-

NAME
          tr − transliterate

SYNOPSIS
          tr [−**cds**] [string1 [string2]]

DESCRIPTION
          Tr copies the standard input to the standard output with substitution or deletion of selected char-
          acters. Input characters found in *string1* are mapped into the corresponding characters of
          string2. If *string2* is short, it is padded with corresponding characters from *string1*. Any combi-
          nation of the options −**cds** may be used. −**c** complements the set of characters in *string1* with re-
          spect to the universe of characters whose ascii codes are 001 through 377 octal. −**d** deletes all
          input characters in *string1*. −**s** squeezes all strings of repeated output characters that are in
          string2 to single characters.

          The following abbreviation conventions may be used to introduce ranges of characters or re-
          peated characters into the strings:

          [*a*−*b*] stands for the string of characters whose ascii codes run from character *a* to character *b*.

          [*a**n], where *n* is an integer or empty, stands for *n*-fold repetition of character *a*. *n* is taken to be
          octal or decimal according as its first digit is or is not zero. A zero or missing *n* is taken to be
          huge; this facility is useful for padding *string2*.

          The escape character '\' may be used as in *sh* to remove special meaning from any character in a
          string. In addition, '\' followed by 1, 2 or 3 octal digits stands for the character whose ascii code
          is given by those digits.

          The following example creates a list of all the words in 'file1' one per line in 'file2', where a
          word is taken to be a maximal string of alphabetics. The strings are quoted to protect the special
          characters from interpretation by the Shell; 012 is the ascii code for newline.

                              tr −cs "[A−Z][a−z]" "[\012*]" <file1 >file2

SEE ALSO
          sh (I), ed (I), ascii (VII)

BUGS
          Won't handle ascii NUL in *string1* or *string2;* always deletes NUL from input.