read (only because of the subject matter), it is a masterpiece.
Of course, the subsequent program development is a mixture
of my own experience and ideas, influenced by my interpreta-
tion of Eijikstra's ideas. I apologize for any misunderstand-
ings that might arise because of this interpretation, and urge you to
go back to the source—his new book.

II. DEVELOPMENT OF AN ALGORITHM

The problem I have chosen to discuss is a realistic one that
occurs in any "text editor" or "format" program. At least
initially, it does not seem very "mathematical," because it has
to do with processing strings or words rather than integers. I
do not remember solving it before, although I did give it as an
"extra credit" part of a programming project many years ago,
which few students actually completed. The development de-
scribed below is an idealized version of the actual development
that occurred when I tackled the problem in the spring of
1976. I did try to discipline my development, and indeed, the
real development followed this idealized version quite closely,
although it was naturally a bit rougher.

A line justifier inserts extra blanks between the words on a
line so that the last character of the last word appears in the
last column of the line. For example, the three lines

\[
\text{justifying lines by inserting extra blanks is the task of a line editor.} \tag{1}
\]

would appear right-justified on a 26-character line as

\[
\text{justifying lines by inserting blanks is the task of a line editor.} \tag{2}
\]

The number of blanks between different pairs of words on a
line may differ by no more than 1. Secondly, in order to
lessen the impact of extra blanks on the reader, the justifier
uses an alternating technique for insertion: when necessary,
for odd (even) lines more blanks are inserted toward the right
(left) of the line. For example, in line 1 of (2) above, the
number of blanks between successive pairs of words is 4 and 5;
on line 2 it is 2, 1, and 1.

We want an algorithm for performing the more difficult part
of justification. Given the column numbers where the words
begin on the unjustified line, we want to determine the column
numbers where they begin on the justified line. For line
1 of (1) above, the words begin in columns (1, 12, 18); judging
by line 1 of (2), this should be changed to (1, 15, 25). For
line 2, (1, 11, 17, 24) should be changed to (1, 12, 18, 25).
For line 3, (1, 15, 10, 13, 15, 20) remains unchanged.

The following would probably be typical of the kind of
documentation produced by today's programmer for this
problem (if any is written at all):

\begin{verbatim}
proc justify (n, x, z: integer;
  var b: array[*] of integer);
  {Line z has n words on it. They begin in columns b[1],
    \ldots, b[n]. Exactly one blank separates each pair. Parameter
    n is the number of spaces which must be inserted be-
    tween words in order to justify the line.}
\end{verbatim}