txt
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 4412 Mar 27 00:18
'Leila_USK_01_file(2ns and 3rd assignment).ipynb'
-rw-r--r-- 1 jupyter-official-nanakai jupyter-official-nanakai 1408 Mar 27 00:11
Untitled.ipynb

```

```
[22]: ls -ld leila123
```

```

dr----- 2 jupyter-official-nanakai jupyter-official-nanakai 4096 Apr  8 20:18
leila123

```

```
[23]: chmod 777 leila123
```

```
[24]: ls -ld leila123
```

```

drwxrwxrwx 2 jupyter-official-nanakai jupyter-official-nanakai 4096 Apr  8 20:18
leila123

```

```
[25]: cd leila123
```

```
[26]: ls

[27]: touch filter.txt

[28]: whoami

jupyter-official-nanakai

[29]: pwd

/home/jupyter-official-nanakai/Leila/leila123

[30]: ##Filter

[31]: file filter

filter: cannot open `filter' (No such file or directory)

[32]: file filter.txt

filter.txt: empty

[33]: head filter.txt

[34]: file filter.txt

filter.txt: ASCII text

[35]: head filter.txt

Random things:
blouse
paper
helmet
street lights
box
book
nail file
twister
cinder block

[36]: head -15 filter.txt

Random things:
blouse
paper
helmet
street lights
```

box
book
nail file
twister
cinder block
photo album
sidewalk
bananas
mop
needle

```
[37]: tail -5 filter.txt
```

door
chalk
chalk
flowers
candle

```
[38]: cat filter.txt
```

Random things:
blouse
paper
helmet
street lights
box
book
nail file
twister
cinder block
photo album
sidewalk
bananas
mop
needle
thermostat
chapter book
chapter book
television
bow
house
helmet
magnet
canvas
monitor
bowl
flowers

```
fake flowers
outlet
door
chalk
chalk
flowers
candle
```

[39]: `sort filter.txt`

```
bananas
blouse
book
bow
bowl
box
candle
canvas
chalk
chalk
chapter book
chapter book
cinder block
door
fake flowers
flowers
flowers
helmet
helmet
house
magnet
monitor
mop
nail file
needle
outlet
paper
photo album
Random things:
sidewalk
street lights
television
thermostat
twister
```

[40]: `man sort`

SORT(1)

User Commands

SORT(1)

NAME

sort - sort lines of text files

SYNOPSIS

```
sort [OPTION]... [FILE]...
sort [OPTION]... --files0-from=F
```

DESCRIPTION

Write sorted concatenation of all FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too. Ordering options:

- b, --ignore-leading-blanks
ignore leading blanks
- d, --dictionary-order
consider only blanks and alphanumeric characters
- f, --ignore-case
fold lower case to upper case characters
- g, --general-numeric-sort
compare according to general numerical value
- i, --ignore-nonprinting
consider only printable characters
- M, --month-sort
compare (unknown) < 'JAN' < ... < 'DEC'
- h, --human-numeric-sort
compare human readable numbers (e.g., 2K 1G)
- n, --numeric-sort
compare according to string numerical value
- R, --random-sort
shuffle, but group identical keys. See shuf(1)
- random-source=FILE
get random bytes from FILE
- r, --reverse
reverse the result of comparisons

`--sort=WORD`
 sort according to WORD: general-numeric `-g`, human-numeric `-h`,
 month `-M`, numeric `-n`, random `-R`, version `-V`

`-V, --version-sort`
 natural sort of (version) numbers within text

Other options:

`--batch-size=NMERGE`
 merge at most NMERGE inputs at once; for more use temp files

`-c, --check, --check=diagnose-first`
 check for sorted input; do not sort

`-C, --check=quiet, --check=silent`
 like `-c`, but do not report first bad line

`--compress-program=PROG`
 compress temporaries with PROG; decompress them with PROG `-d`

`--debug`
 annotate the part of the line used to sort, and warn about questionable usage to stderr

`--files0-from=F`
 read input from the files specified by NUL-terminated names in file F; If F is `-` then read names from standard input

`-k, --key=KEYDEF`
 sort via a key; KEYDEF gives location and type

`-m, --merge`
 merge already sorted files; do not sort

`-o, --output=FILE`
 write result to FILE instead of standard output

`-s, --stable`
 stabilize sort by disabling last-resort comparison

`-S, --buffer-size=SIZE`
 use SIZE for main memory buffer

`-t, --field-separator=SEP`
 use SEP instead of non-blank to blank transition

-T, --temporary-directory=DIR
 use DIR for temporaries, not \$TMPDIR or /tmp; multiple options
 specify multiple directories

--parallel=N
 change the number of sorts run concurrently to N

-u, --unique
 with -c, check for strict ordering; without -c, output only the
 first of an equal run

-z, --zero-terminated
 line delimiter is NUL, not newline

--help display this help and exit

--version
 output version information and exit

KEYDEF is F[.C][OPTS][,F[.C][OPTS]] for start and stop position, where F is a field number and C a character position in the field; both are origin 1, and the stop position defaults to the line's end. If neither -t nor -b is in effect, characters in a field are counted from the beginning of the preceding whitespace. OPTS is one or more single-letter ordering options [bdfgiMhnRrV], which override global ordering options for that key. If no key is given, use the entire line as the key. Use --debug to diagnose incorrect key usage.

SIZE may be followed by the following multiplicative suffixes: % 1% of memory, b 1, K 1024 (default), and so on for M, G, T, P, E, Z, Y.

*** WARNING *** The locale specified by the environment affects sort order. Set LC_ALL=C to get the traditional sort order that uses native byte values.

AUTHOR

Written by Mike Haertel and Paul Eggert.

REPORTING BUGS

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>
Report sort translation bugs to <<http://translationproject.org/team/>>

COPYRIGHT

Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<http://gnu.org/licenses/gpl.html>>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

shuf(1), uniq(1)

Full documentation at: <<http://www.gnu.org/software/coreutils/sort>>
or available locally via: info '(coreutils) sort invocation'

GNU coreutils 8.28

January 2018

SORT(1)

```
[41]: sort -R filter.txt
```

```
photo album
house
sidewalk
monitor
mop
street lights
needle
magnet
nail file
cinder block
helmet
helmet
chalk
chalk
blouse
Random things:
paper
box
flowers
flowers
bow
outlet
television
fake flowers
candle
chapter book
chapter book
book
thermostat
door
bowl
canvas
twister
bananas
```

```
[42]: sort -u filter.txt
```

```
bananas
```

blouse
book
bow
bowl
box
candle
canvas
chalk
chapter book
cinder block
door
fake flowers
flowers
helmet
house
magnet
monitor
mop
nail file
needle
outlet
paper
photo album
Random things:
sidewalk
street lights
television
thermostat
twister

```
[43]: sort -V filter.txt
```

Random things:
bananas
blouse
book
bow
bowl
box
candle
canvas
chalk
chalk
chapter book
chapter book
cinder block
door
fake flowers

flowers
flowers
helmet
helmet
house
magnet
monitor
mop
nail file
needle
outlet
paper
photo album
sidewalk
street lights
television
thermostat
twister

[44]: nl filter.txt

```
1 Random things:
2 blouse
3 paper
4 helmet
5 street lights
6 box
7 book
8 nail file
9 twister
10 cinder block
11 photo album
12 sidewalk
13 bananas
14 mop
15 needle
16 thermostat
17 chapter book
18 chapter book
19 television
20 bow
21 house
22 helmet
23 magnet
24 canvas
25 monitor
26 bowl
27 flowers
```

```
28 fake flowers
29 outlet
30 door
31 chalk
32 chalk
33 flowers
34 candle
```

```
[ ]: clear
```

```
[1]: file filter.txt
```

```
filter.txt: cannot open `filter.txt' (No such file or directory)
```

```
[2]: pwd
```

```
/home/jupyter-official-nanakai/Unisat HW
```

```
[3]: cd
```

```
[4]: cd Leila
```

```
[5]: pwd
```

```
/home/jupyter-official-nanakai/Leila
```

```
[6]: cd leila123
```

```
[7]: ls
```

```
filter.txt
```

```
[8]: file filter.txt
```

```
filter.txt: ASCII text
```

```
[9]: nl -s ' ' -w 10 filter.txt
```

```
1. Random things:
2. blouse
3. paper
4. helmet
5. street lights
6. box
7. book
8. nail file
9. twister
10. cinder block
11. photo album
```

- 12. sidewalk
- 13. bananas
- 14. mop
- 15. needle
- 16. thermostat
- 17. chapter book
- 18. chapter book
- 19. television
- 20. bow
- 21. house
- 22. helmet
- 23. magnet
- 24. canvas
- 25. monitor
- 26. bowl
- 27. flowers
- 28. fake flowers
- 29. outlet
- 30. door
- 31. chalk
- 32. chalk
- 33. flowers
- 34. candle

```
[10]: nl -i filter.txt
```

```
nl: invalid line number increment: 'filter.txt'
```

```
[11]: nl -n rz filter.txt
```

```
000001 Random things:
000002 blouse
000003 paper
000004 helmet
000005 street lights
000006 box
000007 book
000008 nail file
000009 twister
000010 cinder block
000011 photo album
000012 sidewalk
000013 bananas
000014 mop
000015 needle
000016 thermostat
```

```
000017 chapter book
000018 chapter book
000019 television
000020 bow
000021 house
000022 helmet
000023 magnet
000024 canvas
000025 monitor
000026 bowl
000027 flowers
000028 fake flowers
000029 outlet
000030 door
000031 chalk
000032 chalk
000033 flowers
000034 candle
```

```
[13]: nl -v 5 filter.txt #How to make nl use a different starting line number?
```

```
5 Random things:
6 blouse
7 paper
8 helmet
9 street lights
10 box
11 book
12 nail file
13 twister
14 cinder block
15 photo album
16 sidewalk
17 bananas
18 mop
19 needle
20 thermostat
21 chapter book
22 chapter book
23 television
24 bow
25 house
26 helmet
27 magnet
28 canvas
29 monitor
30 bowl
31 flowers
```



```
32 fake flowers
33 outlet
34 door
35 chalk
36 chalk
37 flowers
38 candle
```

```
[14]: wc filter.txt
```

```
33 42 280 filter.txt
```

```
[15]: #33 lines 42 words 280 charaters
```

```
[16]: wc -l filter.txt
```

```
33 filter.txt
```

```
[17]: wc -w filter.txt
```

```
42 filter.txt
```

```
[18]: wc -m filter.txt
```

```
280 filter.txt
```

```
[19]: cut -f 1 -d ' ' filter.txt
```

```
Random
blouse
paper
helmet
street
box
book
nail
twister
cinder
photo
sidewalk
bananas
mop
needle
thermostat
chapter
chapter
television
bow
```

house
helmet
magnet
canvas
monitor
bowl
flowers
fake
outlet
door
chalk
chalk
flowers
candle

[22]: `man cut`

```
CUT(1)                                User Commands                                CUT(1)

NAME
    cut - remove sections from each line of files

SYNOPSIS
    cut OPTION... [FILE]...

DESCRIPTION
    Print selected parts of lines from each FILE to standard output.

    With no FILE, or when FILE is -, read standard input.

    Mandatory arguments to long options are mandatory for short options
    too.

    -b, --bytes=LIST
        select only these bytes

    -c, --characters=LIST
        select only these characters

    -d, --delimiter=DELIM
        use DELIM instead of TAB for field delimiter

    -f, --fields=LIST
        select only these fields; also print any line that contains no
        delimiter character, unless the -s option is specified

    -n
        (ignored)
```

--complement
 complement the set of selected bytes, characters or fields

-s, --only-delimited
 do not print lines not containing delimiters

--output-delimiter=STRING
 use STRING as the output delimiter the default is to use the
 input delimiter

-z, --zero-terminated
 line delimiter is NUL, not newline

--help display this help and exit

--version
 output version information and exit

Use one, and only one of -b, -c or -f. Each LIST is made up of one range, or many ranges separated by commas. Selected input is written in the same order that it is read, and is written exactly once. Each range is one of:

N N'th byte, character or field, counted from 1

N- from N'th byte, character or field, to end of line

N-M from N'th to M'th (included) byte, character or field

-M from first to M'th (included) byte, character or field

AUTHOR

Written by David M. Ihnat, David MacKenzie, and Jim Meyering.

REPORTING BUGS

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>
 Report cut translation bugs to <<http://translationproject.org/team/>>

COPYRIGHT

Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<http://gnu.org/licenses/gpl.html>>.
 This is free software: you are free to change and redistribute it.
 There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

Full documentation at: <<http://www.gnu.org/software/coreutils/cut>>
 or available locally via: info '(coreutils) cut invocation'

```
[25]: touch 09cd.txt
```

```
[26]: pwd
```

```
/home/jupyter-official-nanakai/Leila/leila123
```

```
[27]: ls
```

```
09cd.txt  filter.txt
```

```
[28]: cut -f 1, 2 -d ' ' 09cd.txt
```

```
cut: fields are numbered from 1
Try 'cut --help' for more information.
```

```
[29]: cut --help
```

```
Usage: cut OPTION... [FILE]...
```

```
Print selected parts of lines from each FILE to standard output.
```

```
With no FILE, or when FILE is -, read standard input.
```

```
Mandatory arguments to long options are mandatory for short options too.
```

-b, --bytes=LIST	select only these bytes
-c, --characters=LIST	select only these characters
-d, --delimiter=DELIM	use DELIM instead of TAB for field delimiter
-f, --fields=LIST	select only these fields; also print any line that contains no delimiter character, unless the -s option is specified
-n	(ignored)
--complement	complement the set of selected bytes, characters or fields
-s, --only-delimited	do not print lines not containing delimiters
--output-delimiter=STRING	use STRING as the output delimiter the default is to use the input delimiter
-z, --zero-terminated	line delimiter is NUL, not newline
--help	display this help and exit
--version	output version information and exit

Use one, and only one of -b, -c or -f. Each LIST is made up of one range, or many ranges separated by commas. Selected input is written in the same order that it is read, and is written exactly once.

Each range is one of:

N N'th byte, character or field, counted from 1

N- from N'th byte, character or field, to end of line
N-M from N'th to M'th (included) byte, character or field
-M from first to M'th (included) byte, character or field

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>
Full documentation at: <<http://www.gnu.org/software/coreutils/cut>>
or available locally via: info '(coreutils) cut invocation'

```
[30]: cut -f 1-2 -d ' ' 09cd.txt
```

```
Jimmy Carter.  
George H.  
Gerald Ford.  
Ronald Reagan.  
John Adams.  
Herbert Hoover.
```

```
[31]: cat filter.txt
```

```
Random things:  
blouse  
paper  
helmet  
street lights  
box  
book  
nail file  
twister  
cinder block  
photo album  
sidewalk  
bananas  
mop  
needle  
thermostat  
chapter book  
chapter book  
television  
bow  
house  
helmet  
magnet  
canvas  
monitor  
bowl  
flowers  
fake flowers  
outlet
```

```
door
chalk
chalk
flowers
candle
```

```
[33]: uniq filter.txt #to see the unqie things in the list
```

```
Random things:
blouse
paper
helmet
street lights
box
book
nail file
twister
cinder block
photo album
sidewalk
bananas
mop
needle
thermostat
chapter book
television
bow
house
helmet
magnet
canvas
monitor
bowl
flowers
fake flowers
outlet
door
chalk
flowers
candle
```

```
[34]: tac filter.txt
```

```
candleflowers
chalk
chalk
door
outlet
```

fake flowers
flowers
bowl
monitor
canvas
magnet
helmet
house
bow
television
chapter book
chapter book
thermostat
needle
mop
bananas
sidewalk
photo album
cinder block
twister
nail file
book
box
street lights
helmet
paper
blouse
Random things:

[37]: *## how to make unique and tac together?*

[38]: date

Wed Apr 8 21:10:31 UTC 2020

[]: