`grep`, print lines matching a pattern

version 2.5.3, 18 August 2006

Alain Magloire et al.

Published by the Free Software Foundation,
51 Franklin Street – Fifth Floor,
Boston, MA 02110–1301, USA

1 Introduction

`grep` searches the input files for lines containing a match to a given pattern list. When it finds a match in a line, it copies the line to standard output (by default), or produces whatever other sort of output you have requested with options.

Though `grep` expects to do the matching on text, it has no limits on input line length other than available memory, and it can match arbitrary characters within a line. If the final byte of an input file is not a newline, `grep` silently supplies one. Since newline is also a separator for the list of patterns, there is no way to match newline characters in a text.

2 Invoking `grep`

The general synopsis of the `grep` command line is

```
grep options pattern input_file_names
```

There can be zero or more *options*. *pattern* will only be seen as such (and not as an *input_file_name*) if it wasn't already specified within *options* (by using the ‘`-e pattern`’ or ‘`-f file`’ options). There can be zero or more *input_file_names*.

2.1 Command-line Options

`grep` comes with a rich set of options: some from POSIX.2 and some being GNU extensions. Long option names are always a GNU extension, even for options that are from POSIX specifications. Options that are specified by POSIX, under their short names, are explicitly marked as such to facilitate POSIX-portable programming. A few option names are provided for compatibility with older or more exotic implementations.

Several additional options control which variant of the `grep` matching engine is used. See Chapter 4 [grep Programs], page 10.

2.1.1 Generic Program Information

‘`--help`’
Print a usage message briefly summarizing the command-line options and the bug-reporting address, then exit.

‘`-V`’
‘`--version`’
Print the version number of `grep` to the standard output stream. This version number should be included in all bug reports.

2.1.2 Matching Control

‘`-e pattern`’
‘`--regexp=pattern`’
Use *pattern* as the pattern; useful to protect patterns beginning with a ‘`-`’. (‘`-e`’ is specified by POSIX.)

‘`-f file`’
‘`--file=file`’
Obtain patterns from *file*, one per line. The empty file contains zero patterns, and therefore matches nothing. (‘`-f`’ is specified by POSIX.)

‘`-i`’
‘`-y`’
‘`--ignore-case`’

Ignore case distinctions in both the pattern and the input files. ‘`-y`’ is an obsolete synonym that is provided for compatibility. (‘`-i`’ is specified by POSIX.)

‘`-v`’
‘`--invert-match`’

Invert the sense of matching, to select non-matching lines. (‘`-v`’ is specified by POSIX.)

‘`-w`’
‘`--word-regexp`’

Select only those lines containing matches that form whole words. The test is that the matching substring must either be at the beginning of the line, or preceded by a non-word constituent character. Similarly, it must be either at the end of the line or followed by a non-word constituent character. Word-constituent characters are letters, digits, and the underscore.

‘`-x`’
‘`--line-regexp`’

Select only those matches that exactly match the whole line. (‘`-x`’ is specified by POSIX.)

2.1.3 General Output Control

‘`-c`’
‘`--count`’

Suppress normal output; instead print a count of matching lines for each input file. With the ‘`-v`’, ‘`--invert-match`’ option, count non-matching lines. (‘`-c`’ is specified by POSIX.)

‘`--color[=`*WHEN*`]`’
‘`--colour[=`*WHEN*`]`’

Surround the matched (non-empty) strings, matching lines, context lines, file names, line numbers, byte offsets, and separators (for fields and groups of context lines) with escape sequences to display them in color on the terminal. The colors are defined by the environment variable *GREP_COLORS* and default to ‘`ms=01;31:mc=01;31:sl=:cx=:fn=35:ln=32:bn=32:se=36`’ for bold red matched text, magenta file names, green line numbers, green byte offsets, cyan separators, and default terminal colors otherwise. The deprecated environment variable *GREP_COLOR* is still supported, but its setting does not have priority; it defaults to ‘01;31’ (bold red) which only covers the color for matched text. *WHEN* is ‘`never`’, ‘`always`’, or ‘`auto`’.

‘`-L`’
‘`--files-without-match`’

Suppress normal output; instead print the name of each input file from which no output would normally have been printed. The scanning of every file will stop on the first match.

‘`-l`’
‘`--files-with-matches`’

Suppress normal output; instead print the name of each input file from which output would normally have been printed. The scanning of every file will stop on the first match. (‘`-l`’ is specified by POSIX.)

‘`-m` *num*’
‘`--max-count=`*num*’

Stop reading a file after *num* matching lines. If the input is standard input from a regular file, and *num* matching lines are output, `grep` ensures that the standard input is positioned just after the last matching line before exiting, regardless of the presence of trailing context lines. This enables a calling process to resume a search. For example, the following shell script makes use of it:

```
while grep -m 1 PATTERN
do
  echo xxxx
done < FILE
```

But the following probably will not work because a pipe is not a regular file:

```
# This probably will not work.
cat FILE |
while grep -m 1 PATTERN
do
  echo xxxx
done
```

When `grep` stops after *num* matching lines, it outputs any trailing context lines. Since context does not include matching lines, `grep` will stop when it encounters another matching line. When the ‘`-c`’ or ‘`--count`’ option is also used, `grep` does not output a count greater than *num*. When the ‘`-v`’ or ‘`--invert-match`’ option is also used, `grep` stops after outputting *num* non-matching lines.

‘`-o`’
‘`--only-matching`’

Print only the matched (non-empty) parts of matching lines, with each such part on a separate output line.

‘`-q`’
‘`--quiet`’
‘`--silent`’

Quiet; do not write anything to standard output. Exit immediately with zero status if any match is found, even if an error was detected. Also see the ‘`-s`’ or ‘`--no-messages`’ option. (‘`-q`’ is specified by POSIX.)

‘`-s`’
‘`--no-messages`’

Suppress error messages about nonexistent or unreadable files. Portability note: unlike GNU `grep`, 7th Edition Unix `grep` did not conform to POSIX, because it lacked ‘`-q`’ and its ‘`-s`’ option behaved like GNU `grep`’s ‘`-q`’ option. USG-style `grep` also lacked ‘`-q`’ but its ‘`-s`’ option behaved like GNU `grep`’s. Portable shell scripts should avoid both '`-q`' and '`-s`' and should redirect standard and error output to '`/dev/null`' instead. ('`-s`' is specified by POSIX.)

2.1.4 Output Line Prefix Control

When several prefix fields are to be output, the order is always file name, line number, and byte offset, regardless of the order in which these options were specified.

'`-b`'
'`--byte-offset`'

Print the 0-based byte offset within the input file before each line of output. If '`-o`' ('`--only-matching`') is specified, print the offset of the matching part itself. When `grep` runs on MS-DOS or MS-Windows, the printed byte offsets depend on whether the '`-u`' ('`--unix-byte-offsets`') option is used; see below.

'`-H`'
'`--with-filename`'

Print the file name for each match. This is the default when there is more than one file to search.

'`-h`'
'`--no-filename`'

Suppress the prefixing of file names on output. This is the default when there is only one file (or only standard input) to search.

'`--label=`*LABEL*'

Display input actually coming from standard input as input coming from file *LABEL*. This is especially useful for tools like `zgrep`; e.g.:

```
gzip -cd foo.gz | grep --label=foo something
```

'`-n`'
'`--line-number`'

Prefix each line of output with the 1-based line number within its input file. ('`-n`' is specified by POSIX.)

'`-T`'
'`--initial-tab`'

Make sure that the first character of actual line content lies on a tab stop, so that the alignment of tabs looks normal. This is useful with options that prefix their output to the actual content: '`-H`', '`-n`', and '`-b`'. In order to improve the probability that lines from a single file will all start at the same column, this also causes the line number and byte offset (if present) to be printed in a minimum-size field width.

'`-u`'
'`--unix-byte-offsets`'

Report Unix-style byte offsets. This option causes `grep` to report byte offsets as if the file were a Unix-style text file, i.e., the byte offsets ignore the `CR` characters that were stripped. This will produce results identical to running `grep` on a Unix machine. This option has no effect unless the '`-b`' option is also used; it has no effect on platforms other than MS-DOS and MS-Windows.

‘`-Z`’
‘`--null`’ Output a zero byte (the ASCII `NUL` character) instead of the character that normally follows a file name. For example, ‘`grep -lZ`’ outputs a zero byte after each file name instead of the usual newline. This option makes the output unambiguous, even in the presence of file names containing unusual characters like newlines. This option can be used with commands like ‘`find -print0`’, ‘`perl -0`’, ‘`sort -z`’, and ‘`xargs -0`’ to process arbitrary file names, even those that contain newline characters.

2.1.5 Context Line Control

Regardless of how these options are set, `grep` will never print any given line more than once. If the ‘`-o`’ or ‘`--only-matching`’ option is specified, these options have no effect and a warning is given upon their use.

‘`-A` *num*’
‘`--after-context=`*num*’
Print *num* lines of trailing context after matching lines.

‘`-B` *num*’
‘`--before-context=`*num*’
Print *num* lines of leading context before matching lines.

‘`-C` *num*’
‘`-`*num*’
‘`--context=`*num*’
Print *num* lines of leading and trailing output context.

Matching lines normally use ‘`:`’ as a separator between prefix fields and actual line content. Context (i.e., non-matching) lines use ‘`-`’ instead. When no context is specified, matching lines are simply output one right after another. When nonzero context is specified, lines that are adjacent in the input form a group and are output one right after another, but disjoint groups of lines are separated by a ‘`--`’ without any prefix and on a line of its own. Each group may contain several matching lines when they are close enough to each other that two otherwise adjacent but divided groups connect and can just merge into a single contiguous one.

2.1.6 File and Directory Selection

‘`-a`’
‘`--text`’ Process a binary file as if it were text; this is equivalent to the ‘`--binary-files=text`’ option.

‘`--binary-files=`*type*’
If the first few bytes of a file indicate that the file contains binary data, assume that the file is of type *type*. By default, *type* is ‘`binary`’, and `grep` normally outputs either a one-line message saying that a binary file matches, or no message if there is no match. If *type* is ‘`without-match`’, `grep` assumes that a binary file does not match; this is equivalent to the ‘`-I`’ option. If *type* is ‘`text`’, `grep` processes a binary file as if it were text; this is equivalent to the ‘`-a`’ option. *Warning:* ‘`--binary-files=text`’ might output binary garbage,

which can have nasty side effects if the output is a terminal and if the terminal driver interprets some of it as commands.

‘`-D` *`action`*’
‘`--devices=`*`action`*’
If an input file is a device, FIFO, or socket, use *action* to process it. By default, *action* is ‘`read`’, which means that devices are read just as if they were ordinary files. If *action* is ‘`skip`’, devices, FIFOs, and sockets are silently skipped.

‘`-d` *`action`*’
‘`--directories=`*`action`*’
If an input file is a directory, use *action* to process it. By default, *action* is ‘`read`’, which means that directories are read just as if they were ordinary files (some operating systems and file systems disallow this, and will cause `grep` to print error messages for every directory or silently skip them). If *action* is ‘`skip`’, directories are silently skipped. If *action* is ‘`recurse`’, `grep` reads all files under each directory, recursively; this is equivalent to the ‘`-r`’ option.

‘`--exclude=`*`glob`*’
Skip files whose base name matches *glob* (using wildcard matching). A file-name glob can use ‘`*`’, ‘`?`’, and ‘`[`’...‘`]`’ as wildcards, and `\` to quote a wildcard or backslash character literally.

‘`--exclude-from=`*`file`*’
Skip files whose base name matches any of the file-name globs read from *file* (using wildcard matching as described under ‘`--exclude`’).

‘`--exclude-dir=`*`dir`*’
Exclude directories matching the pattern *dir* from recursive directory searches.

‘`-I`’ Process a binary file as if it did not contain matching data; this is equivalent to the ‘`--binary-files=without-match`’ option.

‘`--include=`*`glob`*’
Search only files whose base name matches *glob* (using wildcard matching as described under ‘`--exclude`’).

‘`-r`’
‘`-R`’
‘`--recursive`’
For each directory mentioned on the command line, read and process all files in that directory, recursively. This is the same as the ‘`--directories=recurse`’ option.

2.1.7 Other Options

‘`--line-buffered`’
Use line buffering on output. This can cause a performance penalty.

‘`--mmap`’ If possible, use the `mmap` system call to read input, instead of the default `read` system call. In some situations, ‘`--mmap`’ yields better performance. However, ‘`--mmap`’ can cause undefined behavior (including core dumps) if an input file shrinks while `grep` is operating, or if an I/O error occurs.

‘`-U`’
‘`--binary`’

Treat the file(s) as binary. By default, under MS-DOS and MS-Windows, `grep` guesses the file type by looking at the contents of the first 32kB read from the file. If `grep` decides the file is a text file, it strips the `CR` characters from the original file contents (to make regular expressions with `^` and `$` work correctly). Specifying ‘`-U`’ overrules this guesswork, causing all files to be read and passed to the matching mechanism verbatim; if the file is a text file with `CR/LF` pairs at the end of each line, this will cause some regular expressions to fail. This option has no effect on platforms other than MS-DOS and MS-Windows.

‘`-z`’
‘`--null-data`’

Treat the input as a set of lines, each terminated by a zero byte (the ASCII `NUL` character) instead of a newline. Like the ‘`-Z`’ or ‘`--null`’ option, this option can be used with commands like ‘`sort -z`’ to process arbitrary file names.

2.2 Environment Variables

The behavior of `grep` is affected by the following environment variables.

The locale for category `LC_`*`foo`* is specified by examining the three environment variables `LC_ALL`, `LC_`*`foo`*, and `LANG`, in that order. The first of these variables that is set specifies the locale. For example, if `LC_ALL` is not set, but `LC_MESSAGES` is set to ‘`pt_BR`’, then the Brazilian Portuguese locale is used for the `LC_MESSAGES` category. The ‘`C`’ locale is used if none of these environment variables are set, if the locale catalog is not installed, or if `grep` was not compiled with national language support (NLS).

`GREP_OPTIONS`

This variable specifies default options to be placed in front of any explicit options. For example, if `GREP_OPTIONS` is ‘`--binary-files=without-match --directories=skip`’, `grep` behaves as if the two options ‘`--binary-files=without-match`’ and ‘`--directories=skip`’ had been specified before any explicit options. Option specifications are separated by whitespace. A backslash escapes the next character, so it can be used to specify an option containing whitespace or a backslash.

`GREP_COLOR`

This variable specifies the color used to highlight matched (non-empty) text. It is deprecated in favor of `GREP_COLORS`, but still supported. The ‘`mt`’, ‘`ms`’, and ‘`mc`’ capabilities of `GREP_COLORS` have priority over it. It can only specify the color used to highlight the matching non-empty text in any matching line (a selected line when the ‘`-v`’ command-line option is omitted, or a context line when ‘`-v`’ is specified). The default is ‘`01;31`’, which means a bold red foreground text on the terminal’s default background.

`GREP_COLORS`

This variable specifies the colors and other attributes used to highlight various parts of the output. Its value is a colon-separated list of capabilities that defaults to ‘`ms=01;31:mc=01;31:sl=:cx=:fn=35:ln=32:bn=32:se=36`’ with the ‘`rv`’ and ‘`ne`’ boolean capabilities omitted (i.e., false). Supported capabilities are as follows.

`sl=`	SGR substring for whole selected lines (i.e., matching lines when the ‘`-v`’ command-line option is omitted, or non-matching lines when ‘`-v`’ is specified). If however the boolean ‘`rv`’ capability and the ‘`-v`’ command-line option are both specified, it applies to context matching lines instead. The default is empty (i.e., the terminal’s default color pair).
`cx=`	SGR substring for whole context lines (i.e., non-matching lines when the ‘`-v`’ command-line option is omitted, or matching lines when ‘`-v`’ is specified). If however the boolean ‘`rv`’ capability and the ‘`-v`’ command-line option are both specified, it applies to selected non-matching lines instead. The default is empty (i.e., the terminal’s default color pair).
`rv`	Boolean value that reverses (swaps) the meanings of the ‘`sl=`’ and ‘`cx=`’ capabilities when the ‘`-v`’ command-line option is specified. The default is false (i.e., the capability is omitted).
`mt=01;31`	SGR substring for matching non-empty text in any matching line (i.e., a selected line when the ‘`-v`’ command-line option is omitted, or a context line when ‘`-v`’ is specified). Setting this is equivalent to setting both ‘`ms=`’ and ‘`mc=`’ at once to the same value. The default is a bold red text foreground over the current line background.
`ms=01;31`	SGR substring for matching non-empty text in a selected line. (This is only used when the ‘`-v`’ command-line option is omitted.) The effect of the ‘`sl=`’ (or ‘`cx=`’ if ‘`rv`’) capability remains active when this kicks in. The default is a bold red text foreground over the current line background.
`mc=01;31`	SGR substring for matching non-empty text in a context line. (This is only used when the ‘`-v`’ command-line option is specified.) The effect of the ‘`cx=`’ (or ‘`sl=`’ if ‘`rv`’) capability remains active when this kicks in. The default is a bold red text foreground over the current line background.
`fn=35`	SGR substring for file names prefixing any content line. The default is a magenta text foreground over the terminal’s default background.
`ln=32`	SGR substring for line numbers prefixing any content line. The default is a green text foreground over the terminal’s default background.
`bn=32`	SGR substring for byte offsets prefixing any content line. The default is a green text foreground over the terminal’s default background.
`se=36`	SGR substring for separators that are inserted between selected line fields (‘`:`’), between context line fields (‘`-`’), and between groups of

adjacent lines when nonzero context is specified (‘`--`’). The default is a cyan text foreground over the terminal’s default background.

`ne` Boolean value that prevents clearing to the end of line using Erase in Line (EL) to Right (‘`\33[K`’) each time a colorized item ends. This is needed on terminals on which EL is not supported. It is otherwise useful on terminals for which the `back_color_erase` (`bce`) boolean terminfo capability does not apply, when the chosen highlight colors do not affect the background, or when EL is too slow or causes too much flicker. The default is false (i.e., the capability is omitted).

Note that boolean capabilities have no ‘`=`’... part. They are omitted (i.e., false) by default and become true when specified.

See the Select Graphic Rendition (SGR) section in the documentation of your text terminal for permitted values and their meaning as character attributes. These substring values are integers in decimal representation and can be concatenated with semicolons. `grep` takes care of assembling the result into a complete SGR sequence (‘`\33[`’...‘`m`’). Common values to concatenate include ‘`1`’ for bold, ‘`4`’ for underline, ‘`5`’ for blink, ‘`7`’ for inverse, ‘`39`’ for default foreground color, ‘`30`’ to ‘`37`’ for foreground colors, ‘`90`’ to ‘`97`’ for 16-color mode foreground colors, ‘`38;5;0`’ to ‘`38;5;255`’ for 88-color and 256-color modes foreground colors, ‘`49`’ for default background color, ‘`40`’ to ‘`47`’ for background colors, ‘`100`’ to ‘`107`’ for 16-color mode background colors, and ‘`48;5;0`’ to ‘`48;5;255`’ for 88-color and 256-color modes background colors.

`LC_ALL`
`LC_COLLATE`
`LANG` These variables specify the locale for the `LC_COLLATE` category, which determines the collating sequence used to interpret range expressions like ‘`[a-z]`’.

`LC_ALL`
`LC_CTYPE`
`LANG` These variables specify the locale for the `LC_CTYPE` category, which determines the type of characters, e.g., which characters are whitespace.

`LC_ALL`
`LC_MESSAGES`
`LANG` These variables specify the locale for the `LC_MESSAGES` category, which determines the language that `grep` uses for messages. The default ‘`C`’ locale uses American English messages.

`POSIXLY_CORRECT`

If set, `grep` behaves as POSIX.2 requires; otherwise, `grep` behaves more like other GNU programs. POSIX.2 requires that options that follow file names must be treated as file names; by default, such options are permuted to the front of the operand list and are treated as options. Also, POSIX.2 requires that unrecognized options be diagnosed as “illegal”, but since they are not really against the law the default is to diagnose them as “invalid”. `POSIXLY_CORRECT` also disables `_`*N*`_GNU_nonoption_argv_flags_`, described below.

`_N_GNU_nonoption_argv_flags_`

(Here *N* is `grep`'s numeric process ID.) If the *i*th character of this environment variable's value is '`1`', do not consider the *i*th operand of `grep` to be an option, even if it appears to be one. A shell can put this variable in the environment for each command it runs, specifying which operands are the results of file name wildcard expansion and therefore should not be treated as options. This behavior is available only with the GNU C library, and only when `POSIXLY_CORRECT` is not set.

3 Exit Status

Normally, the exit status is 0 if selected lines are found and 1 otherwise. But the exit status is 2 if an error occurred, unless the '`-q`' or '`--quiet`' or '`--silent`' option is used and a selected line is found. Note, however, that POSIX only mandates, for programs such as `grep`, `cmp`, and `diff`, that the exit status in case of error be greater than 1; it is therefore advisable, for the sake of portability, to use logic that tests for this general condition instead of strict equality with 2.

4 grep programs

`grep` searches the named input files (or standard input if no files are named, or the file name '`-`' is given) for lines containing a match to the given pattern. By default, `grep` prints the matching lines. There are four major variants of `grep`, controlled by the following options.

'`-G`'
'`--basic-regexp`'

Interpret the pattern as a basic regular expression (BRE). This is the default.

'`-E`'
'`--extended-regexp`'

Interpret the pattern as an extended regular expression (ERE). ('`-E`' is specified by POSIX.)

'`-F`'
'`--fixed-strings`'

Interpret the pattern as a list of fixed strings, separated by newlines, any of which is to be matched. ('`-F`' is specified by POSIX.)

'`-P`'
'`--perl-regexp`'

Interpret the pattern as a Perl regular expression. This is highly experimental and '`grep -P`' may warn of unimplemented features.

In addition, two variant programs `egrep` and `fgrep` are available. `egrep` is the same as '`grep -E`'. `fgrep` is the same as '`grep -F`'. Direct invocation as either `egrep` or `fgrep` is deprecated, but is provided to allow historical applications that rely on them to run unmodified.

5 Regular Expressions

A *regular expression* is a pattern that describes a set of strings. Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions. `grep` understands two different versions of regular expression syntax: “basic”(BRE) and “extended”(ERE). In GNU `grep`, there is no difference in available functionality using either syntax. In other implementations, basic regular expressions are less powerful. The following description applies to extended regular expressions; differences for basic regular expressions are summarized afterwards.

5.1 Fundamental Structure

The fundamental building blocks are the regular expressions that match a single character. Most characters, including all letters and digits, are regular expressions that match themselves. Any meta-character with special meaning may be quoted by preceding it with a backslash.

A regular expression may be followed by one of several repetition operators:

‘`.`’	The period ‘`.`’ matches any single character.
‘`?`’	The preceding item is optional and will be matched at most once.
‘`*`’	The preceding item will be matched zero or more times.
‘`+`’	The preceding item will be matched one or more times.
‘`{`*n*`}`’	The preceding item is matched exactly *n* times.
‘`{`*n*`,}`’	The preceding item is matched *n* or more times.
‘`{,`*m*`}`’	The preceding item is matched at most *m* times.
‘`{`*n*`,`*m*`}`’	The preceding item is matched at least *n* times, but not more than *m* times.

Two regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating two substrings that respectively match the concatenated expressions.

Two regular expressions may be joined by the infix operator ‘`|`’; the resulting regular expression matches any string matching either alternalte expression.

Repetition takes precedence over concatenation, which in turn takes precedence over alternation. A whole expression may be enclosed in parentheses to override these precedence rules and form a subexpression.

5.2 Character Classes and Bracket Expressions

A *bracket expression* is a list of characters enclosed by ‘`[`’ and ‘`]`’. It matches any single character in that list; if the first character of the list is the caret ‘`^`’, then it matches any character **not** in the list. For example, the regular expression ‘`[0123456789]`’ matches any single digit.

Within a bracket expression, a *range expression* consists of two characters separated by a hyphen. It matches any single character that sorts between the two characters, inclusive, using the locale’s collating sequence and character set. For example, in the default C locale, ‘`[a-d]`’ is equivalent to ‘`[abcd]`’. Many locales sort characters in dictionary order, and in these locales ‘`[a-d]`’ is typically not equivalent to ‘`[abcd]`’; it might be equivalent to ‘`[aBbCcDd]`’, for example. To obtain the traditional interpretation of bracket expressions, you can use the ‘`C`’ locale by setting the `LC_ALL` environment variable to the value ‘`C`’.

Finally, certain named classes of characters are predefined within bracket expressions, as follows. Their interpretation depends on the `LC_CTYPE` locale; the interpretation below is that of the ‘`C`’ locale, which is the default if no `LC_CTYPE` locale is specified.

‘`[:alnum:]`’

  Alphanumeric characters: ‘`[:alpha:]`’ and ‘`[:digit:]`’.

‘`[:alpha:]`’

  Alphabetic characters: ‘`[:lower:]`’ and ‘`[:upper:]`’.

‘`[:blank:]`’

  Blank characters: space and tab.

‘`[:cntrl:]`’

  Control characters. In ASCII, these characters have octal codes 000 through 037, and 177 (`DEL`). In other character sets, these are the equivalent characters, if any.

‘`[:digit:]`’

  Digits: `0 1 2 3 4 5 6 7 8 9`.

‘`[:graph:]`’

  Graphical characters: ‘`[:alnum:]`’ and ‘`[:punct:]`’.

‘`[:lower:]`’

  Lower-case letters: `a b c d e f g h i j k l m n o p q r s t u v w x y z`.

‘`[:print:]`’

  Printable characters: ‘`[:alnum:]`’, ‘`[:punct:]`’, and space.

‘`[:punct:]`’

  Punctuation characters: ``! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~``.

‘`[:space:]`’

  Space characters: tab, newline, vertical tab, form feed, carriage return, and space.

‘`[:upper:]`’

  Upper-case letters: `A B C D E F G H I J K L M N O P Q R S T U V W X Y Z`.

‘`[:xdigit:]`’

  Hexadecimal digits: `0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f`.

For example, ‘`[[:alnum:]]`’ means ‘`[0-9A-Za-z]`’, except the latter depends upon the ‘`C`’ locale and the ASCII character encoding, whereas the former is independent of locale and character set. (Note that the brackets in these class names are part of the symbolic names, and must be included in addition to the brackets delimiting the bracket expression.)

Most meta-characters lose their special meaning inside bracket expressions.

‘`]`’	ends the bracket expression if it’s not the first list item. So, if you want to make the ‘`]`’ character a list item, you must put it first.
‘`[.`’	represents the open collating symbol.
‘`.]`’	represents the close collating symbol.
‘`[=`’	represents the open equivalence class.
‘`=]`’	represents the close equivalence class.
‘`[:`’	represents the open character class symbol, and should be followed by a valid character class name.
‘`:]`’	represents the close character class symbol.
‘`-`’	represents the range if it’s not first or last in a list or the ending point of a range.
‘`^`’	represents the characters not in the list. If you want to make the ‘`^`’ character a list item, place it anywhere but first.

5.3 The Backslash Character and Special Expressions

The ‘\’ character, when followed by certain ordinary characters, takes a special meaning:

‘‘`\b`’’	Match the empty string at the edge of a word.
‘‘`\B`’’	Match the empty string provided it’s not at the edge of a word.
‘‘`\<`’’	Match the empty string at the beginning of word.
‘‘`\>`’’	Match the empty string at the end of word.
‘‘`\w`’’	Match word constituent, it is a synonym for ‘`[[:alnum:]]`’.
‘‘`\W`’’	Match non-word constituent, it is a synonym for ‘`[^[:alnum:]]`’.

For example, ‘`\brat\b`’ matches the separate word ‘`rat`’, ‘`\Brat\B`’ matches ‘`crate`’ but not ‘`furry rat`’.

5.4 Anchoring

The caret ‘`^`’ and the dollar sign ‘`$`’ are meta-characters that respectively match the empty string at the beginning and end of a line.

5.5 Back-references and Subexpressions

The back-reference ‘`\`*n*’, where *n* is a single digit, matches the substring previously matched by the *n*th parenthesized subexpression of the regular expression. For example, ‘`(a)\1`’ matches ‘`aa`’. When used with alternation, if the group does not participate in the match then the back-reference makes the whole match fail. For example, ‘`a(.)|b\1`’ will not match ‘`ba`’. When multiple regular expressions are given with ‘`-e`’ or from a file (‘`-f file`’), back-references are local to each expression.

5.6 Basic vs Extended Regular Expressions

In basic regular expressions the meta-characters ‘`?`’, ‘`+`’, ‘`{`’, ‘`|`’, ‘`(`’, and ‘`)`’ lose their special meaning; instead use the backslashed versions ‘`\?`’, ‘`\+`’, ‘`\{`’, ‘`\|`’, ‘`\(`’, and ‘`\)`’.

Traditional `egrep` did not support the ‘`{`’ meta-character, and some `egrep` implementations support ‘`\{`’ instead, so portable scripts should avoid ‘`{`’ in ‘`grep -E`’ patterns and should use ‘`[{]`’ to match a literal ‘`{`’.

GNU `grep -E` attempts to support traditional usage by assuming that ‘`{`’ is not special if it would be the start of an invalid interval specification. For example, the command ‘`grep -E '{1'`’ searches for the two-character string ‘`{1`’ instead of reporting a syntax error in the regular expression. POSIX.2 allows this behavior as an extension, but portable scripts should avoid it.

6 Usage

Here is an example command that invokes GNU `grep`:

```
grep -i 'hello.*world' menu.h main.c
```

This lists all lines in the files ‘`menu.h`’ and ‘`main.c`’ that contain the string ‘`hello`’ followed by the string ‘`world`’; this is because ‘`.*`’ matches zero or more characters within a line. See Chapter 5 [Regular Expressions], page 11. The ‘`-i`’ option causes `grep` to ignore case, causing it to match the line ‘`Hello, world!`’, which it would not otherwise match. See Chapter 2 [Invoking], page 1, for more details about how to invoke `grep`.

Here are some common questions and answers about `grep` usage.

1. How can I list just the names of matching files?

   ```
   grep -l 'main' *.c
   ```

   lists the names of all C files in the current directory whose contents mention ‘`main`’.

2. How do I search directories recursively?

   ```
   grep -r 'hello' /home/gigi
   ```

   searches for ‘`hello`’ in all files under the ‘`/home/gigi`’ directory. For more control over which files are searched, use `find`, `grep`, and `xargs`. For example, the following command searches only C files:

   ```
   find /home/gigi -name '*.c' -print0 | xargs -0r grep -H 'hello'
   ```

   This differs from the command:

   ```
   grep -rH 'hello' *.c
   ```

   which merely looks for ‘`hello`’ in all files in the current directory whose names end in ‘`.c`’. Here the ‘`-r`’ is probably unnecessary, as recursion occurs only in the unlikely event that one of ‘`.c`’ files is a directory. The ‘`find ...`’ command line above is more similar to the command:

   ```
   grep -rH --include='*.c' 'hello' /home/gigi
   ```

3. What if a pattern has a leading ‘`-`’?

   ```
   grep -e '--cut here--' *
   ```

   searches for all lines matching ‘`--cut here--`’. Without ‘`-e`’, `grep` would attempt to parse ‘`--cut here--`’ as a list of options.

4. Suppose I want to search for a whole word, not a part of a word?

   ```
   grep -w 'hello' *
   ```

   searches only for instances of '`hello`' that are entire words; it does not match '`Othello`'. For more control, use '`\<`' and '`\>`' to match the start and end of words. For example:

   ```
   grep 'hello\>' *
   ```

   searches only for words ending in '`hello`', so it matches the word '`Othello`'.

5. How do I output context around the matching lines?

   ```
   grep -C 2 'hello' *
   ```

   prints two lines of context around each matching line.

6. How do I force `grep` to print the name of the file?

   Append '`/dev/null`':

   ```
   grep 'eli' /etc/passwd /dev/null
   ```

   gets you:

   ```
   /etc/passwd:eli:x:2098:1000:Eli Smith:/home/eli:/bin/bash
   ```

   Alternatively, use '`-H`', which is a GNU extension:

   ```
   grep -H 'eli' /etc/passwd
   ```

7. Why do people use strange regular expressions on `ps` output?

   ```
   ps -ef | grep '[c]ron'
   ```

   If the pattern had been written without the square brackets, it would have matched not only the `ps` output line for `cron`, but also the `ps` output line for `grep`. Note that on some platforms, `ps` limits the output to the width of the screen; `grep` does not have any limit on the length of a line except the available memory.

8. Why does `grep` report "Binary file matches"?

   If `grep` listed all matching "lines" from a binary file, it would probably generate output that is not useful, and it might even muck up your display. So GNU `grep` suppresses output from files that appear to be binary files. To force GNU `grep` to output lines even from files that appear to be binary, use the '`-a`' or '`--binary-files=text`' option. To eliminate the "Binary file matches" messages, use the '`-I`' or '`--binary-files=without-match`' option.

9. Why doesn't '`grep -lv`' print non-matching file names?

   '`grep -lv`' lists the names of all files containing one or more lines that do not match. To list the names of all files that contain no matching lines, use the '`-L`' or '`--files-without-match`' option.

10. I can do OR with '`|`', but what about AND?

    ```
    grep 'paul' /etc/motd | grep 'franc,ois'
    ```

    finds all lines that contain both '`paul`' and '`franc,ois`'.

11. How can I search in both standard input and in files?

    Use the special file name '`-`':

    ```
    cat /etc/passwd | grep 'alain' - /etc/motd
    ```

12. How to express palindromes in a regular expression?

    It can be done by using back-references; for example, a palindrome of 4 characters can be written with a BRE:

```
grep -w -e '\(.\)\(.\).\2\1' file
```

It matches the word "radar" or "civic".

Guglielmo Bondioni proposed a single RE that finds all palindromes up to 19 characters long using 9 subexpressions and 9 back-references:

```
grep -E -e '^(.?)(.?)(.?)(.?)(.?)(.?)(.?)(.?)(.?).?\9\8\7\6\5\4\3\2\1$' file
```

Note this is done by using GNU ERE extensions; it might not be portable to other implementations of `grep`.

13. Why is this back-reference failing?

    ```
    echo 'ba' | grep -E '(a)\1|b\1'
    ```

    This gives no output, because the first alternate '`(a)\1`' does not match, as there is no '`aa`' in the input, so the '`\1`' in the second alternate has nothing to refer back to, meaning it will never match anything. (The second alternate in this example can only match if the first alternate has matched – making the second one superfluous.)

14. What do `grep`, `fgrep`, and `egrep` stand for?

    The name `grep` comes from the way line editing was done on Unix. For example, `ed` uses the following syntax to print a list of matching lines on the screen:

    ```
    global/regular expression/print
    g/re/p
    ```

    `fgrep` stands for Fixed `grep`; `egrep` stands for Extended `grep`.

7 Reporting bugs

Email bug reports to bug-grep@gnu.org, a mailing list whose web page is http://lists.gnu.org/mailman/listinfo/bug-grep. The Savannah bug tracker for `grep` is located at http://savannah.gnu.org/bugs/?group=grep.

7.1 Known Bugs

Large repetition counts in the '`{n,m}`' construct may cause `grep` to use lots of memory. In addition, certain other obscure regular expressions require exponential time and space, and may cause `grep` to run out of memory.

Back-references are very slow, and may require exponential time.

8 Copying

GNU grep is licensed under the GNU GPL, which makes it *free software*.

Please note that "free" in "free software" refers to liberty, not price. As some GNU project advocates like to point out, think of "free speech" rather than "free beer". The exact and legally binding distribution terms are spelled out below; in short, you have the right (freedom) to run and change grep and distribute it to other people, and even—if you want—charge money for doing either. The important restriction is that you have to grant your recipients the same rights and impose the same restrictions.

This method of licensing software is also known as *open source* because, among other things, it makes sure that all recipients will receive the source code along with the program, and be able to improve it. The GNU project prefers the term "free software" for reasons outlined at http://www.gnu.org/philosophy/free-software-for-freedom.html.

The exact license terms are defined by this paragraph and the GNU General Public License it refers to:

> GNU grep is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
>
> GNU grep is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> A copy of the GNU General Public License is included as part of this manual; if you did not receive it, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

In addition to this, this manual is free in the same sense:

> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being "GNU General Public License" and "GNU Free Documentation License", with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

The full texts of the GNU General Public License and of the GNU Free Documentation License are available below.

8.1 GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

1. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

2. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

3. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
   c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

   Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

   In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

4. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
   a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
   c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

5. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
6. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
7. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
8. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

9. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
10. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

    Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
11. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

12. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
13. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR

LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.
Copyright (C) 19yy  name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands ‘`show w`’ and ‘`show c`’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘`show w`’ and ‘`show c`’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

8.2 GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the

Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.1
or any later version published by the Free Software Foundation;
with the Invariant Sections being list their titles, with the
Front-Cover Texts being list, and with the Back-Cover Texts being list.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being *list*"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Concept Index

This is a general index of all issues discussed in this manual, with the exception of all specific `grep` command-line options, environment variables, color capabilities, and regular expression constructs, which are covered in their own index.

A

after context 5
alphabetic characters 12
alphanumeric characters 12
anchoring 13
asterisk 11

B

back-reference 13
backslash 13
basic regular expressions 14
before context 5
binary files 5
binary files, MS-DOS/MS-Windows 7
blank characters 12
braces, first argument omitted 11
braces, one argument 11
braces, second argument omitted 11
braces, two arguments 11
bracket expression 11
Bugs, known 16
Bugs, reporting 16
byte offset 4
byte offsets, on MS-DOS/MS-Windows 4

C

case insensitive search 2
changing name of standard input 4
character class 11
character classes 12
character type 9
classes of characters 12
context 5
context lines, after match 5
context lines, before match 5
control characters 12
Copying 16
counting lines 2

D

default options environment variable 7
device search 6
digit characters 12
directory search 6
dot 11

E

environment variables 7
exclude directories 6
exclude files 6

F

FAQ about `grep` usage 14
FDL, GNU Free Documentation License 24
files which don't match 2

G

GPL, GNU General Public License 17
graphic characters 12

H

hexadecimal digits 12
highlight markers 7
highlight, color, colour 2

I

include files 6
interval specifications 14
invert matching 2

L

language of messages 9
line buffering 6
line numbering 4
lower-case letters 12

M

match expression at most *m* times 11
match expression at most once 11
match expression from *n* to *m* times 11
match expression *n* or more times 11
match expression *n* times 11
match expression one or more times 11
match expression zero or more times 11
match the whole line 2
matching basic regular expressions 10
matching extended regular expressions 10
matching fixed strings 10
matching Perl regular expressions 10
matching whole words 2

max-count 3
memory mapped input 6
message language 9
MS-DOS/MS-Windows binary files 7
MS-DOS/MS-Windows byte offsets 4

N

names of matching files 3
national language support 9
NLS 9
no filename prefix 4
no warranty 21
numeric characters 12

O

only matching 3

P

palindromes 15
pattern from file 1
pattern list 1
period 11
plus sign 11
print non-matching lines 2
printable characters 12
punctuation characters 12

Q

question mark 11
quiet, silent 3

R

range expression 11
recursive search 6
regular expressions 11

S

searching directory trees 6
Searching for a pattern 1
space characters 12
subexpression 13
suppress binary data 5
suppress error messages 3

T

tab-aligned content lines 4
translation of message language 9

U

upper-case letters 12
Usage summary, printing 1
Usage, examples 14
Using `grep`, Q&A 14

V

Version, printing 1

W

whitespace characters 12
with filename prefix 4

X

xdigit class 12

Z

zero-terminated file names 5
zero-terminated lines 7

Index

This is an lexicographical list of all `grep` command-line options, environment variables, color capabilities, and regular expression constructs.

`*`

`*` 11

`+`

`+` 11

`-`

`--after-context` 5
`--basic-regexp` 10
`--before-context` 5
`--binary` 7
`--binary-files` 5
`--byte-offset` 4
`--color` 2
`--colour` 2
`--context` 5
`--count` 2
`--devices` 6
`--directories` 6
`--exclude` 6
`--exclude-dir` 6
`--exclude-from` 6
`--extended-regexp` 10
`--file` 1
`--files-with-matches` 3
`--files-without-match` 2
`--fixed-strings` 10
`--help` 1
`--ignore-case` 2
`--include` 6
`--initial-tab` 4
`--invert-match` 2
`--label` 4
`--line-buffered` 6
`--line-number` 4
`--line-regexp` 2
`--max-count` 3
`--mmap` 6
`--no-filename` 4
`--no-messages` 3
`--null` 5
`--null-data` 7
`--only-matching` 3
`--perl-regexp` 10
`--quiet` 3
`--recursive` 6
`--regexp=`*pattern* 1
`--silent` 3
`--text` 5
`--unix-byte-offsets` 4
`--version` 1
`--with-filename` 4
`--word-regexp` 2
`-a` 5
`-A` 5
`-b` 4
`-B` 5
`-c` 2
`-C` 5
`-d` 6
`-D` 6
`-e` 1
`-E` 10
`-f` 1
`-F` 10
`-G` 10
`-h` 4
`-H` 4
`-i` 2
`-l` 3
`-L` 2
`-m` 3
`-n` 4
`-`*num* 5
`-o` 3
`-P` 10
`-q` 3
`-r` 6
`-s` 3
`-T` 4
`-u` 4
`-U` 7
`-v` 2
`-V` 1
`-w` 2
`-x` 2
`-y` 2
`-z` 7
`-Z` 5

`.`

`.` 11

`?`

`?` 11

_

`_N_GNU_nonoption_argv_flags_` environment variable 10

{

`{,m}` 11
`{n,}` 11
`{n,m}` 11
`{n}` 11

A

`alnum` character class 12
`alpha` character class 12

B

`blank` character class 12
`bn` `GREP_COLORS` capability 8

C

`cntrl` character class 12
`cx` `GREP_COLORS` capability 8

D

`digit` character class 12

F

`fn` `GREP_COLORS` capability 8

G

`graph` character class 12
`GREP_COLOR` environment variable 7
`GREP_COLORS` environment variable 7
`GREP_OPTIONS` environment variable 7

L

`LANG` environment variable 9
`LC_ALL` environment variable 9
`LC_COLLATE` environment variable 9
`LC_CTYPE` environment variable 9
`LC_MESSAGES` environment variable 9
`ln` `GREP_COLORS` capability 8
`lower` character class 12

M

`mc` `GREP_COLORS` capability 8
`ms` `GREP_COLORS` capability 8
`mt` `GREP_COLORS` capability 8

N

`ne` `GREP_COLORS` capability 9

P

`POSIXLY_CORRECT` environment variable 9
`print` character class 12
`punct` character class 12

R

`rv` `GREP_COLORS` capability 8

S

`sl` `GREP_COLORS` capability 8
`space` character class 12

U

`upper` character class 12

X

`xdigit` character class 12

Table of Contents

1 Introduction 1

2 Invoking `grep` 1

- 2.1 Command-line Options 1
  - 2.1.1 Generic Program Information 1
  - 2.1.2 Matching Control 1
  - 2.1.3 General Output Control 2
  - 2.1.4 Output Line Prefix Control 4
  - 2.1.5 Context Line Control 5
  - 2.1.6 File and Directory Selection 5
  - 2.1.7 Other Options 6
- 2.2 Environment Variables 7

3 Exit Status 10

4 `grep` programs 10

5 Regular Expressions 11

- 5.1 Fundamental Structure 11
- 5.2 Character Classes and Bracket Expressions 11
- 5.3 The Backslash Character and Special Expressions 13
- 5.4 Anchoring 13
- 5.5 Back-references and Subexpressions 13
- 5.6 Basic vs Extended Regular Expressions 14

6 Usage 14

7 Reporting bugs 16

- 7.1 Known Bugs 16

8 Copying 16

- 8.1 GNU General Public License 17
  - Preamble 17
  - TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION 18
  - How to Apply These Terms to Your New Programs 23
- 8.2 GNU Free Documentation License 24
  - ADDENDUM: How to use this License for your documents 29

Concept Index 30

Index 32