**Filtering text (tr, uniq, tac, nl, wc, cut, sort, sed)**

All of the following commands read from **stdin** and write to **stdout**. (Use cat to start a *piped* solution.)

**tr OPTIONS SET1 SET2** - Translate, squeeze, and/or delete *characters*

OPTIONS

|  |  |
| --- | --- |
|  | Translation occurs if -d is not given and both SET1 and SET2 appear.  SET2 is extended to the length of SET1 by repeating its last character as necessary. Excess characters of SET2 are ignored. |
| -t | first truncate SET1 to length of SET2 |
| -c | complement SET1, e.g., if SET1 is [:digit:], SET1 is all characters except digits 0-9 |
| -d | delete characters in SET1, do not translate |
| -s | replace any character that is in SET1 and sequentially repeated with a single occurrence of that character. uses SET1 if not translating nor deleting; else uses SET2 and occurs after translation or deletion. |

SETS – (enclose in quotes, either ' or ", if you use character classes)

|  |  |  |
| --- | --- | --- |
| \\ - backslash  \a - alert  \b - backspace  \f - form feed  \r - return  \t - horizontal tab  \v - vertical tab  \E - escape | [: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 |

Examples:

|  |  |
| --- | --- |
| Create a list of all the words, one per line | tr -cs A-Za-z '\n' tr -cs "[:alpha:]" "\n" |
| Translate braces into parentheses | tr '{}' '()' |
| Remove all non-printable characters from the file | tr -cd "[:print:]" |
| Delete all NULL characters from a file | tr -d '\0' |
| Replace every sequence of one or more new lines with a single new line | tr -s '\n' |
| Replace every nonprinting character, other than valid control characters, with a ? | tr -c '[:print:][:cntrl:]' '?' |
| Replace successive white space characters (space & tab) with a single blank | tr -s '[:blank:]' ' ' |

**uniq [OPTIONS]** - filter out duplicate *lines*, count duplicate *lines*, or find unique *lines*.

OPTIONS

|  |  |
| --- | --- |
|  | Matching lines are merged to the first occurrence |
| -c | Prefix an output line with a number representing how many times the line occurred. |
| -d | Only print duplicated lines. |
| -f N | Avoid comparing the first N fields of a line before determining uniqueness. A field is a group of characters, delimited by whitespace. |
| -i | Ignore case. |
| -s N | Do not compare the first N characters of each line. Like -f, but skips N characters rather than fields. |
| -u | Only prints unique lines. |

**tac** – cat backwards! – reverse the lines of stdin and outputs them to stdout.

**nl [OPTIONS]** – number the *lines*. (There are many formatting options.)

**wc [OPTIONS]** - count the number of lines, words, and characters in a stream

OPTIONS

|  |  |  |
| --- | --- | --- |
| -c | Print byte count | Default is: wc –l –w -c |
| -m | Print the character count | Order of output: |
| -l | Print the line count | lines, words, characters, bytes, maximum line length |
| -L | Print the length of the longest line |  |
| -w | Print the word count |  |

**cut [OPTIONS]** - cut out sections of each line. (Assumes data is in tabular format.)

OPTIONS

|  |  |  |  |
| --- | --- | --- | --- |
| -b N | Select specific bytes from a line. | How to specify the bytes, characters or fields: | |
| -c N | Select specific characters from a line. | N | The Nth one, counted from 1 |
| -f N | Select specific fields. | N- | The Nth one to the end of the line |
| -d DELIM | Field delimiter character(s) (default is TAB) | N-M | From the Nth to the Mth one |
| --complement | Complement the selected set. | -M | From the first to the Mth one. |
|  |  | N,M,P | The Nth, Mth and Pth one. |

**paste file1 file2** - sends the two files to stdout side by side (opposite of cut)

**sort [OPTIONS]** - Sort the *lines* of a text stream

OPTIONS

|  |  |
| --- | --- |
| -b | ignore leading blanks |
| -f | fold lower case to upper case characters |
| -g | compare according to general numerical value |
| -M | sort on months; compare (unknown) < 'JAN' < ... < 'DEC' |
| -h | compare human readable numbers (e.g., 2K, 1G) |
| -n | compare according to string numerical value |
| -R | random shuffle, but group identical keys. |
| -r | reverse |
| -t SEP | field separator; use SEP instead of non-blank to blank transition |
| -k N | N is F[.C][,F[.C]] 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. |

**sed [OPTIONS] [SCRIPT]** - "stream editor" that allows you to filter and transform text.

Most common usage is to search and replace text. For example:

sed 's/test/example/g' substitute every occurrence of 'test' with 'example