

**NAME**

total - sum up columns

**SYNOPSIS**

**total** [ **-m** ] [ **-sE** | **-p** | **-u** | **-l** ] [ **-i{fld}** ] **[N]** [ **-o{fld}** ] [ **-in M** ] [ **-on M** ] [ **-tC** ] [ **-N** [ **-r** ] ] [ file .. ]

**DESCRIPTION**

*Total* sums up columns of real numbers from one or more files and prints out the result on its standard output.

By default, *total* computes the straight sum of each input column, but multiplication can be specified instead with the *-p* option. Likewise, the *-u* option means find the upper limit (maximum), and *-l* means find the lower limit (minimum).

Sums of powers can be computed by giving an exponent with the *-s* option. (Note that there is no space between the *-s* and the exponent.) This exponent can be any real number, positive or negative. The absolute value of the input is always taken before the power is computed in order to avoid complex results. Thus, *-sl* will produce a sum of absolute values. The default power (zero) is interpreted as a straight sum without taking absolute values.

The *-m* option can be used to compute the mean rather than the total. For sums, the arithmetic mean is computed. If a power is also specified using the *-s* option, the inverse power will be applied to the averaged result. For products, the geometric mean is computed. (A logarithmic sum of absolute values is used to avoid overflow, and zero values are silently ignored.)

If the input data is binary, the *-id* or *-if* option may be given for 64-bit double or 32-bit float values, respectively. Either option may be followed immediately by an optional count, which defaults to 1, indicating the number of double or float binary values to read per record on the input file. (There can be no space between the option and this count.) Similarly, the *-od* and *-of* options specify binary double or float output, respectively. These options do not need a count, as this will be determined by the number of input channels.

A count can be given as the number of lines to read before computing a result. Normally, *total* reads each file to its end before producing its result, but this behavior may be overridden by inserting blank lines in the input. For each blank input line, *total* produces a result as if the end-of-file had been reached. If two blank lines immediately follow each other, *total* closes the file and proceeds to the next one (after reporting the result). The *-N* option (where N is a decimal integer) tells *total* to produce a result and reset the calculation after every N input lines. In addition, the *-r* option can be specified to override reinitialization and thus give a running total every N lines (or every blank line). This option also turns off the usual output flushing at each total. If the end of file is reached, the current total is printed and the calculation is reset before the next file (with or without the *-r* option).

The *-in* option if present, will limit the number of input records read (per input file). The *-on* option may be used to limit the total number of output records produced.

The *-tC* option can be used to specify the input and output tab character. The default tab character is TAB.

If no files are given, the standard input is read.

**EXAMPLES**

To compute the RMS value of colon-separated columns in a file:

```
total -t: -m -s2 input
```

To produce a running product of values from a file:

```
total -p -l -r input
```

**BUGS**

If the input files have varying numbers of columns, mean values will certainly be off. *Total* will ignore missing column entries if the tab separator is a non-white character, but cannot tell where a missing column should have been if the tab character is white.

TOTAL(1)

TOTAL(1)

**AUTHOR**

Greg Ward

**SEE ALSO**

cnt(1), neaten(1), rcalc(1), rcollate(1), rlam(1), rsplit(1), tabfunc(1)