Processing is similar to stream processing of a continuous series of ASCII characters, delimited by a newline character.

In a more realistic scenario, any one (or more) of the network-centric control characters may be substituted for the carriage-control (carriage-control/linefeed pair) along with big-endian/little-endian concerns (byte-address ordering).

Table

Description automatically generated

****

In the application, intermediate computations for the diff value are shown to illustrate the progressive nature of processing each "tranch" of numbers.

The diff is accumulated following the concept of a "control break" situation – specifically, encountering a newline character within the "str-weam" (stream).

Sadly, I did a fair number of COBOL reports leveraging "control break" logic for Milwaukee Electric Tool – like: part count by plant, within region, within division, etc).

Control break logic is not x-actly "easy", yeezy.

But it allows single-pass logic (and saves much wall-time, if not actual CPU time).

The logic was tricky – the COBOL did not enhance the coding joy (which lasted until 1995-6).

I did some other 4-GL language leveraging COBOL shit, then the advent of Java!

****

Text

Description automatically generated

Text

Description automatically generated

Line 37 is the conditional branch decision point.

Text

Description automatically generated

Line 51 does most of the bookkeeping; line 66 does the final trick, when neither conditional (situation) holds true.

****

THIS TOOK 2 HOURS TO SLOG THROUGH