

Name:K.Soujanya
Section:CSE-Sec A
Roll No:422163

Implementation Of Commands

Grep Command

```
Activities Terminal ▾ Feb 14 14:57
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cd Desktop/422163
bash: cd: Desktop/422163: No such file or directory
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cd 422163
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
cat: file1.txt: No such file or directory
student@ai-HP-ProDesk-600-G4-MT:~/422163$ touch file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat>file1.txt
This is beautiful
It is Gorgeous
They are birds
These are animals
They are people
We are humans student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
This is beautiful
It is Gorgeous
They are birds
These are animals
They are people
We are humans student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat>file1.txt
It is delicate
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
It is delicate
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
It is delicate
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat>file1.txt
This is beautiful
It is Gorgeous
They are birds
These are animals
They are people^C
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
It is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peoplestudent@ai-HP-ProDesk-600-G4-MT:~/422163$ cat -n file1.txt
 1 It is delicateThis is beautiful
 2 It is Gorgeous
 3 They are birds
student@ai-HP-ProDesk-600-G4-MT:~/422163$
```

```
Activities Terminal ▾ Feb 14 14:57
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
It is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peoplestudent@ai-HP-ProDesk-600-G4-MT:~/422163$ cat -n file1.txt
 1 It is delicateThis is beautiful
 2 It is Gorgeous
 3 They are birds
 4 These are animals
 5 They are peoplestudent@ai-HP-ProDesk-600-G4-MT:~/422163$ cat -s file1.txt
It is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peoplestudent@ai-HP-ProDesk-600-G4-MT:~/422163$ cat -d file1.txt
cat: invalid option -- 'd'
Try 'cat --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat --d file1.txt
cat: unrecognized option '--d'
Try 'cat --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -i connect file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -i lt file1.txt
It is delicateThis is beautiful
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -c this file1.txt
0
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -i this file1.txt
It is delicate It is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -w it file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -w This file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -0 lt file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -1 lt file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -1 lt file1.txt
It is Gorgeous
```

```
Activities Terminal ▾ Feb 14 14:57
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -2 it file1.txt
It is Gorgeous
It is delicateThis is beautiful
It is Gorgeous
They are birds
It is delicateThis is beautiful
1:It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -n it file1.txt
1:It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ gerp -n They file1.txt
Command 'gerp' not found, did you mean:
  command 'grep' from deb grep (3.4-1)

Try: sudo apt install <deb name>

student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -n They file1.txt
1:They are birds
2:They are people
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep--help
grep--help: command not found
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERNs in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNs can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
-E, --extended-regexp  PATTERNs are extended regular expressions
-F, --fixed-strings    PATTERNs are strings
-G, --basic-regexp    PATTERNs are basic regular expressions
-P, --perl-regexp     PATTERNs are Perl regular expressions
-e, --regexp=PATTERN  use PATTERN for matching
-f, --file=FILE        take PATTERNs from FILE
-i, --ignore-case     ignore case distinctions in patterns and data
-n, --no-ignore-case   do not ignore case distinctions (default)
-w, --word-regexp     match only whole words
-X, --line-regexp     match only whole lines

Activities Terminal ▾ Feb 14 14:57
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
grep --help: command not found
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERNs in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNs can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
-E, --extended-regexp  PATTERNs are extended regular expressions
-F, --fixed-strings    PATTERNs are strings
-G, --basic-regexp    PATTERNs are basic regular expressions
-P, --perl-regexp     PATTERNs are Perl regular expressions
-e, --regexp=PATTERN  use PATTERN for matching
-f, --file=FILE        take PATTERNs from FILE
-i, --ignore-case     ignore case distinctions in patterns and data
-n, --no-ignore-case   do not ignore case distinctions (default)
-w, --word-regexp     match only whole words
-X, --line-regexp     match only whole lines
```

```
Activities Terminal ▾ Feb 14 14:57
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
grep --help: command not found
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERNs in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNs can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
-E, --extended-regexp  PATTERNs are extended regular expressions
-F, --fixed-strings    PATTERNs are strings
-G, --basic-regexp    PATTERNs are basic regular expressions
-P, --perl-regexp     PATTERNs are Perl regular expressions
-e, --regexp=PATTERN  use PATTERN for matching
-f, --file=FILE        take PATTERNs from FILE
-i, --ignore-case     ignore case distinctions in patterns and data
-n, --no-ignore-case   do not ignore case distinctions (default)
-w, --word-regexp     match only whole words
-X, --line-regexp     match only whole lines

Miscellaneous:
-s, --no-messages    suppress error messages
-v, --invert-match   select non-matching lines
-V, --version         display version information and exit
--help               display this help text and exit

Output control:
-m, --max-count=NUM  stop after NUM selected lines
-b, --byte-offset    print the byte offset with output lines
-n, --line-number    print line number with output lines
-H, --with-filename  print file name with output lines
-h, --no-filename    suppress the file name prefix on output
--label=LABEL        use LABEL as the standard input file name prefix
-o, --only-matching  show only nonempty parts of lines that match
-q, --quiet, --silent suppress all normal output
--binary-files=TYPE assume that binary files are TYPE;
TYPE is 'binary', 'text', or 'without-match'
-a, --text           equivalent to --binary-files=text
-I                 equivalent to --binary-files=without-match
-d, --directories=ACTION how to handle directories;
ACTION is 'read', 'recurse', or 'skip'
-D, --devices=ACTION how to handle devices, FIFOs and sockets;
ACTION is 'read' or 'skip'
-r, --recursive      like --directories=recurse
-R, --dereference-recursive likewise, but follow all symlinks
--include=GLOB       search only files that match GLOB (a file pattern)
--exclude=GLOB      skip files that match GLOB
--exclude-from=FILE skip files that match any file pattern from FILE
--exclude-dir=GLOB  skip directories that match GLOB
-L, --files-without-match print only names of FILES with no selected lines
-l, --files-with-matches print only names of FILES with selected lines
-c, --count          print only a count of selected lines per FILE
-T, --initial-tab    make tabs line up (if needed)

Activities Terminal ▾ Feb 14 14:57
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
grep --help: command not found
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERNs in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNs can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
-E, --extended-regexp  PATTERNs are extended regular expressions
-F, --fixed-strings    PATTERNs are strings
-G, --basic-regexp    PATTERNs are basic regular expressions
-P, --perl-regexp     PATTERNs are Perl regular expressions
-e, --regexp=PATTERN  use PATTERN for matching
-f, --file=FILE        take PATTERNs from FILE
-i, --ignore-case     ignore case distinctions in patterns and data
-n, --no-ignore-case   do not ignore case distinctions (default)
-w, --word-regexp     match only whole words
-X, --line-regexp     match only whole lines
```

```
Activities Terminal ▾ Feb 14 14:57
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
grep --help: command not found
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --help
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERNs in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNs can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
-E, --extended-regexp  PATTERNs are extended regular expressions
-F, --fixed-strings    PATTERNs are strings
-G, --basic-regexp    PATTERNs are basic regular expressions
-P, --perl-regexp     PATTERNs are Perl regular expressions
-e, --regexp=PATTERN  use PATTERN for matching
-f, --file=FILE        take PATTERNs from FILE
-i, --ignore-case     ignore case distinctions in patterns and data
-n, --no-ignore-case   do not ignore case distinctions (default)
-w, --word-regexp     match only whole words
-X, --line-regexp     match only whole lines

Context control:
-B, --before-context=NUM print NUM lines of leading context
-A, --after-context=NUM  print NUM lines of trailing context
-C, --context=NUM        print NUM lines of output context
-NUM                     same as --context=NUM
--color[=WHEN],          use markers to highlight the matching strings;
--colour[=WHEN]          WHEN is 'always', 'never', or 'auto'
-U, --binary             do not strip CR characters at EOL (MSDOS/Windows)

When FILE is '-', read standard input. With no FILE, read '.' if
recursive, '-' otherwise. With fewer than two FILES, assume '-'.
Exit status is 0 if any line (or file if -L) is selected, 1 otherwise;
if any error occurs and -q is not given, the exit status is 2.

Report bugs to: bug-grep@gnu.org
GNU grep home page: <http://www.gnu.org/software/grep/>
General help using GNU software: <https://www.gnu.org/gethelp/>
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -a it file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -h file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -h it file1.txt
Command 'grep' not found, but can be installed with:
sudo apt install grep

student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -h it file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep -u it file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT:~/422163$ grep --version
grep (GNU grep) 3.4
Copyright (C) 2020 Free Software Foundation, Inc.
```

Activities Terminal Feb 14 14:57 student@ai-HP-ProDesk-600-G4-MT: ~/422163

```
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.

Written by Mike Haertel and others; see
<https://git.sv.gnu.org/cgit/grep.git/tree/AUTHORS>.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -d file1.txt
grep: invalid argument 'file1.txt' for '--directories'
Valid arguments are:
- 'read'
- 'recurse'
- 'skip'
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -d It file1.txt
grep: invalid argument 'It' for '--directories'
Valid arguments are:
- 'read'
- 'recurse'
- 'skip'
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep --d It file1.txt
grep: option '-d' is ambiguous; possibilities: '--devices' '--directories' '--dereference-recursive'
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m It file1.txt
grep: invalid max count
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -l "It" *
file1.txt
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -w "It" file1.txt
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -o "It" file1.txt
It
It
```

Activities Terminal Feb 14 16:34 student@ai-HP-ProDesk-600-G4-MT: ~/422163

```
When FILE is '-', read standard input. With no FILE, read '.' if
recursive, '-' otherwise. With fewer than two FILEs, assume -h.
Exit status is 0 if any line (or file if -L) is selected, 1 otherwise;
if any error occurs and -q is not given, the exit status is 2.

Report bugs to: bug-grep@gnu.org
GNU grep home page: <http://www.gnu.org/software/grep/>
General help using GNU software: <https://www.gnu.org/gethelp/>
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -a It file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -h file1.txt
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -h It file1.txt

Command 'grep' not found, but can be installed with:
sudo apt install grep

student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -h It file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -u It file1.txt
It is delicateThis is beautiful
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep --version
grep (GNU grep) 3.4
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.

Written by Mike Haertel and others; see
<https://git.sv.gnu.org/cgit/grep.git/tree/AUTHORS>.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -d file1.txt
grep: invalid argument 'file1.txt' for '--directories'
Valid arguments are:
- 'read'
- 'recurse'
- 'skip'
```

Activities Terminal Feb 14 14:57 student@ai-HP-ProDesk-600-G4-MT: ~/422163

```
grep: invalid argument 'lt' for '--directories'
Valid arguments are:
- 'read'
- 'recurse'
- 'skip'
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -d It file1.txt
grep: option '-d' is ambiguous; possibilities: '--devices' '--directories' '--dereference-recursive'
Usage: grep [OPTION]... PATTERN [FILE]...
Try 'grep --help' for more information.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m It file1.txt
grep: invalid max count
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -l "It" *
file1.txt
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -w "It" file1.txt
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -o "It" file1.txt
It
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -v "It" file1.txt
it is delicateThis is beautiful
They are birds
These are animals
They are people
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^It" file1.txt
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^This" file1.txt
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "oss" file1.txt
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "ousS" file1.txt
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -e "It" file1.txt
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -R "It" file1.txt
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT: ~/422163$
```

```
Activities Terminal ▾ Feb 14 16:34 student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep: invalid argument 'file1.txt' for '--directories'  
Valid arguments are:  
- 'read'  
- 'reurse'  
- 'skip'  
Usage: grep [OPTION]... PATTERNS [FILE]...  
Try 'grep --help' for more information.  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -d it file1.txt  
grep: invalid argument 'lt' for '--directories'  
Valid arguments are:  
- 'read'  
- 'reurse'  
- 'skip'  
Usage: grep [OPTION]... PATTERNS [FILE]...  
Try 'grep --help' for more information.  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep --d it file1.txt  
grep: option '-d' is ambiguous; possibilities: '--devices' '--directories' '--dereference-recursive'  
Usage: grep [OPTION]... PATTERNS [FILE]...  
Try 'grep --help' for more information.  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m it file1.txt  
grep: invalid max count  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -l "It" *  
file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -w It file1.txt  
It is Gorgeous  
It is Gorgeous  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -o "It" file1.txt  
It  
It  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -v "It" file1.txt  
it is delicateThis is beautiful  
They are birds  
These are animals  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^\It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^\This" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^\oss" file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^\ous\$" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -e "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -R "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -E "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -f "It" file1.txt  
grep: It: No such file or directory  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -F "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -G "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -P "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -x "It" file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -z "They" file1.txt  
It is delicateThis is beautiful  
It is Gorgeous  
They are birds  
These are animals  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -x "it is delicate" file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -x "It is Gorgeous" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -z file1.txt  
^C  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -z file1.txt  
^C  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -s "They" file1.txt  
They are birds  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ cat file1.txt  
it is delicateThis is beautiful  
It is Gorgeous
```

```
Activities Terminal ▾ Feb 14 16:34 student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep '^This' file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^\oss" file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep "^\ous\$" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -e "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -R "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -E "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -f "It" file1.txt  
grep: It: No such file or directory  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -F "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -G "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -P "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -x "It" file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -z "They" file1.txt  
It is delicateThis is beautiful  
It is Gorgeous  
They are birds  
These are animals  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -x "it is delicate" file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -x "It is Gorgeous" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -z file1.txt  
^C  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -z file1.txt  
^C  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -s "They" file1.txt  
They are birds  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ cat file1.txt  
it is delicateThis is beautiful  
It is Gorgeous
```

```
Activities Terminal ▾ Feb 14 16:34 student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ cat file1.txt  
They are birds  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m 3 file1.txt  
it is delicateThis is beautiful  
It is Gorgeous  
They are birds  
These are animals  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m 3 file1.txt  
^C  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -3 file1.txt  
^C  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -b 5 file1.txt  
grep: file1.txt: No such file or directory  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -b 5 file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -b 5 file1.txt  
^C  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -3 "It" file1.txt  
it is delicateThis is beautiful  
It is Gorgeous  
They are birds  
These are animals  
They are people  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m 3 "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m 5 "It" file1.txt  
It is Gorgeous  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ grep -m 10 "uo" file1.txt  
student@student@ai-HP-ProDesk-600-G4-MT: ~/422163$ head --help  
Usage: head [OPTION]... [FILE]...  
Print the first 10 lines of each FILE to standard output.  
With more than one FILE, precede each with a header giving the file name.  
With no FILE, or when FILE is -, read standard input.  
Mandatory arguments to long options are mandatory for short options too.  
-c, --bytes=[-]NUM print the first NUM bytes of each file;  
with the leading '-' , print all but the last
```

```
student@ai-HP-ProDesk-600-G4-MT: ~/422163
With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.
  -c, --bytes=[-]NUM      print the first NUM bytes of each file;
                        with the leading '-', print all but the last
                        NUM bytes of each file
  -n, --lines=[-]NUM      print the first NUM lines instead of the first 10;
                        with the leading '-', print all but the last
                        NUM lines of each file
  -q, --quiet, --silent  never print headers giving file names
  -v, --verbose          always print headers giving file names
  -z, --zero-terminated   line delimiter is NUL, not newline
  -h, --help              display this help and exit
  -V, --version           output version information and exit

NUM may have a multiplier suffix:
b 512, kB 1000, K 1024, MB 1000*1000, M 1024*1024,
GB 1000*1000*1000, G 1024*1024*1024, and so on for T, P, E, Z, Y.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/HEAD>
or available locally via: info '(coreutils) head invocation'
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ head file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peoplestudent@ai-HP-ProDesk-600-G4-MT: ~/422163$ head -n 3 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
They are birds
These are animals
They are peopleYou are INtelligent
He is Smartstudent@ai-HP-ProDesk-600-G4-MT: ~/422163$ head -q file1.txt file2.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleYou are INtelligent
He is Smartstudent@ai-HP-ProDesk-600-G4-MT: ~/422163$ head -v file1.txt
=> file1.txt <==
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleYou are INtelligent
He is Smartstudent@ai-HP-ProDesk-600-G4-MT: ~/422163$ head -z file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleYou are INtelligent
He is Smartstudent@ai-HP-ProDesk-600-G4-MT: ~/422163$ tail --help
Usage: tail [OPTION]... [FILE]...
Print the last 10 lines of each FILE to standard output.
With more than one FILE, precede each with a header giving the file name.

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

Mandatory arguments to long options are mandatory for short options too.
  -c, --bytes=[+NUM]      output the last NUM bytes; or use -c +NUM to
                        output starting with byte NUM of each file
  -f, --follow[=(name|descriptor)]  output appended data as the file grows;
                        an absent option argument means 'descriptor'
  -F, --follow-fd[=FD]     same as --follow=name --retry
  -n, --lines=[+NUM]       output the last NUM lines, instead of the last 10;
```

Head Command

```
student@ai-HP-ProDesk-600-G4-MT: ~/422163
it is
You are INtelligent
He is Smart
It is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleYou are INtelligent
He is Smart
=> file1.txt <==
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleYou are INtelligent
He is Smart
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ head -v file2.txt
=> file2.txt <==
You are INtelligent
He is Smart
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ head -z file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleYou are INtelligent
He is Smart
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ tail --help
Usage: tail [OPTION]... [FILE]...
Print the last 10 lines of each FILE to standard output.
With more than one FILE, precede each with a header giving the file name.

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

Mandatory arguments to long options are mandatory for short options too.
  -c, --bytes=[+NUM]      output the last NUM bytes; or use -c +NUM to
                        output starting with byte NUM of each file
  -f, --follow[=(name|descriptor)]  output appended data as the file grows;
                        an absent option argument means 'descriptor'
  -F, --follow-fd[=FD]     same as --follow=name --retry
  -n, --lines=[+NUM]       output the last NUM lines, instead of the last 10;
```

Tail command

```
student@ai-HP-ProDesk-600-G4-MT: ~/422163
-n, --lines=[+NUM]      output the last NUM lines, instead of the last 10;
                        or use -n +NUM to output starting with line NUM
--max-unchanged-stats=N
                        with --follow=name, reopen a FILE which has not
                        changed size after N (default 5) iterations
                        to see if it has been unlinked or renamed
                        (this is the usual case of rotated log files);
                        with inotify, this option is rarely useful
--pid=PID              with -f, terminate after process ID, PID dies
-q, --quiet, --silent  never output headers giving file names
--retry                keep trying to open a file if it is inaccessible
-s, --sleep-interval=N with -f, sleep for approximately N seconds
                        (default 1.0) between iterations;
                        with inotify and --pid=P, check process P at
                        least once every N seconds
--v, --verbose          always output headers giving file names
--z, --zero-terminated  line delimiter is NUL, not newline
--help                 display this help and exit
--version              output version information and exit

NUM may have a multiplier suffix:
b 512, kB 1000, K 1024, MB 1000*1000, M 1024*1024,
GB 1000*1000*1000, G 1024*1024*1024, and so on for T, P, E, Z, Y.

With --follow (-f), tail defaults to following the file descriptor, which
means that even if a tail'ed file is renamed, tail will continue to track
its end. This default behavior is not desirable when you really want to
track the actual name of the file, not the file descriptor (e.g., log
rotation). Use --follow=name in that case. That causes tail to track the
named file in a way that accommodates renaming, removal and creation.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/tail>
or available locally via: info '(coreutils) tail invocation'
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ tail file2.txt
You are INtelligent
He is Smart
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ cat >>file1.txt
```

```
Activities Terminal ▾ Feb 14 16:35
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat >>file1.txt
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ TAIL FILE1.TXT
TAIL: command not found
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail file1.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail file1.txt file2.txt
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
==> file2.txt <==
You are INtelligent
He is Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail file1.txt file2.txt
==> file1.txt <==
It is Gorgeous
They are birds

```

```
Activities Terminal ▾ Feb 14 16:35
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat >>file1.txt
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
==> file2.txt <==
You are INtelligent
He is Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -q file1.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -q file1.txt file2.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -n 3 file1.txt
Bangalore
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -n +4 file1.txt
These are animals
```

```
Activities Terminal ▾ Feb 14 16:35 student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -n 3 file1.txt
He is Smart
Bangalore
Mumbai
PUNE
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
apore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -n +4 file1.txt
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -c 45 file1.txt
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -c 12 file1.txt
beautiful
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -v file1.txt
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -v file1.txt
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -s 4 file1.txt
tail: invalid number of seconds: 'file1.txt'
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -s 4 file1.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f 3 file1.txt
tail: cannot open '3' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f 0 file1.txt
tail: cannot open '0' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f 1 file1.txt
tail: cannot open '1' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
```

```
Activities Terminal ▾ Feb 14 16:35 student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -v file1.txt
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -v file1.txt
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
tail: invalid number of seconds: 'file1.txt'
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -s 4 file1.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
tail: cannot open '3' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
tail: cannot open '0' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
tail: cannot open '1' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
```

```
Activities Terminal ▾ Feb 14 16:35 student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f 3 file1.txt
tail: cannot open '3' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f 0 file1.txt
tail: cannot open '0' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f 1 file1.txt
tail: cannot open '1' for reading: No such file or directory
==> file1.txt <==
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
```

```
Activities Terminal ▾ Feb 14 16:35
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f 1 file1.txt
tail: cannot open '1' for reading: No such file or directory
=> file1.txt <==

It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
C
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tail -f "it" file1.txt
tail: cannot open 'it' for reading: No such file or directory
=> file1.txt <==

It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
C
student@ai-HP-ProDesk-600-G4-MT:~/422163$ 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: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/cut>
```

```
Activities Terminal ▾ Feb 14 16:35
student@ai-HP-ProDesk-600-G4-MT:~/422163$ 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: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/cut>
```

Cut command

```
Activities Terminal ▾ Feb 14 16:35
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -b 1 file1.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
C
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -b 1,2,3,4,5 file1.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
C
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -b 1-3,5-7 file1.txt
It is Gorgeous
They are birds
These are animals
They are people
Indonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
C
```

```
Activities Terminal ▾ Feb 14 16:35
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -b 1-3,5-7 file1.txt
Mumba
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -b 1- file1.txt
It s d
It s G
The ar
Thee a
The ar
Sinapo
Hydrab
Chenal
Banalo
Munai
PUN
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -c 1 file1.txt
i
I
T
T
S
H
C
B
M
P
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -c 1 file1.txt
i
I
T
T
S
H
C
B
M
P
```

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -c 1 file1.txt
i
I
T
T
S
H
C
B
M
P
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -c 4,6 file1.txt
i
l
ya
s
ya
qp
ea
na
gl
bi
E
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -f 1 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -f 1 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
```

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -f 2 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d -f 1 file1.txt
cut: the delimiter must be a single character
Try 'cut --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d -f 1 file1.txt
cut: the delimiter must be a single character
Try 'cut --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d "delimiter" -f 1 file1.txt
cut: the delimiter must be a single character
Try 'cut --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d 1 file1.txt
cut: you must specify a list of bytes, characters, or fields
Try 'cut --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d DELIM file1.txt
cut: the delimiter must be a single character
Try 'cut --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -dDELIM file1.txt
cut: the delimiter must be a single character
Try 'cut --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d "" -f 1 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
```

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d "" -f 1 file1.txt
1 It is delicateThis is beautiful
2 It is Gorgeous
3 They are birds
4 These are animals
5 They are peopleIndonesia
6 Singapore
7 Hyderabad
8 Chennai
9 Bangalore
10 Mumbai
11 PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut -d "" -f 1-4 file1.txt
1 It is delicateThis is beautiful
2 It is Gorgeous
3 They are birds
4 These are animals
5 They are peopleIndonesia
6 Singapore
7 Hyderabad
8 Chennai
9 Bangalore
10 Mumbai
11 PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cut --version
cut (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.

Written by David M. Ihnat, David MacKenzie, and Jim Meyering.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste --help
Usage: paste [OPTION]... [FILE]...
Write lines consisting of the sequentially corresponding lines from
each FILE, separated by TABs, to standard output.
```

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste --help
Usage: paste [OPTION]... [FILE]...
Write lines consisting of the sequentially corresponding lines from
each FILE, separated by TABs, to standard output.

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

Mandatory arguments to long options are mandatory for short options too.
-d, --delimiters=LIST  reuse characters from LIST instead of TABs
-s, --serial           paste one file at a time instead of in parallel
-z, --zero-terminated  line delimiter is NUL, not newline
--help                display this help and exit
--version             output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/paste>
or available locally via: info '(coreutils) paste invocation'
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste number file1.txt
paste: number: No such file or directory
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat > number
1
2
3
4
5
6
7
8
9
10
11@student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste number file1
paste: file1: No such file or directory
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste number file1.txt
1
2 It is delicateThis is beautiful
3 It is Gorgeous
4 They are birds
5 These are animals
```

Paste Command

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste number file1.txt
1
2 It is delicateThis is beautiful
3 It is Gorgeous
4 They are birds
5 These are animals
6 They are peopleIndonesia
7 Singapore
8 Hyderabad
9 Chennai
10 Bangalore
11 Mumbai
12 PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste -d "||" number file1.txt
1||It is delicateThis is beautiful
2||It is Gorgeous
3||They are birds
4||These are animals
5||They are peopleIndonesia
6||Singapore
7||Hyderabad
8||Chennai
9||Bangalore
10||Mumbai
11||PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste -d ",," number file1.txt
1,It is delicateThis is beautiful
2,It is Gorgeous
3,They are birds
4,These are animals
5,They are peopleIndonesia
6,Singapore
7,Hyderabad
8,Chennai
9,Bangalore
10,Mumbai
11,PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste -d "|," number file1.txt
```

```
Activities Terminal Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste -d "|," number file1.txt
> ^
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste -s number file1.txt
1 2 3 4 5 6 7 8 9 10 11
it is delicateThis is beautifulIt is Gorgeous They are birds These are animals They are peopleIndonesia Sin
gapore Hyderabad Chennai Bangalore Mumbai PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste -s -d ":" number file1.txt
1:2:3:4:5:6:7:8:9:10:11
it is delicateThis is beautifulIt is Gorgeous:They are birds:These are animals:They are peopleIndonesia:Singapore:Hyderabad
d:Chennai:Bangalore:Mumbai:PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste --version
paste (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.

Written by David M. Ihnat and David MacKenzie.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq --help
Usage: uniq [OPTION]... [INPUT [OUTPUT]]
Filter adjacent matching lines from INPUT (or standard input),
writing to OUTPUT (or standard output).

With no options, matching lines are merged to the first occurrence.

Mandatory arguments to long options are mandatory for short options too.
-c, --count           prefix lines by the number of occurrences
-d, --repeated        only print duplicate lines, one for each group
-D                   print all duplicate lines
--all-repeated[=METHOD] like -D, but allow separating groups
                     with an empty line;
                     METHOD=(none(default),prepend,separate)
-f, --skip-fields=N  avoid comparing the first N fields
```

```
Activities Terminal Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ paste --version
paste (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.

Written by David M. Ihnat and David MacKenzie.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq --help
Usage: uniq [OPTION]... [INPUT [OUTPUT]]
Filter adjacent matching lines from INPUT (or standard input),
writing to OUTPUT (or standard output).

With no options, matching lines are merged to the first occurrence.

Mandatory arguments to long options are mandatory for short options too.
-c, --count           prefix lines by the number of occurrences
-d, --repeated        only print duplicate lines, one for each group
-D                   print all duplicate lines
--all-repeated[=METHOD] like -D, but allow separating groups
                     with an empty line;
                     METHOD=(none(default),prepend,separate)
-f, --skip-fields=N  avoid comparing the first N fields
--group[=METHOD]      show all items, separating groups with an empty line;
                     METHOD=(separate(default),prepend,append,both)
-i, --ignore-case     ignore differences in case when comparing
-s, --skip-chars=N    avoid comparing the first N characters
-u, --unique          only print unique lines
-z, --zero-terminated line delimiter is NUL, not newline
-w, --check-chars=N   compare no more than N characters in lines
--help                display this help and exit
--version             output version information and exit

A field is a run of blanks (usually spaces and/or TABs), then non-blank
characters. Fields are skipped before chars.
```

Unique Command

```
Activities Terminal Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq
or available locally via: info '(coreutils) uniq invocation'
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -c file1.txt
1 it is delicateThis is beautiful
1 It is Gorgeous
1 They are birds
1 These are animals
1 They are peopleIndonesia
1 Singapore
1 Hyderabad
1 Chennai
1 Bangalore
1 Mumbai
1 PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -d file1.txt
Mumbai
PUNE
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat>file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -d file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
```

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -d file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
It is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -d file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -d file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -D file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat>file1.txt
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -D file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -u file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals

```

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -D file1.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -u file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -f 2 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
PUNE
It is Gorgeous
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -s 3 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -w 2 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -l file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
PUNE
It is Gorgeous
```

```
Activities Terminal ▾ Feb 14 16:36
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -s 3 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -w 2 file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
PUNE
It is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -l file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
Mumbai
PUNE
It is Gorgeous
PUNE
It is Gorgeous
```

```
Activities Terminal Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -i file1.txt
It is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
PUNEIt is GorgeousPUNEIt is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -z file1.txt
it is delicateThis is beautiful
It is Gorgeous
They are birds
These are animals
They are peopleIndonesia
Singapore
Hyderabad
Chennai
Bangalore
Mumbai
PUNE
PUNEIt is GorgeousPUNEIt is Gorgeous
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -i file2.txt
You are INtelligent
You are Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file2.txt
You are INtelligent
You are Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tr --help
Usage: tr [OPTION]... SET1 [SET2]
    translate, squeeze, and/or delete characters from standard input,
    writing to standard output.

    -c, -C, --complement      use the complement of SET1
    -d, --delete              delete characters in SET1, do not translate
```

```
Activities Terminal Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ uniq -i file2.txt
You are INtelligent
You are Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file2.txt
You are INtelligent
You are Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tr --help
Usage: tr [OPTION]... SET1 [SET2]
    Translate, squeeze, and/or delete characters from standard input,
    writing to standard output.

    -c, -C, --complement      use the complement of SET1
    -d, --delete              delete characters in SET1, do not translate
    -s, --squeeze-repeats    replace each sequence of a repeated character
                             that is listed in the last specified SET,
                             with a single occurrence of that character
    -t, --truncate-set1     first truncate SET1 to length of SET2
    --help                  display this help and exit
    --version               output version information and exit

SETs are specified as strings of characters.  Most represent themselves.
Interpreted sequences are:

\NNN      character with octal value NNN (1 to 3 octal digits)
\\      backslash
\a      audible BEL
\b      backspace
\f      form feed
\n      new line
\r      return
\t      horizontal tab
\v      vertical tab
CHAR1-CHAR2  all characters from CHAR1 to CHAR2 in ascending order
[CHAR*]    in SET2, copies of CHAR until length of SET1
[CHAR*REPEAT] REPEAT copies of CHAR, REPEAT octal if starting with 0
[:alnum:]  all letters and digits
[:alpha:]  all letters
```

tr Command

```
Activities Terminal Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ tr --help
SETs are specified as strings of characters.  Most represent themselves.
Interpreted sequences are:

\NNN      character with octal value NNN (1 to 3 octal digits)
\\      backslash
\a      audible BEL
\b      backspace
\f      form feed
\n      new line
\r      return
\t      horizontal tab
\v      vertical tab
CHAR1-CHAR2  all characters from CHAR1 to CHAR2 in ascending order
[CHAR*]    in SET2, copies of CHAR until length of SET1
[CHAR*REPEAT] REPEAT copies of CHAR, REPEAT octal if starting with 0
[:alnum:]  all letters and digits
[:alpha:]  all letters
[:blank:]  all horizontal whitespace
[:cntrl:]  all control characters
[:digit:]  all digits
[:graph:]  all printable characters, not including space
[:lower:]  all lower case letters
[:print:]  all printable characters, including space
[:punct:]  all punctuation characters
[:space:]  all horizontal or vertical whitespace
[:upper:]  all upper case letters
[:xdigit:] all hexadecimal digits
[=CHAR=]   all characters which are equivalent to CHAR

Translation occurs if -d is not given and both SET1 and SET2 appear.
-t may be used only when translating.  SET2 is extended to length of
SET1 by repeating its last character as necessary.  Excess characters
of SET2 are ignored.  Only [:lower:] and [:upper:] are guaranteed to
expand in ascending order; used in SET2 while translating, they may
only be used in pairs to specify case conversion.  -s uses the last
specified SET, and occurs after translation or deletion.
```

```
Activities Terminal Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt | tr [a-z][A-Z]
tr: missing operand after '[a-z][A-Z]'
Two strings must be given when translating.
Try 'tr --help' for more information.
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt | tr [a-z] [A-Z]
IT IS DELICATETHIS IS BEAUTIFUL
IT IS GORGEOUS
THEY ARE BIRDS
THESE ARE ANIMALS
THEY ARE PEOPLEINDONESIA
SINGAPORE
HYDERABAD
CHENNAI
BANGALORE
MUMBAI
PUNEMBAI
PUNEIT IS GORGEOUSPUNEIT IS GORGEOUSstudent@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt | tr [:lower:] [:upper:]
IT IS DELICATETHIS IS BEAUTIFUL
IT IS GORGEOUS
THEY ARE BIRDS
THESE ARE ANIMALS
THEY ARE PEOPLEINDONESIA
SINGAPORE
HYDERABAD
CHENNAI
BANGALORE
MUMBAI
PUNEMBAI
PUNEIT IS GORGEOUSPUNEIT IS GORGEOUSstudent@ai-HP-ProDesk-600-G4-MT:~/422163$ echo "Welcome To GeeksforGeeks" | tr [:space:]
] "t"
Welcome To GeeksforGeeks student@ai-HP-ProDesk-600-G4-MT:~/422163$ echo "Welcome To GeeksforGeeks" | tr -s " "
Welcome To GeeksforGeeks
student@ai-HP-ProDesk-600-G4-MT:~/422163$ echo "welcome to the party" | tr -d e
wlcom to the party
student@ai-HP-ProDesk-600-G4-MT:~/422163$ echo "-cd [:digit:] <<> "It is 1987"
-cd [:digit:]
student@ai-HP-ProDesk-600-G4-MT:~/422163$ echo "it is 1987" | tr -cd [:digit:]
```

Sort Command

```

Activities Terminal ▾ Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
PUNEIt is GorgeousPUNEIt is Gorgeous:student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
Usage: sort [OPTION]... [FILE]...
      or: sort [OPTION]... -f[ILE]0:from=F
      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
  -l, --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)
  -r, --reverse                reverse the result of comparisons
  --random-source=FILE        get random bytes from FILE
  --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=NERGE          merge at most NERGE 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

Activities Terminal ▾ Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
PUNEIt is GorgeousPUNEIt is Gorgeous:student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
Usage: sort [OPTION]... [FILE]...
      or: sort [OPTION]... -f[ILE]0:from=F
      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
  -l, --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)
  -r, --reverse                reverse the result of comparisons
  --random-source=FILE        get random bytes from FILE
  --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=NERGE          merge at most NERGE 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][,.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

```

```

Activities Terminal ▾ Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
PUNEIt is GorgeousPUNEIt is Gorgeous:student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
Usage: sort [OPTION]... [FILE]...
      or: sort [OPTION]... -f[ILE]0:from=F
      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
  -l, --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)
  -r, --reverse                reverse the result of comparisons
  --random-source=FILE        get random bytes from FILE
  --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=NERGE          merge at most NERGE 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][,.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 [bdgMhNrRv],
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.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/sort>
or available locally via: info '(coreutils) sort invocation'
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort file1.txt
Bangalore
Chennai

```

```

Activities Terminal ▾ Feb 14 16:37
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
PUNEIt is GorgeousPUNEIt is Gorgeous:student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort --help
Usage: sort [OPTION]... [FILE]...
      or: sort [OPTION]... -f[ILE]0:from=F
      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
  -l, --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)
  -r, --reverse                reverse the result of comparisons
  --random-source=FILE        get random bytes from FILE
  --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=NERGE          merge at most NERGE 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][,.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 [bdgMhNrRv],
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.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/sort>
or available locally via: info '(coreutils) sort invocation'
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort file1.txt
Bangalore
Chennai

```

```
Activities Terminal ▾ Feb 14 16:37 student@ai-HP-ProDesk-600-G4-MT:~/422163
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort < file1.txt
Bangalore
Chennai
Hyderabad
It is delicateThis is beautiful
It is Gorgeous
Mumbai
PUNEIt is GorgeousPUNEIt is Gorgeous
PUNEumbai
Singapore
These are animals
They are birds
They are peopleIndonesia
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -r file1.txt
They are peopleIndonesia
They are birds
These are animals
Singapore
PUNEumbai
PUNEIt is GorgeousPUNEIt is Gorgeous
Mumbai
It is Gorgeous
It is delicateThis is beautiful
Hyderabad
Chennai
Bangalore
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -k3 file1.txt
Bangalore
Chennai
Hyderabad
Mumbai
PUNEumbai
Singapore
These are animals
They are birds
They are peopleIndonesia
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -k1 file1.txt
Bangalore
Chennai
Hyderabad
It is delicateThis is beautiful
It is Gorgeous
Mumbai
PUNEIt is GorgeousPUNEIt is Gorgeous
They are peopleIndonesia
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -c file1.txt
sort: file1.txt:4: disorder: These are animals
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -nr file1.txt
They are peopleIndonesia
They are birds
These are animals
Singapore
PUNEumbai
PUNEIt is GorgeousPUNEIt is Gorgeous
Mumbai
It is Gorgeous
It is Gorgeous
```

```
Activities Terminal ▾ Feb 14 16:37 student@ai-HP-ProDesk-600-G4-MT:~/422163
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -k3 file1.txt
Bangalore
Chennai
Hyderabad
Mumbai
PUNEumbai
Singapore
These are animals
They are birds
They are delicateThis is beautiful
It is Gorgeous
PUNEIt is GorgeousPUNEIt is Gorgeous
They are peopleIndonesia
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -k1 file1.txt
Bangalore
Chennai
Hyderabad
It is delicateThis is beautiful
It is Gorgeous
Mumbai
PUNEIt is GorgeousPUNEIt is Gorgeous
PUNEumbai
Singapore
These are animals
They are birds
They are peopleIndonesia
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -c file1.txt
sort: file1.txt:4: disorder: These are animals
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -nr file1.txt
They are peopleIndonesia
They are birds
These are animals
Singapore
PUNEumbai
PUNEIt is GorgeousPUNEIt is Gorgeous
Mumbai
It is Gorgeous
It is Gorgeous
```

```
Activities Terminal ▾ Feb 14 16:37 student@ai-HP-ProDesk-600-G4-MT:~/422163
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -nr file1.txt
They are peopleIndonesia
They are birds
These are animals
Singapore
PUNEumbai
PUNEIt is GorgeousPUNEIt is Gorgeous
Mumbai
It is Gorgeous
It is delicateThis is beautiful
Hyderabad
Chennai
Bangalore
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -o file1.txt file3.txt
sort: cannot read: file3.txt: No such file or directory
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -o file1.txt file2.txt
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -n file1.txt
He is Smart
You are INtelligent
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -n file2.txt
He is Smart
You are INtelligent
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file1.txt
He is Smart
You are INtelligent
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat file2.txt
You are INtelligent
He is Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -u file2.txt
He is Smart
You are INtelligent
student@ai-HP-ProDesk-600-G4-MT:~/422163$ cat>file1.txt
He is Smart
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -u file1.txt
He is Smart
You are INtelligent
student@ai-HP-ProDesk-600-G4-MT:~/422163$ sort -u file1.txt
He is Smart
You are INtelligent
```

```
Activities Terminal ▾ Feb 14 16:37
student@al-HP-ProDesk-600-G4-MT: ~/422163
[1] 11:42:16 sort -u file1.txt
He is Smart
You are INtelligent
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort -z file1.txt
He is Smart
You are INtelligent
[2] 11:42:16 sort -m file1.txt file2.txt
He is Smart
You are INtelligent
He is Smart(student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort -m file1.txt file2.txt
He is Smart
You are INtelligent
He is Smart
You are INtelligent
He is Smart
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort -s file1.txt
He is Smart
He is Smart
You are INtelligent
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort -s file2.txt
He is Smart
You are INtelligent
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort -b file1.txt
He is Smart
He is Smart
You are INtelligent
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort -b 3 file1.txt
sort: cannot read 3: No such file or directory
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort -v file1.txt
sort: invalid option -- 'v'
Try 'sort --help' for more information.
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort --version
sort (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
```

```
Activities Terminal ▾ Feb 14 16:37
student@al-HP-ProDesk-600-G4-MT: ~/422163
[1] 11:42:16 sort --version
Try 'sort --help' for more information.
student@al-HP-ProDesk-600-G4-MT: ~/422163$ sort --version
sort (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.

Written by Mike Haertel and Paul Eggert.
student@al-HP-ProDesk-600-G4-MT: ~/422163$ pr --help
Usage: pr [OPTION]... [FILE]...
Paginate or columnate FILE(s) for printing.

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

Mandatory arguments to long options are mandatory for short options too.
  -FIRST_PAGE[:LAST_PAGE], --pages=FIRST_PAGE[:LAST_PAGE]
    begin [stop] printing with page FIRST_[:LAST_]PAGE
  -COLUMN, --columns=COLUMN
    output COLUMN columns and print columns down,
    unless -a is used. Balance number of lines in the
    columns on each page
  -a, --across
    print columns across rather than down, used together
    with -COLUMN
  -c, --show-control-chars
    use hat notation (^G) and octal backslash notation
  -d, --double-space
    double space the output
  -D, --date-format=FORMAT
    use FORMAT for the header date
  -e[CHAR[WIDTH]], --expand-tabs[=CHAR[WIDTH]]
    expand input CHARs (TABS) to tab WIDTH (8)
  -F, -f, --form-feed
    use form feeds instead of newlines to separate pages
    (by a 3-line page header with -F or a 5-line header
    and trailer without -F)
  -h, --header=HEADER
```

Pr Command

```
Activities Terminal ▾ Feb 14 16:38 student@ai-HP-ProDesk-600-G4-MT: ~/422163
With no FILE, or when FILE is -, read standard input.

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

  --FIRST_PAGE[:LAST_PAGE], --pages=FIRST_PAGE[:LAST_PAGE]
    begin [stop] printing with page FIRST_[:LAST_]PAGE
  -COLUMN, --columns=COLUMN
    output COLUMN columns and print columns down,
    unless -a is used. Balance number of lines in the
    columns on each page
  -a, --across
    print columns across rather than down, used together
    with -COLUMN
  -c, --show-control-chars
    use hat notation (^G) and octal backslash notation
  -d, --double-space
    double space the output
  -D, --date-format=FORMAT
    use FORMAT for the header date
  -e[CHAR[WIDTH]], --expand-tabs[=CHAR[WIDTH]]
    expand input CHARs (TABS) to tab WIDTH (8)
  -F, -f, --form-feed
    use form feeds instead of newlines to separate pages
    (by a 3-line page header with -F or a 5-line header
    and trailer without -F)
  -h, --header=HEADER
    use a centered HEADER instead of filename in page header,
    -h "" prints a blank line, don't use -h ""
  -i[CHAR[WIDTH]], --output-tabs[=CHAR[WIDTH]]
    replace spaces with CHARs (TABS) to tab WIDTH (8)
  -J, --join-lines
    merge full lines, turns off -W line truncation, no column
    alignment, --sep-string[=STRING] sets separators
  -l, --length=PAGE_LENGTH
    set the page length to PAGE_LENGTH (66) lines
    (default number of lines of text 56, and with -F 63).
    implies -t if PAGE_LENGTH <= 10
  -m, --merge
    print all files in parallel, one in each column,
    truncate lines, but join lines of full length with -J
```

```
Activities Terminal ▾ Feb 14 16:38 student@ai-HP-ProDesk-600-G4-MT: ~/422163
      set the page length to PAGE_LENGTH (66) lines
      (default number of lines of text 56, and with -F 63).
      implies -t if PAGE_LENGTH <= 10
  -m, --merge
    print all files in parallel, one in each column,
    truncate lines, but join lines of full length with -J
  -n[SEP[DIGITS]], --number-lines[=SEP[DIGITS]]
    number lines, use DIGITS (5) digits, then SEP (TAB),
    default counting starts with 1st line of input file
  -N, --first-line-number=NUMBER
    start counting with NUMBER at 1st line of first
    page printed (see +FIRST_PAGE)
  -o, --indent=MARGIN
    offset each line with MARGIN (zero) spaces, do not
    affect -w or -W, MARGIN will be added to PAGE_WIDTH
  -r, --no-file-warnings
    omit warning when a file cannot be opened
  -s[CHAR], --separator[=CHAR]
    separate columns by a single character, default for CHAR
    is the <TAB> character without -w and 'no char' with -w.
    -s[CHAR] turns off line truncation of all 3 column
    options (-COLUMN|-a -COLUMN|-m) except -w is set
  -S[STRING], --sep-string[=STRING]
    separate columns by STRING,
    without -S, default separator <TAB> with -J and <space>
    otherwise (same as -S " "), no effect on column options
  -t, --omit-header
    omit page headers and trailers;
    implied if PAGE_LENGTH <= 10
  -T, --omit-pagination
    omit page headers and trailers, eliminate any pagination
    by form feeds set in input files
  -v, --show-nonprinting
    use octal backslash notation
  -w, --width=PAGE_WIDTH
    set page width to PAGE_WIDTH (72) characters for
    multiple text-column output only, -s[char] turns off (72)
  -W, --page-width=PAGE_WIDTH
    set page width to PAGE_WIDTH (72) characters always,
```

```
Activities Terminal ▾ Feb 14 16:38 student@ai-HP-ProDesk-600-G4-MT: ~/422163
  -s[CHAR], --separator[=CHAR]
    separate columns by a single character, default for CHAR
    is the <TAB> character without -w and 'no char' with -w.
    -s[CHAR] turns off line truncation of all 3 column
    options (-COLUMN|-a -COLUMN|-m) except -w is set
  -S[STRING], --sep-string[=STRING]
    separate columns by STRING,
    without -S, default separator <TAB> with -J and <space>
    otherwise (same as -S " "), no effect on column options
  -t, --omit-header
    omit page headers and trailers;
    implied if PAGE_LENGTH <= 10
  -T, --omit-pagination
    omit page headers and trailers, eliminate any pagination
    by form feeds set in input files
  -v, --show-nonprinting
    use octal backslash notation
  -w, --width=PAGE_WIDTH
    set page width to PAGE_WIDTH (72) characters for
    multiple text-column output only, -s[char] turns off (72)
  -W, --page-width=PAGE_WIDTH
    set page width to PAGE_WIDTH (72) characters always,
    truncate lines, except -J option is set, no interference
    with -S or -s
  --help
    display this help and exit
  --version
    output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/pr>
or available locally via: info '(coreutils) pr invocation'
student@ai-HP-ProDesk-600-G4-MT: ~/422163 pr -3 file1.txt

2024-02-14 16:17          file1.txt          Page 1

  He is Smart      You are INtelligent      He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163
or available locally via: info '(coreutils) pr invocation'
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -3 file1.txt

2024-02-14 16:17           file1.txt          Page 1

He is Smart      You are INtelligent      He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163
or available locally via: info '(coreutils) pr invocation'
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -t file1.txt

2024-02-14 16:17           file1.txt          Page 1

He is Smart
You are INtelligent
He is Smart

student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -d file1.txt
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163
or available locally via: info '(coreutils) pr invocation'
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -n file1.txt

2024-02-14 16:17           file1.txt          Page 1

1 He is Smart
2 You are INtelligent
3 He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr --version
pr (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://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.

Written by Pete TerMaat and Roland Huebner.
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -a file1.txt

2024-02-14 16:17          file1.txt          Page 1

He is Smart
You are INtelligent
He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -d file1.txt

2024-02-14 16:17          file1.txt          Page 1

He is Smart
You are INtelligent
He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -m file1.txt file2.txt

2024-02-14 16:27          Page 1

He is Smart          You are INtelligent
You are INtelligent          He is Smart
He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -w file1.txt
pr: '-w PAGE_WIDTH' invalid number of characters: 'file1.txt'
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -w 7 file1.txt

2024-02-14 16:17 file1.txt Page 1

He is Smart
You are INtelligent
He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -e file1.txt

2024-02-14 16:17           file1.txt           Page 1

He is Smart
You are INtelligent
He is Smart
```

```
Activities Terminal ▾ Feb 14 16:38
student@ai-HP-ProDesk-600-G4-MT: ~/422163
student@ai-HP-ProDesk-600-G4-MT: ~/422163$ pr -n file1.txt

2024-02-14 16:17           file1.txt           Page 1

1 He is Smart
2 You are INtelligent
3 He is Smart
```