Rules

Symbol	Description
.	Matches any single character
?	Preceding character is matched zero or once.
+	Preceding character is matched one or more times
*	Preceding character is optional and matched zero or more times
{n}	Preceding character is matched *n* times
{n,}	Preceding character is matched *n* or more times
{n,m}	Preceding character is matched at least *n* times, but no more than *m* times
{,m}	Preceding character is matched no more than *m* times

Range Expressions

Range	Description
[a-z]	Lower Case
[A-Z]	Upper Case
[a-d]	All lower case letters: {a,b,c,d}
[0-9]	All digits
[A-Z0-9]	Combined. All uppercase letters and digits
'witch'	Searches for character sequence 'witch'
[a-z][a-z]	Searches for two letter sequences (that may be part of a longer sequence) in lower case
[A-Z][a-z]	Search for two letter sequences (that may be part of a longer sequence), starting with an uppercase letter and followed by a lower case letter.

Classes

Range	Description
[:alpha:]	Alphabetical characters [a-Z]
[:digit:]	Numerical characters [0-9]
[:alnum:]	Alpha-numerical characters[a-Z0-9]
[:punct:]	Punctuation characters
[:blank:]	Blank characters
[:lower:]	Lower case [a-z]
[:upper:]	Upper case [A-Z]
[:xdigit:]	Hexidecimal digits
[:print:]	Printable characters

Meta Characters

Range	Description
[]	Bracketed expression
-	range, e.g., a-z
^	new line
^	Not in range, e.g., not a vowel [^aeiou]
$	End of line

Backslash

Range	Description
\b	Match empty string at edge of word
\B	Match empty string providing it is not at edge of a word
\ <	Match empty string at beginning of word
\ >	Match empty string at end of word
\w	Match word constituent
\W	Match non-word constituent
\s	Match space
\S	Match non-space
\\|	Logical OR

grep –help I

```
Usage: grep [OPTION]... PATTERN [FILE]...
Search for PATTERN in each FILE.
Example: grep -i 'hello world' menu.h main.c

Pattern selection and interpretation:
  -E, --extended-regexp     PATTERN is an extended regular expression
  -F, --fixed-strings       PATTERN is a set of newline-separated strings
  -G, --basic-regexp        PATTERN is a basic regular expression (default)
  -P, --perl-regexp         PATTERN is a Perl regular expression
  -e, --regexp=PATTERN      use PATTERN for matching
  -f, --file=FILE           obtain PATTERN from FILE
  -i, --ignore-case         ignore case distinctions
  -w, --word-regexp         force PATTERN to match only whole words
  -x, --line-regexp         force PATTERN to match only whole lines
  -z, --null-data           a data line ends in 0 byte, not newline

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
      --line-buffered       flush output on every line
  -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
```

grep –help II

```
  -o, --only-matching       show only the part of a line matching PATTERN
  -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=FILE_PATTERN  search only files that match FILE_PATTERN
      --exclude=FILE_PATTERN  skip files and directories matching FILE_PATTERN
      --exclude-from=FILE   skip files matching any file pattern from FILE
      --exclude-dir=PATTERN  directories that match PATTERN will be skipped.
  -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)
  -Z, --null                print 0 byte after FILE name

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],
      --colour[=WHEN]       use markers to highlight the matching strings;
                            WHEN is ’always’, ’never’, or ’auto’
```

grep –help III

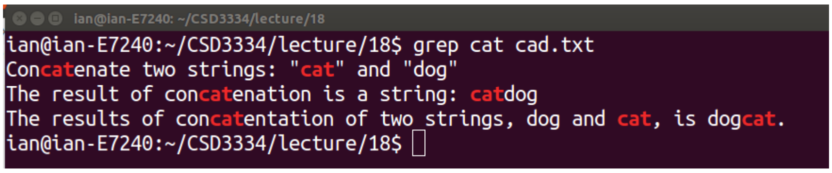
```
  -U, --binary              do not strip CR characters at EOL (MSDOS/Windows)

When FILE is ’-’, read standard input.  With no FILE, read ’.’ if
recursive, ’-’ otherwise.  With fewer than two FILEs, assume -h.
Exit status is 0 if any line 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: <http://www.gnu.org/gethelp/>
```

Example text

filename: `cad.txt`

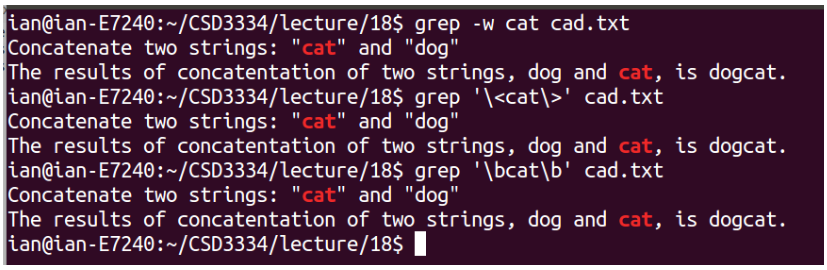
```
Concatenate two strings: "cat" and "dog"
The result of concatenation is a string: catdog
The results of concatentation of two strings, dog and cat, is dogcat.
```

Find occurrences of “cat”



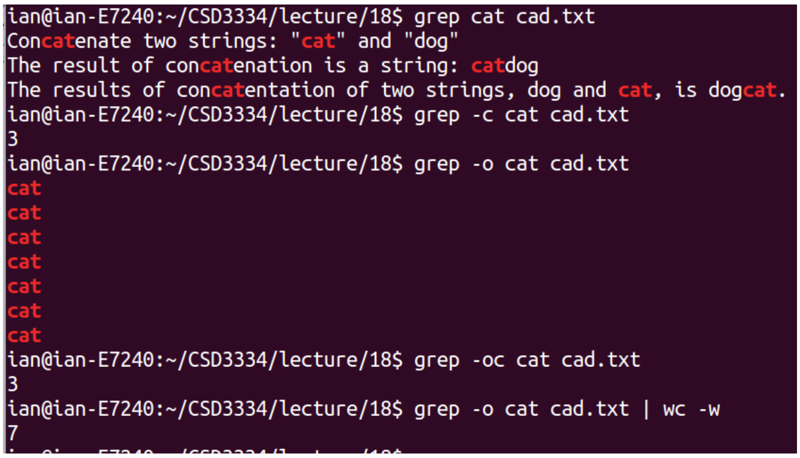
```
ian@ian-E7240:~/CSD3334/lecture/18$ grep cat cad.txt
Concatenate two strings: "cat" and "dog"
The result of concatenation is a string: catdog
The results of concatentation of two strings, dog and cat, is dogcat.
ian@ian-E7240:~/CSD3334/lecture/18$ 
```

Occurrences of cat string exact, empty string either side, not in a word like concatenate



```
ian@ian-E7240:~/CSD3334/lecture/18$ grep -w cat cad.txt
Concatenate two strings: "cat" and "dog"
The results of concatentation of two strings, dog and cat, is dogcat.
ian@ian-E7240:~/CSD3334/lecture/18$ grep '\<cat\>' cad.txt
Concatenate two strings: "cat" and "dog"
The results of concatentation of two strings, dog and cat, is dogcat.
ian@ian-E7240:~/CSD3334/lecture/18$ grep '\bcat\b' cad.txt
Concatenate two strings: "cat" and "dog"
The results of concatentation of two strings, dog and cat, is dogcat.
ian@ian-E7240:~/CSD3334/lecture/18$ 
```

Count occurrences of “cat”



```
ian@ian-E7240:~/CSD3334/lecture/18$ grep cat cad.txt
Concatenate two strings: "cat" and "dog"
The result of concatenation is a string: catdog
The results of concatentation of two strings, dog and cat, is dogcat.
ian@ian-E7240:~/CSD3334/lecture/18$ grep -c cat cad.txt
3
ian@ian-E7240:~/CSD3334/lecture/18$ grep -o cat cad.txt
cat
cat
cat
cat
cat
cat
cat
ian@ian-E7240:~/CSD3334/lecture/18$ grep -oc cat cad.txt
3
ian@ian-E7240:~/CSD3334/lecture/18$ grep -o cat cad.txt | wc -w
7
ian@ian-E7240:~/CSD3334/lecture/18$
```