```
  CHAR1-CHAR2     all characters from CHAR1 to CHAR2 in ascending order
  [CHAR*]         in SET2, copies of CHAR until length of SET1
  [CHAR*REPEAT]   REPEAT copies of CHAR, REPEAT octal if starting with 0
  [:alnum:]       all letters and digits
  [:alpha:]       all letters
  [:blank:]       all horizontal whitespace
  [:cntrl:]       all control characters
  [:digit:]       all digits
  [:graph:]       all printable characters, not including space
  [:lower:]       all lower case letters
  [:print:]       all printable characters, including space
  [:punct:]       all punctuation characters
  [:space:]       all horizontal or vertical whitespace
  [:upper:]       all upper case letters
  [:xdigit:]      all hexadecimal digits
  [=CHAR=]        all characters which are equivalent to CHAR

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

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/tr>
or available locally via: info '(coreutils) tr invocation'
bandit11@bandit:~$ tr 'a-zA-Z' 'n-mN-M' data.txt
tr: extra operand ‘data.txt’
Try 'tr --help' for more information.
bandit11@bandit:~$ cat data.txt | tr 'a-zA-Z' 'n-mN-M'
tr: range-endpoints of 'n-m' are in reverse collating sequence order
bandit11@bandit:~$ cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'
The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv
bandit11@bandit:~$
```