MTR-2119

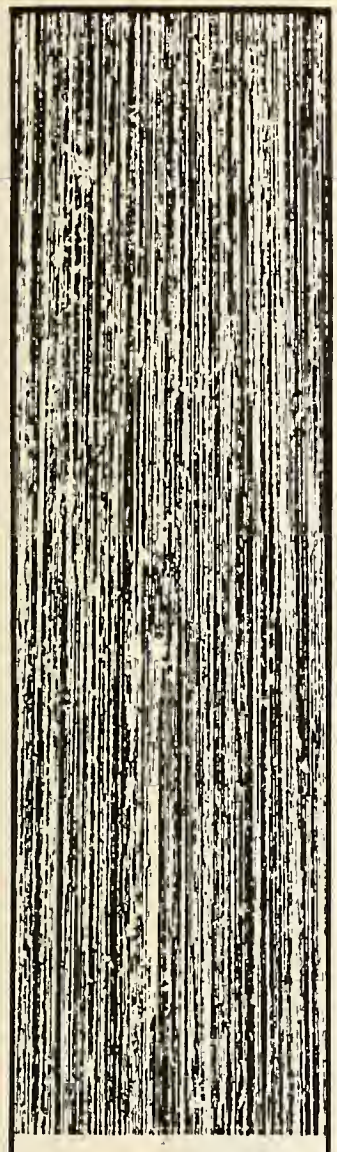
DOTSYS III

a portable program for grade II braille translation

by w.r.gerhart j.k.millen j.e. sullivan

the MITRE corporation

PRINT
655.38
M
c. 1
Mitre Corporation
DOTSYS III: A PORTABLE
PROGRAM FOR GRADE 2
BRAILLE TRANSLATION



THE MITRE CORPORATION
BEDFORD, MASSACHUSETTS

MITRE Technical Report

MTR- 2119
No. Vol. Series Rev. Supp. Corr.

Subject: DOTSYS III: A Portable Program for Grade 2 Braille Translation

Author: W. Reid Gerhart, Dr. Jonathan K. Millen, Joseph E. Sullivan
Dept.: D-73

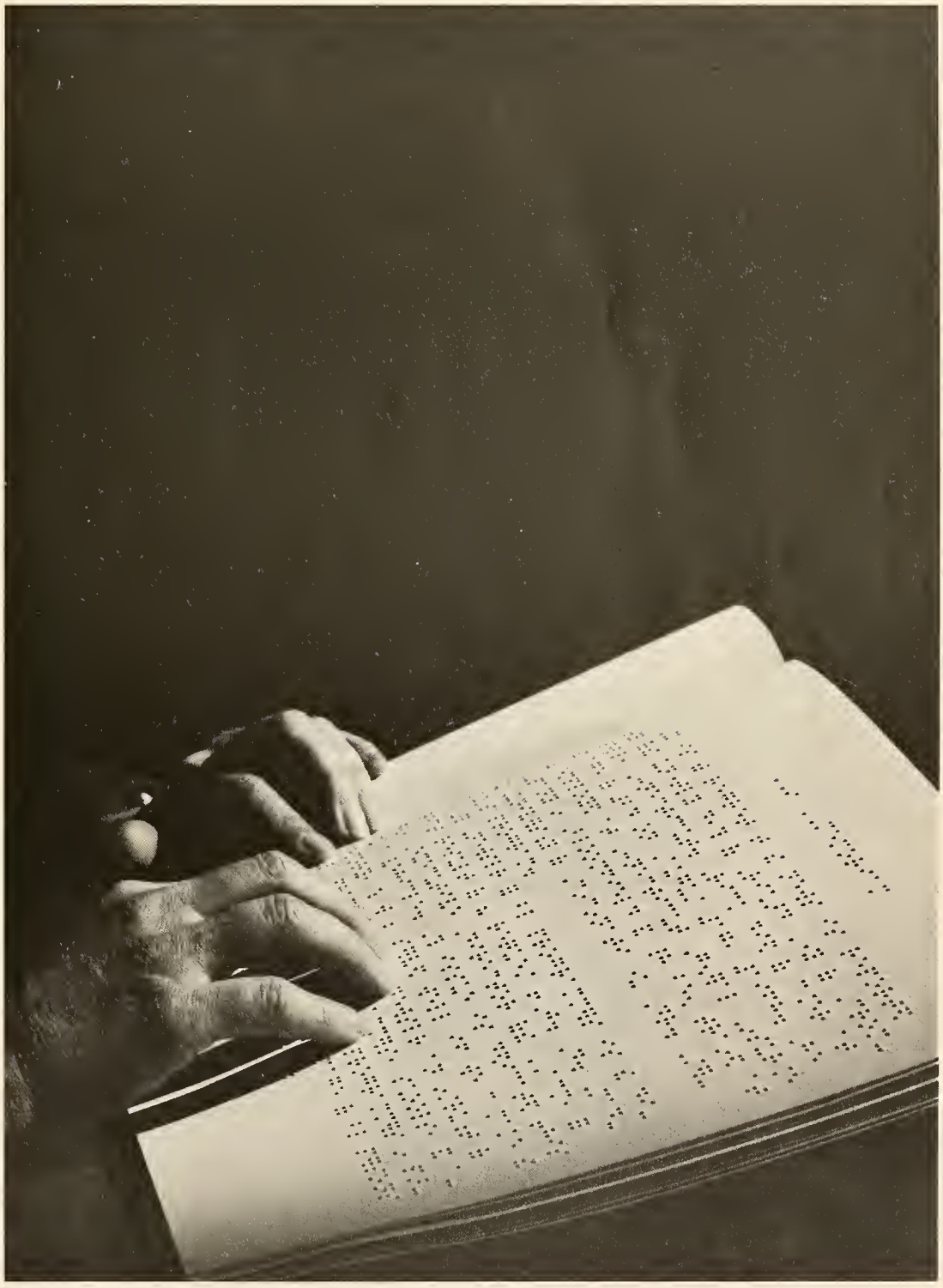
Date: 14 May 1971
Contract No.: SR 21761
Contract Sponsor: Sensory Aids Evaluation and Development Center, Massachusetts Institute of Technology
Project: 1420
Issued at: Bedford, Massachusetts

Department Approval: E. L. Lafferty

MITRE Project Approval: R. A. J. Gildea

Page 1 of 150 Pages

This document has been approved for public release.



ABSTRACT

This document describes DOTSYS III, a table-driven COBOL program for the translation of English text into grade 2 (or grade 1) braille. While general acquaintence with computer programming and the subject of braille translation would be helpful in reading the document, no special knowledge in these areas is presupposed. The program's method of operation, together with detailed instructions on using the program, on modifying or extending the translation heuristics, as defined in the tables, and on transferring the program to a new computer environment are all presented.

W. R. Gerhart
Intelligence and
Information Systems

J. K. Millen
Intelligence and
Information Systems

J. E. Sullivan
Intelligence and
Information Systems

WRG:JKM:JES/ces

PREFACE

This document is intended to serve the needs of many different levels of interest. Those interested only in what the program will do, for example, need read only the introduction. Persons with a general interest in the subject of braille translation should add the section on method. Those who actually wish to use the program -- transcribers, editors, and keypunchers -- will, of course, want to read the appropriate parts of the Usage Section and may or may not be interested in method. A systems programmer or other person whose main interest is to get the program running in a new environment ought to read the "Transfer" Section. Finally, a programmer who may wish to modify the program should read the whole document, including the final section on maintenance.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

Braille, as it is used today in almost all contexts other than primer textbooks, is almost a system of shorthand and not simply a scheme for representing individual inkprint symbols in a tactile code. This system, called grade 2 braille, is defined in Reference 1. (A more direct transliteration is called grade 1 braille.) The chief device used to reduce the number of braille signs in grade 2 is the contraction. A contraction is the representation of an inkprint letter-group or whole word by a relatively short sequence of braille signs; for example, the word "receiving" is represented by the four braille signs for r, c, v, and g in succession. There are 189 letter-groups that may be contracted.

Unfortunately (at least from the standpoint of automating the translation process), a given contractable letter-group is not necessarily contracted wherever it appears. Moreover, the rules governing the use of contractions frequently involve such matters as pronunciation and meaning. For example, in the word "disease," the "dis" and the "ea" are normally contracted. However, the (now obsolete) meaning "lack of ease" is also defined for this word, and when "disease" is used with that meaning, the "ea" should not be contracted. This example, although admittedly farfetched, illustrates that in some circumstances even a human transcriber might have difficulty applying the rules. Cases routinely arise that are easy for a human transcriber, but still difficult for an essentially mechanical process -- for example, distinguishing the musical note "do" from the verb "do." For these reasons, automatic natural-language to braille translation algorithms tend to be heuristic, which is to say fallible.

DOTSYS III is a computer program embodying such a natural-language-to-braille translation algorithm. As its name implies, it is an outgrowth of an earlier program, DOTSYS II, described in References 2 and 3.* The basic translation algorithm remains

---

* In order to make the present document as self-contained as possible, portions of References 2 and 3 have been incorporated with little or no change.

essentially the same, but a number of new features have been added, the translation quality has been improved, and better internal processing methods have been introduced. These improvements in capability have been offset by the improved methods, so that the overall speed of DOTSYS III remains about the same as DOTSYS II (1300 wpm on an IBM 360/50, 3330 wpm on a 360/65). DOTSYS II, in turn, owes the fundamentals of its translation algorithm both to previous work in the field of braille translation by computer and to elementary concepts in the theory of automata, logic design, and formal languages. The background references and relationships discussed in Reference 2 are relevant to DOTSYS III also.

The source language used in coding DOTSYS III is ASA Standard COBOL level 2 (Reference 4), with the COMPUTE verb the only feature used that is not level 1. (COMPUTE statements may easily be recast as level 1 statements if required.) This should enable DOTSYS III to be run on any computer having a COBOL compiler and a modest amount of core storage (approximately equivalent to 64,000 bytes of IBM 360 storage).

The input text is presented to DOTSYS III in the form of 80-character (punched card image) records. These can be manually keypunched directly from inkprint or produced by another program according to a fairly straightforward set of conventions. The output is the sequence of braille signs equivalent to the input text. Output can be produced in any of several forms, including "proof" output for a sighted editor and tactile (embossed) braille.

DOTSYS III is almost completely table-driven; i.e., details of the translation algorithm are determined by tables read in at execution time rather than by the program itself. DOTSYS III, as described in this document, comprises both the program and a standard set of tables. In principle, with modified tables, the DOTSYS III program would be capable of processing different kinds of text, such as text containing mathematical or technical notation, languages other than English, or text containing nonstandard symbols for format control.

SECTION II

METHOD

GENERAL

DOTSYS III comprises five cooperating processors: the primary input section, the translator, the stacker, the braille line composer, and the final output writer. It is the translator, as its name implies, that does most of the work of braille translation, and in fact, this section forms a sort of main loop or program, the others being in the form of subprograms called by the translator to do their work at the appropriate time. However, in order to follow the processing of a given piece of text from inkprint form to braille form, it is easiest to imagine the processors running sequentially, each one in turn operating on the output, or a collection of outputs, from the previous processor.

The following descriptions of these processors are idealized to some extent, so that the discussion does not become hopelessly mired in detail.

THE PRIMARY INPUT PROCESSOR

The primary input section reads the 80-characters (card-image) records. Columns 73-80 are discarded, so that the first character of a record logically follows character 72 from the previous record. This stream of characters is further collapsed by deleting all instances of the vertical bar character (|), and by deleting all consecutive blanks after the second in a series following a period (.) and all after the first in any other context. This collapsed stream of characters, one at a time, is the output of this section.

In the "self-checking" mode (described in detail in a later section), the primary input processor prepares the correct translation words for later comparison against the translator output.

THE TRANSLATOR*

The Buffer

The translation section operates on the stream of characters issued by primary input. As a first step, these are collected into a ten-character sliding window, called the "buffer," so that a group of characters can be examined.

The Alphabet Table Search

The leftmost character in the buffer is looked up in the alphabet table. In this table, there is exactly one entry for each symbol that may appear in the text, containing, among other things: (a) a code denoting the braille sign for that symbol, (b) the symbol's "input class," and (c) an index to that portion of the contraction table which pertains to this initial letter. An input class is an arbitrary numerical code with three distinct purposes, as will be seen.

The Contraction Table Search

The next step is to search that section of the contraction table indicated for this initial letter. In principle, this search may be visualized as a simple top-to-bottom entry-by-entry sequential search, although in fact a much faster tree-search algorithm (described in detail under "Maintenance") is used. An entry in the contraction table consists of:(a) a string of up to nine characters (ten, counting the implied initial letter); (b) a "right-context class" designator; (c) an input class code; (d) a shift count; and (e) a set of up to four braille sign codes.

For a match to occur on a particular entry, three conditions must be satisfied. First, the entire string for that entry must correspond to the buffer, or left substring thereof. Secondly, the input class for the entry determines a set of "state variable" conditions that must be satisfied. A state variable is a logical switch, having a

---

*The translator operates essentially as described in Section III of Reference 2, except that (1) the contraction table search has been speeded up considerably by using a modified binary search, which affects the table ordering rules slightly, (2) the STRING field delimiter in the contraction table has been changed from "$" to "|" (vertical bar), and (3) the decision table permits a new decision symbol "F" meaning "unconditional no."

value "yes" or "no" according to its meaning and the text already translated. For example, a state variable whose meaning is "after a digit" would have the value "yes" if the character immediately preceding the one leftmost in the buffer had been one of the digits 0-9 and would have the value "no" in all other circumstances, including initially. "Meaning" is, strictly speaking, defined completely by entries in another table which governs the setting of these switches, called the transition table. This table will be discussed presently. The mechanism by which the input class selects a set of state variable conditions to be tested is called the decision table; this table is also discussed in more detail in a separate section.

The third condition for a contraction table match to occur is that the right-context class be correct. If the right-context designator for the entry is blank, no right-context condition is imposed. Otherwise, the character immediately to the right of the matched string in the buffer is looked up in the alphabet table to determine its input class. Another table, called the right-context table is consulted to determine whether that input class is one of those listed as acceptable for this designator--e.g., the designator "P" (for "punctuation") may apply to input classes 2, 3, 13, and 14.

The contraction table search stops when a match occurs or the end of the table section for that particular initial letter is reached. In the latter event the shift count is taken to be 1, the input class and braille sign code revert to those found for the initial letter, and processing proceeds.

The Buffer Shift

The buffer is then "shifted" left by the amount of the shift count, with new characters from the input stream entering on the right.

Braille Sign Output

The braille sign code(s) are sent to the stacker section, constituting the output of the translator. These may directly represent braille signs, or they may be special control codes as is discussed in the section describing the stacker.

The State Transition

Finally, the input class is used to determine a set of transitions to be applied to the state variables. For each variable, the transition may be specified as "no change", "toggle" (change "yes" to "no" and "no" to "yes"), "set to yes" or "set to no."

After the state variable transition, the process is repeated, beginning with the alphabet table search.

The Decision Table

The decision table determines whether the current settings of the state variables permit the use of a given contraction table entry. It is best represented by a tableau in two parts, as depicted in Figure 1. The upper, or decision portion, has a row for each input class; the lower, or condition portion has one row per state variable. The number of columns is the same for both sections, and this number will depend upon the number of independent tests that may have to be made. The elements of the array are single characters, as given in Figure 1.

Processing of the table for a particular input class consists of a left-to-right scan of the associated row in the upper portion. If a "G" ("Go") symbol is encountered, processing stops with a "yes" decision. If an "F" ("Fail") is encountered, processing is likewise terminated, but with a "no" decision.

If one of the symbols "Y" or "N" is reached, then the corresponding column in the lower portion is compared against the current settings of the state variables. A "Y" in the i^{th} row implies that the i^{th} state variable must have the value "yes"; an "N" demands the value "no" and a dash is satisfied by either setting. If the specified conditions are satisfied for all the state variables, then processing is terminated with a "yes" decision if the upper portion had a "Y" symbol, and "no" if it was an "N". If one or more state variables does not have the value specified, then scanning of the decision section row is resumed.

When a dash is reached in the scan, the scan simply continues with the next column. If the scan encounters the end of the row without otherwise reaching a decision, then the decision becomes "no."

THE STACKER

The stacker operates upon the braille sign codes issued by the translator. In the simplest case, these codes are merely accumulated until the code for the blank braille sign is received--i.e., a braille word is finished. This word, constituting one entry in a stack, is normally then removed from the stack and issued as output to the braille line composer. In exceptional circumstances, two or more words may be held in a stack before release. This may be because the order

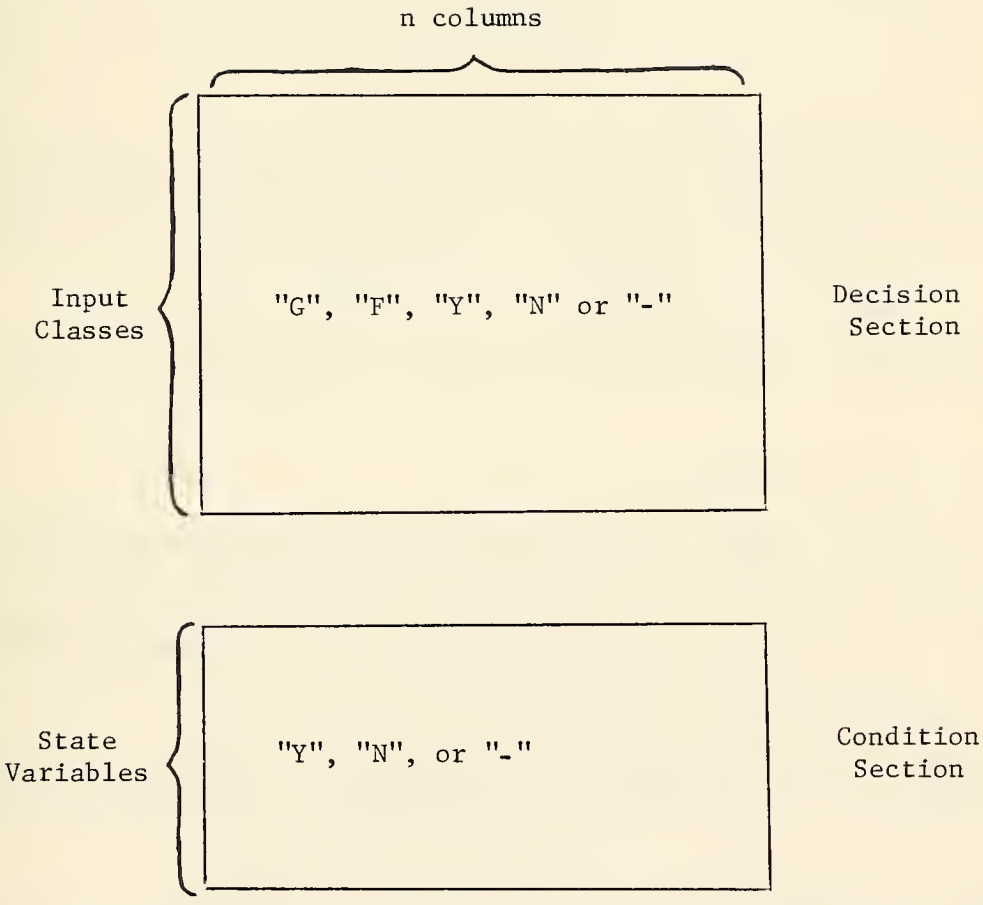


Figure 1. Decision Table Schematic

may have to be changed (in braille, units of measure are placed before the associated numeric quantity), or because the overall length of several words must be computed before issuing the first one--as when centering headings.

All special operations by the stacker, including delayed release and interchange of order, are directed by control codes issued by the translator. These are distinguishable from ordinary braille sign codes in that they have a value greater than 64. (See Appendix VII.)

The stacker maintains up to three distinct stacks, sharing a common area by borrowing entry fields from a linked free storage list. One stack is used for normal output; a second stack is used to collect the words in a heading (such as a chapter title); the third is used to hold the title (running head) if it is present.

In the "self-checking" mode (described in a later section), it is the stacker that compares each braille word against the correct translation prepared by primary input.

THE BRAILLE LINE COMPOSER

The line composer operates on braille words (stack entries) produced by the stacker. In each case, the length of the word will determine whether it can fit on the current braille line, an internal output buffer. If so, it is simply added thereto. Otherwise, the current line is sent to the final output writer, cleared, and started anew with the current braille word.

The line composer also concerns itself with counting lines to determine a new page condition, page numbering and titling, and similar matters.

THE FINAL OUTPUT WRITER

The output writer is actually a collection of processors, one for each distinct mode of output. For each mode selected, the associated processor produces actual output corresponding to the current braille line set up by the line composer.

SECTION III

USAGE

INPUT

Deck Setup

The complete input deck read by DOTSYS III consists of four main sections, in order:

(1) the echo option card;

(2) the tables;

(3) the run control cards; and

(4) the text.

Sections 1-3 must be sequence-numbered (in columns 73-80), in strictly increasing order.

The Echo Option Card

The echo option card determines whether a literal listing of the tables and run control cards will be printed on the SYSPRINT (normal printer) output device. The echo card itself is always so printed; printing of the text is controlled by one of the run control cards.

If column 1 of the echo card contains an "E," (for "echo") the listing is produced; if it contains an "N," (for "no echo") it is not. Columns 2-80 are ignored. For production use, the "N" option would be usual.

The Tables

Table Data in General

The tables are supplied with the DOTSYS III program and form an integral part of the process described by this document. However, circumstances may arise such that the user may wish to alter or augment the tables. One such circumstance might be a word found to be incorrectly translated by DOTSYS III, that occurs so often in a given text that it is impractical to force the correct translation by

special treatment (see the "$/", "/_" and "_/" control symbols under "Forcing and Preventing Contractions" in "Text Input", below). In such a case, an addition to the contraction table is called for.

The tables and associated dimensioning cards are to be arranged in the following order:

(1) the alphabetical contraction list;

(2) the alphabet table;

(3) the contraction table;

(4) the card specifying the number of right-context classes;

(5) the right-context table;

(6) the card specifying the number of state variables;

(7) the card specifying the number of input classes;

(8) the transition table;

(9) the card specifying the number of decision table columns;

(10) the decision table; and

(11) the sign table.

The formats of the tables and cards listed above follow. All two-column numerical fields must contain two digits; in particular, a number less than 10 must be given a leading zero when used in a two-column field. Numerical limitations stated in the form: "this number may not exceed ..." are appropriate for the table sizes used in the version of the DOTSYS III program listed in Appendix I. Refer to "TABLE SIZE BOUNDS" under "PROGRAM MAINTENANCE" (Section V) if changes in table size are contemplated.

The Alphabetical Contraction List

This table contains exactly 189 cards, one for each contraction defined in grade 2 English braille. The order of input must be alphabetical. The table is used in support of the self-checking feature (described under "Text Input" below). The format is as given in Table I.

Table I

Alphabetical Contraction List Card Format

Columns	Contents
1-10	The inkprint contractable letter sequence, left-justified.
24-31	Four 2-digit braille sign codes (Cf. Appendix VI) representing the equivalent braille sign(s). 99's are used for filler on the right.
32-33	Presently ignored, but reserved for a fifth sign code.

The Alphabet Table

The alphabet table identifies all legal text input characters. The order of input is arbitrary, but for the sake of efficiency should normally be by decreasing frequency of occurrence of the symbol. (Note also that the order is related to that of the contraction table, q.v.) A dummy entry, with 99 in columns 18-19, should follow the last actual alphabet table entry. Including this card, the total number may not exceed 64. The card format is given in Table II.

Table II

Alphabet Table Card Format

Columns	Contents
1	The symbol being defined.
18-19	The input class (must be in the range from 01 to the number of input classes). Note: A card with 99 in this field signals that the end of the alphabet table has been reached.
21-22	The sign code to use when the symbol occurs in a computer-braille string (see "Text Input," below and Appendix VI).

Columns	Contents
24-25	The sign code to use in normal context (Cf. Appendix VI).
30-31	01 if the symbol may be used as one of a pair of symbols bracketing a letter denoting itself; 00 otherwise.

The Contraction Table

The contraction table contains not only contractions but many other sequences related to the heuristics of the translation process, and the definition of special control symbols.

Entries in the contraction table beginning with the same symbol must be grouped together, and these groups must be arranged in the same order as the corresponding symbols in the alphabet table. Also, symbols in the alphabet table that have no corresponding section in the contraction table are grouped, in any order, at the end of the alphabet table.

Within a section of the contraction table determined by a common initial letter, all entries having a given second character must also be grouped together. The order of input of these subsections is completely arbitrary and usually will vary from section to section as a function of conditional frequency. This grouping scheme is continued right up to the 10th character. In general, if two entries agree through the nth character, then all entries between them must also agree with them through the nth character.

A dummy entry, with 99 in columns 18-19, should follow the last actual entry. The total number of cards, including the dummy, cannot exceed 1,000.

Table III gives details of the contraction table card layout.

Table III

Contraction Table Card Format

Columns	Contents
1-10	Character sequence, left-justified. If the string is shorter than 10 characters, then a vertical bar (|) should follow the last character. (Thus blanks are permitted in the string.)
16	Right-context class designation may be specified (non-blank) only if the string is shorter than 10 characters.
18-19	Input class. Note: A card with 99 in this field signifies that the end of the table has been reached.
21-22	Shift count.
24-31	Four two-digit sign codes (Cf. Appendix VI). 99's are used for filler on the right.

The Card Specifying the Number of Right-Context Classes

As its name implies, this card contains the number of right-context classes that are to be defined. This figure is punched in columns 18-19. It cannot exceed 02.

The Right-Context Table

This table contains one card per right-context class; i.e., there are as many cards as signified on "the Number of Right-Context Classes" card, just previously described. The layout is given in Table IV.

Table IV

Right-Context Class Card Format

Columns	Contents
16	A single-character designator for the class being defined--e.g., "P" for "punctuation."
24-31	Up to four input class codes. All symbols having one of the listed input classes are implied to have the right context class being defined. If there are fewer than four input class codes, the last one is simply repeated to make four.

The Card Specifying the Number of State Variables

The number of state variables is punched in columns 18-19. This number may not exceed 10.

The Card Specifying the Number of Input Classes

The number of input classes is punched in columns 18-19. This number may not exceed 25.

The Transition Table

This table contains one card per state variable. On each such card, the i^{th} column contains a character signifying how the state variable is to be set when input class i is processed. An "R" signifies "reset" (set to "no"), an "S" means "set" (to "yes"), a dash (-) means "leave," and "T" means "toggle" (change to opposite state).

The Card Specifying the Number of Decision Table Columns

The number of columns in the decision table is punched in columns 18-19. This number cannot exceed 15.

The Decision Table

This table contains one card per decision table column. The layout of each card is given in Table V.

Table V

Decision Table Card Format

Columns	Contents
1 - nic*	The i^{th} card column contains the upper (decision) section symbol for the input class i. It must be G, F, Y, N or -. (See "The Decision Table.")
nic - (nic+nsv)*	The (nic + j)th card column contains the symbol denoting the condition imposed on the j^{th} state variable. It must be Y, N or -.

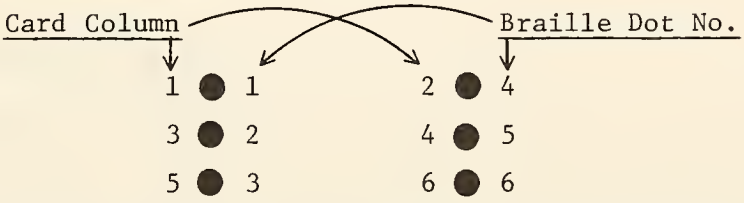
The Sign Table

The sign table contains exactly 64 cards, one for each "ordinary" braille sign code. (Codes 00 and 64 both refer to sign 64.) The i^{th} card defines code i. The layout is given in Table VI.

Table VI

Sign Table Card Format

Columns	Contents
1-6	Dots, keypunched as periods, corresponding to the braille dots embossed for this character. The correspondence is :



7-9	Three characters to be printed on proof output, denoting the most common meaning for this sign.

* nic is the number of input classes, nsv the number of state variables.

The Run Control Cards

The run control cards select options that remain in effect throughout the translation of the text--i.e., for the entire "run," or job step.*

There are 7 run control cards. The first four select the output modes (described under "output," below); any combination is permitted. On these, only column 1 is read; it should contain an "N" if the output mode is not wanted or a letter associated with the mode otherwise. Specifically, the options cards are, in order:

(1) Proof output (P or N in column 1).

(2) "Elastomer" braille (B or N in column 1).

(3) "RPQ" braille (R or N in column 1).

(4) Punched-card output (P or N in column 1).

The fifth and sixth control cards select the braille page dimensions. In both cases the required number is punched in columns 18-19:

(5) The number of braille signs per line (not less than 2 nor greater than 40).

(6) The number of lines per 11-inch page. Table VII gives suggested values for this item.

---

* On the IBM 360 under OS, it is possible to use IBM Operating System Job Control Language to run part of the text input with one set of DOTSYS III control cards, and change control cards for a succeeding part of the input. This could be done simply by using the IBM utility IEBGENER to copy the tabular input onto a temporary disk data set, say &&TABLES. Then, instead of executing the catalogued procedure COBFLG just once, do it once for each part of the text input. The SYSIN data set to use is the concatenation of &&TABLES with a DD * data set consisting of the control cards and the text of that part. Quite possibly, similar facilities exist on other systems.

Table VII

Suggested Number of Braille Lines Per Page

	Line Printer Setting Lines/Inch		
	6	8	10
Proof Output	10	13	16
Elastomer Braille Output	15	20	25

The last card determines whether or not automatic titling and page numbering is to occur:

(7) If the top line of each braille page is to be reserved for a running title and automatically produced page number, a "P" should be punched in column 1 and the starting page number should be punched in columns 32-35. Otherwise, a "N" should be punched in column 1. In this latter case, no automatic page numbering is done and running titles, while still permitted in the input text, will not appear on the output.

Text Input

General Keypunching Rules

Text is punched in columns 1-72 of each text input card using an EBCDIC keypunch (e.g., IBM 029) or the equivalent multiple punches on another model. Letters, numbers, and punctuation are reproduced as they appear in normal typewritten English text, with the exception of quotation marks, accent marks, mathematical symbols, and brackets ([,]). However, additional symbols must be added to the text to indicate capitalization, italics , and format controls, to force or prevent improper translations in exceptional cases, and to define the correct translation for self-checking purposes.

Column 72 of a card is considered to be adjacent to column 1 of the next card. In general, any number of spaces will be collapsed automatically into a single space (refer to the description of the

primary input processor), except following a period, where a number of spaces greater than or equal to 2 will be interpreted as 2 spaces; (e.g.:

THE START. THE END.

is interpreted as

THE START. THE END.)

Capitalization

If the first letter of the inkprint word is capitalized, the key-punched word must be preceded immediately by a logical-not sign (¬). If the whole inkprint word is capitalized, the key-punched word must be preceded immediately by two logical-not signs (¬¬). When Roman numerals are written as capital letters, a single logical-not sign must be used before a single letter and two logical-not signs must be used before numerals containing two or more letters.

Italics

If only one, two, or three successive inkprint words are italicized, each of the words must be preceded immediately by an underline(_) when keypunched.

If four or more successive inkprint words are italicized, the first word must be preceded immediately by two underlines (_ _) and the last by one underline.

Indicating Contractions in Self-Checking Mode

If the self-checking feature of DOTSYS III is to be used, then all contractions which should be made must be surrounded by vertical bars (|). Refer to the section entitled "Self-Checking", below.

Ordering

When two or more special signs must be punched (e.g., an italicized capital letter), the ordering should be:

Italic sign (_)

Capital sign(s) (¬ or ¬¬)

Accent sign (¢)

Vertical bar (|)

Forcing and Preventing Contractions

The form of any contractable letter group can be forced to occur, regardless of context, by surrounding the letter group with the symbols "/_" and "_/". For example:

a/_dd_/

would cause the "dd" contraction to be used even though otherwise the program would not (and should not) use it at the end of a word.

A contraction that would otherwise occur can be prevented by introducing the null replacement symbol $/ (see below). For example,

DISE$/ASE

will prevent the "EA" contraction.

Replacement Symbols for Special Characters

Mathematical Symbols. Symbols such as + (plus), - (minus), = (equal*), > (greater than), and < (less than) must be spelled out as words, even though they appear on the keypunch.

Quotation Marks. The single quote character on the keypunch (') is always taken to mean an apostrophe; thus, special symbols must be used for single quotes.

$' is punched for a left single quote.

$'R is punched for a right single quote.

Ordinarily, the double quote (") on the keypunch may be used for both left and right double quotes. However, double quotes within the scope of another pair of double quotes must be represented by special symbols.

$" is punched for a left double quote within quoted text.

$"R is punched for a right double quote within quoted text.

Accent Marks. Any accent or other diacritical mark used with a letter (such as é, è, ê, ä, ç) is represented by preceding the keypunched letter with a cents sign (¢). For example, Abbé is keypunched ¬ABB¢E.

---

* A keypunched equals sign is interpreted as a letter sign.

Brackets.

< is punched for a left bracket ([).

> is punched for a right bracket (]).

Short Syllable Sign. To insert a short syllable sign, the special symbol $SV is used.

Long Syllable Sign. To insert a long syllable sign, the special symbol $LV is used.

End of Poetry Foot Sign. To insert an end of poetry foot sign, the special symbol $FT is used.

Caesura Sign. To insert a caesura sign, the special symbol $CS is used.

Null Symbol. The symbol $/ is a null replacement symbol (not a blank) and is generally used to prevent contractions (see above).

Forced Blanks. To produce multiple blanks or otherwise control their placement, the symbol $B is provided. One blank is produced for each occurrence of the symbol (e.g., ABB$BC produces A C).

Format and Mode Control Symbols

Keypunching Rule for Spaces Around Format Controls. Unlike replacement symbols where spaces before or after the symbol are interpreted as spaces (i.e., word dividers), any spaces before or after format symbols are ignored. However, the general rule is to provide spaces around each format symbol to avoid possible ambiguity (e.g., $LVOICE would be interpreted as $LV OICE even though $L VOICE may have been intended). Otherwise, the control symbols will usually operate properly regardless of the presence or absence of blanks.

Paragraphs. The symbol $P must be keypunched to indicate the beginning of a new paragraph. The text following the $P is started on the next line after two spaces (that is, starting in the third cell position).

New Line. The symbol $L may be used to begin a new output line. Caution: at most one line will be skipped, even if the $L symbol is repeated. To skip more than one line, see "SKIP MULTIPLE LINES."

Skip Multiple Lines. The symbol $SLnnb, where b is a blank and nn is a two digit number (with a leading zero, if necessary) will skip the number of lines indicated, and output will continue from the left margin.

New Page. The symbol $PG may be used to begin a new page.

One-Time Tabulation. If the symbol $TABnnb is punched, where n's represent digits and b represents a blank, the text beginning immediately after the blank will be started at column nn. Tabulation is implemented by the automatic insertion of spaces into the output and will therefore proceed to the next line if nn is less than or equal to the present column number. A leading zero must be supplied if the column number is 9 or less.

Permanent Tabs. Numbered tabs can be set so that the user can right, left, or decimal point justify a word or number on any column. For example, the symbol $STB4L03 means set tab 4 to left justify on column 3. The symbol $STB means to set tab, followed by the one digit number of the tab to be set, followed by an L for left justification (alignment of the left most character), R for right justification or a D for decimal justification. The last two digit numbers are the column on which justification is to take place. After a tab has been set, it may be executed any number of times by inserting the symbol $# followed by the one digit number of the tab to be executed. (Example: $STB4L03 $#4 LEFT--will left-justify the word LEFT on column 3. The symbol $#4 can be used any number of times.) If a tab is called in a cell position less than or equal to the present position, output will begin on the next line.

The definition of a given tab number may be changed as often as desirable in the text. Initially, all tabs are set to left-justify on column (5 x tab number).

Titles. Titles are placed between the symbols $TLS (Title START) and $TLE (Title END), (e.g., $TLS THIS IS A SAMPLE TITLE $TLE.) After a title has been inserted, each subsequent page has a centered title as its first line (the same line which contains the page number). Pagination must be turned on for titles to appear on output (see "The Run Control Cards"). If a new title is entered, it takes the place of the old on all future pages. Entering the title does not automatically turn to a new page.

Headings. These are similar to titles except that the control symbols are $HDS and $HDE and that headings are a one time occurrence. Headings are centered on the next line with subsequent input beginning in column 1 of the line after the heading. Headings that overflow begin in Column 4 of successive lines.

Poetry. The symbols $PTYS (Poetry START) and $PTYE (Poetry END) are placed before and after all poetry text input. In this mode, the continuations of all poetry lines which exceed the physical length of the output line will be indented two spaces.

Turning the Self-Checking Mode On or Off. The symbol $SCON$/$/$/$/$/ is used to turn self-checking on and $SCOFF is used to turn self-checking off. (See "Self-Checking" below.)

Self-Checking

Self-checking is a feature that allows DOTSYS III to be used to check itself (and the contraction table) against a text specially marked to indicate the correct translation. This is accomplished by automatically comparing the results of two translations:

(1) the normal process, ignoring the special markings, and

(2) a trivial special process, wherein the markings are used to determine where contractions occur.

Discrepancies are flagged on the output, so that the clerical effort required to check the program's correctness is greatly reduced.

A deck containing 5,808 typical and problem words and phrases, taken from the Transcribers' Guide to English Braille (Reference 6), has been punched with the special markings necessary for self-checking.

In order to turn on self-checking, a control symbol ($SCON$/$/$/$/$/) must be included in the text. The text following this symbol is punched normally except that vertical bars (|) should immediately surround any sequence of letters that should be contracted in a word. For example,

ever y th/ing

should be punched |EVER|Y|TH||ING|

Note: A vertical bar at the end of the word and followed by a blank may be omitted; i.e.,

|EVER|Y|TH||ING is an acceptable variation.

---

* Throughout this report, an underlined sequence of two or more letters in an example denotes a group contractable in braille. Adjacent contractable letter groups are separated by a slash. This is consistent with common practice (Refs. 1 and 6).

Those braille words not passing the comparison test are output as they ordinarily would be but flagged by appending two braille "for" symbols (all dots). In the proof output, the comment "TSK nnn" appears to the right of any line containing a flagged word. (If the first word in a line is in error, the TSK appears on the preceding line. In this context, a word is any sequence of non-blank braille symbols.) The nnn is simply a sequence counter, incremented by one for each discrepancy.

There is only one known situation where, assuming correct input, the program will fail to flag a word that is incorrectly translated. That is when a letter sign should be added by the translator and is not; for example, the "ab" of

ab initio (italicized),

punched

_AB _|IN|ITIO,

should be preceded by a letter sign in braille. These cases are indicated by an asterisk in the Transcribers' Guide, so that manual checking of these is fairly easy.

More commonly, words are flagged as wrong that are in fact correct. There are a number of situations where this can occur:

(1) Synchronization: When, for whatever reason, the correspondence between inkprint (punched) words (sequences of non-blanks) and braille words is not one-for-one, a possibly spurious error due to synchronization will be flagged. For example, the first space in the phrase "by the by" is removed in braille; the word by will not compare with the "word" bythe, causing a synchronization error. Control inputs ($TAB, etc.) also cause spurious errors of this class, but controls are not normally used in problem word lists. Typically, the word where the synchronization failed and the one after it are flagged before proper synchronization is reestablished.

(2) "Perceiving": For the sake of saving space, only the first four braille sign codes are read into the alphabetical contraction table, even though five are punched on the table input card. This means that "perceiving," the only 5-sign contraction, is not properly represented in the list and so words containing this contraction will be improperly flagged.

(3) Periods: Words containing periods will be flagged because the alphabet table assumes that periods are decimal points unless the contraction table logic explicitly overrides this assumption.

All of the spurious error flagging (or failing to flag), except possibly for the synchronization errors, could be corrected with somewhat more logic and/or storage cost in the program. At least in the case of the 5,808 problem words, these problems have not been sufficiently bothersome to warrant the expenditure.

Octal Braille

Octal braille permits the user to generate arbitrary braille cells directly. A particular braille cell is selected by a two-digit code. This code is the sum of the numbers associated with the dots in the cell, according to the following association scheme:

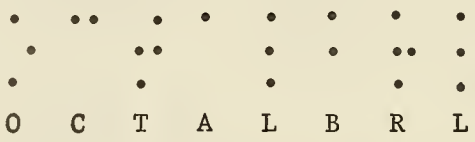
Braille Cell Dots	
10	• • 1
20	• • 2
40	• • 4

The codes for the desired braille cells are placed in sequence, following the special sign $OCT. The first blank after $OCT terminates the sequence.

The code is actually an octal representation of the braille cell. Example:

$OCT521163107000307270

will translate into



Computer Braille

Computer braille is similar to octal braille except that it is driven by characters rather than two-digit numbers and is initiated by the special sign $CPB. When using computer braille, a character is represented by the two-digit braille sign code punched in columns 21-22 of the corresponding alphabet table input card. This two-digit number is determined by the sum of the numbers associated with each braille dot in the table:

```
1 . .  8
2 . . 16
4 . . 32
```

Example:

Entry in alphabet table:

```
Column 1          18  21  24  30
                  ↓   ↓   ↓   ↓
%                 01  41  15  00
```

Text Input:

```
$CPB%%%%
```

will cause output

```
..  ..  ..  ..

 .   .   .   .
```

Notes to the Braille Editor

Role of the Editor

This section indicates where the intervention of the editor is required and presents the tools available to the editor to implement his intervention. Generally speaking, the nineteen problem situations listed in the Memorandum on Braille translation (Reference 5) by R. L. Haynes summarize where the intervention of the braille editor is needed.

There are basically two ways in which the editor can ensure proper translation in problem situations: (1) instructing the keypunch operator to add certain symbols to the input text; and (2) modifying the tables which control DOTSYS III. Modification of tables is explained under "Tables" above; the only time it would be done under normal circumstances would be when a problem word occurs often in a particular body of text; in that case, its correct translation would be added to the contraction table.

In the remainder of this section we shall discuss the special symbols which can be inserted in the input text and the situations in which they are helpful or necessary. These symbols are the letter sign (=), the grade switch ($G), the division symbol ($/), the

forced-contraction symbols (/_, _/), octal braille sign ($OCT ...), and the termination symbol ($T). The editor's attention is also directed to the sections on computer braille and the self-checking feature.

Letter Sign (=)

The editor must insert the letter sign when:

(1) a letter which means a letter stands alone and is not followed by a period indicating an abbreviation and is not italicized or surrounded by double quotes or parentheses.

(2) the capitalized or uncapitalized letter "a", "i", or "o" requires a letter sign in the braille, except when used after a number or as a word.

(3) combinations of letters that could be confused with short-form words appear in the input text. (It is suggested that a contraction table entry be used to insert the letter sign if such a letter combination occurs frequently. The tables already contain a number of frequently used abbreviations.)

In other situations the program inserts the letter sign automatically where necessary (where one has not already been inserted in the input).

Grade Switch ($G)

An occurrence of the grade switch, $G, changes the mode of translation from grade 2 to grade 1 or vice versa. It must precede and follow any sequence of characters in which no contractions are to be used, such as a foreign word.

Division Symbol ($/)

The division symbol is the most useful and versatile tool of the braille editor, although its operation is simple: it merely prevents contraction of a letter group which crosses it. For example, for the lisped word "thentury", the "the" contraction is avoided by inserting the division symbol after the "th", thus: TH$/ENTURY. It may be placed between the parts of compound words, such as NUT$/HATCH. It may be placed between prefixes and stems, as in BI$/NOMIAL, or indicate syllable division, as in PERITO$/NEUM and SKI$/DADDLE.

It must be used in a time interval after the hyphen separating minutes from hours, as in 9:30 - $/10:30, in order to produce a number sign correctly.

The division symbol may also be used in cases where the symbols to be translated are coincidentally DOTSYS III control symbols. For example, $$/TAB22 will be translated as "$TAB22" literally, avoiding the control function "tab to column 22." Even "$/" may be generated for output by supplying "$$//" as input.

Forced-Contraction Symbols (/_, _/)

These symbols are used to force a contractible letter group to be contracted. See "Text Input" for a complete description.

Octal Braille Sign ($OCT ...)

This facility allows an arbitrary sequence of braille signs to be inserted in the output. This is an alternative to $/ (q.v.) when the symbols to be translated into braille happen to be DOTSYS III input control symbols, and the control function is not desired. Another use would be in the preparation of special tactile effects, such as drawing diagrams using braille dots. See "Text Input" for a complete description.

Termination Symbol ($T)

This translates into the double sign dot-6, dot-3, in the output where required (Reference 1, Rule II.11a).

OUTPUT

Echo Output

When selected by the echo card (q.v.), a literal listing of the table input and run control cards is produced following the image of the echo card itself, which is always printed. This printout is placed on the SYSPRINT logical device, preceding the proof output (q.v.), if any. Error messages (q.v.) may be interspersed with the echo output.

Proof Output

Though proof output is optional (see "Run Control Cards"), it is normally called for except, perhaps, in final production runs. It is essentially a printout for the use of a sighted braille editor; the braille signs are represented as printed dots. This printout occurs on the same logical device (SYSPRINT) as the echo output and error messages (q.v.); error messages may be interspersed with the normal proof output.

The first three pages of proof output display the contents of the right context table, the decision table, and the transition table in a form much easier to read than the literal echo listing.

The remainder of the output is primarily a page-by-page, line-by-line representation of the braille output, using periods for braille dots. Below the braille sign are up to three characters intended to help identify the sign. If the sign represents a letter, for example, the letter itself is printed below the sign. The identification characters depend only on the sign (as determined by the sign table, q.v.) and not its context. For example, the sign for "K" has the identification character "K" even when it represents the word "knowledge." Below the identification characters is printed the braille sign code in the range 0 through 63, which may be used to check the punched output. Figure 2 contains a sample of proof output.

A literal listing of the text input cards, as read, is also produced between the lines of braille output (where a "braille output line" is actually a group of five printed lines). These card images will generally occur somewhat sporadically, with two or more occurring together at times and none between pairs of braille lines in other cases. (For separation, a blank line is printed in the latter case.) Typically, a given input word and the corresponding output are separated by about two braille lines.

If self-checking is in effect, the word TSK with a number immediately under it may appear on the far right of the page, to call attention to mistranslations. See the section entitled "Self-Checking."

Elastomer Braille Output

Elastomer braille output consists of the braille signs (dots) only, with each line produced in mirror-image. This output is suitable for a form of embossing on a line printer which has been modified by placing an elastic band between the print hammers and the paper. The elastic band might be a rubber band, soft polyurethane, a garter belt, or whatever the user's ingenuity can provide. It can be held on by tape or wire at the ends, or, on an IBM 1403 line printer, by hooks obtainable free of charge from IBM if requested under the number RPQ 818047. The number of lines per inch should be set as close as possible to ten.

This output, if selected, is placed on logical device SYSBRL. Figure 3 contains a sample of Elastomer braille as it would appear on the printed, or depressed dot (as opposed to raised dot) side.

```
    . .  .. ...   .     .  .  .     .     .
   .        .  .   .    .  .. .     .   .  .
 . .     .  .  .          .  .     .  .
   S  A  M  P  L  E     B  R  L     L  IN E
32 14 01 13 15 07 17 00 03 23 07 00 07 20 17 00 00 00 00 00
```

Figure 2. Sample of Proof Output

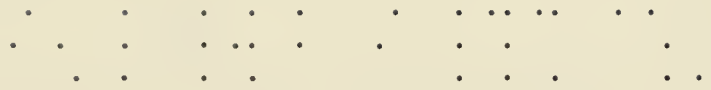


Figure 3. Elastomer Braille Output for Line in Figure 2

RPQ Braille Output

RPQ braille is similar to elastomer braille (q.v.) in most respects, with an additional modification to the print chain (IBM RPQ F19229) so that spacing of the dots is closer to the braille standard. This output, if selected, is placed on logical device SYSRPQ.

Punched Card Output

Punched card output makes possible the processing of DOTSYS III's braille output by other programs--for example, to produce tactile braille on a computer system which can drive an embosser but cannot support a suitably modified DOTSYS III.

Each card represents one braille line; page boundaries are not represented. Forty two-digit braille sign codes are punched on each card, with 00's filling out the line on the right. Figure 4 illustrates part of a card containing punched output.

Punched output is placed on logical device SYSPUNCH. Of course, on systems supporting a logical file concept, this device need not be physically a card punch but could be, for instance, a tape containing equivalent card images.

Error Messages

Error messages are unconditionally printed on SYSPRINT, thereby appearing intermixed with echo and proof output, if any. The following is a list of the possible messages. For each entry, "A" is an explanation, "B" is the program's automatic action, and "C" is the suggested user response.

(1) NEW CHAR X

A. The character printed (X) has appeared in the input text but does not have an alphabet table entry.

B. The character is replaced by an asterisk and processing proceeds. (Note: processing stops if asterisk is not in the alphabet table.)

C. Correct the text input or provide an alphabet table entry.

```
3214011315071700032307000720170000000000000000000000000000000000
```

Figure 4. Punched Card Output for Line in Figure 2

(2) ** TABLE SEQUENCE ERROR

A. A contraction table section is out-of-sequence with respect to the alphabet table order.

B. The "echo" option is turned on, if not on already, so that at least the remainder of the tables will be printed. Reading of the tables is continued but processing of the text is suppressed.

C. Refer to "Input" for a discussion of the relationship between the alphabet table arrangement and that of the contraction table, and rearrange accordingly.

(3) **FOLLOWING CARD(S) OUT OF SEQUENCE

A. An out-of-order card was found in the section of the input containing the echo card, the tables and the run control cards.

B. Processing of this section is continued but processing of the text is suppressed.

C. Reestablish correct order or correct the sequence number.

SECTION IV

PROGRAM TRANSFER

This section contains a checklist of steps required to bring DOTSYS III into operation on any computer having a Standard COBOL compiler and sufficient core storage to execute DOTSYS III. 'Standard COBOL' means COBOL according to Reference 4 having the level 1 nucleus, table handling, and sequential access. The core storage required depends on the compiler and on the size of the contraction table. With the IBM 360 level F compiler and 1,000 contraction table entries, about 59,000 bytes are required.

It is assumed here that the program and tables will be keypunched from the listings appended to this manual. If a card deck or the equivalent has been provided, information about how it was produced, and how it is to be used, should accompany it.

The program can be made about fifteen per cent faster, without affecting its translation capability, by replacing the alphabet search by a machine-dependent indexing scheme, at the expense of reducing further transferability and increasing the program size. The method of doing this is explained under "Maintenance."

Here is the checklist of steps to transfer DOTSYS III:

(1) Rewrite the environment division of the program in accordance with the COBOL manual for your computer and peripheral units. Keep in mind that SYSINPUT is the input file, and SYSPRINT, SYSPUNCH, SYSBRL, and SYSRPQ are output files. Other information about them is in the file description (FD) entries in the data division.

(2) Check the file description entries (in the file section of the data division) to ensure that the block size and label records are specified correctly for your installation.

Punched output records may or may not begin with a control character (for pocket selecting), and the control character, if present, varies with the installation. DOTSYS III is set up to place a control character in the beginning of each punched output record; the control character used is the value of the data item POCKET-SELECT in the

working storage section. If no control character is to be used, remove the 02-level data item FILLER in CODED-OUTPUT, and change the sentence

```
WRITE CODED-OUTPUT AFTER POCKET-SELECT
```

in the BREAK6 paragraph of the output section to

```
WRITE CODED-OUTPUT
```

(3) Check the COBOL character set at your installation against the characters in the program. Some characters, such as the single quote ('), greater than (>), less than (<), and equals (=), may have to be replaced by other characters or expressions.

(4) Check the character set of your computer and keypunch against the characters in the tables. Remove or replace table entries containing illegal characters. Inkprint characters having no single-character machine-readable form may be represented by multiple-character symbols, just as in the case of single quotation marks. They are implemented via contraction table entries. Be sure to inform the keypunch operator of your choices in this matter.

(5) If your COBOL does not permit the COMPUTE verb, replace each occurrence of it by an equivalent sequence of individual arithmetic operations. For example, the sentence "COMPUTE I = N * M + J." may be replaced by the two sentences

```
MULTIPLY N BY M GIVING I.

ADD J TO I.
```

(6) Compile the program to obtain an object deck. If the object program is too large, try leaving out the table display routine, from TABLE-DISPLAY through SKIP-DISPLAY. If the self-checking feature is not to be used, another possibility would be to shorten the alphabetic contraction table, or even to remove it and all associated code. As as last resort, decrease the contraction table bound (and shorten the table input accordingly).

(7) Execute the program, using standard test case input if possible.

SECTION V

PROGRAM MAINTENANCE

WHEN MAINTENANCE MAY BE REQUIRED

This section presents certain details of the COBOL implementation of DOTSYS III. It is for use in making changes in the tabular input, and in the program itself, that may be made necessary or desirable by the transfer of DOTSYS III to another installation, a change in peripheral units, a change in the braille translation rules, or a desire to improve the conformity of the translation with the braille ideal.

TABLE-SIZE BOUNDS

The maximum capacities for a number of tables are conceptually variable, in that only a few OCCURS clauses or other references need be changed and the program recompiled, to effect an increase or decrease. These items are listed in Table VIII.

Table VIII

Variable Table Capacity References

Table	Program Card Numbers
Alphabetic Contraction	00033100, 00039800, 00091700
Alphabet	00024100
Contraction	00020800
Right Context	00025000
State Variables	00025900, 00026200
Input Classes	00025800, 00026300
Decision Table (width)	00025700

Table VIII (Concluded)

Variable Table Capacity References

Table	Program Card Numbers
Stacks (no. of entries)	00022500, 00036000, 00036100
Stack (entry width)	00023000, 00101000, 00102600, 00133100
Self-Checking Ring (entry width)	00023500, 00088700, 00093200, 00094100

Note that increasing the size of a table in such a way that the number of digits required to index it is increased (e.g., a change from 99 to 101) may require that the PICTURE clause for data items used as indices (subscripts) to that table may have to be changed (e.g., from S99 to S999). Note also that in some cases a dummy (terminator) entry is actually stored in the table.

REPLACING THE ALPHABET TABLE SEARCH BY DIRECT INDEXING

The first step in translating the current contents of the buffer is to find the alphabet table entry for the leftmost character in the buffer. In order to preserve machine-independence, DOTSYS III does this by searching the alphabet table linearly from the top for a match with the leftmost character in the buffer. If the alphabet table has been ordered with the higher-frequency items nearer the top, an average of about nine entries is tested. On the IBM 360/50, that takes about one millisecond, or roughly fifteen per cent of the average time spent per character.

On some machines, including the IBM 360, it is possible to reduce drastically the time required to find the appropriate alphabet table entry, by using the test character itself as an index. This is done by moving the character to a data item defined as USAGE DISPLAY (the default case) but redefined as USAGE COMPUTATIONAL. For example, on the IBM 360, the data item below describes a half-word integer whose value is determined by the character moved into its right-hand byte; its left-hand byte is binary zeroes.

```
01   CHARACTER-INDEX.
     02   LEFT                PICTURE X, VALUE LOW-VALUE
     02   RIGHT               PICTURE X.
01   N REDEFINES CHARACTER-INDEX,
                               PICTURE S999, USAGE COMPUTATIONAL.
```

Thus, if a character is moved to RIGHT, then N is a number determined by that character, and can be used as an index into an array of pointers to the appropriate places in the alphabet table. For example, suppose the array of pointers is called ARRAY-OF-POINTERS and the left most character in the buffer is a space, whose alphabet table entry comes first. Moving the space to RIGHT gives N a value of hexadecimal 40 or decimal 64; the 64th entry in ARRAY-OF-POINTERS should be 1 since the space entry is first in the alphabet table. Thus, the two sentences,

```
MOVE RL1 TO RIGHT.

MOVE ARRAY-OF-POINTERS (N) TO LETTER.
```

replace the alphabet table search, provided that ARRAY-OF-POINTERS has been properly initialized, which takes only two sentences in the READ-ALPHABET loop during initialization. The same thing can be done to replace the search occurring in the paragraphs following TEST-NON-TERMINAL (which check right context) in the translation section.

Note that with indexing, there is no longer any need to order the alphabet table according to frequency; however, the order of contraction table sections must still match the order of the associated alphabet table entries.

The price paid for the reduced search time is an increased storage requirement, since ARRAY-OF-POINTERS must have 2^k entries, where k is the number of bits in a character. On the IBM 360, ARRAY-OF-POINTERS would take up $2^8 x 2 = 512$ bytes.

CONTRACTION TABLE SEARCH ALGORITHM

General Rationale

A form of tree search has been implemented for comparison of a string against the contraction table. This algorithm is inherently much more efficient than a linear search for any size table. Moreover, the proportionate cost (in time) for new entries is reduced: the time increases only logarithmically, rather than directly.

The method takes advantage of the fact that often a comparison failure on, say,the third character implies that quite a few additional entries (with the identical first three characters) can be skipped around. The method also conserves space in that only one numeric field must be added to each table entry to support the algorithm.

The basic idea is quite simple. In a given initial-letter section of the table, all the entries which start a new subsection (wherein the first two letters are identical) are chained together, using the added "branch" field as a forward link. This forms the "level 1" chain; a level 2 chain may be formed and so forth. The implicit structure is that of a binary tree. During search, one either follows the branch (continued on the current chain) if comparison failure occurs at the character whose index equals the current level, or else goes to the next entry and one level higher.

The main complication with this process is that, having reached the highest point in the tree and still not having found a match, it still may be necessary to return to the lower level. For example, "AB" may be at level 2 and would probably follow "ABCDE"--at, say, level 4--in the table. The key "ABCDF" would thus reach at least level 4 and yet may not actually match until the "AB" is encountered. The handling of this situation in the algorithm is facilitated by indicating the level of the next entry whenever the current level chain ends. This level is entered in the branch field, in negative form so as not to be confused with a forward pointer and also to indicate that the forward pointer is null.

It should be noted that the table must be properly ordered on input in order for the algorithms to work. The proper ordering condition is that entries whose first p letters correspond must be together in the table; formally, using the notation defined below: if there exist two entries, with indices q and r ($q < r$) and an integer $p > 0$ such that $C_{qj} = C_{rj}$ for all j in the range $1 \leq j \leq p$, then for all t in the range $q \leq t \leq r$ it is also true that $C_{qj} = C_{tj}$ for all j in the range $1 \leq j \leq p$.

Notation

The notation used in the flow charts (Figures 5 and 6) is as follows:

LET

C_{ij} be the j^{th} character in the i^{th} entry of the contraction table.

b_i be the value of the branch field for the i^{th} entry.

s_k be the index of the first entry in the contraction table for the k^{th} initial letter.

e_k be the index of the last entry for the k^{th} initial letter.

L be the index for the lowest entry to be considered at the current level.

H be the index for the highest entry to be considered at the current level.

V be the highest index for the current initial letter.

n be the current level.

A_n be the upper index bound for the n^{th} level.

m be the number of entries in the contraction table, not counting the extra "end of table" entry.

M be the number of initial letters.

w_j be the j^{th} character in the current input string (string being looked up).

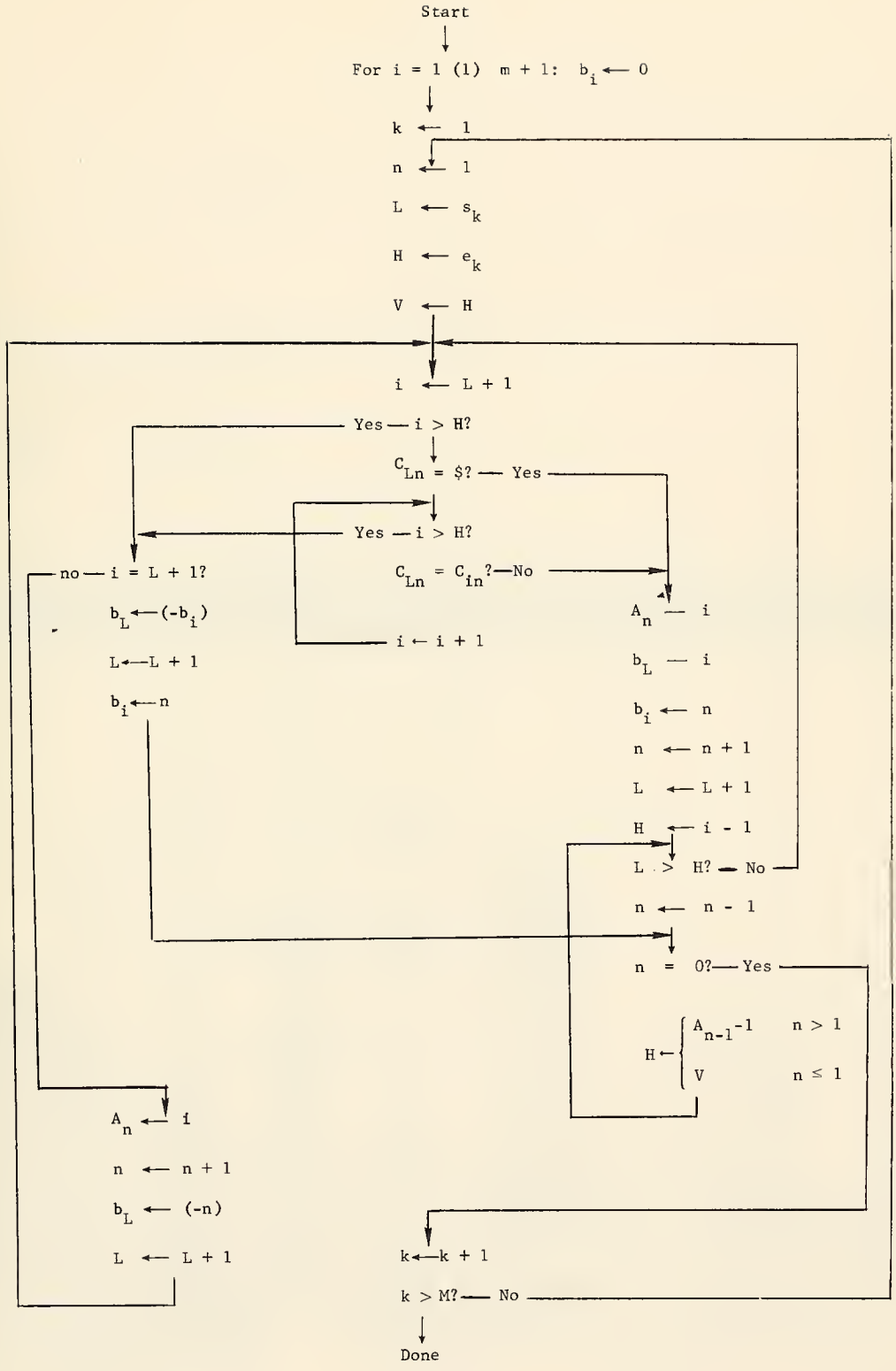


Figure 5. Set-Up Algorithm

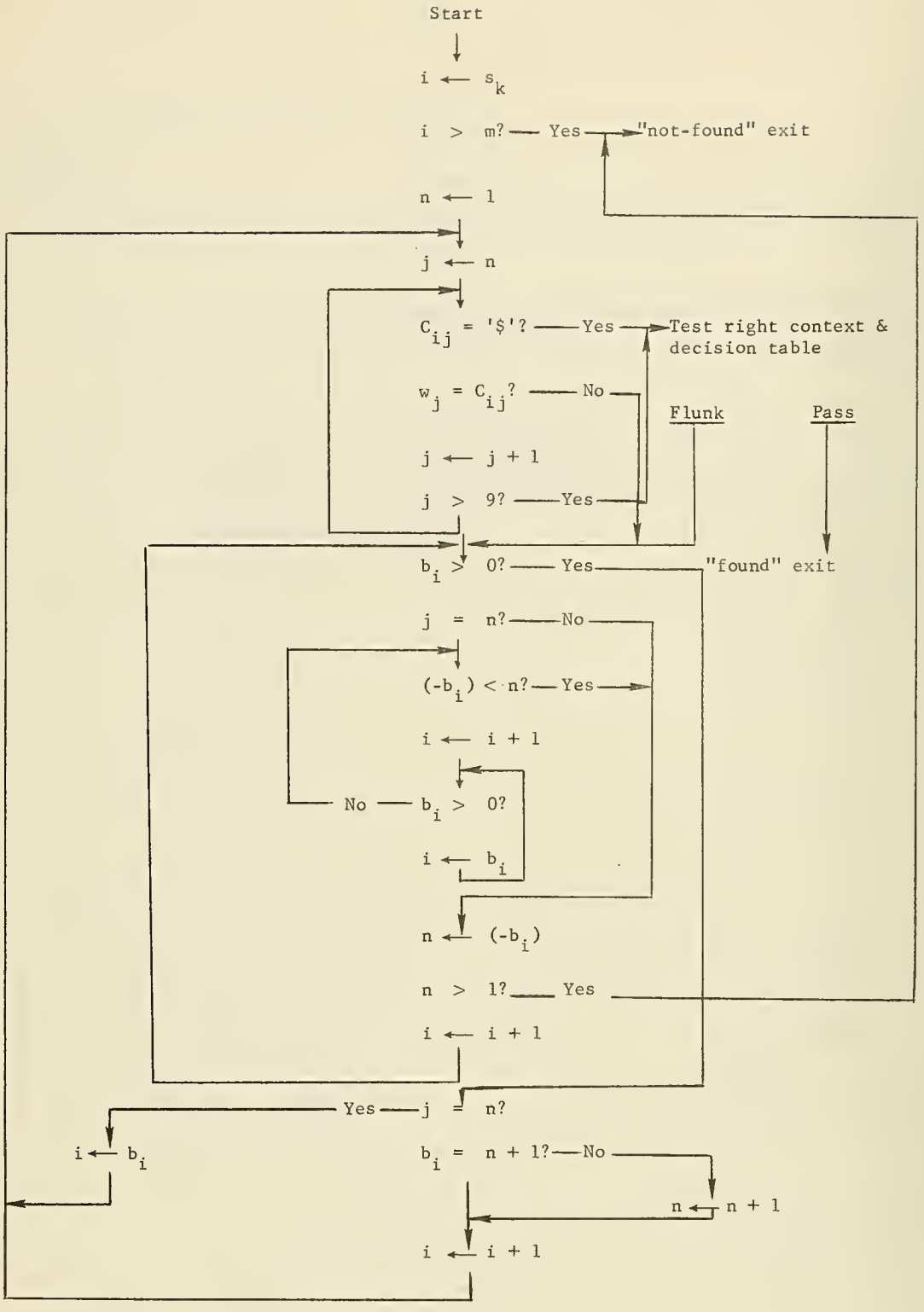


Figure 6. Look-Up Algorithm

LISTING OF COMPLETE IBM 360 JOB DECK


```
//COB EXEC PGM=IEQC8L00,REGION=86K,                                  X0000010C
// PARM='NOBASIS,NCCCPY,SEQ,CLIST,NCMAP,DECK'
//SYSUT1 DD UNIT=SYSDA,SPACE=(46C,I700,100))                          00000300
//SYSUT2 OD UNIT=SYSDA,SPACE=(46C,(700,100))                          00000400
//SYSUT3 OO UNIT=SYSDA,SPACE=(460,I700,100))                          00000500
//SYSUT4 OD UNIT=SYSDA,SPACE=(46C,I700,100))                          00000600
//SYSLIN DD DSNAME=&LDAOSET,DISP=(MOD,PASS),UNIT=SYSDA,               X0000C700
//  SPACE=(80,(500,100)),DCB=BLKSIZE=80                               00000800
//SYSPRINT DD SYSOUT=A,DCB=IRECFM=FBA,LRECL=121,BLKSIZE=121)          00000900
//SYSPUNCH OD SYSOUT=B,DCB=BLKSIZE=80
//SYSIN DD *                                                          00001100
       PROGRAM-ID.  'COTSYS3'                                        00001200
       AUTHOR.  W. R. GERHART, J. K. MILLEN, J. E. SULLIVAN.         00001300
       INSTALLATION.  MIT, MITRE.                                    0000140C
       DATE-WRITTEN.  AUGUST 1970.                                   00001420
       DATE-COMPILED.  1970.                                         00001430
       REMARKS.                                                      00001450
           FIRST REVISION OCTOBER 1970.                              00001451
             CHANGE IN STACKING LOGIC FOR UNITS-OF-MEASURE REVERSAL. 00001452
             PERMIT DYNAMIC SELF-CHECKING SWITCH.                    00001453
             SEVERAL MINOR FIXES AND MODIFICATIONS.                  00001454
                                                                     C0001500
       ENVIRONMENT DIVISION.                                         00001600
                                                                     00001700
       CONFIGURATION SECTION.                                        000018C0
       SOURCE-COMPUTER.  IBM-360.                                    00001900
       OBJECT-COMPUTER.  IBM-360.                                    00002000
                                                                     00002100
       INPUT-OUTPUT SECTION.                                         00002200
       FILE-CONTROL.                                                 00002300
           SELECT SYSINPUT, ASSIGN TO 'SYSIN' UTILITY.               00002400
           SELECT SYSPRINT, ASSIGN TO 'SYSPRINT' UTILITY.            00002500
           SELECT PUNCH, ASSIGN TO 'SYSPUNCH' UTILITY.               00002600
           SELECT SYSBRL, ASSIGN TO 'SYSBRL' UTILITY.                00002700
           SELECT SYSRPQ, ASSIGN TO 'SYSRPQ' UTILITY.                00002800
                                                                     00002900
       DATA DIVISION.                                                00003000
                                                                     00003100
                                                                     00003200
       FILE SECTION.                                                 00003300
                                                                     00003400
       FD  SYSINPUT                                                  00003500
           LABEL RECORDS ARE STANDARD,                               00003600
           RECORDING MODE IS F,                                      00003700
           RECORD CONTAINS 80 CHARACTERS,                            00003800
           BLOCK CONTAINS 0 RECORDS,                                 00003900
           DATA RECORD IS INPUT-RECORD.                              00004000
                                                                     00004100
       01  INPUT-RECORD.                                             00004200
           02  TEXT.                                                 00004300
              03  CHAR OCCURS 80 TIMES, PICTURE X.                   00004400
           02  TABLE-INFO REDEFINES TEXT.                            00004500
               03  FIELD1.                                           00004600
```

```
           04  FIELD11   PICTURE X(1).                          00004700
           04  FIELD12   PICTURE X(9).                          00004800
       03  FIELD1WHOLE REDEFINES FIELD1.                        00004900
           04  FIELD1W PICTURE X(10).                           00005000
       03  FILLER        PICTURE X(5).                          00005100
       03  FIELD2        PICTURE X.                             00005200
       03  FILLER        PICTURE A.                             00005300
       03  FIELD3        PICTURE 99.                            00005400
       03  FILLER        PICTURE A.                             00005500
       03  FIELD4        PICTURE 99.                            00005600
       03  FILLER        PICTURE A.                             00005700
       03  FIELD5.                                              00005800
           04  FIELD51   PICTURE 99.                            00005900
           04  FIELD52   PICTURE 99.                            00006000
           04  FIELD53   PICTURE 99.                            00006100
           04  FIELD54   PICTURE 99.                            00006200
     03  FIELD6 PICTURE 9999.                                   00006300
     03  FILLER PICTURE X(37).                                  00006400
     03  CARD-XX PICTURE 99999999.                              00006410
                                                                00006500
FD  SYSPRINT                                                    00006600
    LABEL RECORDS ARE STANDARD,                                 00006700
    RECORDING MODE IS F,                                        00006800
    DATA RECORD IS OUTPUT-LINE.                                 00006900
                                                                00007000
01  OUTPUT-LINE.                                               00007100
    02  FILLER            PICTURE A.                           00007200
    02  OUT.                                                   00007300
        03  OUT-PLACE     OCCURS 120 TIMES, PICTURE X.         00007400
                                                                00007500
FD  PUNCH                                                       00007600
    LABEL RECORDS ARE STANDARD,                                 00007700
    RECORDING MODE IS F,                                        00007800
    DATA RECORD IS CODED-OUTPUT.                                00007900
                                                                00008000
01  CODED-OUTPUT.                                              00008100
    02  FILLER            PICTURE X.                           00008200
    02  CODED-OUT         PICTURE X(80).                       00008300
                                                                00008400
FD  SYSRL                                                       00008500
    LABEL RECORDS ARE STANDARD,                                 00008600
    RECORDING MODE IS F,                                        00008700
    DATA RECORD IS OUTPUT-RL.                                   00008800
01  OUTPUT-RL.                                                 00008900
    02  FILLER PICTURE A.                                      00009000
    02  OUT-RL.                                                00009100
      03  OUT-PLACE-RL OCCURS 120 TIMES, PICTURE X.            00009200
                                                                00009300
FD  SYSRPQ                                                      00009400
    LABEL RECORDS ARE STANDARD,                                 00009500
    RECORDING MODE IS F,                                        00009600
    DATA RECORD IS OUTPUT-RPQ.                                  00009700
01  OUTPUT-RPQ.                                                00009800
```

```
    02  FILLER PICTURE X.                                          00009900
    02 OUT-RPQ.                                                    00010000
       03 OUT-RPQ-BEGIN PICTURE X.                                 00010100
       03 OUT-PLACE-RPQ OCCURS 24 TIMES.                           00010200
          04 OUT-RPQ-ONE PICTURE XX.                               00010300
          04 OUT-RPQ-TWO PICTURE XXX.                              00010400
WORKING-STORAGE SECTION.                                          00010500
                                                                  00010600
01  GENERAL-VARIABLES.                                           00010610
    07  LEVEL-ARRAY.                                             00010611
    08  L-ARRAY PICTURE S9999, USAGE COMPUTATIONAL,               00010612
    OCCURS 10 TIMES.                                              00010614
    07  SNAP-I PICTURE S9999, USAGE COMPUTATIONAL.                00010680
    07  FOUND-SECOND-SYMBOL  PICTURE X.                          00010700
    07  RCC PICTURE X, VALUE '|'.                                00010710
    07  I                    PICTURE S999, USAGE COMPUTATIONAL.  00010800
    07  INDEX                PICTURE S999, USAGE COMPUTATIONAL.  00010900
    07  INITIAL-STATE        PICTURE X(10), VALUE 'NNNNNNNNNN'.  00011000
    07  INPTR PICTURE S99, USAGE COMPUTATIONAL.                  00011100
    07  J                    PICTURE S999, USAGE COMPUTATIONAL.  00011200
    07  K                    PICTURE S999, USAGE COMPUTATIONAL.  00011300
    07  LETTER               PICTURE S99, USAGE COMPUTATIONAL.   00011400
    07  LINECOUNT            PICTURE S99, USAGE COMPUTATIONAL.   00011500
    07  LPG                  PICTURE S99, USAGE COMPUTATIONAL.   00011700
    07  M                    PICTURE S99, USAGE COMPUTATIONAL.   00011800
    07  N                    PICTURE S999, USAGE COMPUTATIONAL.  00011900
    07  NALPHABET            PICTURE S999, USAGE COMPUTATIONAL.  00012000
    07  NOTC                 PICTURE S99, USAGE COMPUTATIONAL.   00012200
    07  NIC                  PICTURE S99, USAGE COMPUTATIONAL.   00012300
    07  NN                   PICTURE S999, USAGE COMPUTATIONAL.  00012400
    07  NNT                  PICTURE S99, USAGE COMPUTATIONAL.   00012500
    07  NSV                  PICTURE S99, USAGE COMPUTATIONAL.   00012600
    07  NXTCHR PICTURE X.                                        00012700
    07  OUTPTR PICTURE S99,          USAGE COMPUTATIONAL.        00012800
    07  OUTPTRSV PICTURE S99, USAGE COMPUTATIONAL.               00012810
    07  OUTL                 PICTURE S99, USAGE COMPUTATIONAL.   00012900
    07  OUTSIGN              PICTURE S99, USAGE COMPUTATIONAL.   00013100
    07  POCKET-SELECT        PICTURE X, VALUE 'V'.               00013200
    07  POINTER              PICTURE S99, USAGE COMPUTATIONAL.   00013300
    07  BRAILLE PICTURE X.                                       00013400
    07  RPQ PICTURE X.                                           00013500
    07  PROOF-OUT PICTURE X.                                     00013600
    07  PUNCHED-OUTPUT       PICTURE X.                          00013700
    07  TABULATE             PICTURE S99, USAGE COMPUTATIONAL.   00013800
    07  TEMP PICTURE X(9).                                       00014000
    07  TEMPL                PICTURE X.                          00014100
    07  THREE-SPACES         PICTURE AAA, VALUE SPACES.          00014200
    07  MAXEXTENT PICTURE S99, USAGE COMPUTATIONAL.              00014300
    07  II PICTURE S999, USAGE COMPUTATIONAL.                    00014600
    07  TITLE-LENGTH PICTURE S99, USAGE COMPUTATIONAL.           00014700
    07  WORD-ERROR PICTURE S9, USAGE COMPUTATIONAL.              00014800
    07  ECHO PICTURE X.                                          00014900
    07  VHIGH PICTURE S999, USAGE COMPUTATIONAL.                 00015000
```

```
    C7  STACK-INDICATOR PICTURE S99, USAGE COMPUTATIONAL.          00015100
    C7  SAVE-CURRENT PICTURE S99, USAGE COMPUTATIONAL.             00015200
    C7  TAB-COL PICTURE S99, USAGE COMPUTATIONAL.                  00015300
    C7  CARDS-PRINTED PICTURE S999, USAGE COMPUTATIONAL.           00015310
    C7  POETRY-INDICATOR PICTURE S9, USAGE COMPUTATIONAL.          00015400
    C7  SELF-TEST PICTURE S9, USAGE COMPUTATIONAL.                 00015500
    C7  CC-HIGH PICTURE S99, USAGE COMPUTATIONAL, VALUE 80.        00015600
    C7  SC-HIGH PICTURE S99, USAGE COMPUTATIONAL, VALUE 90.        00015700
    C7  HEAD-FREE-STACK PICTURE S99, USAGE COMPUTATIONAL.          00015800
    C7  CURRENT-TYPE PICTURE S9, USAGE COMPUTATIONAL.              00015900
    07  CURRENT-TRANS PICTURE S99, USAGE COMPUTATIONAL.           00016000
    C7  NXT-LEVEL PICTURE S9, USAGE COMPUTATIONAL.                 00016100
    C7  HS PICTURE S99, USAGE COMPUTATIONAL.                       00016200
    C7  CS PICTURE S99, USAGE COMPUTATIONAL.                       00016300
    07  TS PICTURE S99, USAGE COMPUTATIONAL.                      00016400
    C7  PS PICTURE S99, USAGE COMPUTATIONAL.                       00016500
    C7  X PICTURE S99, USAGE COMPUTATIONAL.                        00016600
    C7  HIGH-NO PICTURE S999, USAGE COMPUTATIONAL.                 00016700
    C7  LOW-NO  PICTURE S999, USAGE COMPUTATIONAL.                 00016800
    C7  NCS PICTURE S99, USAGE COMPUTATIONAL.                      00016900
    07  TRANS-CLASS PICTURE S99, USAGE COMPUTATIONAL.             00017000
    C7  XYZ         PICTURE S99, USAGE COMPUTATIONAL.              00017100
    C7  SPACE-COUNT  PICTURE S99, USAGE COMPUTATIONAL.             00017200
    07  CURRENT-IN-TRANS PICTURE S99, USAGE COMPUTATIONAL.        00017300
    07  L-NXTCHR PICTURE S999, USAGE COMPUTATIONAL.               00017400
    C7 PRIOR-SPACE PICTURE S9, USAGE COMPUTATIONAL.                00017500
    07 IN-CONTRACT PICTURE S9, USAGE COMPUTATIONAL.                00017600
    C7 C-L-LOW PICTURE S99, USAGE COMPUTATIONAL.                   00017700
    07 C-L-HIGH PICTURE S99, USAGE COMPUTATIONAL.                  00017800
    07 C-L-CURR PICTURE S99, USAGE COMPUTATIONAL.                  00017900
    C7 BRL-E-I PICTURE S99, USAGE COMPUTATIONAL.                   00018000
    C7 N-O-E-T PICTURE S99, USAGE COMPUTATIONAL.                   00018100
    C7  XX PICTURE S999, USAGE COMPUTATIONAL.                      00018200
    C7  NCT  PICTURE S999, USAGE COMPUTATIONAL.                    00018300
    C7  PAGINATION PICTURE X.                                      00018400
    C7    SEQ-ERR-IND PICTURE X.                                   00018410
    C7     PREV-CARD-NO PICTURE 99999999.                          00018420
    07     CARD-NO PICTURE 99999999.                               00018440
    C7  SC-ERROR-CT PICTURE 999.                                   00018445
    07 HEAD-INO PICTURE 9.                                         00018450
01  BUFFER.                                                       00018500
    02  L.                                                        00018600
       03  RL9 PICTURE X(9).                                      00018700
       03  RR1 PICTURE X.                                         00018800
    02  R REDEFINES L.                                           00018900
       03  RL1 PICTURE X.                                         00019000
       03  RR9 PICTURE X(9).                                      00019100
    C2  RCHARS REDEFINES L.                                       00019200
       03  FIRSTCHAR PICTURE X.                                   00019300
       03  RCHAR OCCURS 9 TIMES, PICTURE X.                       00019400
    C2 TAB-DEF REDEFINES L.                                       00019500
       03 FRSTCHARS PICTURE 99.                                  00019600
       03 SCNDCHARS PICTURE 99.                                  00019700
```

```
      03 LASTCHARS PICTURE X(6).                                    00019800
    02 XXXX REDEFINES L.                                           00019900
      03 ONECHAR PICTURE 9.                                         00020000
      03  RESTCHAR OCCURS 9 TIMES, PICTURE 9.                      00020100
                                                                    00020200
01  STATE-VECTOR.                                                  00020300
    02  STATE-VARIABLE    OCCURS 10 TIMES, DEPENDING ON NSV,       00020400
                           PICTURE X.                               00020500
                                                                    00020600
01  CONTRACTION-TABLE.                                             00020700
    02  TABLE-ENTRY   OCCURS 1000 TIMES.                           00020800
        03  STRING.                                                00020900
            04  TABLE-CHAR OCCURS 9 TIMES, PICTURE X.              00021000
        03  RIGHT-CONTEXT PICTURE X.                               00021100
        03  INPUT-CLASS   PICTURE S99, USAGE COMPUTATIONAL.        00021200
        03  SHIFT         PICTURE S99, USAGE COMPUTATIONAL.        00021300
        03  SIGNS.                                                 00021400
            04  SIGN      OCCURS 4 TIMES, PICTURE S99,             00021500
                           USAGE COMPUTATIONAL.                     00021600
        03  BRANCH PICTURE S999, USAGE COMPUTATIONAL.              00021700
                                                                    00021800
01  STACK-TYPE.                                                    00021900
    02 STACK-PTR OCCURS 2 TIMES.                                   00022000
      03 HEAD-S PICTURE S9, USAGE COMPUTATIONAL.                    00022100
      03 TAIL-S PICTURE S9, USAGE COMPUTATIONAL.                    00022200
      03 CURRENT-S PICTURE S9, USAGE COMPUTATIONAL.                 00022300
01  PUSH-POP-STACKS.                                               00022400
    02 STACK OCCURS 19 TIMES.                                      00022500
      03 PREVIOUS-STACK PICTURE S9, USAGE COMPUTATIONAL.            00022600
      03 NEXT-STACK PICTURE S9, USAGE COMPUTATIONAL.                00022700
      03 SPECIAL-CONTROL PICTURE S99, USAGE COMPUTATIONAL.          00022710
      03 NO-OF-ELEMENTS       PICTURE S99,                          00022800
         USAGE COMPUTATIONAL.                                       00022900
      03 ELEMENT OCCURS 30 TIMES, PICTURE S99, USAGE                00023000
         COMPUTATIONAL.                                             00023001
01  CORRECT-TRANSLATIONS.                                          00023100
    02 TRANS OCCURS 6 TIMES.                                       00023200
      03 NO-OF-ELEMENTS-TRANS PICTURE S99,                          00023300
         USAGE COMPUTATIONAL.                                       00023400
      03 ELEMENT-TRANS OCCURS 30 TIMES, PICTURE S99,                00023500
         USAGE COMPUTATIONAL.                                       00023501
01  TAB-TABLE.                                                     00023600
    02 TAB OCCURS 9 TIMES.                                         00023700
      03 TAB-TYPE PICTURE X.                                        00023800
      03 TAB-COLUMN PICTURE S99, USAGE COMPUTATIONAL.               00023900
01  ALPHABET.                                                      00024000
    02  ALPHABETIC-ENTRY OCCURS 64 TIMES.                          00024100
        03  SYMBOL         PICTURE X.                              00024200
        03  CHAR-CLASS      PICTURE S99, USAGE COMPUTATIONAL.      00024300
        03  SINGLE-SIGN   PICTURE S99, USAGE COMPUTATIONAL.        00024400
        03  EXTENT         PICTURE S999, USAGE COMPUTATIONAL.      00024500
      03  COMP-8  PICTURE S99, USAGE COMPUTATIONAL.                00024600
       03  ISO-LETTER  PICTURE S9, USAGE COMPUTATIONAL.            00024700
```

```
                                                                00024800
01  RIGHT-CONTEXT-TABLE.                                       00024900
    02  RIGHT-CONTEXT-TABLE-ENTRY OCCURS 2 TIMES.              00025000
        03  NON-TERMINAL  PICTURE X.                           00025100
        03  RIGHT-CONTEXT-CLASSES.                             00025200
            04  RIGHT-CONTEXT-CLASS OCCURS 4 TIMES,            00025300
                        PICTURE S99, USAGE COMPUTATIONAL.       00025400
                                                                00025500
01  DECISION-TABLE.                                            00025600
    02  DECISION-TABLE-COLUMN OCCURS 15 TIMES.                 00025700
        03  DECISION OCCURS 25 TIMES, PICTURE X.               00025800
        03  CONDITION     OCCURS 10 TIMES, PICTURE X.          00025900
                                                                00026000
01  TRANSITION-TABLE.                                          00026100
    02  TRANSITION-TABLE-COLUMN OCCURS 10 TIMES.               00026200
        03  TRANSITION   OCCURS 25 TIMES, PICTURE X.           00026300
                                                                00026400
01  SIGN-TABLE.                                                00026500
    02  SIGN-TABLE-ENTRY OCCURS 64 TIMES.                      00026600
        03  OCTS-1-4      PICTURE XX.                          00026700
        03  OCTS-2-5      PICTURE XX.                          00026800
        03  OCTS-3-6      PICTURE XX.                          00026900
        03  PROOF-CHARACTERS PICTURE XXX.                      00027000
                                                                00027100
01  OUTPUT-WORK-AREA.                                          00027200
    02  BRAILLE-LINES.                                         00027300
        03  BRAILLE-LINE  OCCURS 3 TIMES.                      00027400
            04  BRAILLE-TEXT PICTURE X(120).                   00027500
    02  BRAILLE-LINES-INDEXED REDEFINES BRAILLE-LINES.         00027600
        03  BRAILLE-LINE-INDEXED OCCURS 3 TIMES.               00027700
            04  BRAILLE-SIGN OCCURS 40 TIMES, PICTURE XXX.     00027800
    02  BRAILLE-LINE-CHARS REDEFINES BRAILLE-LINES.            00027900
        03  LINE-CHARS    OCCURS 3 TIMES.                      00028000
            04  LINE-CHAR OCCURS 120 TIMES, PICTURE X.         00028100
    02  PROOF-LINE.                                            00028200
        03  PROOF-TEXT    PICTURE X(120).                      00028300
        03  OUTPUT-TEXT   REDEFINES PROOF-TEXT.                00028400
            04  OUTPUT-COLUMN OCCURS 120 TIMES, PICTURE X.     00028500
        03  OUTPUT-COLUMN-PAIRS REDEFINES PROOF-TEXT.          00028600
            04  COLUMN-PAIR OCCURS 60 TIMES, PICTURE 99.       00028700
    02  PROOF-LINE-INDEXED REDEFINES PROOF-LINE.               00028800
       03  PROOF OCCURS 40 TIMES, PICTURE XXX.                 00028900
    02  CODE-LINE.                                             00029000
        03  CODED-TEXT    PICTURE X(80).                       00029100
    02  CODE-LINE-INDEXED REDEFINES CODE-LINE.                 00029200
        03  CODED-SIGN OCCURS 40 TIMES, PICTURE 99.            00029300
    02 TITLE-LINES.                                           00029400
       03 TITLE-LINE OCCURS 3 TIMES.                          00029500
            04  TITLE-SIGN OCCURS 40 TIMES, PICTURE XXX.       00029600
    02 TITLE-PROOF.                                           00029700
        03  TITLE-PROOF-SIGN OCCURS 40 TIMES, PICTURE XXX.     00029800
    02  TITLE-SIGN-CODE.                                       00029900
        03  TITLE-SIGNS-CODE OCCURS 40 TIMES, PICTURE 99.      00030000
```

```
                                                                        00030100
01  LETTER-TO-DIGIT-CODE.                                              00030200
    02  DIGIT-SIGNS.                                                   00030300
        03  00 PICTURE S99, USAGE COMPUTATIONAL, VALUE 26.             00030400
        03  01 PICTURE S99, USAGE COMPUTATIONAL, VALUE 1.              00030500
        03  02 PICTURE S99, USAGE COMPUTATIONAL, VALUE 3.              00030600
        03  03 PICTURE S99, USAGE COMPUTATIONAL, VALUE 9.              00030700
        03  04 PICTURE S99, USAGE COMPUTATIONAL, VALUE 25.             00030800
        03  05 PICTURE S99, USAGE COMPUTATIONAL, VALUE 17.             00030900
        03  06 PICTURE S99, USAGE COMPUTATIONAL, VALUE 11.             00031000
        03  07 PICTURE S99, USAGE COMPUTATIONAL, VALUE 27.             00031100
        03  08 PICTURE S99, USAGE COMPUTATIONAL, VALUE 19.             00031200
        03  09 PICTURE S99, USAGE COMPUTATIONAL, VALUE 10.             00031300
    02  DIGIT-SIGNS-INDEXED REDEFINES DIGIT-SIGNS.                     00031400
        03  DIGIT OCCURS 10 TIMES, PICTURE S99,                        00031500
                    USAGE COMPUTATIONAL.                              00031600
                                                                        00031700
01  PAGE-NO-HEAD.                                                      00031800
    02  CRT-PAGE PICTURE 9999.                                         00031900
    02  CRT-DIGIT REDEFINES CRT-PAGE.                                  00032000
       03  CRT-PAGE-DIGIT PICTURE 9, OCCURS 4 TIMES.                   00032100
                                                                        00032200
01  CONTRACT-FORM-STRUC.                                               00032300
    02  C-F-COUNT PICTURE S99, USAGE COMPUTATIONAL.                    00032400
    02  CONTRACT-FORM-WHOLE.                                           00032500
        03  CONTRACT-FORM PICTURE X(10).                               00032600
    02  C-F-LETTERS REDEFINES CONTRACT-FORM-WHOLE.                     00032700
        03  CONTRACT-FORM-CHAR OCCURS 10 TIMES, PICTURE X.             00032800
                                                                        00032900
01  CONTRACTION-LIST.                                                  00033000
    02  LIST-ENTRY OCCURS 189 TIMES.                                   00033100
        03  INKPRINT PICTURE X(10).                                    00033200
        03 BRAILLE-EQUIV OCCURS 4 TIMES, PICTURE S99,                  00033300
            USAGE COMPUTATIONAL.                                        00033400
                                                                        00033500
                                                                        00033600
PROCEDURE DIVISION.                                                     00033700
                                                                        00033800
INITIALIZATION SECTION.                                                 00033900
                                                                        00034000
OPEN-FILES.                                                             00034100
    OPEN INPUT SYSINPUT.                                                00034200
    OPEN OUTPUT SYSPRINT.                                               00034300
    OPEN OUTPUT SYSBRL.                                                 00034400
    OPEN OUTPUT SYSRPQ.                                                 00034500
    MOVE SPACES TO OUT.                                                 00034600
    WRITE OUTPUT-LINE AFTER ADVANCING 0.                                00034700
    MOVE SPACES TO OUT-BRL.                                             00034800
    WRITE OUTPUT-BRL AFTER 0.                                           00034900
    MOVE SPACES TO OUT-RPQ.                                             00035000
    WRITE OUTPUT-RPQ AFTER 0.                                           00035100
    MOVE 'N' TO SEQ-ERR-IND.                                            00035200
    MOVE ZERO TO PREV-CARD-NO.                                          00035210
```

```
    MCVE 'E' TO ECHO.                                          00035220
    PERFORM READ-CARD.                                         00035230
    MCVE FIELD11 TO ECHO.                                      00035300
    MCVE 0 TC N.                                               00035400
LOOP1.                                                         00035500
    ADD 1 TO N.                                                00035600
    MCVE 0 TO NO-OF-ELEMENTS (N).                              00035700
    MOVE  0  TO SPECIAL-CONTROL (N).                           00035710
    CCMPUTE PREVIOUS-STACK (N) = N - 1.                        00035800
    CCMPUTE NEXT-STACK (N) = N + 1.                            00035900
    )F N NOT = 19 THEN GO TC LOOP1.                            00036000
    MOVE 0 TO NEXT-STACK (19).                                 00036100
    MOVE 0 TO NEXT-STACK (1).                                  00036200
    MOVE 0 TO PREVIOUS-STACK (2).                              00036300
    MCVE 3 TO HEAD-FREE-STACK.                                 00036400
    MOVE 1 TO CURRENT-TYPE.                                    00036500
    MCVE 1 TO HEAD-S (1).                                      00036600
    MCVE 1 TO TAIL-S (1).                                      00036700
    MCVE 1 TO CURRENT-S (1).                                   00036800
    MCVE 2 TO TAIL-S (2).                                      00036900
    MCVE 2 TO CURRENT-S (2).                                   00037000
    MOVE 2 TO HEAD-S (2).                                      00037100
    MCVE 0 TO PREVIOUS-STACK (3).                              00037200
    MOVE 0 TO NEXT-STACK (2).                                  00037300
    MOVE C TO N.                                               00037400
LCOP1A.                                                        00037410
    ADD 1 TC N.                                                00037420
    MOVE 'L' TO TAB-TYPE (N).                                  00037430
    CCMPUTE TAB-COLUMN (N) = 5 * N.                            00037440
    )F N NOT = 9 THEN GO TO LOOP1A.                            00037450
    MOVE 0 TO LINECOUNT.                                       00037460
    MCVE 0 TO NALPHABET.                                       00037470
    MOVE 1 TO OUTPTR.                                          00037480
    MCVE 0 TO TABULATE.                                        00037490
    MOVE 0 TO SPACE-CCUNT.                                     00037500
    MOVE 0 TO XX.                                              00037510
    MOVE C TO HEAD-INO.                                        00037520
    MOVE 'X' TO TEMPL.                                         00037530
    MOVE 2 TO STACK-1NDICATOR.                                 00037540
    MOVE 0 TO SC-ERROR-CT.                                     00037550
    MCVE C TO SELF-TEST.                                       00037560
    MCVE 0 TO CARCS-PRINTED.                                   00037570
    MOVE 1 TO CURRENT-IN-TRANS.                                00037600
    MCVE 0 TO PRIOR-SPACE.                                     00037700
    MOVE 0 TO IN-CCNTRACT.                                     00037800
    ALTER S-T-B1 TO PRCCEED TO S-T-NC1.                        00038100
    MCVE 0 TO N.                                               00038300
LOOP2.                                                         00038400
    ADD 1 TC N.                                                00038500
    MCVE 0 TO NO-OF-ELEMENTS-TRANS (N).                        00038600
    IF N < 4 GO TO LOOP2.                                      00038700
    MOVE 0 TO CURRENT-TRANS.                                   00038800
    MOVE 0 TO N.                                               00038900
```

```
LOOP2A.                                                          00039000
    PERFORM READ-CARD.                                           00039100
    ADD 1 TO N.                                                  00039200
    MOVE FIELD1W TO INKPRINT (N).                                00039300
    MOVE FIELD51 TO BRAILLE-EQUIV (N, 1).                        00039400
    MOVE FIELD52 TO BRAILLE-EQUIV (N, 2).                        00039500
    MOVE FIELD53 TO BRAILLE-EQUIV (N, 3).                        00039600
    MOVE FIELD54 TO BRAILLE-EQUIV (N, 4).                        00039700
    IF N < 189 THEN GO TO LOOP2A.                                00039800
                                                                 00040500
    MOVE 999 TO EXTENT (1).                                      00040600
    MOVE 1 TO N.                                                 00040700
READ-ALPHABET.                                                   00040800
    MOVE EXTENT (1) TO EXTENT (N).                               00040900
    PERFORM READ-CARD.                                           00041000
    ADD 1 TO NALPHABET.                                          00041100
    IF FIELD3 = '99' THEN GO TO LOOP63.                          00041200
    MOVE FIELD11 TO SYMBOL (N).                                  00041300
    MOVE FIELD3 TO CHAR-CLASS (N).                               00041400
    MOVE FIELD51 TO SINGLE-SIGN (N).                             00041500
    MOVE FIELD54 TO ISC-LETTER (N).                              00041600
    MOVE FIELD4 TO COMP-8 (N).                                   00041900
    ADD 1 TO N.                                                  00042100
    GO TO READ-ALPHABET.                                         00042200
                                                                 00042300
LOOP63.                                                          00042400
    MOVE 1 TO I.                                                 00042500
    MOVE 1 TO N.                                                 00042600
FILL-TABLE.                                                      00042700
    PERFORM READ-CARD.                                           00042800
    IF FIELD11 = TEMP1 THEN GO TO MOVE-FIELDS.                   00042900
    MOVE FIELD1 TO TEMP1.                                        00043000
    MOVE N TO EXTENT (I).                                        00043100
    IF SYMBOL (I) = TEMP1 THEN GO TO ADD1I.                      00043110
    IF FIELD3 = '99' THEN GO TO ADD1I.                           00043115
    MOVE SPACES TO OUT.                                          00043120
    MOVE 'E' TO ECHO.                                            00043130
    MOVE '** TABLE SEQUENCE ERROR.' TO OUT.                      00043140
    WRITE OUTPUT-LINE AFTER ADVANCING 1.                         00043150
    MOVE 'Y' TO SEQ-ERR-IND.                                     00043160
ADD1I.                                                           00043170
    ADD 1 TO I.                                                  00043200
MOVE-FIELDS.                                                     00043300
    MOVE FIELD12 TO STRING (N).                                  00043400
    MOVE FIELD2 TO RIGHT-CONTEXT (N).                            00043500
    MOVE FIELD3 TO INPUT-CLASS (N).                              00043600
    MOVE FIELD4 TO SHIFT (N).                                    00043700
    MOVE FIELD51 TO SIGN (N, 1).                                 00043800
    MOVE FIELD52 TO SIGN (N, 2).                                 00043900
    MOVE FIELD53 TO SIGN (N, 3).                                 00044000
    MOVE FIELD54 TO SIGN (N, 4).                                 00044100
    ADD 1 TO N.                                                  00044200
    IF FIELD3 NOT = '99' THEN GO TO FILL-TABLE.                  00044300
```

```
    COMPUTE NCT = N - 2.                                          00044400
    COMPUTE J = I - 1.                                            00044500
    COMPUTE EXTENT (I) = EXTENT (J) + 1.                          00044600
    MOVE J TO MAXEXTENT.                                          00044700
    MOVE 1 TO I.                                                  00044800
LABEL1A.                                                          00044900
    MOVE ZEROS TO BRANCH (I).                                     00045000
    ADD 1 TO I.                                                   00045100
    IF I < NCT + 2 THEN GO TO LABEL1A.                            00045200
    MOVE 1 TO I.                                                  00045300
LABEL1O.                                                          00045400
    MOVE I TO N.                                                  00045500
    MOVE EXTENT (I) TO LOW-NO.                                    00045600
    COMPUTE J = I + 1.                                            00045700
    COMPUTE HIGH-NO = EXTENT (J) - 1.                             00045800
    MOVE HIGH-NO TO VHIGH.                                        00045900
LABEL1.                                                           00046000
    COMPUTE II = LOW-NO + 1.                                      00046100
    IF II > HIGH-NO THEN GO TO LABEL5.                            00046200
    IF TABLE-CHAR (LOW-NO, N) = RCC THEN GO TO LABEL2.            00046300
LABEL11.                                                          00046400
    IF II > HIGH-NO THEN GO TO LABEL5.                            00046500
    IF TABLE-CHAR (LOW-NO, N) NOT = TABLE-CHAR (II, N)            00046600
    THEN GO TO LABEL2.                                            00046700
    ADD 1 TO II.                                                  00046800
    GO TO LABEL11.                                                00046900
LABEL2.                                                           00047000
    MOVE II TO L-ARRAY (N).                                       00047100
    MOVE II TO BRANCH (LOW-NO).                                   00047200
    MOVE N TO BRANCH (II).                                        00047300
    ADD 1 TO N.                                                   00047400
    ADD 1 TO LOW-NO.                                              00047500
    COMPUTE HIGH-NO = II - 1.                                     00047600
LABEL3.                                                           00047700
    IF LOW-NO NOT > HIGH-NO THEN GO TO LABEL1.                    00047800
    SUBTRACT 1 FROM N.                                            00047900
LABEL4.                                                           00048000
    IF N = 0 THEN GO TO LABEL6.                                   00048100
    COMPUTE K = N - 1.                                            00048200
    IF N > 1 THEN COMPUTE HIGH-NO = L-ARRAY (K) - 1               00048300
    ELSE MOVE VHIGH TO HIGH-NO.                                   00048400
    GO TO LABEL3.                                                 00048500
LABEL5.                                                           00048600
    IF II NOT = LOW-NO + 1 THEN GO TO LABEL9.                     00048700
    COMPUTE BRANCH (LOW-NO) = - BRANCH (II).                      00048800
    ADD 1 TO LOW-NO.                                              00048900
    MOVE N TO BRANCH (II).                                        00049000
    GO TO LABEL4.                                                 00049100
LABEL9.                                                           00049200
    MOVE II TO L-ARRAY (N).                                       00049300
    ADD 1 TO N.                                                   00049400
    COMPUTE BRANCH (LOW-NO) = - N.                                00049500
    ADD 1 TO LOW-NO.                                              00049600
```

```
    GO TO LABEL1.                                                    00049700
LABEL6.                                                              00049800
    ADD 1 TO I.                                                      00049900
    IF I > MAXEXTENT THEN GO TO READ-RIGHT-CONTEXT-TABLE.            00050000
    GO TO LABEL10.                                                   00050100
READ-RIGHT-CONTEXT-TABLE.                                            00050200
    PERFORM READ-CARD.                                               00050300
    MOVE FIELD3 TO NNT.                                              00050400
    MOVE 1 TO N.                                                     00050500
READ-NON-TERMINALS.                                                  00050600
    PERFORM READ-CARD.                                               00050700
    MOVE FIELD2 TO NON-TERMINAL (N).                                 00050800
    MOVE FIELD51 TO RIGHT-CONTEXT-CLASS (N, 1).                      00050900
    MOVE FIELD52 TO RIGHT-CONTEXT-CLASS (N, 2).                      00051000
    MOVE FIELD53 TO RIGHT-CONTEXT-CLASS (N, 3).                      00051100
    MOVE FIELD54 TO RIGHT-CONTEXT-CLASS (N, 4).                      00051200
    ADD 1 TO N.                                                      00051300
    IF N NOT > NNT THEN GO TO READ-NON-TERMINALS.                    00051400
                                                                     00051500
READ-STATE-TABLES.                                                   00051600
    PERFORM READ-CARD.                                               00051700
    MOVE FIELD3 TO NSV.                                              00051800
    PERFORM READ-CARD.                                               00051900
    MOVE FIELD3 TO NIC.                                              00052000
    MOVE 1 TO N.                                                     00052100
READ-TRANSITION-TABLE.                                               00052200
    PERFORM READ-CARD.                                               00052300
    MOVE TABLE-INFO TO TRANSITION-TABLE-COLUMN (N).                  00052400
    ADD 1 TO N.                                                      00052500
    IF N NOT > NSV THEN GO TO READ-TRANSITION-TABLE.                 00052600
    PERFORM READ-CARD.                                               00052700
    MOVE FIELD3 TO NOTC.                                             00052800
    MOVE 1 TO N.                                                     00052900
READ-DECISION-TABLE.                                                 00053000
    PERFORM READ-CARD.                                               00053100
    MOVE TABLE-INFO TO DECISION-TABLE-COLUMN (N).                    00053200
    MOVE NIC TO I.                                                   00053300
    MOVE 0 TO J.                                                     00053400
READ-CONDITIONS.                                                     00053500
    ADD 1 TO I.                                                      00053600
    ADD 1 TO J.                                                      00053700
    MOVE CHAR (I) TO CONDITION (N, J).                               00053800
    IF I < NIC + NSV THEN GO TO READ-CONDITIONS.                     00053900
    ADD 1 TO N.                                                      00054000
    IF N NOT > NOTC THEN GO TO READ-DECISION-TABLE.                  00054100
    MOVE INITIAL-STATE TO STATE-VECTOR.                              00054200
                                                                     00054300
READ-SIGN-TABLE.                                                     00054400
    MOVE 1 TO N.                                                     00054500
READ-SIGN-TABLE-ENTRY.                                               00054600
    PERFORM READ-CARD.                                               00054700
    MOVE TABLE-INFO TO SIGN-TABLE-ENTRY (N).                         00054800
    ADD 1 TO N.                                                      00054900
```

```
    IF N NOT > 64 THEN GO TO READ-SIGN-TABLE-ENTRY.                  00055000
                                                                     00055100
READ-CONTROL-CARDS.                                                  00055200
    PERFORM READ-CARD.                                               00055300
    MOVE FIELD11 TO PROOF-OUT.                                       00055400
    PERFORM READ-CARD.                                               00055500
    MOVE FIELD11 TO BRAILLE.                                         00055600
    PERFORM READ-CARD.                                               00055700
    MOVE FIELD11 TO RPG.                                             00055800
    PERFORM READ-CARD.                                               00055900
    MOVE FIELD11 TO PUNCHED-OUTPUT.                                  00056000
    PERFORM READ-CARD.                                               00056100
    MOVE FIELD3 TO OUTL.                                             00056200
    PERFORM READ-CARD.                                               00056300
    MOVE FIELD3 TO LPG.                                              00056400
    IF PUNCHED-OUTPUT NOT = 'N' THEN OPEN OUTPUT PUNCH.              00056500
    PERFORM READ-CARD.                                               00056600
    MOVE FIELD6 TO CRT-PAGE.                                         00056700
    MOVE FIELD11 TO PAGINATION.                                      00056800
                                                                     00056900
                                                                     00057000
TABLE-DISPLAY.                                                       00057100
                                                                     00057200
    IF PROOF-OUT NOT = 'P' THEN GO TO SKIP-DISPLAY.                  00057300
    MOVE SPACES TO OUT.                                              00057400
    WRITE OUTPUT-LINE AFTER 0.                                       00057500
    MOVE '                RIGHT CONTEXT TABLE' TO OUT.               00057600
    WRITE OUTPUT-LINE AFTER 2.                                       00057700
    MOVE 'NON-TERMINAL   INPUT CLASSES' TO OUT.                      00057800
    WRITE OUTPUT-LINE AFTER 2.                                       00057900
    MOVE 1 TO K.                                                     00058000
DISP14.  MOVE SPACES TO OUTPUT-TEXT.                                 00058100
    MOVE NON-TERMINAL (K) TO OUTPUT-COLUMN (6).                      00058200
    MOVE RIGHT-CONTEXT-CLASS (K, 1) TO COLUMN-PAIR (8).              00058300
    MOVE RIGHT-CONTEXT-CLASS (K, 2) TO COLUMN-PAIR (10).             00058400
    MOVE RIGHT-CONTEXT-CLASS (K, 3) TO COLUMN-PAIR (12).             00058500
    MOVE RIGHT-CONTEXT-CLASS (K, 4) TO COLUMN-PAIR (14).             00058600
    MOVE OUTPUT-TEXT TO OUT.                                         00058700
    WRITE OUTPUT-LINE AFTER 2.                                       00058800
    ADD 1 TO K.                                                      00058900
    IF K NOT > NNT THEN GO TO DISP14.                                00059000
DISP13.  MOVE SPACES TO OUT.                                         00059100
    WRITE OUTPUT-LINE AFTER 0.                                       00059200
    MOVE '                DECISION TABLE' TO OUT.                    00059300
    WRITE OUTPUT-LINE AFTER 3.                                       00059400
    MOVE 'COLUMN' TO OUTPUT-TEXT.                                    00059500
    PERFORM DISP1 VARYING K FROM 1 BY 1 UNTIL K = NOTC.              00059600
DISP1.  COMPUTE I = 9 + 2 * K.                                       00059700
    MOVE K TO COLUMN-PAIR (I).                                       00059800
END-DISP1.                                                           00059900
    MOVE OUTPUT-TEXT TO OUT.                                         00060000
    WRITE OUTPUT-LINE AFTER 2.                                       00060100
    MOVE SPACES TO OUT.                                              00060110
```

```
    WRITE OUTPUT-LINE AFTER 1.                                          00060120
DISP5.                                                                  00060200
    MOVE I TO N.                                                        00060300
    MOVE 'INPUT CLASS' TO OUTPUT-TEXT.                                  00060400
DISP6.                                                                  00060410
    MOVE N TO COLUMN-PAIR (9).                                          00060500
    MOVE I TO K.                                                        00060600
DISP8.  COMPUTE I = 18 + 4 * K.                                         00060700
    MOVE DECISION (K, N) TO OUTPUT-COLUMN (I).                          00060800
    ADD 1 TO K.                                                         00061000
    IF K NOT > NOTC THEN GO TO DISP8.                                   00061100
    MOVE OUTPUT-TEXT TO OUT.                                            00061110
    MOVE SPACES TO OUTPUT-TEXT.                                         00061120
DISP7.  WRITE OUTPUT-LINE AFTER 1.                                      00061200
    ADD 1 TO N.                                                         00061300
    IF N NOT > NIC THEN GO TO DISP6.                                    00061400
    MOVE SPACES TO OUT.                                                 00061500
    WRITE OUTPUT-LINE AFTER 1.                                          00061600
    MOVE I TO N.                                                        00061700
    MOVE 'STATE VARIABLE' TO OUTPUT-TEXT.                               00061800
DISP2.                                                                  00061810
    MOVE N TO COLUMN-PAIR (9).                                          00061900
    MOVE 1 TO K.                                                        00062000
DISP4.  COMPUTE I = 18 + 4 * K.                                         00062100
    MOVE CONDITION (K, N) TO OUTPUT-COLUMN (I).                         00062200
    ADD 1 TO K.                                                         00062400
    IF K NOT > NOTC THEN GO TO DISP4.                                   00062500
    MOVE OUTPUT-TEXT TO OUT.                                            00062510
    MOVE SPACES TO OUTPUT-TEXT.                                         00062520
DISP3.  WRITE OUTPUT-LINE AFTER 1.                                      00062600
    ADD 1 TO N.                                                         00062700
    IF N NOT > NSV THEN GO TO DISP2.                                    00062800
DISP9.  MOVE SPACES TO OUT.                                             00062900
    WRITE OUTPUT-LINE AFTER 0.                                          00063000
    MOVE '                TRANSITION TABLE' TO OUT.                     00063100
    WRITE OUTPUT-LINE AFTER 3.                                          00063200
    MOVE 'STATE VARIABLE' TO OUTPUT-TEXT.                               00063300
    PERFORM DISP1 VARYING K FROM 1 BY 1 UNTIL K > NSV.                  00063400
    MOVE OUTPUT-TEXT TO OUT.                                            00063500
    WRITE OUTPUT-LINE AFTER 2.                                          00063600
    MOVE SPACES TO OUT.                                                 00063610
    WRITE OUTPUT-LINE AFTER 1.                                          00063620
    MOVE 'INPUT CLASS' TO OUTPUT-TEXT.                                  00063630
    PERFORM DISP10 THRU DISP11 VARYING N FROM 1 BY 1                    00063700
    UNTIL N = NIC.                                                      00063800
DISP1C.                                                                 00063900
    MOVE N TO COLUMN-PAIR (9).                                          00064000
    MOVE 1 TO K.                                                        00064100
DISP12.  COMPUTE I = 18 + 4 * K.                                        00064200
    MOVE TRANSITION (K, N) TO OUTPUT-COLUMN (I).                        00064300
    ADD 1 TO K.                                                         00064500
    IF K NOT > NSV THEN GO TO DISP12.                                   00064600
    MOVE OUTPUT-TEXT TO OUT.                                            00064610
```

```
    MOVE SPACES TO OUTPUT-TEXT.                                          00064620
DISP11.  WRITE OUTPUT-LINE AFTER 1.                                      00064700
END-DISP11.  MOVE SPACES TO OUT.                                         00064800
    WRITE OUTPUT-LINE AFTER 0.                                           00064900
SKIP-DISPLAY.                                                            00065000
                                                                         00065100
    IF SEC-ERR-IND = 'Y' THEN GO TO CLOSE-FILES.                         00065110
READ-FIRST-RECORD.                                                       00065200
    PERFORM READ-NEW-RECORD.                                             00065300
    PERFORM SHIFT-LEFT-ONE 10 TIMES.                                     00065400
    MOVE SPACES TO BRAILLE-LINES.                                        00065500
    MOVE SPACES TO PROOF-LINE.                                           00065600
    MOVE ZEROS TO CODE-LINE.                                             00065700
    GO TO TRANSLATION.                                                   00065800
NOTHING.  EXIT.                                                          00065900
END-INITIALIZATION.                                                      00066000
                                                                         00066100
                                                                         00066200
TRANSLATION SECTION.                                                     00066300
                                                                         00066400
                                                                         00066500
INSPECT-BUFFER.                                                          00066600
    MOVE 1 TO LETTER.                                                    00066700
IDENTIFY-FIRST-CHARACTER.                                                00066800
    IF SYMBOL (LETTER) = RL1 THEN GO TO LOOKUP.                          00066900
    IF LETTER = ALPHABET THEN GO TO TEST-RCC.                            00067000
    ADD 1 TO LETTER.                                                     00067100
    GO TO IDENTIFY-FIRST-CHARACTER.                                      00067200
                                                                         00067210
TEST-RCC.                                                                00067220
    IF RL1 NOT = RCC AND RL1 NOT = '*' THEN GO TO ERROR-1.               00067230
    MOVE 58 TO OUTSIGN.                                                  00067240
    PERFORM OUTPUT-SIGN.                                                 00067250
    NOTE RUN WILL STOP WITH ABOVE PERFORM.                               00067260
                                                                         00067300
LOOKUP.                                                                  00067400
    IF ISO-LETTER (LETTER) NOT = 1 THEN GO TO LOOP80.                    00067500
    IF (RL1 = '_') AND ((RCHAR (1) = 'A') OR (RCHAR (1) = 'I')           00067600
       OR (RCHAR (1) = 'O') OR (RCHAR (1) = '_'))                        00067700
       THEN GO TO LOOP80.                                                00067800
    MOVE 1 TO N.                                                         00067900
    MOVE 1 TO K.                                                         00068000
LOOP81.                                                                  00068100
    IF RCHAR (N) = '¬' THEN GO TO LOOP88.                                00068200
    IF RCHAR (N) = '=' THEN GO TO LOOP80.                                00068300
    IF RCHAR (N) = ' ' THEN GO TO LOOP88.                                00068400
    IF RCHAR (N) = '_' THEN GO TO LOOP82.                                00068500
    MOVE 1 TO X.                                                         00068600
IDEN-CHAR.                                                               00068700
    IF SYMBOL (X) = RCHAR (N) THEN GO TO LOOP83.                         00068800
    ADD 1 TO X.                                                          00068900
    GO TO IDEN-CHAR.                                                     00069000
LOOP83.                                                                  00069100
```

```
    IF CHAR-CLASS (X) NOT = 1 THEN GO TO LOOP80.                00069200
    COMPUTE M = N + 1.                                          00069300
    IF RL1 = '_' THEN GO TO LOOP90.                             00069400
    IF SYMBOL (LETTER) = '(' THEN GO TO LOOP84.                 00069500
    IF SYMBOL (LETTER) = RCHAR (M) THEN GO TO LOOP85.           00069600
    GO TO LOOP80.                                               00069700
LOOP84.                                                         00069800
    IF RCHAR (M) NOT = ')' THEN GO TO LOOP80.                   00069900
LOOP85.                                                         00070000
    PERFORM SHIFT-CHAR K TIMES.                                 00070100
    PERFORM SHIFT-LEFT-ONE M TIMES.                             00070200
    MOVE '=' TO RL1.                                            00070300
    GO TO INSPECT-BUFFER.                                       00070400
LOOP88.                                                         00070500
    ADD 1 TO K.                                                 00070600
LOOP82.                                                         00070700
    ADD 1 TO N.                                                 00070800
    GO TO LOOP81.                                               00070900
LOOP90.                                                         00071000
    MOVE 1 TO X.                                                00071100
LOOP91.                                                         00071200
    IF SYMBOL (X) = RCHAR (M) THEN GO TO LOOP92.                00071300
    ADD 1 TO X.                                                 00071400
    GO TO LOOP91.                                               00071500
LOOP92.                                                         00071600
    IF CHAR-CLASS (X) = 1 THEN GO TO LOOP80.                    00071700
    IF N > K THEN PERFORM SHIFT-LEFT-ONE.                       00071800
    MOVE '=' TO RL1.                                            00071900
    GO TO INSPECT-BUFFER.                                       00072000
SHIFT-CHAR.                                                     00072100
    MOVE RCHAR (N) TO RCHAR (M).                                00072200
    SUBTRACT 1 FROM N.                                          00072300
    SUBTRACT 1 FROM M.                                          00072400
END-SHIFT-CHAR.                                                 00072500
LOOP80.                                                         00072600
    COMPUTE NN = LETTER + 1.                                    00072700
    MOVE EXTENT (LETTER) TO INDEX.                              00072800
    IF INDEX > NOT THEN GO TO MOVE-SINGLE-SIGN.                 00072900
    MOVE 1 TO NXT-LEVEL.                                        00073000
MATCH.                                                          00073100
    MOVE NXT-LEVEL TO POINTER.                                  00073200
COMPARE.                                                        00073300
    IF TABLE-CHAR (INDEX, POINTER) = RCC                        00073400
    THEN GO TO TEST-NON-TERMINAL.                               00073500
    IF RCHAR (POINTER) NOT = TABLE-CHAR (INDEX, POINTER)        00073600
    THEN GO TO COMPARE-FAIL.                                    00073700
    ADD 1 TO POINTER.                                           00073800
    IF POINTER > 9 THEN GO TO TEST-NON-TERMINAL.                00073900
    GO TO COMPARE.                                              00074000
MISMATCH.                                                       00074100
COMPARE-FAIL.                                                   00074200
    IF BRANCH (INDEX) > 0 THEN GO TO SKIP.                      00074300
    IF POINTER NOT = NXT-LEVEL THEN GO TO NO-MATCH-LEVEL.       00074400
```

```
LAMP1.                                                                  00074500
    IF - IBRANCH IINOEX)) < NXT-LEVEL THEN GO TO NO-MATCH-LEVEL.        00074600
    ADD 1 TO INOEX.                                                     00074700
LAMP2.                                                                  00074800
    IF BRANCH (INOEX) NOT > 0 THEN GO TO LAMP1.                         00074900
    MOVE BRANCH (INOEX) TO INOEX.                                       00075000
    GO TO LAMP2.                                                        00075100
NO-MATCH-LEVEL.                                                         00075200
    COMPUTE NXT-LEVEL = - BRANCH (INOEX).                               00075300
    IF NXT-LEVEL NOT > 1 THEN GO TO MOVE-SINGLE-SIGN.                   00075400
    ADD 1 TO INOEX.                                                     00075500
    GO TO MATCH.                                                        00075600
SKIP.                                                                   00075700
    IF POINTER = NXT-LEVEL THEN GO TO JUMP.                             00075800
    IF BRANCH IINOEX) NOT = INOEX + 1 THEN ADD 1 TO NXT-LEVEL.          00075900
    ADD 1 TO INOEX.                                                     00076000
    GO TO MATCH.                                                        00076100
JUMP.                                                                   00076200
    MOVE BRANCH IINOEX) TO INOEX.                                       00076300
    GO TO MATCH.                                                        00076400
                                                                        00076500
TEST-NON-TERMINAL.                                                      00076600
    IF RIGHT-CONTEXT IINOEX) = SPACE THEN GO TO OUTPUT-LOGIC.           00076700
    MOVE 1 TO N.                                                        00076800
TNT1.                                                                   00076900
    IF SYMBOL IN) = RCHAR (PCINTER) THEN GO TO TNT2.                    00077000
    IF N = NALPHABET THEN GO TO MISMATCH.                               00077100
    ADD 1 TO N.                                                         00077200
    GO TO TNT1.                                                         00077300
TNT2.                                                                   00077400
    MOVE 1 TO K.                                                        00077500
    IF RIGHT-CONTEXT IINOEX) NOT = NON-TERMINAL (K)                     00077600
    THEN GO TO TNT3.                                                    00077700
TNT5.                                                                   00077800
    MOVE I TO J.                                                        00077900
TNT4.                                                                   00078000
    IF CHAR-CLASS (N) = RIGHT-CONTEXT-CLASS (K, J)                      00078100
    THEN GO TO OUTPUT-LOGIC.                                            00078200
    IF RIGHT-CONTEXT-CLASS (K, J) = 99 THEN GO TO MISMATCH.             00078300
    ADD 1 TO J.                                                         00078400
    IF J NOT > 4 THEN GO TO TNT4 ELSE GO TO MISMATCH.                   00078500
TNT3.                                                                   00078600
    ADD 1 TO K.                                                         00078700
    IF K NOT > NNT THEN GO TO TNT5 ELSE GO TO MISMATCH.                 00078800
                                                                        00078900
OUTPUT-LOGIC.                                                           00079000
                                                                        00079100
    MOVE INPUT-CLASS IINOEX) TO I.                                      00079200
    MOVE I TO TRANS-CLASS.                                              00079300
    MOVE 1 TO K.                                                        00079400
LOGIC1.                                                                 00079500
    IF DECISION (K, I) = '-' THEN GO TO LOGIC4.                         00079600
    IF DECISION (K, I) = 'G' THEN GO TO TRANSOUCE.                      00079700
```

```
    IF DECISION (K, I) = 'F' THEN GO TO MISMATCH.                          00079750
    MOVE 1 TO N.                                                           00079800
LOGIC2.                                                                    00079900
    IF CONDITION (K, N) = '-' THEN GO TO LOGIC3.                           00080000
    IF STATE-VARIABLE (N) NOT = CONDITION (K, N)                           00080100
    THEN GO TO LOGIC9.                                                     00080200
LOGIC3.                                                                    00080300
    IF N = NSV THEN GO TO LOGIC5.                                          00080400
    ADD 1 TO N.                                                            00080500
    GO TO LOGIC2.                                                          00080600
LOGIC4.                                                                    00080700
    IF K = NOTC THEN GO TO MISMATCH.                                       00080800
    ADD 1 TO K.                                                            00080900
    GO TO LOGIC1.                                                          00081000
LOGIC5.                                                                    00081100
    IF DECISION (K, I) = 'Y' THEN GO TO TRANSOUCE,                         00081200
    ELSE GO TO MISMATCH.                                                   00081300
LOGIC9.                                                                    00081400
    IF DECISION (K, I) = 'Y' THEN GO TO MISMATCH,                          00081500
    ELSE GO TO TRANSOUCE.                                                  00081600
                                                                           00081700
TRANSOUCE.                                                                 00081800
    MOVE SHIFT (INDEX) TO J.                                               00081900
    PERFORM SHIFT-LEFT-ONE J TIMES.                                        00082100
    MOVE 1 TO J.                                                           00082400
PUT-OUT-SIGNS.                                                             00082500
    IF SIGN (INDEX, J) = 99 THEN GO TO END-OUTPUT-SIGNS.                   00082600
    MOVE SIGN (INDEX, J) TO OUTSIGN.                                       00082700
    PERFORM OUTPUT-SIGN.                                                   00082800
    ADD 1 TO J.                                                            00082900
    IF J NOT > 4 THEN GO TO PUT-OUT-SIGNS.                                 00083000
END-OUTPUT-SIGNS.                                                          00083100
                                                                           00083200
                                                                           00083300
TRANSITION-LOGIC.                                                          00083400
                                                                           00083500
    MOVE 1 TO N.                                                           00083600
    MOVE TRANS-CLASS TO I.                                                 00083700
LOGIC6.                                                                    00083800
    IF TRANSITION (N, I) = '-' THEN NEXT SENTENCE,                         00083900
    ELSE IF TRANSITION (N, I) = 'S'                                        00084000
    OR (TRANSITION (N, I) = 'T' AND STATE-VARIABLE (N) = 'N')              00084100
    THEN MOVE 'Y' TO STATE-VARIABLE (N),                                   00084200
    ELSE MOVE 'N' TO STATE-VARIABLE (N).                                   00084300
    ADD 1 TO N.                                                            00084400
    IF N NOT > NSV THEN GO TO LOGIC6.                                      00084500
    GO TO TRANSLATION.                                                     00084600
                                                                           00084700
MOVE-SINGLE-SIGN.                                                          00084800
    MOVE SINGLE-SIGN (LETTER) TO OUTSIGN.                                  00084900
    PERFORM OUTPUT-SIGN.                                                   00085000
    MOVE CHAR-CLASS (LETTER) TO TRANS-CLASS.                               00085100
    PERFORM SHIFT-LEFT-ONE.                                                00085200
```

```
    GO TO END-OUTPUT-SIGNS.                                          00085300
                                                                     00085400
                                                                     00085500
                                                                     00085600
ERROR-1.                                                             00085700
    MOVE 'NEW CHAR' TO CL1.                                          00085800
    MOVE RL1 TO OUT-PLACE (12).                                      00085900
    WRITE OUTPUT-LINE AFTER 2.                                       00086000
    MOVE SPACES TO OUT.                                              00086010
    MOVE '*' TO RL1.                                                 00086020
    GO TO TRANSLATION.                                               00086100
                                                                     00086200
                                                                     00086300
INPUT-CHAR SECTION.                                                  00086400
                                                                     00086500
MOVE-NEXT-CHAR.                                                      00086600
    IF INPTR > 72 THEN PERFORM READ-NEW-RECORD.                      00086700
    MOVE CHAR (INPTR) TO NXTCHR.                                     00086800
    ADD 1 TO INPTR.                                                  00086900
    IF NXTCHR NOT = ' ' THEN GO TO RESET-PRIOR-SPACE.                00087000
    IF PRIOR-SPACE = 0 THEN GO TO MOVE-NEXT-CHAR.                    00087100
    SUBTRACT 1 FROM PRIOR-SPACE.                                     00087200
    GO TO S-T-81.                                                    00087300
RESET-PRIOR-SPACE.                                                   00087400
    IF NXTCHR = '.' THEN MOVE 2 TO PRIOR-SPACE, ELSE                 00087410
    MOVE 1 TO PRIOR-SPACE.                                           00087500
S-T-E1.  GO TO.                                                      00087600
    NOTE SELF-TEST OPTION SWITCH.                                    00087700
S-T-NO1.                                                             00087800
    IF NXTCHR = RCC THEN GO TO MOVE-NEXT-CHAR,                       00087900
    ELSE GO TO END-INPUT.                                            00088000
S-T-YES1.                                                            00088100
    IF IN-CONTRACT = 1 THEN GO TO YES-IN-CONTRACT.                   00088200
NOT-IN-CONTRACT.                                                     00088300
    IF NXTCHR = RCC THEN GO TO SET-IN-CONTRACT.                      00088400
    IF NXTCHR = ' ' THEN GO TO INC-C-I-T.                            00088500
    MOVE NO-OF-ELEMENTS-TRANS (CURRENT-IN-TRANS) TO N-O-E-T.         00088600
    IF N-O-E-T < 30 THEN ADD 1 TO N-O-E-T.                           00088700
    MOVE N-O-E-T TO NO-OF-ELEMENTS-TRANS (CURRENT-IN-TRANS).         00088800
    MOVE 1 TO L-NXTCHR.                                              00088900
LKUP-NXTCHR.                                                         00089000
    IF SYMBOL (L-NXTCHR) = NXTCHR THEN GO TO STORE1-E-T.             00089100
    IF L-NXTCHR = NALPHABET THEN GO TO STORE1-E-T-99.                00089200
    ADD 1 TO L-NXTCHR.                                               00089300
    GO TO LKUP-NXTCHR.                                               00089400
STORE1-E-T-99.                                                       00089500
    MOVE 99 TO ELEMENT-TRANS (CURRENT-IN-TRANS, N-O-E-T).            00089600
    GO TO END-INPUT.                                                 00089700
STORE1-E-T.                                                          00089800
    MOVE SINGLE-SIGN (L-NXTCHR) TO ELEMENT-TRANS                     00089900
    (CURRENT-IN-TRANS, N-O-E-T).                                     00090000
    GO TO END-INPUT.                                                 00090100
SET-IN-CONTRACT.                                                     00090200
```

```
    MOVE 0 TO C-F-COUNT.                                                    00090300
    MOVE SPACES TO CONTRACT-FORM.                                           00090400
    MOVE 1 TO IN-CONTRACT.                                                  00090500
    GO TO MOVE-NEXT-CHAR.                                                   00090600
YES-IN-CONTRACT.                                                            00090700
    IF NXTCHR = RCC THEN GO TO LKUP-CC.                                     00090800
    IF NXTCHR = ' ' THEN GO TO LKUP-CC.                                     00090900
    IF C-F-COUNT < 10 THEN ADD 1 TO C-F-COUNT.                              00091000
    MOVE NXTCHR TO CONTRACT-FORM-CHAR (C-F-COUNT).                          00091100
    GO TO END-INPUT.                                                        00091200
LKUP-CC.                                                                    00091300
    IF C-F-COUNT = 0 THEN GO TO RESET-IN-CONTRACT.                          00091400
    MOVE NO-OF-ELEMENTS-TRANS (CURRENT-IN-TRANS) TO N-O-E-T.                00091500
    MOVE 1 TO C-L-LOW.                                                      00091600
    MOVE 185 TO C-L-HIGH.                                                   00091700
C-L-BIN-SEARCH-LOOP.                                                        00091800
    COMPUTE C-L-CURR = (C-L-LOW + C-L-HIGH) / 2.                            00091900
    IF INKPRINT (C-L-CURR) > CONTRACT-FORM THEN GO TO BOTT-HALF.            00092000
    IF INKPRINT (C-L-CURR) < CONTRACT-FORM THEN GO TO TOP-HALF.             00092100
    GO TO C-L-FOUND.                                                        00092200
BOTT-HALF.                                                                  00092300
    IF C-L-LOW NOT < C-L-CURR THEN GO TO C-L-NOT-FOUND.                     00092400
    COMPUTE C-L-HIGH = C-L-CURR - 1.                                        00092500
    GO TO C-L-BIN-SEARCH-LOOP.                                              00092600
TOP-HALF.                                                                   00092700
    IF C-L-HIGH NOT > C-L-CURR THEN GO TO C-L-NOT-FOUND.                    00092800
    COMPUTE C-L-LOW = C-L-CURR + 1.                                         00092900
    GO TO C-L-BIN-SEARCH-LOOP.                                              00093000
C-L-NOT-FOUND.                                                              00093100
    IF N-O-E-T < 30 THEN ADD 1 TO N-O-E-T.                                  00093200
    MOVE N-O-E-T TO NO-OF-ELEMENTS-TRANS (CURRENT-IN-TRANS).                00093300
    MOVE 99 TO ELEMENT-TRANS (CURRENT-IN-TRANS, N-O-E-T).                   00093400
    GO TO RESET-IN-CONTRACT.                                                00093500
C-L-FOUND.                                                                  00093600
    MOVE 1 TO BRL-E-I.                                                      00093700
BRL-E-MOVE.                                                                 00093800
    IF BRAILLE-EQUIV (C-L-CURR, BRL-E-I) = 99 THEN GO TO                    00093900
        STORE-NOET.                                                         00094000
    IF N-O-E-T < 30 THEN ADD 1 TO N-O-E-T.                                  00094100
    MOVE BRAILLE-EQUIV (C-L-CURR, BRL-E-I)                                  00094200
    TO ELEMENT-TRANS (CURRENT-IN-TRANS, N-O-E-T).                           00094300
    ADD 1 TO BRL-E-I.                                                       00094400
    IF BRL-E-I NOT > 4 THEN GO TO BRL-E-MOVE.                               00094500
STORE-NOET.                                                                 00094600
    MOVE N-O-E-T TO NO-OF-ELEMENTS-TRANS (CURRENT-IN-TRANS).                00094700
RESET-IN-CONTRACT.                                                          00094800
    MOVE 0 TO IN-CONTRACT.                                                  00094900
    IF NXTCHR = RCC THEN GO TO MOVE-NEXT-CHAR.                              00095000
INC-C-I-T.                                                                  00095100
    ADD 1 TO CURRENT-IN-TRANS.                                              00095200
    IF CURRENT-IN-TRANS > 6 THEN MOVE 1 TO CURRENT-IN-TRANS.                00095300
    MOVE 0 TO NO-OF-ELEMENTS-TRANS (CURRENT-IN-TRANS).                      00095400
    GO TO END-INPUT.                                                        00095500
```

```
FIN.                                                                    00095600
    ADD 1 TO SPACE-COUNT.                                               00095700
    IF (SPACE-COUNT = 1) AND (PRIOR-SPACE > 0)                          00095800
       THEN MOVE ' ' TO NXTCHR                                          00095900
       ELSE MOVE ROO TO NXTCHR.                                         00096000
    GO TO END-INPUT.                                                    00096100
                                                                        00096200
READ-NEW-RECORD.                                                        00096300
    IF SPACE-COUNT > 0 THEN GO TO FIN.                                  00096400
    READ SYSINPUT, AT END GO TO FIN.                                    00096500
    MOVE 1 TO INPTR.                                                    00096600
    IF PROOF-OUT NOT = 'P' THEN GO TO END-INPUT.                        00096610
    MOVE SPACES TO OUT.                                                 00096620
    MOVE TEXT TO OUT.                                                   00096630
    WRITE OUTPUT-LINE AFTER ADVANCING 1.                                00096640
    ADD 1 TO CARDS-PRINTED.                                             00096650
END-INPUT.  EXIT.                                                       00096700
                                                                        00096800
                                                                        00096900
OUTPUT-SIGN SECTION.                                                    00097000
                                                                        00097100
CHARACTER-TEST.                                                         00097200
    MOVE CURRENT-S (CURRENT-TYPE) TO CS.                                00097300
    MOVE HEAD-S (CURRENT-TYPE) TO HS.                                   00097400
    MOVE TAIL-S (CURRENT-TYPE) TO TS.                                   00097500
    IF OUTSIGN < 64 THEN GO TO ORD-CHAR-OUT.                            00097700
    IF OUTSIGN = 64 THEN GO TO SPACE-OUT.                               00097800
    IF OUTSIGN NOT > CC-HIGH THEN GO TO CARRIAGE-CONTROL.               00097900
    IF OUTSIGN NOT > SC-HIGH THEN GO TO STACK-CONTROL.                  00098000
    GO TO MODE-CONTROL.                                                 00098100
                                                                        00098200
SPACE-OUT.                                                              00098300
    IF SELF-TEST = 0 THEN GO TO SKIP-SELF-TEST.                         00098500
    ADD 1 TO CURRENT-TRANS.                                             00098600
    IF CURRENT-TRANS > 6 THEN MOVE 1 TO CURRENT-TRANS.                  00098700
    PERFORM TRANS-TEST.                                                 00098800
    IF WORD-ERROR = 0 THEN GO TO SKIP-SELF-TEST.                        00098900
    MOVE CURRENT-TRANS TO M.                                            00099100
LOOP4.                                                                  00099200
    ADD 1 TO CURRENT-TRANS.                                             00099300
    IF CURRENT-TRANS > 6 THEN MOVE 1 TO CURRENT-TRANS.                  00099400
    IF M = CURRENT-TRANS THEN GO TO MARK-ERROR.                         00099500
    PERFORM TRANS-TEST.                                                 00099600
    IF WORD-ERROR = 0 THEN GO TO MARK-ERROR.                            00099700
     GO TO LOOP4.                                                       00099800
MARK-ERROR.                                                             00099810
    ADD 1 TO SC-ERROR-CT.                                               00099820
    MOVE 'ISK' TO PROOF (40).                                           00099830
    PERFORM MOVE-ERROR 2 TIMES.                                         00099890
SKIP-SELF-TEST.                                                         00099900
    IF STACK-INDICATOR = 2 THEN PERFORM CLEAR-T8,                       00100000
    ELSE PERFORM STACK-T8.                                              00100300
    NOTE STACK-INDICATOR IS ALWAYS 2, 4, OR 5.                          00100400
```

```
    GO TO END-OUTPUT.                                                    00100700
                                                                         00100800
ORD-CHAR-OUT.                                                            00100900
    IF NO-OF-ELEMENTS (CS) = 30                                          00101000
    THEN GO TO OVER-FLOW-STACK.                                          00101100
    ADD 1 TO NO-OF-ELEMENTS (CS).                                        00101200
CONTINUE-ORD.                                                            00101300
    MOVE NO-OF-ELEMENTS (CS) TO NCS.                                     00101400
    MOVE OUTSIGN TO ELEMENT (CS, NCS).                                   00101500
    GO TO END-OUTPUT.                                                    00101600
OVER-FLOW-STACK.                                                         00101700
    PERFORM STACK-TB.                                                    00101800
    MOVE 1 TO NO-OF-ELEMENTS (CS).                                       00101900
    GO TO CONTINUE-ORD.                                                  00102000
                                                                         00102100
CARRIAGE-CONTROL.                                                        00102200
    IF NO-OF-ELEMENTS (CS) NOT = 0 THEN PERFORM STACK-TB.                00102300
    MOVE 30 TO NO-OF-ELEMENTS (CS).                                      00102600
    MOVE OUTSIGN TO SPECIAL-CONTROL (CS).                                00102700
    IF OUTSIGN NOT = 72 THEN GO TO LOOP72.                               00102800
    MOVE ONECHAR TO ELEMENT (CS, 31).                                    00102900
    PERFORM SHIFT-LEFT-ONE.                                              00103000
    IF STACK-INDICATOR = 2 THEN MOVE 5 TO STACK-INDICATOR.               00103100
    GO TO END-OUTPUT.                                                    00103200
LOOP72.                                                                  00103300
    IF (OUTSIGN NOT = 68) AND (OUTSIGN NOT = 71)                         00103400
    THEN GO TO END-OUTPUT.                                               00103500
    MOVE FRSTCHARS TO ELEMENT (CS, 3).                                   00103600
    PERFORM SHIFT-LEFT-ONE 2 TIMES.                                      00103700
    GO TO END-OUTPUT.                                                    00103800
                                                                         00103900
STACK-CONTROL.                                                           00104000
    IF OUTSIGN = 86 THEN GO TO UNIT-OF-MEASURE.                          00104400
    IF OUTSIGN = 88 THEN GO TO BEGIN-TITLE.                              00104600
    IF OUTSIGN = 89 THEN GO TO END-TITLE.                                00104700
    IF OUTSIGN = 82 THEN GO TO END-HEADING.                              00104800
    IF OUTSIGN = 81 THEN GO TO BEGIN-HEADING.                            00104900
    IF OUTSIGN = 84 THEN IF STACK-INDICATOR = 2                          00104910
       THEN MOVE 5 TO STACK-INDICATOR.                                   00104920
    GO TO END-OUTPUT.                                                    00105000
                                                                         00105100
MODE-CONTROL.                                                            00105200
    IF OUTSIGN = 91 THEN GO TO SELF-TEST-OFF, ELSE                       00105210
    IF OUTSIGN = 92 THEN GO TO SELF-TEST-ON, ELSE                        00105220
    IF OUTSIGN = 93 THEN PERFORM SET-TAB, ELSE                           00105300
    IF OUTSIGN = 94 THEN PERFORM COMPLETE-CONTROL, ELSE                  00105400
    IF OUTSIGN = 95 THEN MOVE 1 TO POETRY-INDICATOR, ELSE                00105500
    IF OUTSIGN = 96 THEN MOVE 0 TO POETRY-INDICATOR, ELSE                00105600
    IF OUTSIGN = 97 THEN PERFORM COMPUTER-BRAILLE, ELSE                  00105700
    IF OUTSIGN = 98 THEN GO TO WRAP-UP.                                  00105800
    GO TO END-OUTPUT.                                                    00105900
                                                                         00106000
BEGIN-HEADING.                                                           00106100
```

```
    IF NO-OF-ELEMENTS (CS) NOT = 0                                          00106200
    PERFORM CLEAR-T8.                                                       00106300
    MOVE 65 TO OUTSIGN.                                                     00106500
    PERFORM CARRIAGE.                                                       00106600
    MOVE 4 TO STACK-INDICATOR.                                              00106700
    GO TO END-OUTPUT.                                                       00106800
                                                                            00106900
END-HEADING.                                                                00107000
    MOVE HEAD-S (1) TO N.                                                   00107100
    MOVE 0 TO M.                                                            00107200
LOOP51.                                                                     00107300
    ADD NO-OF-ELEMENTS (N) TO M.                                            00107400
    IF SPECIAL-CONTROL (N) = 0 AND NO-OF-ELEMENTS (N) > 0 THEN              00107510
    ADD 1 TO M.                                                             00107600
    IF N = TAIL-S (1) THEN GO TO LOOP52.                                    00107610
    MOVE NEXT-STACK (N) TO N.                                               00107700
    GO TO LOOP51.                                                           00107800
LOOP52.                                                                     00107900
    IF M > OUTL THEN MOVE 1 TO OUTPTR                                       00108000
       ELSE COMPUTE OUTPTR = ((OUTL - M + 2) / 2) + 1.                      00108100
    MOVE 2 TO STACK-INDICATOR.                                              00108200
    MOVE 1 TO HEAD-IND.                                                     00108210
    PERFORM CLEAR-T8.                                                       00108300
    MOVE 0 TO HEAD-IND.                                                     00108310
    MOVE 65 TO OUTSIGN.                                                     00108400
    PERFORM CARRIAGE.                                                       00108500
    GO TO END-OUTPUT.                                                       00108600
                                                                            00108700
BEGIN-TITLE.                                                                00108800
    MOVE 2 TO CURRENT-TYPE.                                                 00108900
    MOVE HEAD-S (2) TO HS.                                                  00109000
    MOVE TAIL-S (2) TO TS.                                                  00109100
    MOVE CURRENT-S (2) TO CS.                                               00109200
    MOVE 4 TO STACK-INDICATOR.                                              00109300
    MOVE HEAD-S (2) TO M.                                                   00109400
    IF NO-OF-ELEMENTS (M) = 0 THEN GO TO END-OUTPUT.                        00109600
    MOVE 0 TO NO-OF-ELEMENTS (M).                                           00109700
LOOP41.                                                                     00109800
    IF HEAD-S (2) NOT = TAIL-S (2) THEN PERFORM FREE-LAST-STACK             00109900
       ELSE GO TO END-OUTPUT.                                               00110000
    MOVE HEAD-S (2) TO CURRENT-S (2).                                       00110100
    GO TO LOOP41.                                                           00110200
    GO TO END-OUTPUT.                                                       00110300
                                                                            00110400
END-TITLE.                                                                  00110500
    MOVE 0 TO TITLE-LENGTH.                                                 00110600
    MOVE HEAD-S (2) TO N.                                                   00110700
LOOP12.                                                                     00110800
    ADD NO-OF-ELEMENTS (N) TO TITLE-LENGTH.                                 00110900
    IF SPECIAL-CONTROL (N) = 0 AND NO-OF-ELEMENTS (N) > 0 THEN              00111010
    ADD 1 TO TITLE-LENGTH.                                                  00111100
    IF N = TAIL-S (2) THEN GO TO TITLE2.                                    00111110
    MOVE NEXT-STACK (N) TO N.                                               00111200
```

```
    GC TO LCOP12.                                                    00111300
TITLE2.                                                              00111400
    MCVE 1 TO CURRENT-TYPE.                                          00111500
    MCVE 2 TO STACK-INDICATCR.                                       00111700
    GC TO END-OUTPUT.                                                00111800
                                                                     00111905
SELF-TEST-CFF.                                                       00111910
    ALTER S-T-B1 TO PRCCEED TO S-T-NO1.                              00111915
    MCVE 0 TO SELF-TEST.                                             00111920
    GO TO END-OUTPUT.                                                00111925
                                                                     00111930
SELF-TEST-CN.                                                        00111935
    ALTER S-T-B1 TO PRCCEED TO S-T-YES1.                             00111940
    MCVE 1 TO SELF-TEST.                                             00111945
    MCVE 6 TO CURRENT-TRANS.                                         00111950
    MOVE 1 TO CURRENT-IN-TRANS.                                      00111955
    MCVE 0 TO NO-OF-ELEMENTS-TRANS (1).                              00111960
    MOVE C TO IN-CCNTRACT.                                           00111965
    GC TO END-OUTPUT.                                                00111970
                                                                     00111975
UNIT-CF-MEASURE.                                                     00111980
    MOVE 1 TO SPECIAL-CONTROL (CS).                                  00111985
    PERFORM TAIL-SWAP.                                               00111990
    IF RL1 = 'S' THEN PERFORM SHIFT-LEFT-ONE.                        00111991
    IF RL1 NOT = '.' THEN GO TO END-OUTPUT.                          00111992
    IF RCHAR (1) NOT = ' ' OR RCHAR (2) NOT = ' '                    00111993
       THEN PERFORM SHIFT-LEFT-CNE.                                  00111994
    GO TO END-OUTPUT.                                                00111995
                                                                     00111996
WRAP-UP.                                                             00111999
    MCVE 65 TO OUTSIGN.                                              00112000
    FERFORM CARRIAGE.                                                00112100
CLOSE-FILES.                                                         00112200
    CLCSE SYSINPUT.                                                  00112300
    CLCSE SYSPRINT.                                                  00112400
    CLOSE SYSRL.                                                     00112500
    CLCSE SYSRPQ.                                                    00112600
    IF PUNCHED-OUTPUT NOT = 'N' THEN CLCSE PUNCH.                    00112700
    STOP RUN.                                                        00112800
                                                                     00112900
END-OUTPUT.  EXIT.                                                   00113000
                                                                     00113100
                                                                     00113200
                                                                     00113300
CARRIAGE SECTION.                                                    00113400
CARRIAGE-PARAGRAPH.                                                  00113500
BREAK4.                                                              00113600
    IF LINECOLNT < LPG                                               00113700
    THEN GO TO BREAK5.                                               00113800
    MCVE SPACES TO OUT.                                              00113900
    IF PRCOF-OUT = 'P' THEN                                          00113910
    WRITE OUTPUT-LINE AFTER C.                                       00114000
    MOVE SPACES TO OUT-RPQ.                                          00114100
```

```
    IF RPC = 'R' THEN                                                   00114200
    WRITE OUTPUT-RPQ AFTER 0.                                           00114300
    MOVE SPACES TO CUT-BRL.                                             00114400
    IF BRAILLE = 'B' THEN                                               00114500
    WRITE OUTPUT-BRL AFTER 0.                                           00114600
    MOVE 0 TO LINECCUNT.                                                00114700
    MCVE 0 TO CARDS-PRINTED.                                            00114710
    PERFORM PAGE-HEAD.                                                  00114800
BREAK5.                                                                 00114900
    IF OUTPTR = 1 THEN IF CUTSIGN = 65 OR OUTSIGN = 67                  00115000
    THEN GO TO BREAK12.                                                 00115100
    PERFORM CLT-OUTPUT.                                                 00115200
BREAK12.                                                                00115300
    IF OUTSIGN = 65 OR CUTSIGN = 69 THEN MOVE 1 TO OUTPTR,              00115400
    ELSE IF OUTSIGN = 67 THEN MOVE 3 TO OUTPTR,                         00115500
    ELSE IF OUTPTR = CUTL + 1 THEN MOVE 1 TO OUTPTR,                    00115600
    ELSE IF OUTPTR = 2 * CUTL THEN MOVE CUTL TO OUTPTR,                 00115700
    ELSE COMPLTE OUTPTR = OUTPTR - M.                                   00115800
END-CARRIAGE. EXIT.                                                     00115900
                                                                        00116000
CLEAR-BT SECTION.                                                       00116100
CLEAR-BT-PARAGRAPH.                                                     00116200
    MOVE 2 TO STACK-INDICATOR.                                          00116300
    MCVE 0 TO I.                                                        00116400
    PERFORM CLEAR-STACKS.                                               00116500
    MCVE HS TO CURRENT-S (CURRENT-TYPE).                                00116600
END-CLEAR-BT. EXIT.                                                     00116700
                                                                        00116800
CLEAR-STACKS SECTION.                                                   00116900
CLEAR-STACKS-PARAGRAPH.                                                 00117000
    IF I = C THEN MCVE TS TO PS ELSE MOVE HS TO PS.                     00117100
    MCVE 0 TO M.                                                        00117200
    IF SPECIAL-CONTRCL (PS) < 64                                        00117300
    THEN GO TO LOOP14.                                                  00117400
    MCVE SPECIAL-CONTRCL (PS) TO OUTSIGN.                               00117500
    IF OUTSIGN NOT = 68 THEN GC TO LCOP20.                              00117600
    MCVE ELEMENT (PS, 3) TC TABULATE.                                   00117700
    PERFORM TABULATICN.                                                 00117800
    GC TO LCOP22.                                                       00117900
LOOP20.                                                                 00118000
    IF OUTSIGN NOT = 71 THEN GO TO LOOP30.                              00118100
    IF OUTPTR = 1 THEN GC TC LOOP54.                                    00118200
    MOVE 65 TO OUTSIGN.                                                 00118300
    PERFORM CARRIAGE.                                                   00118400
LCOP54.                                                                 00118500
    MCVE SPACES TO BRAILLE-LINES.                                       00118600
    MCVE SPACES TO PROCF-LINE.                                          00118700
    MOVE ZEROS TO CODE-LINE.                                            00118800
LCOP27.                                                                 00118900
    IF M = ELEMENT (PS, 3) THEN GO TO LOOP22.                           00119000
    CCMPUTE OUTPTR = OLTL + 1.                                          00119100
    PERFCRM CARRIAGE.                                                   00119200
    ADD 1 TC M.                                                         00119300
```

```
    GO TO LOOP27.                                                    00119400
LOOP3C.                                                              00119500
    IF OUTSIGN NOT = 72 THEN GO TO LOOP26.                           00119600
    MOVE ELEMENT (PS, 3) TO N.                                       00119700
    IF TAB-TYPE (M) = 'R' THEN GO TO LOOP73.                         00119800
    IF TAB-TYPE (M) = 'C' THEN GO TO LOOP74.                         00119900
    MOVE TAB-COLUMN (M) TO TABULATE.                                 00120000
    PERFORM TABULATION.                                              00120100
    GO TO LOOP22.                                                    00120200
LOOP73.                                                              00120300
    MOVE NEXT-STACK (PS) TO XX.                                      00120400
    COMPUTE TABULATE = TAB-COLUMN (M) - NO-OF-ELEMENTS (XX) + 1.     00120500
    PERFORM TABULATION.                                              00120600
    GO TO LOOP22.                                                    00120700
LOOP74.                                                              00120800
    MOVE NEXT-STACK (PS) TO XX.                                      00120900
    MOVE 1 TO XYZ.                                                   00121000
LOOP75.                                                              00121100
    IF ELEMENT (XX, XYZ) =  40  THEN GO TO LOOP76.                   00121200
    ADD 1 TO XYZ.                                                    00121300
    IF XYZ NOT > NO-OF-ELEMENTS (XX) THEN GO TO LOOP75.              00121400
LOOP76.                                                              00121500
    COMPUTE TABULATE = TAB-COLUMN (M) - XYZ + 1.                     00121600
    PERFORM TABULATION.                                              00121700
    GO TO LOOP22.                                                    00121800
LOOP26.                                                              00121900
    PERFORM CARRIAGE.                                                00122000
    GO TO LOOP22.                                                    00122100
LOOP14.                                                              00122200
    IF NO-OF-ELEMENTS (PS) = 0 THEN GO TO LOOP22.                    00122300
    IF OUTPTR + NO-OF-ELEMENTS (PS) < OUTL + 2                       00122400
    THEN GO TO LOOP5.                                                00122500
    COMPUTE OUTPTR = CLTL + 1.                                       00122600
    PERFORM CARRIAGE.                                                00122700
    IF POETRY-INDICATOR = 1 THEN PERFORM MOVE-SPACE 2 TIMES.         00122800
    IF HEAD-ING = 1 THEN PERFORM MOVE-SPACE 3 TIMES.                 00122810
LOOP5.                                                               00122820
    MOVE PS TO N.                                                    00122830
LOOP5R.                                                              00122900
    ADD 1 TO M.                                                      00123000
    PERFORM MOVE-ELEMENT.                                            00123200
    IF M NOT = NO-OF-ELEMENTS (N) THEN GO TO LOOP5R.                 00123300
    IF SPECIAL-CONTROL (N) = 1 THEN GO TO LOOP22.                    00123400
    PERFORM MOVE-SPACE.                                              00123500
LOOP22.                                                              00123600
    IF HS = TS THEN GO TO DONE-CLEAR-STACKS.                         00123700
    IF I = 0 THEN PERFORM FREE-LAST-STACK                            00124100
    ELSE PERFORM FREE-HEAD-STACK.                                    00124200
    GO TO CLEAR-STACKS-PARAGRAPH.                                    00124300
DONE-CLEAR-STACKS.                                                   00124310
    MOVE 0 TO NO-OF-ELEMENTS (HS).                                   00124310
    MOVE 0 TO SPECIAL-CONTROL (HS).                                  00124320
END-CLEAR-STACKS. EXIT.                                              00124400
```

```
                                                                        00124500
CLEAR-TB SECTION.                                                       00124600
CLEAR-TB-PARAGRAPH.                                                     00124700
    MOVE 1 TO I.                                                        00124800
    PERFORM CLEAR-STACKS.                                               00124900
    MOVE HS TO CURRENT-S (CURRENT-TYPE).                                00125000
END-CLEAR-TB. EXIT.                                                     00125100
                                                                        00125200
COMPLETE-CONTROL SECTION.                                               00125300
COMPLETE-CONTROL-PARAGRAPH.                                             00125400
    IF RLI = ' ' THEN GO TO END-COMPLETE-CONTROL.                       00125500
    COMPUTE X = ONECHAR + 8 * RESTCHAR (1).                             00125600
    IF X = 0 THEN GO TO LOOP87.                                         00125700
    ADD 1 TO NO-OF-ELEMENTS (CS).                                       00125800
    MOVE NO-OF-ELEMENTS (CS) TO NCS.                                    00125900
    MOVE X TO ELEMENT (CS, NCS).                                        00126000
LOOP86.                                                                 00126100
    PERFORM SHIFT-LEFT-ONE.                                             00126200
    PERFORM SHIFT-LEFT-ONE.                                             00126300
    GO TO COMPLETE-CONTROL-PARAGRAPH.                                   00126400
LOOP87.                                                                 00126500
    PERFORM STACK-TB.                                                   00126600
    GO TO LOOP86.                                                       00126700
END-COMPLETE-CONTROL. EXIT.                                             00126800
                                                                        00126900
COMPUTER-BRAILLE SECTION.                                               00127000
COMP-BRAILLE.                                                           00127100
    IF RLI = ' ' THEN GO TO END-COMP-BRAILLE.                           00127200
    MOVE 1 TO LETTER.                                                   00127300
LOOP61.                                                                 00127400
    IF SYMBOL (LETTER) = RLI THEN GO TO LOOP62.                         00127500
    IF LETTER NOT < ALPHABET THEN GO TO COMP-BRAILLE-ERROR.             00127510
    ADD 1 TO LETTER.                                                    00127600
    GO TO LOOP61.                                                       00127700
COMP-BRAILLE-ERROR.                                                     00127710
    IF RLI = ACC THEN GO TO END-COMP-BRAILLE.                           00127720
    MOVE 'NEW CHAR' TO OUT.                                             00127730
    MOVE RLI TO OUT-PLACE (12).                                         00127740
    WRITE OUTPUT-LINE AFTER 2.                                          00127750
    MOVE SPACES TO OUT.                                                 00127760
    MOVE '*' TO RLI.                                                    00127770
    GO TO COMP-BRAILLE.                                                 00127780
LOOP62.                                                                 00127800
    ADD 1 TO NO-OF-ELEMENTS (CS).                                       00127900
    MOVE NO-OF-ELEMENTS (CS) TO NCS.                                    00128000
    IF COMP-B (LETTER) = 0 THEN MOVE 64 TO ELEMENT (CS, NCS)            00128100
       ELSE MOVE COMP-B (LETTER) TO ELEMENT (CS, NCS).                  00128200
    PERFORM SHIFT-LEFT-ONE.                                             00128300
    GO TO COMP-BRAILLE.                                                 00128400
END-COMP-BRAILLE. EXIT.                                                 00128500
                                                                        00128600
FREE-HEAD-STACK SECTION.                                                00128700
FREE-HEAD-STACK-PARAGRAPH.                                              00128800
```

```
    MOVE 0 TO NO-OF-ELEMENTS (HS).                                  00128810
    MOVE 0 TO SPECIAL-CONTROL (HS).                                 00128820
    IF NEXT-STACK (HS) = 0 THEN GO TO END-FREE-HEAD-STACK.          00128900
    MOVE HS TO M.                                                   00129000
    MOVE HEAD-FREE-STACK TO N.                                      00129200
    MOVE NEXT-STACK (M) TO HEAD-S (CURRENT-TYPE).                   00129300
    MOVE HEAD-S (CURRENT-TYPE) TO HS.                               00129400
    MOVE M TO HEAD-FREE-STACK.                                      00129500
    MOVE 0 TO PREVIOUS-STACK (HS).                                  00129700
    MOVE N TO NEXT-STACK (M).                                       00129800
END-FREE-HEAD-STACK. EXIT.                                          00129900
                                                                    00130000
FREE-LAST-STACK SECTION.                                            00130100
FREE-LAST-STACK-PARAGRAPH.                                          00130200
    MOVE 0 TO NO-OF-ELEMENTS (TS).                                  00130210
    MOVE 0 TO SPECIAL-CONTROL (TS).                                 00130220
    IF PREVIOUS-STACK (TS) = 0 THEN GO TO END-FREE-LAST-STACK.      00130300
    MOVE TS TO M.                                                   00130400
    MOVE HEAD-FREE-STACK TO N.                                      00130600
    MOVE PREVIOUS-STACK (M) TO TAIL-S (CURRENT-TYPE).               00130700
    MOVE TAIL-S (CURRENT-TYPE) TO TS.                               00130800
    MOVE 0 TO NEXT-STACK (TS).                                      00130900
    MOVE N TO NEXT-STACK (M).                                       00131100
    MOVE M TO HEAD-FREE-STACK.                                      00131300
END-FREE-LAST-STACK. EXIT.                                          00131400
                                                                    00131500
MOVE-ELEMENT SECTION.                                               00131600
MOVE-ELEMENT-PARAGRAPH.                                             00131700
    MOVE ELEMENT (N, M) TO X.                                       00131800
MOVE-SPECIAL-SIGN.                                                  00131900
    MOVE X TO CODED-SIGN (OUTPTR).                                  00132000
    IF X = 0 THEN MOVE 64 TO X.                                     00132010
    MOVE DOTS-1-4 (X) TO BRAILLE-SIGN (1, OUTPTR).                  00132100
    MOVE DOTS-2-5 (X) TO BRAILLE-SIGN (2, OUTPTR).                  00132200
    MOVE DOTS-3-6 (X) TO BRAILLE-SIGN (3, OUTPTR).                  00132300
    MOVE PROOF-CHARACTERS (X) TO PROOF (OUTPTR).                    00132400
    ADD 1 TO OUTPTR.                                                00132500
END-MOVE-ELEMENT. EXIT.                                             00132600
                                                                    00132700
MOVE-ERROR SECTION.                                                 00132800
MOVE-ERROR-PARAGRAPH.                                               00132900
    MOVE NO-OF-ELEMENTS (CS) TO N.                                  00133000
    IF N < 30 THEN ADD 1 TO N.                                      00133100
    MOVE 63 TO ELEMENT (CS, N).                                     00133200
    MOVE N TO NO-OF-ELEMENTS (CS).                                  00133300
END-MOVE-ERROR. EXIT.                                               00133800
MOVE-SPACE SECTION.                                                 00133900
MOVE-SPACE-PARAGRAPH.                                               00134000
    IF OUTPTR > OUTL THEN GO TO END-MOVE-SPACE.                     00134010
    MOVE 00 TO CODED-SIGN (OUTPTR).                                 00134100
    MOVE DOTS-1-4 (64) TO BRAILLE-SIGN (1, OUTPTR).                 00134200
    MOVE DOTS-2-5 (64) TO BRAILLE-SIGN (2, OUTPTR).                 00134300
    MOVE DOTS-3-6 (64) TO BRAILLE-SIGN (3, OUTPTR).                 00134400
```

```
    MOVE PROOF-CHARACTERS (64) TO PROOF (OUTPTR).                00134500
    ADD 1 TO OUTPTR.                                             00134600
END-MOVE-SPACE. EXIT.                                            00134700
                                                                 00134800
OUT-OUTPUT SECTION.                                              00134900
OUT-OUT.                                                         00135000
    IF PROOF-OUT NOT = 'P' THEN GO TO BREAK21.                   00135100
    MOVE SPACES TO OUT.                                          00135200
    IF CAROS-PRINTED = 0 THEN                                    00135210
    WRITE OUTPUT-LINE AFTER 1.                                   00135300
    MOVE 0 TO CAROS-PRINTED.                                     00135310
    MOVE 1 TO NN.                                                00135400
BREAK9.                                                          00135500
    MOVE BRAILLE-TEXT (NN) TO OUT.                               00135600
    WRITE OUTPUT-LINE AFTER 1.                                   00135700
    ADD 1 TO NN.                                                 00135800
    IF NN NOT > 3 THEN GO TO BREAK9.                             00135900
    MOVE PROOF-TEXT TO OUT.                                      00136000
    WRITE OUTPUT-LINE AFTER 1.                                   00136100
    MOVE 1 TO N.                                                 00136200
    IF PROOF (40) = 'TSK' THEN MOVE SC-ERROR-CT TO PROOF (40).   00136210
BREAK14.                                                         00136300
    MOVE CODED-SIGN (N) TO PROOF (N).                            00136400
    ADD 1 TO N.                                                  00136500
    IF N NOT > OUTL THEN GO TO BREAK14.                          00136600
    MOVE PROOF-TEXT TO OUT.                                      00136700
    WRITE OUTPUT-LINE AFTER 1.                                   00136800
BREAK21.                                                         00136900
    IF BRAILLE NOT = 'B' THEN GO TO BREAK31.                     00137000
    MOVE SPACES TO OUT-BRL.                                      00137100
    WRITE OUTPUT-BRL AFTER 1.                                    00137200
    MOVE 1 TO NN.                                                00137300
    MOVE 1 TO N.                                                 00137400
    COMPUTE K = 3 * OUTL.                                        00137500
BREAK10.                                                         00137600
    MOVE LINE-CHAR (NN, N) TO OUT-PLACE-BRL (K).                 00137700
    ADD 1 TO N.                                                  00137800
    SUBTRACT 1 FROM K.                                           00137900
    IF N NOT > OUTL * 3 THEN GO TO BREAK10.                      00138000
    WRITE OUTPUT-BRL AFTER 1.                                    00138100
    ADD 1 TO NN.                                                 00138200
    MOVE 1 TO N.                                                 00138300
    COMPUTE K = 3 * OUTL.                                        00138400
    IF NN NOT > 3 THEN GO TO BREAK10.                            00138500
BREAK31.                                                         00138600
    IF RPO NOT = 'R' THEN GO TO BREAK41.                         00138700
    MOVE SPACES TO OUT-RPO.                                      00138800
    WRITE OUTPUT-RPO AFTER 1.                                    00138900
    MOVE 1 TO NN.                                                00139000
BREAK33.                                                         00139100
    MOVE 1 TO K.                                                 00139200
    MOVE 1 TO N.                                                 00139300
    MOVE SPACE TO OUT-RPO-BEGIN.                                 00139400
```

```
BREAK32.                                                        00139500
    MOVE BRAILLE-SIGN (NN, N) TO OUT-RPQ-ONE (K).               00139600
    ADD 1 TO N.                                                 00139700
    MOVE BRAILLE-SIGN (NN, N) TO OUT-RPQ-TWO (K).               00139800
    ADD 1 TO N.                                                 00139900
    ADD 1 TO K.                                                 00140000
    IF N NOT > OUTL THEN GO TO BREAK32.                         00140100
    WRITE OUTPUT-RPQ AFTER 1.                                   00140200
    ADD 1 TO NN.                                                00140300
    IF NN NOT > 3 THEN GO TO BREAK33.                           00140400
BREAK41.                                                        00140500
    IF PUNCHED-OUTPUT = 'N' THEN GO TO BREAK51.                 00140600
    MOVE CODED-TEXT TO CODED-OUT.                               00140700
    WRITE CODED-OUTPUT AFTER POCKET-SELECT.                     00140800
BREAK51.                                                        00140900
    MOVE SPACES TO PRCOF-TEXT.                                  00141000
    MOVE ZEROS TO CODED-TEXT.                                   00141100
    MOVE SPACES TO BRAILLE-TEXT (1).                            00141200
    MOVE SPACES TO BRAILLE-TEXT (2).                            00141300
    MOVE SPACES TO BRAILLE-TEXT (3).                            00141400
    IF OUTSIGN = 69 THEN MOVE LPG TO LINECOUNT                  00141500
    ELSE ADD 1 TO LINECOUNT.                                    00141600
END-OUT-OUT. EXIT.                                              00141700
                                                                00141800
PAGE-HEAD SECTION.                                              00141900
PAGE-HEAD-PARAGRAPH.                                            00142000
    IF PAGINATION = 'N' THEN GO TO END-PAGE-HEAD.               00142100
    COMPUTE N = ((OUTL - TITLE-LENGTH + 2) / 2) + 1.            00142200
    IF N < 1 THEN MOVE 1 TO N.                                  00142300
    MOVE BRAILLE-LINES TO TITLE-LINES.                          00142400
    MOVE PROOF-LINE TO TITLE-PROOF.                             00142500
    MOVE CODE-LINE TO TITLE-SIGN-CODE.                          00142600
    MOVE OUTPTR TO OUTPTRSV.                                    00142610
    MOVE SPACES TO BRAILLE-LINES.                               00142700
    MOVE SPACES TO PROOF-LINE.                                  00142800
    MOVE ZEROS TO CODE-LINE.                                    00142900
    MOVE N TO OUTPTR.                                           00143000
    MOVE 0 TO XYZ.                                              00143100
    MOVE HEAD-S (2) TO N.                                       00143200
LOOP42.                                                         00143300
    IF NO-OF-ELEMENTS (N) = 0 OR SPECIAL-CONTROL (N) NOT < 64   00143400
       THEN GO TO LOOP42-CONT.                                  00143410
LOOP42A.                                                        00143420
    ADD 1 TO XYZ.                                               00143500
    MOVE ELEMENT (N, XYZ) TO X.                                 00143600
    PERFORM MOVE-SPECIAL-SIGN.                                  00143700
    IF XYZ NOT = NO-OF-ELEMENTS (N) THEN GO TO LOOP42A.         00143800
    IF SPECIAL-CONTROL (N) = 0 THEN PERFORM MOVE-SPACE.         00143810
LOOP42-CONT.                                                    00143820
    IF N = TAIL-S (2) THEN GO TO LOOP44.                        00143900
    MOVE 0 TO XYZ.                                              00144000
    MOVE NEXT-STACK (N) TO N.                                   00144100
    IF OUTPTR + NO-OF-ELEMENTS (N) > OUTL - 5 THEN GO TO LOOP44. 00144210
```

```
    GO TO LOOP42.                                                    00144300
LOOP44.                                                              00144400
    MOVE I TO N.                                                     00144500
    COMPUTE OUTPTR = OUTL - 4.                                       00144600
LOOP46.                                                              00144700
    IF CRT-PAGE-DIGIT (N) NOT = 0 THEN GO TO LOOP45.                 00144800
    ADD 1 TO N.                                                      00144900
    ADD 1 TO OUTPTR.                                                 00145000
    GO TO LOOP46.                                                    00145100
LOOP45.                                                              00145200
    MOVE 60 TO X.                                                    00145300
    PERFORM MOVE-SPECIAL-SIGN.                                       00145400
LOOP47.                                                              00145500
    COMPUTE XYZ = CRT-PAGE-DIGIT (N) + 1.                            00145600
    MOVE DIGIT (XYZ) TO X.                                           00145700
    PERFORM MOVE-SPECIAL-SIGN.                                       00145800
    ADD 1 TO N.                                                      00145900
    IF N NOT > 4 THEN GO TO LOOP47.                                  00146000
    ADD 1 TO CRT-PAGE.                                               00146100
LOOP43.                                                              00146200
    PERFORM OUT-OUTPUT.                                              00146300
    MOVE OUTPTRSV TO OUTPTR.                                         00146310
    MOVE TITLE-LINES TO BRAILLE-LINES.                               00146400
    MOVE TITLE-PROOF TO PROOF-LINE.                                  00146500
    MOVE TITLE-SIGN-CODE TO CODE-LINE.                               00146600
END-PAGE-HEAD.   EXIT.                                               00146700
                                                                     00146800
PUSH SECTION.                                                        00146900
STACK-RT.                                                            00147000
    IF HEAD-FREE-STACK = 0 THEN GO TO END-PUSH.                      00147100
    MOVE HEAD-FREE-STACK TO M.                                       00147200
    MOVE NEXT-STACK (HEAD-FREE-STACK) TO HEAD-FREE-STACK.            00147300
    MOVE HS TO N.                                                    00147400
    MOVE C                    TO PREVIOUS-STACK (M).                 00147500
    MOVE M TO PREVIOUS-STACK (N).                                    00147600
    MOVE      M TO HEAD-S (CURRENT-TYPE).                            00147800
    MOVE M                      TO HS.                               00147900
    MOVE N TO NEXT-STACK (M).                                        00148000
    MOVE M TO CURRENT-S (CURRENT-TYPE).                              00148100
    MOVE M                          TO CS.                           00148200
END-PUSH. EXIT.                                                      00148300
                                                                     00148400
SET-TAB SECTION.                                                     00148500
SET-TAB-PARAGRAPH.                                                   00148600
    MOVE RCHAR (1) TO TAB-TYPE (ONECHAR).                            00148700
    MOVE SCNOCHARS TO TAB-COLUMN (ONECHAR).                          00148800
    PERFORM SHIFT-LEFT-ONE 4 TIMES.                                  00148900
END-SET-TAB. EXIT.                                                   00149000
                                                                     00149100
SHIFT-LEFT-ONE SECTION.                                              00149200
SHIFT-LEFT-ONE-PARAGRAPH.                                            00149300
    MOVE RRS TO TEMP.                                                00149400
    MOVE TEMP TO RL9.                                                00149500
```

```
    PERFORM INPUT-CHAR.                                                  00149600
    MOVE NXTCHR TO RR1.                                                  00149700
END-SHIFT-LEFT-ONE. EXIT.                                                00149800
                                                                         00149900
STACK-TB SECTION.                                                        00150000
STACK-TB-PARAGRAPH.                                                      00150100
    IF HEAD-FREE-STACK = 0 THEN GO TO END-STACK-TB.                      00150200
    MOVE HEAD-FREE-STACK TO N.                                           00150300
    MOVE NEXT-STACK (N) TO HEAD-FREE-STACK.                              00150400
    MOVE 0 TO NEXT-STACK (N).                                            00150600
    MOVE TS TO PREVIOUS-STACK (N).                                       00150700
    MOVE N TO NEXT-STACK (TS).                                           00150710
    MOVE N TO TAIL-S (CURRENT-TYPE).                                     00150800
    MOVE N                          TO TS.                               00150900
    MOVE N TO CURRENT-S (CURRENT-TYPE).                                  00151100
    MOVE N                             TO CS.                            00151200
    IF STACK-INDICATOR = 5 THEN MOVE 2 TO STACK-INDICATOR.               00151400
END-STACK-TB. EXIT.                                                      00151500
                                                                         00151600
TABULATION SECTION.                                                      00151700
TABULATION-PARAGRAPH.                                                    00151800
    SUBTRACT OUTPTR FROM TABULATE.                                       00151900
    IF TABULATE < 0 THEN GO TO LOOP65.                                   00152000
    MOVE TABULATE TO N.                                                  00152100
    PERFORM MOVE-SPACE N TIMES.                                          00152200
    GO TO LOOP66.                                                        00152300
LOOP65.                                                                  00152400
    COMPUTE TABULATE = TABULATE + OUTPTR - 1.                            00152500
    MOVE 65 TO OUTSIGN.                                                  00152600
    PERFORM CARRIAGE.                                                    00152700
    PERFORM MOVE-SPACE TABULATE TIMES.                                   00152800
LOOP66.                                                                  00152900
    MOVE 0 TO TABULATE.                                                  00153000
END-TABULATION. EXIT.                                                    00153100
                                                                         00153110
TAIL-SWAP SECTION.                                                       00153115
TAIL-SWAP-P.                                                             00153120
    MOVE PREVIOUS-STACK (TS) TO N.                                       00153125
    IF N = 0 THEN GO TO END-TAIL-SWAP.                                   00153130
    MOVE PREVIOUS-STACK (N) TO M.                                        00153132
    MOVE PREVIOUS-STACK (N) TO PREVIOUS-STACK (TS).                      00153135
    MOVE N TO NEXT-STACK (TS).                                           00153140
    MOVE TS TO PREVIOUS-STACK (N).                                       00153145
    MOVE 0 TO NEXT-STACK (N).                                            00153150
    MOVE N TO TS.                                                        00153155
    MOVE TS TO TAIL-S      (CURRENT-TYPE).                               00153156
    MOVE N TO CS.                                                        00153157
    MOVE N TO CURRENT-S (CURRENT-TYPE).                                  00153158
    IF N NOT = HS THEN GO TO SET-FWD-LINK.                               00153159
    MOVE PREVIOUS-STACK (N) TO HS.                                       00153161
    MOVE HS TO HEAD-S (CURRENT-TYPE).                                    00153162
    GO TO END-TAIL-SWAP.                                                 00153163
SET-FWD-LINK.                                                            00153164
```

```
           MOVE PREVIOUS-STACK (N) TO NEXT-STACK (M).                    00153165
       END-TAIL-SWAP.  EXIT.                                             00153169
                                                                         00153200
       TRANS-TEST SECTION.                                               00153300
       TRANS-TEST-PARAGRAPH.                                             00153400
           IF NO-OF-ELEMENTS (CS) NOT =                                  00153500
           NO-OF-ELEMENTS-TRANS (CURRENT-TRANS) THEN GO TO FAIL-TRANS.   00153600
           IF NO-OF-ELEMENTS (CS) = 0 THEN GO TO PASS-TRANS.             00153610
           MOVE 0 TO N.                                                  00153700
       LOOP3.                                                            00153800
           ADD 1 TO N.                                                   00153900
           IF ELEMENT (CS, N) NOT =                                      00154000
           ELEMENT-TRANS (CURRENT-TRANS, N)                              00154100
           THEN GO TO FAIL-TRANS.                                        00154200
           IF N < NO-OF-ELEMENTS (CS)                                    00154300
           THEN GO TO LOOP3.                                             00154400
       PASS-TRANS.                                                       00154410
           MOVE 0 TO WORD-ERROR.                                         00154500
           GO TO END-TRANS-TEST.                                         00154600
       FAIL-TRANS.                                                       00154700
           MOVE 1 TO WORD-ERROR.                                         00154800
       END-TRANS-TEST. EXIT.                                             00154900
                                                                         00155000
       READ-CARD SECTION.                                                00155100
       READ-CARD-PARAGRAPH.                                              00155200
           READ SYSINPUT, AT END GO TO FIN.                              00155300
           MOVE CARD-XX TO CARD-NO.                                      00155305
           IF CARD-NO NOT > PREV-CARD-NO THEN GO TO SEQ-ERROR.           00155310
           MOVE CARD-NO TO PREV-CARD-NO.                                 00155320
           IF ECHC NOT = 'E' THEN GO TO END-READ-CARD.                   00155400
           GO TO PRINT-CARD.                                             00155410
       SEQ-ERROR.                                                        00155420
           MOVE 'Y' TO SEQ-ERR-IND.                                      00155430
           MOVE ZERO TO PREV-CARD-NO.                                    00155440
           MOVE SPACES TO OUT.                                           00155450
           MOVE '** FOLLOWING CARD(S) OUT OF SEQUENCE:' TO OUT.          00155460
           WRITE OUTPUT-LINE AFTER ADVANCING 1.                          00155470
       PRINT-CARD.                                                       00155480
           MOVE TEXT TO OUT.                                             00155500
           WRITE OUTPUT-LINE AFTER ADVANCING 1.                          00155600
       END-READ-CARD. EXIT.                                              00155700
//LKED EXEC PGM=IEWL,PARM=(XREF,LIST,LET),                              X00000100
// REGION=96K                                                            00000200
//SYSLIB DD DSNAME=SYS1.COBLIB,DISP=SHR                                 00000400
//SYSLIN DD DSNAME=&LOADSET,DISP=(OLD,DELETE),UNIT=SYSDA                00000300
//SYSUT1 DD UNIT=SYSDA,SPACE=(1024,(50,20))                             00000500
//SYSLMOD DD DSN=&&GOGO(DSYS3),DISP=(NEW,PASS),
//             UNIT=SYSDA,SPACE=(CYL,(1,1,1))
//SYSPRINT DD SYSOUT=A,                                                 00000800
//         DCB=(BLKSIZE=121,LRECL=121,RECFM=FBM)                        00000900
// EXEC PGM=DSYS3
//STEPLIB DD DSN=&&GOGO,DISP=(SHR,PASS)
//SYSOUT DD SYSOUT=A,
```

```
// OCB=(LRECL=120,BLKSIZE=120)
//SYSO(SPL DD SYSOUT=A,
// OCB=(LRECL=120,BLKSIZE=120)
//SYSPRINT DD SYSOUT=A,
// OCB=(RECFM=FBA,LRECL=133,BLKS(ZE=3458)
//SYSRPQ OO SYSOLT=A,
// OCB=(RECFM=FBA,LRECL=132,BLKSIZE=132)
//SYSBRL OO SYSOLT=A,
// OCB=(RECFM=FBA,LRECL=132,BLKSIZE=132)
//SYSABENO CO SYSOUT=A,SPACE=(TRK,(C,BO))
//SYSIN DO *
NCECHO                                                          00000001
ABOUT                   0103999999                              00000010
ABOVE                   0103399995                              00000020
ACCOROING               0109999999                              00000030
ACROSS                  0109239999                              00000040
AFTER                   0111995995                              00000050
AFTERNCCN               0111299999                              00000060
AFTERWARO               0111585995                              00000070
AGA(N                   0127959995                              00000080
AGA(NST                 0127129999                              00000090
ALLY                    3261999999                              00000100
ALMOST                  0107139599                              00000110
ALREAOY                 0107239599                              00000120
ALSO                    0107959595                              00000130
ALTHCUGH                0107579599                              00000140
ALTOGETHER              0107309995                              00000150
ALWAYS                  0107585595                              00000160
ANCE                    4017999999                              00000170
ANO                     4799999995                              00000180
AR                      2899999999                              00000190
AS                      5399999999                              00000200
AT(ON                   3229999999                              00000210
BB                      0699999999                              00000220
BE                      0699959995                              00000230
BECAUSE                 0605999995                              00000240
BEFORE                  0611999999                              00000250
BEHINO                  0619999999                              00000260
BELOW                   0607999999                              00000270
BENEATH                 0629999999                              00000280
BESIOE                  0614955599                              00000290
BETWEEN                 0630959999                              00000300
BEYONO                  0661955999                              00000310
BLE                     6C99995999                              00000320
BL(NO                   0307999999                              00000330
BRAILLE                 0323079999                              00000340
BUT                     0399995599                              00000350
BY                      5299999999                              00000360
CAN                     0999999595                              00000370
CANNOT                  5609999995                              00000380
CC                      1899959995                              00000390
CH                      3399995599                              00000400
CHARACTER               1633999999                              00000410
```

CHILO	3399995999	00000420
CHILDREN	3329995995	00000430
COM	3699959995	00000440
CON	1895595995	00000450
CONCEIVE	1809355995	00000460
CONCEIVING	1809392799	00000470
COULD	0925999999	00000480
DAY	1625995959	00000490
OO	5099999999	00000500
OECEIVE	2505395995	00000510
OECEIVING	2509352795	00000520
OECLARE	2505C75995	00000530
OECLARING	2505072795	00000540
OIS	5099999995	00000550
OO	2595559999	00000560
EA	0299995995	00000570
EO	4399999999	00000580
EITHER	1710555595	00000590
EN	3499995995	00000600
ENCE	4817999595	00000610
ENOUGH	3499555595	00000620
ER	5995999999	00000630
EVER	1617995595	00000640
EVERY	1799999995	00000650
FATHER	1611995995	00000660
FF	2299959595	00000670
FIRST	1112995595	00000680
FOR	6399599595	00000690
FRIENO	1123955595	00000700
FROM	1199959999	00000710
FUL	4807995999	00000720
GG	5499555555	00000730
GH	3599999999	00000740
GO	2799555555	00000750
GOOD	2725995995	00000760
GREAT	2723305995	00000770
HAO	5619995995	00000780
HAVE	1999959999	00000790
HERE	1619995995	00000800
HERSELF	1955119995	00000810
HIM	1913999995	00000820
HIMSELF	1913115595	00000830
HIS	3899955595	00000840
IMMEDIATE	1013135595	00000850
IN	2095555595	00000860
ING	4499995999	00000870
INTO	2022555595	00000880
IT	4599955595	00000890
ITS	4514995599	00000900
ITSELF	4511995595	00000910
ITY	4861995999	00000920
JUST	2695995999	00000930
KNOW	1605995595	00000940

KNOWLEDGE	0599999999	00000950
LESS	4014999999	00000960
LETTER	0723999999	00000970
LIKE	0799999999	00000980
LITTLE	0707999999	00000990
LORD	1607999999	00001000
MANY	5613999999	00001010
MENT	4830999999	00001020
MORE	1399999999	00001030
MOTHER	1613999999	00001040
MUCH	1333999999	00001050
MUST	1312999999	00001060
MYSELF	1361119999	00001070
NAME	1629999999	00001080
NECESSARY	2917099999	00001090
NEITHER	2917109999	00001100
NESS	4814999999	00001110
NOT	2999999999	00001120
O'CLOCK	2108099999	00001130
OF	5599999999	00001140
ONE	1621999999	00001150
ONESELF	1621119999	00001160
ONG	4827999999	00001170
OU	5199999999	00001180
OUGHT	1651999999	00001190
OUND	4025999999	00001200
OUNT	4030999999	00001210
OURSELVES	5123391499	00001220
OUT	5199999999	00001230
OW	4299999999	00001240
PAID	1525999999	00001250
PART	1615999999	00001260
PEOPLE	1599999999	00001270
PERCEIVE	1559093999	00001280
PERCEIVING	1559093927	00001290
PERHAPS	1559199999	00001300
QUESTION	1631999999	00001310
QUICK	3105999999	00001320
QUITE	3199999999	00001330
RATHER	2399999999	00001340
RECEIVE	2309399999	00001350
RECEIVING	2309392799	00001360
REJOICE	2326099999	00001370
REJOICING	2326092799	00001380
RIGHT	1623999999	00001390
SAID	1425999999	00001400
SH	4199999999	00001410
SHALL	4199999999	00001420
SHOULD	4125999999	00001430
SIGN	4029999999	00001440
SO	1499999999	00001450
SOME	1614999999	00001460
SPIRIT	5614999999	00001470

```
ST                    1299999999                    00001480
STILL                 1299999999                    00001490
SUCH                  1433999999                    00001500
TH                    5799999999                    00001510
THAT                  3099999999                    00001520
THE                   4699999999                    00001530
THEIR                 5646999999                    00001540
THEMSELVES            4613391499                    00001550
THERE                 1646999999                    00001560
THESE                 2446999999                    00001570
THIS                  5799999999                    00001580
THOSE                 2457999999                    00001590
THROUGH               1657999999                    00001600
THYSELF               5761119999                    00001610
TIME                  1630999999                    00001620
TION                  4829999999                    00001630
TO                    2299999999                    00001640
TODAY                 3025999999                    00001650
TOGETHER              3027239999                    00001660
TOMORROW              3013999999                    00001670
TONIGHT               3029999999                    00001680
UNDER                 1637999999                    00001690
UPON                  2437999999                    00001700
US                    3799999999                    00001710
VERY                  3999999999                    00001720
WAS                   5299999999                    00001730
WERE                  5499999999                    00001740
WH                    4999999999                    00001750
WHERE                 1649999999                    00001760
WHICH                 4999999999                    00001770
WHOSE                 2449999999                    00001780
WILL                  5899999999                    00001790
WITH                  6299999999                    00001800
WORD                  2458999999                    00001810
WORK                  1658999999                    00001820
WORLD                 5658999999                    00001830
WOULD                 5825999999                    00001840
YOU                   6199999999                    00001850
YOUNG                 1661999999                    00001860
YOUR                  6123999999                    00001870
YOURSELF              6123119999                    00001880
YOURSELVES            6123391499                    00001890
  SPACE         14 00 64     CC                     00002600
E               01 17 17     00                     00002700
A               01 01 01     00                     00002800
S               01 14 14     00                     00002900
T               01 30 30     00                     00003000
I               01 10 10     00                     00003100
O               01 21 21     00                     00003200
L               01 07 07     00                     00003300
R               01 23 23     00                     00003400
N               01 29 29     00                     00003500
C               01 09 09     00                     00003600
```

```
D                   01 25 25    00                  00003700
P                   01 15 15    CC                  00003800
M                   01 13 13    CO                  00003900
U                   01 37 37    CO                  00004000
¬ LOGICAL NOT       03 CC 32    CO                  00004100
. PERIODDECIMAL     03 40 40    00                  00004200
B                   01 03 03    CC                  00004300
F                   01 11 11    CO                  00004400
V                   01 39 39    CO                  00004500
G                   C1 27 27    CO                  00004600
W                   01 58 58    CC                  00004700
Y                   01 61 61    00                  00004800
H                   C1 19 19    CC                  00004900
K                   01 05 05    CO                  00005000
, COMMA             03 42 02    00                  00005100
J                   01 26 26    CC                  00005200
Q                   01 31 31    00                  00005300
X                   C1 45 45    00                  00005400
Z                   01 53 53    CC                  00005500
: COLON             03 45 18    CO                  00005600
- HYPHEN (MINUS)    03 36 36    CC                  00005700
1                   02 02 01    CC                  00005800
0                   02 52 26    CO                  00005900
8                   C2 38 19    CO                  00006000
6                   02 22 11    CO                  00006100
2                   02 06 03    CO                  00006200
5                   02 34 17    CO                  00006300
3                   02 18 09    00                  00006400
4                   02 50 25    00                  00006500
9                   02 20 10    CC                  00006600
7                   02 54 27    CO                  00006700
' APOSTROPHE        03 C4 04    01                  00006800
$ DOLLAR SIGN       03 43 50    CO                  00006900
_ ITALICS           13 56 40    01                  00007000
& AMPERSAND         03 47 47    CO                  00007100
" DOUBLE QUOTE      03 CC 52    01                  00007200
* ASTERISK          03 33 20    00                  00007300
% PERCENT           01 41 15    00                  00007400
< LESS THAN         03 35 54    00                  00007500
> GREATER THAN      03 28 54    CC                  00007600
  ACCENT            01 00 08    CO                  00007700
/ SLASH             03 12 12    CC                  00007710
? QUESTION MARK     03 57 38    00                  00007800
  EXCLAMATION PT    03 46 22    CO                  00007900
; SEMICOLON         03 48 06    CO                  00008000
( LEFT PAREN        03 62 54    01                  00008100
) RIGHT PAREN       03 55 54    CC                  00008200
# NUMBER SIGN       03 60 60    CC                  00008300
= LETTER SIGN       C1 63 48    00                  00008500
END ALPHABET        99 99 64                        00008700
 IN I               01 03 64209999                  00008800
 WAS I              01 04 64529999                  00009100
 WERE I             01 05 64549999                  00009200
```

8E |		01	03	64065555	00009300
HIS |		01	04	64385999	00009400
ENOUGH |		01	07	64185555	00009500
_ A|		01	03	64400855	00013410
|		15	01	64999599	00013420
E|		15	01	48179555	00013500
EC |		01	02	43955559	00013600
ECACIOUS|		18	04	17250105	00013700
ECOWN|		01	05	17254229	00013800
EOCM|		01	01	17999559	00013810
EOITION|		18	03	17251099	00013900
ECICT|		18	01	17999999	00014000
EOENTA|		18	01	17999559	00014100
EOUCE|		18	04	17253705	00014200
ECROP|		01	04	17252321	00014400
EO|		01	02	43955959	00014500
ER |		01	02	59955959	00014600
ERA|	L	18	03	17230159	00014700
ERECT|		18	04	17231705	00014800
ERECI|		06	04	17231709	00014900
EROOM|		01	04	17232121	00015000
ERCO|		18	04	17232125	00015100
EROSION|		18	03	17232199	00015200
ERUPT|		18	04	17233715	00015300
ERUP|		06	04	17233715	00015400
ERUC|		06	04	17233705	00015500
ER|		01	02	59959959	00015600
ENCE|		04	04	48179999	00015800
ENESS|		01	05	174E1499	00015900
ENEOI|		01	04	34172599	00016000
ENU|		06	03	17293799	00016100
ENOR|		06	02	17299599	00016110
ENCUN|		06	05	17255129	00016120
EN|	P	06	02	17259959	00016200
EN|		01	02	34955555	00016300
EAR|		01	03	17285955	00016400
EALLY|		01	05	17326199	00016500
EALOGY|		01	04	17010721	00016600
EADE|	P	01	04	17012517	00016700
EADO|		01	04	02252599	00016800
EAX|		01	03	17014599	00016900
EAPP|		01	04	17011515	00017000
EABLE|		01	05	17016099	00017100
EABLY|		01	04	17010307	00017110
EANCE|		01	05	17401799	00017200
EANO|		01	04	17479999	00017300
EAT.CN|		04	06	17322999	00017400
EAWAY|		04	01	17999955	00017410
EA|	L	04	02	02959955	00017500
EVERYONE|		01	05	16176155	00017510
EVERY|	P	06	05	17999959	00017600
EVERS|		01	04	17355555	00017700
EVERT|		06	05	17355530	00017800

```
EVER|            01 04 16179999    00017900
EITHER|          01 06 17109999    00018000
ETHERE|          01 05 17462399    00018100
EQUINOX|         01 04 17313710    00018310
A|               15 01 48019999    00018400
ARIGHT|          01 06 01162399    00018500
AR|              01 02 28999999    00018600
AND THE |        07 04 47999999    00018700
AND A |          07 04 47999999    00018800
AND OF |         07 04 47999999    00018900
AND WITH |       07 04 47999999    00019000
AND FOR |        07 04 47999999    00019100
AND|             01 03 47999999    00019200
ANTENNA|         01 05 01293034    00019300
ANTERIOR|        01 05 01293059    00019400
ANTE|            17 04 01293017    00019500
ANTINOMY|        01 05 01293020    00019600
ANTI|            17 04 01293010    00019700
ANCE|            04 04 40179999    00019800
ANYONE|          06 03 01296199    00020010
ANEMONE|         19 01 01291799    00020020
ATION|           04 05 32299999    00020100
ATHAM|           01 04 01301901    00020200
ASTHMA|          01 05 01145713    00020300
ASST|            01 03 01141499    00020400
AS|            P 06 02 53999999    00020500
ABOUT|           01 05 01039999    00020600
ABOVE|           01 05 01033999    00020700
AB|            P 06 02 48010399    00020750
AGAINST|         01 07 01271299    00020800
AGAIN|           01 05 01279999    00020900
AGERY|           04 04 01271723    00021000
AG|            P 06 02 48012799    00021050
AFTERNOON|       01 09 01112999    00021100
AFTERWARD|       01 09 01115899    00021200
AFTERE|          01 05 01113059    00021300
AFTERI|          01 05 01113059    00021400
AFTER|           01 05 01119999    00021500
AFORE|           17 05 01631799    00021510
AF|            P 06 02 48011199    00021520
ALLY|            04 04 32619999    00021600
ALWAYS|          01 06 01075899    00021700
ALSO|            01 04 01079999    00021800
ALMOST|          01 06 01071399    00021900
ALM|           P 06 03 48010713    00021950
ALREADY|         01 07 01072399    00022000
ALTHOUGH|        01 08 01075799    00022100
ALTOGETHER       01 10 01073099    00022200
ALT|           P 06 03 48010730    00022210
ALONE|         P 04 04 01072129    00022280
AL|            P 06 02 48010799    00022290
ACROSS|          01 06 01092399    00022300
ACCORDING|       01 09 01099999    00022400
```

```
AC|            P 06 02 48010999                                    00022450
AUNDER|          01 04 01372925                                    00022500
AIRE|            01 04 01102317                                    00022600
AINESS|          01 05 01201714                                    00022700
AEO|             01 02 01179999                                    00022800
AEAN|            01 02 01179999                                    00022900
AER|             01 03 01172399                                    00023000
AENOIO|          01 04 01172921                                    00023005
AEN|             06 02 01179999                                    00023010
ADIS|            20 04 01251014                                    00023020
STILL'S|       P 06 07 12041499                                    00023100
STILL|         P 06 05 12999999                                    00023200
STION|           01 05 14402999                                    00023300
STIME|           01 05 14163099                                    00023400
STHOCD|          01 05 12192121                                    00023500
STHEAO|          01 06 12190225                                    00023600
STH|             01 01 14999999                                    00023700
STOWN|           01 02 14309999                                    00023800
STAK|            01 04 14300105                                    00023900
ST|              01 02 12999999                                    00024000
SHALL|         P 06 05 41999999                                    00024100
SHOULDER|        01 06 41510725                                    00024200
SHCULO|          01 06 41259999                                    00024300
SHOUSE|          04 05 14195114                                    00024400
SHORSE|          04 04 14192123                                    00024500
SHOOO|           04 04 14192121                                    00024600
SHEAO|           04 05 14190225                                    00024700
SH|            P 06 02 14199999                                    00024800
SH|              01 02 41999999                                    00024900
SION|            04 04 40299999                                    00025000
SAID|            01 04 14259999                                    00025100
SCMEO|         P 01 05 14211343                                    00025200
SCMETRY|         01 04 14211317                                    00025300
SCMETRIC|        01 04 14211317                                    00025400
SCMETER|         01 04 14211317                                    00025500
SCMER|         L 01 05 14211359                                    00025600
SOME|            01 04 16149999                                    00025700
SO'S|          P 06 04 14041499                                    00025800
SCNE|            01 02 14219999                                    00025900
SO|            P 06 02 14999999                                    00026000
SEVERE|          01 05 14173959                                    00026100
SEVERITY|        01 05 14173959                                    00026200
SEVER|           06 05 14161799                                    00026210
SECATIV|         01 03 14439999                                    00026300
SECAT|           01 03 14172599                                    00026400
SECUC|           01 03 14172599                                    00026500
SECIT|           01 03 14172599                                    00026600
SEOIV|           01 03 14172599                                    00026700
SENOR|         P 06 04 14172921                                    00026800
SEC|           P 06 03 14170986                                    00026900
SPHER|           01 05 14151959                                    00027000
SPIRIT|          01 06 56149999                                    00027100
SUCH|            01 04 14339999                                    00027200
```

```
SUEDE|          01 04 14371725          00027300
SU8|            01 03 14370399          00027400
SSH|            01 02 14149999          00027500
SWCRO|        P 06 04 14582123          00027600
SQUALLY|        01 04 14313701          00027610
TH |            01 02 57999999          00027700
THE |           01 03 46999999          00027800
THERER|         01 06 57595999          00027900
THERED|         01 06 46234359          00028000
THERE|          06 05 16469999          00028100
THEIR|          01 05 56469599          00028200
THESE|        P 01 05 24469559          00028300
THEMSELVES      01 10 46133914          00028400
THENCE|         01 06 57481799          00028500
THEAST|         01 06 57170112          00028600
THEAO|          01 05 30190225          00028700
THEART|         01 05 30191728          00028800
THE|            01 03 46999999          00028900
THAT'LL|      P 06 07 30040707          00029000
THAT'O|       P 06 06 30042599          00029100
THAT|         P 06 04 30999999          00029200
THIS'|          01 05 57101404          00029210
THIS|         P 06 04 57999999          00029300
THILL|          04 04 30191007          00029400
THROUGH|        01 07 16579995          00029500
THOSE|          04 05 24579595          00029600
THOOK|          04 04 30192121          00029700
THOOO|          04 04 30192121          00029800
THORSE|         04 04 30192123          00029900
THOUSE|         04 01 30999995          00030000
THOLE|          04 01 30999595          00030100
THOLO|          04 04 30192107          00030200
THYSELF|        01 07 57611195          00030300
TH|             01 02 57999999          00030400
TO AND FROS     06 06 30216447          00030500
TC |            07 03 22999559          00030600
TOGETHER|       01 08 30272399          00030700
TODAY|          01 05 30259559          00030800
TOMORROW|       01 08 30139959          00030900
TONIGHT|        01 07 30299959          00031000
TION|           04 04 4E299559          00031100
TIMEN|          01 02 30109995          00031400
TIMETER|        04 04 30101317          00031500
TIME|           01 04 16309999          00031600
TWO|            06 03 30582199          00031700
TRINO|          06 04 30231025          00031800
TRED|           01 03 30231799          00031900
TLED|           01 03 30071799          00032000
TLER|         L 01 03 30071795          00032100
TTLEN|          01 04 30300717          00032200
TTLEOR|         01 04 30300717          00032300
II              15 01 48109999          00032400
IST|            20 03 10143099          00032410
```

```
ISCNE|          P 04 C5 10141621                00032420
IATO |            C7 C5 20229999                00032500
IATO|             01 C4 20302199                00032600
INDIS|            17 C5 20251014                00032610
INDIARUP|         01 C5 2025100 1                00032615
INGENT|         P C4 C5 44349999                 00032620
INGENCE|        P 04 C7 44481799                00032625
INGEN|            C4 C5 2027349S                00032630
INGIT|            01 C4 20271099                00032700
INGRAD|           C1 C5 20272201                00032800
INGALE|           01 C5 20270107                00032900
ING|              04 03 44999999                00033000
INESS|            01 05 1041E499                00033100
INER|             18 C3 20179999                00033200
INOMIAL|          01 C4 10292113                00033300
INCHES|         P 22 C6 2086999S                 00033330
INCH|           P 22 C4 20869999                00033350
IN|             P 22 02 20869999                00033360
IN|                04 C2 20999999              00033400
IN|              L C1 C2 2099999S              00033500
ITY|               04 C3 48619999              00033600
ITSELF|            01 06 45119999              00033700
ITS|               01 C3 45149999              00033800
IT'LL|          P C6 C5 450407C7                00033900
IT'S|           P 06 C4 45041499                00034000
IT'D|           P C6 C4 45042599                00034100
IT|             P 06 02 45999999              00034200
IEVER|             01 C5 1017399S              00034300
IETNAMESE|         01 C4 10173029              00034400
IMMECIATE|         01 C9 10131399              00034500
IR|                17 C2 1023999S              00034700
IONE|              01 C2 10219999              00034800
ICRO|              20 C4 1C092321              00034900
OF THE |           C7 C3 55999999              00035000
OF A |             C7 C3 55999999              00035100
OFOR|              01 C4 21639999              00035110
OF|                01 02 55999999              00035200
OUTOIST|           19 04 51302599              00035300
OUT|            P C6 C3 51999999              00035400
OUNO|              04 C4 40259999              00035500
OUNT|              04 C4 40309999              00035600
OUNCES|         P 22 C6 21538699                 00035640
OUNCE|          P 22 C5 21538699                00035660
OUGHTO|            C1 C5 51353099              00035690
OUGHT|             01 C5 16519999              00035700
OURSELVES|         01 C9 51233914              00035800
OUI|               01 C2 51999999              00035900
OWORK|             01 05 21165899              00036000
OW|                C1 C2 42999999              00036100
ONGRU|             01 04 21292723              00036110
ONG|               04 03 48279999              00036200
ONENESS|           C1 C7 16214814              00036300
ONEN|              01 C2 21299999              00036400
```

```
ONER|          01 04 21295999          00036500
ONED|          01 04 21294399          00036600
ONEG|          06 01 21999999          00036610
ONEI|          06 01 21999999          00036620
ONEISM|      P 01 04 21291710          00036630
ONESS|       P 01 05 21481499          00036700
ONESELF|       01 07 16211199          00036800
ONESE|       P 01 04 21291714          00036900
ONESIA|        01 04 21291714          00037000
ONEST|       P 01 05 21291712          00037100
ONEE|          01 03 21291799          00037200
ONEOUS|        01 05 21291751          00037300
ONEA|          01 02 21299999          00037400
ONEY|        P 01 04 21291761          00037500
ONEUM|         01 04 21291737          00037600
ONET|        P 01 04 21291730          00037700
ONETTE|        01 04 21291730          00037710
ONETS|       P 01 04 21291730          00037720
ONEL|        P 04 01 21999999          00037730
ONE|           01 03 16219999          00037800
OIN|           01 01 21999999          00037900
OII|           01 02 21109999          00038000
OGGON'|        01 05 21542129          00038100
OGGO|          10 03 21272799          00038200
OSCME|         01 04 21142113          00038300
OSST|          01 03 21141499          00038400
OEN|           01 03 21172999          00038500
OED|         L 01 03 21172599          00038600
O'CLOCK|       01 07 21040999          00038700
ORSER|         20 04 21231417          00038800
OVER|          17 04 21395999          00038900
OONE|          01 03 21212999          00039000
OLEA|          06 03 21071799          00039010
OZ|          P 22 02 21538699          00039015
LEDY|          01 03 07172599          00039020
LESS|          04 04 40149999          00039100
LETTER|        18 06 07239999          00039200
LIKE|        P 06 04 07999999          00039300
LITTLE|        01 06 07079999          00039400
LINES|       P 22 05 07869999          00039450
LORO|          01 04 16079999          00039500
LAHAO|         01 05 07015619          00039600
LBI            22 02 07038699          00039700
RIGHT|         01 05 16239999          00039800
RATHER|      P 06 06 23999999          00039900
RAFTER|        01 04 23011120          00040000
RAREO|       P 01 05 23012343          00040100
RANSOME|       06 04 23012914          00040110
RTIMER|        01 04 23301013          00040200
REOOU|         18 03 23172599          00040300
REOU|          18 04 23172537          00040400
REOE|          18 02 23179999          00040500
REDAC|         06 02 23179999          00040510
```

```
RECRA|          C6 C2 23179999      00040520
RECRESS|        18 04 23172523      00040600
REDIST|         19 03 23172999      00040700
REDIN|          18 03 23439999      00040800
REDI|           18 02 23179999      00040900
REACT|          01 C2 23179999      00041000
REACH|          01 03 23029999      00041100
REAC|           C6 02 23179999      00041200
READJ|          01 C2 23179999      00041300
READD|          18 05 23170150      00041400
READM|          06 02 23179999      00041500
READD|          06 C2 23179999      00041600
READV|          06 02 23179999      00041700
REAF|           C1 C2 23179999      00041800
REAG|           06 C2 23179999      00041900
REAB|           01 02 23179999      00042000
REALINE|        C6 C2 23179999      00042100
REALL|          06 C2 23179999      00042200
REAN|           06 C2 23179999      00042300
REAPP|          06 C2 23179999      00042400
REASC|          06 C2 23179999      00042500
REASS|          C6 C2 23179999      00042600
REAWAKE|        01 02 23179999      00042700
REATT|          06 C2 23179999      00042800
REATION|        01 02 23179999      00042810
REAR|           01 02 23179999      00042820
REAV|           06 02 23179999      00042830
REA|            C6 C3 23029999      00042840
RER|            18 C2 23179999      00042900
RENAS|          C6 C2 23179999      00043010
RENAM|          06 02 23179999      00043100
RENAV|          06 C2 23179999      00043200
RENEGA|         06 C3 23349999      00043300
RENE|           06 02 23179999      00043400
RENOV|          C6 C3 23349999      00043500
RENO|           06 C2 23179999      00043600
RENU|           06 C2 23179999      00043610
REN|            C6 C3 23349999      00043620
REVEREN|        01 05 23161799      00043700
REVERIE|        C1 C5 23161799      00043800
REVERY|         01 C6 23161761      00043900
REVER|          C6 C5 23173999      00044000
REJOICE|        01 C7 23260999      00044100
REJOICING|      01 C9 23260927      00044200
RECEIVE|        01 C7 23093999      00044300
RECEIVING|      C1 C9 23093927      00044400
REST|           06 C4 23171299      00044410
NIGHT|          17 C5 29103530      00044500
NOT|          P C6 C3 29999999      00044600
NONE|         P 06 C4 29162199      00044610
NONESS|         C6 C3 29212999      00044615
NONES|          C6 C1 29999999      00044620
NON|            17 C3 29212999      00044700
```

```
NCWISE|          01 04 29215810                00044800
NOWAY|           01 C4 292158C1                00044900
NOWHERE|         01 C2 29219959                00045000
NAMENT|          01 02 29019959                00045010
NAME|            01 C4 16299555                00045100
NESS|            04 04 48149999                00045200
NEITHER|         C1 C7 29171C99                00045300
NECESSARY|       01 05 29170999                00045400
NEMONE|          20 06 13212917                00045410
N8LE|            01 04 29030717                00045500
NGHA|            01 03 29271999                00045600
NDIS|            20 C4 29251014                00045610
CI|              15 01 48095955                00045700
CHILDREN|        01 08 33299999                00045800
CHILD'S|       P 06 C7 33041495                00045900
CHILD|         P 06 05 33999959                00046000
CHARACTER|       01 09 16339999                00046100
CHS|             01 03 33149555                00046200
CHI|             01 02 33999959                00046300
CCM'ERE|         01 04 092113C4                00046400
CCMIN'|          C6 C6 3E1029C4                00046500
CCMI|          L 06 C3 36999999                00046600
CONEST|          C6 C3 18999999                00046700
CONEL|           06 C3 18999959                00046800
CONE|            01 01 09999999                00046900
CONIC|           06 C5 1810C955                00047000
CCNI|            01 04 09212910                00047100
CONO|            06 04 09212921                00047110
CCNN|          P 06 C3 09212999                00047200
CONNEO|          06 03 09212999                00047210
CONNING|         C6 C3 09212999                00047220
CCNATIVE|        06 06 18013010                00047300
CONA|            01 03 09212999                00047400
CONGE|           01 04 09212927                00047410
CONU|            01 C3 09212995                00047500
CCNY|            01 04 092129E1                00047600
CONK|            01 04 092129C5                00047700
CCNCEIVING       06 10 18093927                00047710
CONCEIVE|        C6 C8 18093999                00047720
CCNCH|         P 01 05 09212933                00047800
CONTRAOIS|       19 05 18302399                00047810
CONTE|         P 0E C4 C9212930                00047820
CONI|          L 06 03 18999999                00047900
COULO|           01 C5 09259999                00048000
CCLONEL|         01 04 09210721                00048100
COENZYME|        01 04 09213499                00048200
CAN'T|         P C6 C5 09043099                00048300
CAN'S|         P C6 05 09041495                00048400
CANNOT|          01 C6 56099999                00048500
CAN|           P 06 03 C9999955                00048600
CATION|          06 04 09013010                00048700
CCH|             01 03 C9339995                00048800
CCI|           L 04 02 18959559                00048900
```

```
CITIZENESS        19 C4 09103C10  00049000
CENTS|          P 22 05 05869999  00049040
CENT|           P 22 04 05869999  00049060
CM|             P 22 02 09138699  00049080
0|                15 01 48259999  000491C0
DAY|              01 C3 16259999  00049200
00|             P 06 02 25999999  000493C0
OOLLARS|        P 22 C7 25869999  00049340
OCZEN|          P 22 05 25538699  00049360
DISHC|            18 04 25104199  000494C0
DISHEV|           18 04 25104199  000495C0
OISH|           P 01 04 251C4199  00049600
DISK|             18 04 251C14C5  000497C0
OISC|           P C6 C4 25101409  00049830
OISPIRIT|         01 C8 25105614  00049900
OISULPH|          C6 C2 25109999  00049910
OISTANCE|         04 04 25101430  00049920
OIS|            L C6 C3 50999999  000500C0
DINGHY|           06 06 252C35E1  00050100
DDAY|             01 04 25162599  00050200
00|             L 04 C2 5C999999  000503C0
OECEIVE|          01 C7 25C53999  000504C0
DECEIVING|        01 09 25093927  000505C0
OECLARING|        01 C9 25090727  000506C0
DECLARE|          01 C7 25090799  00050700
OERIVATICN        01 C5 25591039  000508C0
OERE|             C6 C3 25599999  00050810
OERM|             06 C3 25599999  00050820
OEROGATE|         01 C3 25599999  00050830
DEROGATICN        01 C3 25599999  00050835
OEROGATING        01 03 25599999  00050836
OERR|             C6 C3 25599999  00050840
DERV|             06 C3 25599999  00050850
DER|            L 06 C2 25179999  00050900
OEO|            L 06 C2 25179999  00051100
OENAT|            C6 C2 25179999  00051200
OENIT|            C6 C2 25179999  00051300
OENO|             C6 C2 25179999  00051400
DENUDE|           06 C2 25179999  000515C0
OENUN|            C6 C2 25179999  000516C0
OENESS|         P 01 02 25179999  00051700
OENCE|          P 04 05 25481799  00051710
OEN|              01 C3 25349999  000518C0
DESHABILLE        C6 C4 25171419  00052C00
OEAW|             01 04 25170158  000521C0
OEAR|             01 04 25172899  00052110
OEA|              18 03 25029999  00052120
OEBB|             C1 C4 2517C699  000522C0
OEMENT|           06 C6 25174830  00052210
OEPREO|           06 C4 25171523  00052300
OEVER|            C1 05 25161799  000523C0
DEITY|            06 C5 25174E61  00052320
OE|               17 02 25179999  00052400
```

```
PHONES|          01 05 15191621                00052500
PHONE|         L 01 04 15192129                00052510
PHONE|           01 05 15191621                00052520
PHOTO|           19 01 15999999                00052530
PARTO|           06 03 15289999                00052600
PARTAK|          01 03 15269999                00052700
PART|            01 04 16159999                00052800
PARER|           01 04 15281799                00052810
PAID|            01 04 15259999                00052900
PAGES|         P 22 05 15869999                00052950
PEOPLE'S|      P 06 08 15041499                00053000
PEOPLE|        P 06 06 15999999                00053100
PERHAPS|         01 07 15591995                00053200
PERCEIVE|        01 08 15590939                00053300
PERCEIVING       01 09 15590939                00053400
PER CENT|      P 22 08 18158699                00053402
PERIN|           06 04 15591099                00053405
PEST|            01 04 15171299                00053410
PEATER|        P 04 04 15170130                00053420
PREDATOR|        01 05 152343C1                00053500
PREDECES|        06 04 15234399                00053510
PREDICAM|        18 04 15231725                00053600
PREDICA|         01 05 1523431C                00053700
PREACH|          01 04 15230299                00053800
PRENTICE|        06 05 15233430                00053810
PRE|             17 03 15231799                00053900
PROFF|           01 05 15235511                00054000
PROFIT|          01 04 15235599                00054100
PROFL|           01 05 15235507                00054200
PROFANATIO       06 04 15235559                00054210
PROF|          L 06 04 15232111                00054300
PROUN|           06 04 15232137                00054310
POSTIO|          01 03 15211499                00054400
POST|            01 04 15211299                00054500
POUNOS|        P 22 06 07038699                00054540
POUND|         P 22 05 07038699                00054560
PINTS|         P 22 05 15308699                00054570
PINT|          P 22 04 153C8699                00054580
PT|            P 22 02 153C8699                00054582
PPI            P 22 02 15869999                00054584
MENT|            04 04 483C9999                00054600
MENING|          06 06 13342027                00054610
METERS|        P 22 06 13308699                00054700
METER|         P 22 05 133C8699                00054800
MAMA|            06 04 13011901                00054900
MALE|            17 04 13010717                00054910
MANY|            01 04 56139999                00055000
MORE'|           06 04 13212317                00055010
MORE|          P 06 04 13999999                00055100
MOTHER|          01 06 16139999                00055200
MONET|           06 05 13162130                00055300
MONEY|           01 05 13162161                00055400
MONGCO|          01 04 13212927                00055500
```

```
MISTERM|           18 04 13101430          00055600
MISTEM|            18 04 13101430          00055700
MISTEACH|          18 04 13101430          00055710
MISTELL|           18 04 13101430          00055720
MISTA|             18 04 13101430          00055800
MISTIT|            18 04 13101430          00055900
MISTIM|            18 03 13101459          00055910
MISTCC|            18 04 13101430          00056000
MISTRIAL|          18 04 13101430          00056100
MISTREA|           18 04 13101430          00056200
MISTRU|            18 04 13101430          00056210
MISTRAIN|          18 04 13101430          00056220
MISTRANS|          18 04 13101430          00056230
MISTH|             18 03 13101459          00056240
MIST|              01 04 13101299          00056300
MISERABLE|         01 05 13101459          00056310
MISSION|           18 03 13101499          00056400
MIS|               17 03 13101459          00056500
MILES|           P 22 05 13869999          00056600
MILE|            P 22 04 13869999          00056700
MI|              P 22 02 13869999          00056750
MILEAGE|           01 04 13100717          00056800
MILLIMETER         22 10 13138699          00056900
MINUTES|         P 22 07 13208699          00057100
MINUTE|          P 22 06 13208699          00057200
MIN|             P 06 03 13208699          00057300
MICRO|             19 01 13999999          00057400
MUCH|              01 04 13339999          00057500
MLSTN'T|           01 06 13122904          00057600
MLST|            P 01 04 13129999          00057700
MYSELF|            01 06 13611199          00057800
MM|              P 22 03 13138699          00057850
UNDERI|            06 04 37292517          00057900
UNDERO|            06 04.37292517          00058000
UNDER|             17 05 16379999          00058100
UNDER|             01 05 16379999          00058200
UNDIS|             19 01 37999999          00058210
UNEA|              06 03 37291759          00058300
UNLESS|          P 06 06 37294014          00058400
UNITY|             18 05 37294E61          00058500
UN|                17 02 37299999          00058600
US|              P 06 02 37999999          00058700
UPON|              01 04 24379999          00058800
URONE|             01 05 37231621          00058810
¬CONN.|            01 06 32182990          00058900
¬OONG|           P 01 04 32092129          00058910
¬ENOUGH |          01 07 32189999          00059000
¬IN |              01 03 32209999          00059100
¬WAS |             01 04 32529999          00059200
¬WERE |            01 05 32549999          00059300
¬SAID|             01 04 32140110          00059310
¬HIS |             01 04 32389999          00059400
¬BE |              01 03 32069999          00059500
```

```
¬GOOO|          C6 C2 32279999
¬PART|          C6 02 32159999
¬¬US|        P 01 03 32323799
.  |            14 C3 50006499
...|            03 C3 C404C499
.|              04 01 5C999999
B|              15 01 4B039999
BUT|         P 06 C3 C3999999
BBLE|           01 C4 C36C9999
BBANO|          01 C5 03034799
BR|          L 04 C2 06999999
BEB|            06 02 06999999
BEFORE|         01 06 06119999
BEF|            06 02 06999999
BECAUSE|        01 C7 06099999
BECA|           06 02 06999999
BECO|           06 02 C6999999
BECL|           06 02 06999999
BECU|           C6 C2 06999999
BETHC|          06 02 06999999
BETH|           C1 02 03179999
BETT|           01 C2 03179999
BETWEEN|        01 C7 063C9999
BET|         L 06 02 06999999
BEHINO|         01 C6 C6199999
BEH|            06 02 06999999
BESIDE|         01 C6 C6149999
BEST|        P 01 02 03179999
BESTIA|         01 02 03179999
BESTEO|         01 02 03179999
BESS|           01 03 03171499
BES|            06 02 06999999
BEYOND|         01 06 06619999
BELOW|          01 C5 06079999
BELO|           C6 02 C6999999
BELA|           06 02 06999999
BELE|           C6 C2 06999999
BELY|           06 C2 06999999
BELU|           06 C2 06999999
BELI|           06 C2 06999999
BEATING|        06 C1 03999999
BEATR|          06 02 06999999
BEAT|        L C6 C2 06999999
BEGG|           01 04 03175499
BEG|         L 06 C2 06999999
BENEATH|        01 07 06259999
BENEV|          06 02 06999999
BENEFICEN|      06 C2 06999999
BENE|        P 06 C4 06291799
BENA|           06 C2 06999999
BENIG|          06 C2 06999999
BEOA|           06 C2 06999999
BECEC|          06 C2 06999999
```

```
00059510
00059520
00059600
00059700
000598C0
000599C0
00060000
00060100
00060200
000603C0
00060400
00060500
00060800
00060900
00061000
00061010
00061020
00061030
00061040
00061050
00061100
00061110
00061200
00061300
00061400
00061500
00061600
00061610
00061620
00061630
00061700
00061800
00061900
00062000
00062005
00062010
00062100
00062200
00062220
00062230
00062300
00062310
00062320
00062400
00062500
00062600
000627C0
00062800
00062805
00062810
00062820
00062900
00063000
```

BEDEV|		06	02	06999999	00063100
BEDEA|		06	02	06999999	00063110
BEDEL|		06	02	06999999	00063120
BEOI|		06	02	06999999	00063200
BECOG|		06	02	06999999	00063300
BEDRA|		06	02	06999999	00063400
BEING|		06	05	06449999	00063410
BEIN'|		06	05	06102904	00063500
BERA|		06	02	06999999	00063510
BERET|		06	02	06999999	00063520
BEREA|		06	02	06999999	00063600
BEREF|		06	02	06999999	00063700
BEC|		06	02	06999999	00063800
BEW|		06	02	06999999	00063900
BEJ|		06	02	06999999	00063920
BEM|		06	02	06999999	00063930
BED|		06	02	06999999	00063940
BLESS|		01	05	03401499	00064000
BLENOE|		01	04	03073499	00064100
BLEEO|		01	05	03071743	00064200
BLEAU|	P	04	05	03070237	00064210
BLE|		04	03	60999999	00064300
BLINOE|		01	04	03072099	00064400
BLINOI|		01	04	03072099	00064500
BLINCA|		01	04	03072099	00064600
BLIND|		01	05	03079999	00064700
BLUENESS|		18	04	03073717	00064800
BLUE|		17	04	03073717	00064900
BY AND BY|	P	19	01	03616447	00065000
BY AND|		07	03	03616499	00065010
BY THE BY|		06	08	52466403	00065100
BY |		07	03	52999999	00065300
BRAILLE|		01	07	03230799	00065400
F|		15	01	48119999	00065500
FOR THE |		07	04	63999999	00065600
FOR A |		07	04	63999999	00065700
FOREVER|		06	07	63161799	00065710
FORENS|		06	03	63999999	00065720
FORE|		06	04	63179999	00065800
FOR|		01	03	63999999	00065900
FOOT|	P	22	04	11308699	00065950
FRUIT|		01	04	11233710	00066000
FRIENOE|		01	05	11231034	00066100
FRIENOI|		01	05	11231034	00066200
FRIEND|		01	06	11239999	00066300
FROM|	P	06	04	11999999	00066400
FROWA|		06	03	11232199	00066410
FREEDOM|		06	04	11231717	00066500
FIRST|		01	05	11129999	00066600
FIANC|	L	06	04	11100129	00066610
FULL|	P	01	04	11370707	00066700
FUL|		04	03	48079999	00066800
FFOR|		01	04	11639999	00066900

```
FF|              L 04 02 22999999    00067000
FATHER|            01 06 16119999    00067100
FEVER|             01 05 11173999    00067200
FEATER|          P 04 04 11170130    00067210
FEET|            P 22 04 11308699    00067300
FLERY|           P 04 04 11071723    00067400
FT|              P 22 02 11308699    00067450
VERY|            P 06 04 39999999    00067500
VICEN|             06 05 39100934    00067510
VICE|              06 04 39100917    00067520
VALEO|             06 04 39010717    00067600
G|                 15 01 48279999    00067700
GHAM|              01 04 27190113    00067800
GHOUSE|            01 05 27195114    00067900
GHOR|              01 04 27192123    00068000
GHEART|            01 05 27190223    00068100
GHILL|             01 04 27191007    00068200
GH|                01 02 35999999    00068300
GOODAMN|           06 04 27212525    00068310
GOOD|              01 04 27259999    00068400
GO'|               06 03 27210499    00068410
GOSHAWK|           01 04 27211419    00068420
GO|              P 06 02 27999999    00068500
GG|              L 04 02 54999999    00068600
GREAT|             01 05 27233099    00068700
GREA|              01 04 27230299    00068800
GRAMS|           P 22 05 27238699    00068802
GRAM|            P 22 04 27238699    00068803
GEATER|          P 04 04 27170130    00068805
WAF|               01 02 58019999    00068810
WITH THE |         07 05 62999999    00068900
WITH A |           07 05 62999999    00069000
WITH|              01 04 62999999    00069100
WILL'S|          P 06 06 58041499    00069200
WILL|            P 06 04 58999999    00069300
WHICH'|            06 06 49103304    00069310
WHICH|           P 06 05 49999999    00069400
WHEREVER|          01 08 49991617    00069410
WHERE'|            01 06 49591704    00069420
WHERE|             01 05 16499999    00069500
WHOSE|             01 05 24499599    00069700
WH|                01 02 49999999    00069800
WOULO|             01 05 58259995    00069900
WORK|              01 04 16589999    00070000
WORO|              01 04 24569999    00070100
WORLO|             01 05 56589995    00070200
WEETHEART|         01 04 58171730    00070300
WEEVER|            01 04 58171739    00070400
WREA|              01 04 58230299    00070500
YOUNG|             01 05 16619999    00070600
YOURSELF|          01 08 61231199    00070700
YOURSELVES         01 10 61233914    00070800
YOUR|              01 04 61239999    00070900
```

```
YOL'RE|        P C6 C6 61042317        00071000
YOL'LL|        P 06 C6 61040707        00071100
YOL'VE|        P 06 C6 61043917        00071200
YOU'D|           C6 C5 61042595        00071210
YOU'|            06 04 61510459        00071220
YOU|           P 06 C3 61999999        00071300
YONE|            01 02 61219999        00071310
Y AND BY|        20 C8 64036199        00071400
YAROS|         P 22 C5 61258659        00071450
YARD|          P 22 04 61258655        00071460
YORO|            20 04 61252321        00071470
YO|              22 C2 61258699        00071480
H|               15 01 48199999        00071500
HAOO|            01 C4 19015099        00071600
HADE|            06 04 19012517        00071700
HAO|             06 C3 56199995        00071800
HAVE|          P 06 04 19999955        00071900
HIMSELF|         01 07 19131199        00072000
HIM|             01 C3 19139999        00072100
HEART|           01 C4 19172859        00072110
HERESY|          01 03 19599999        00072200
HERETI|          01 04 19591799        00072300
HEREN|           01 03 19599955        00072400
HEREO|           01 03 19599999        00072500
HEREF|           01 03 19559959        00072510
HERER|           01 C3 19599955        00072600
HERE|            01 04 16199999        00072700
HERSELF|         01 C7 19591199        00072800
HYDRO|           19 01 19999999        00072810
HORSER|          19 01 19999999        00072900
HOUSER|          18 05 19511417        00073000
HONEY|           01 05 19162161        00073100
HOTO|            20 04 19213021        00073110
HM|            P 06 C2 48191395        00073150
KNOWLEOGE|     P 06 09 05999999        00073200
KNOW|            01 C4 16059559        00073300
KILO|            17 04 05100721        00073400
,|               03 01 02999959        00073500
J|               15 01 48269555        00073600
JUST|          P 06 04 26999559        00073700
QUICK|           01 05 31059599        00074000
QUITE|         P C6 05 31999599        00074100
QUESTION|        01 08 16319999        00074200
QUEO|            01 04 31371725        00074300
QU|              01 02 31379999        00074400
X|               01 01 45999999        00074500
ZENESS|          20 05 53341714        00074600
Z|               01 01 53999999        00074700
: |              03 02 18006459        00074800
-COM|            01 02 36099999        00074900
-BY|             01 C3 36036199        00075000
--|              14 C2 36369999        00075100
1|               22 00 85649999        00075190
```

```
1|              02 01 60018499                    00075200
0|              22 00 85649999                    00075290
0|              02 C1 60268499                    00075300
8|              22 00 85649999                    00075390
8|              02 C1 60198499                    00075400
6|              22 C0 85649999                    00075490
6|              02 01 60118499                    00075500
2|              22 00 85649999                    00075590
2|              02 01 60038499                    00075600
5|              22 00 85649959                    00075690
5|              02 01 60178499                    00075700
3|              22 00 85649959                    00075790
3|              02 C1 60058499                    00075800
4|              22 00 85649999                    00075890
4|              02 01 60258499                    00075900
9|              22 00 85649999                    00075990
9|              02 01 60108499                    00076000
7|              22 C0 85649999                    00076090
7|              02 01 60278499                    00076100
'ER|          P 06 C3 04172399                    00076200
'OU|            04 C3 04519999                    00076300
'CCM|           01 02 04099959                    00076400
'|              04 01 04999999                    00076500
$LV|            01 03 56999999                    00076600
$L|             03 02 64656499                    00076700
$PG|            03 03 64696499                    00076800
$PTYE|          03 05 64966499                    00076900
$PTYS|          03 05 64956499                    00077000
$P"|            08 03 67643899                    00077100
$P|             08 02 67643899                    00077200
$P|             09 02 67644040                    00077300
$P|             03 02 64676499                    00077400
$G|             05 02 99999999                    00077500
$TAB|           03 04 64686499                    00077600
$TLE|           03 04 89649999                    00077700
$TLS |          03 05 64889999                    00077800
$TLS|           03 04 64889599                    00077810
$T|             01 02 32049959                    00077900
$'R|            03 03 52049999                    00078000
$'|             03 02 32389599                    00078100
$"R|            03 03 52999999                    00078200
$"|             03 02 38999959                    00078300
$/|             01 02 99959599                    00078400
$8|             03 02 00999999                    00078500
$HDS |          03 05 64819999                    00078600
$HDS|           03 04 64819999                    00078610
$HOE|           03 04 82649959                    00078700
$SL|            01 03 64716499                    00078800
$SV|            01 03 24999999                    00078900
$STB|           01 04 64936499                    00079000
$SCON|          01 05 64929959                    00079010
$SCOFF|         01 06 64919999                    00079020
$FT|            01 03 36959999                    00079100
```

```
$CP8|            C1 04 64579999
$CS|             01 03 36369999
$#|              01 02 64729999
$DCT|            C1 C4 64949999
__|              12 02 40409999
_EN|           P C6 C3 40172999
_ENOUGH_|        01 C7 40189999
_IN_|            01 C3 40209999
_WAS_|           C1 C4 40529999
_WERE_|          01 C5 40549999
_HIS_|           01 04 40389999
_BE_|            01 03 40065999
_/|              01 02 99999999
&#"|             C8 C3 67389999
&#|              C8 02 67389999
&#|              C9 C2 67404099
&#|              03 02 67999999
"|               10 01 38999999
"|               11 01 52999999
*|               03 C1 20209999
%|               03 01 18159999
<|               C3 C1 32549999
>|               03 C1 54049999
 |               15 C1 64058699
 EN|             01 03 08172999
 ER|             01 03 08172399
 EO|           L 01 03 08172599
 CNG|            01 02 08219999
 AR|             C1 03 08012399
/_ENOUGH_/       C1 10 34999999
/_EN_/|          01 C6 34999959
/_BE_/|          01 06 06999999
/_BY_/|          01 06 52999599
/_WAS_/|         01 07 52959999
/_WERE_/         01 C8 54999999
/_HIS_/|         01 C7 38999959
/_LETTER_/       01 10 07239999
/_INTO_/|        01 C8 20229955
/_TO_/|          01 06 22999959
/_SH_/|          01 06 41999999
/_|              21 02 99999999
END OF TABLE     99 C0 99999999
                 02          NNT
               P          14030213
               L          01131313
NSV              C8
NIC              22
RS-R-RRRRRR--RRRRRRRR
SRRS-SRRRRRRRRRSSSSRS
----T----------------
---------SR----------
----------SR---------
SRRS-SRRRRRRRRRRS-SSS
```

```
00079200
00079300
00079400
00079500
00079600
00079610
00079700
00079800
00079900
00080000
00080100
00080200
00080210
00080300
00080400
00080500
00080600
00080700
00080800
00080900
00081000
00081100
00081200
00081205
00081210
00081220
00081230
00081240
00081250
00081251
00081252
00081253
00081254
00081255
00081256
00081257
00081258
00081259
00081260
00081261
00081280
00081300
00081400
00081500
00081600
00081700
00081800
00081900
00082000
00082100
00082200
00082300
00082400
```

```
RRRRRRRRRRRRRRRRRSRRR                                    00082500
RRRRRRRRRRRRRRSRRRRRR                                    00082550
NDTC                11                                   00082600
Y-G-G------GGG-Y----G---N---N-                           00082700
F--Y----------F--------N--YN-                            00082800
-Y-F----------N-------N-------                           00082900
-F---YY-------G---Y----NN---N-                           00083000
-----FF---------YYF-----N--NN-                           00083100
-------Y--------FF-------YN---                           00083200
-------FY---------------NY---                           00083300
--------FY---------------N----                          00083400
---------FY-------------Y----                           00083500
----------F--------Y--------Y-                          00083600
------------------F-Y--N---NY                           00083650
.      A                                                 00083700
  .    .                                                 00083800
. .    8                                                 00083900
    .  '                                                 00084000
.    . K                                                 00084100
  . . :                                                  00084200
. . . L                                                  00084300
 .                                                       00084400
..     C                                                 00084500
 ..    I                                                 00084600
...    F                                                 00084700
 .  .  ST                                                00084800
..   . M                                                 00084900
 .. . S                                                  00085000
... . P                                                  00085100
    .    5                                               00085200
.  .   E                                                 00085300
  ..   :                                                 00085400
. ..   H                                                 00085500
   ..  IN                                                00085600
.  ..  O                                                 00085700
  ...  TO                                                00085800
. ...  R                                                 00085900
 . .   45                                                00086000
.. .   D                                                 00086100
 ...   J                                                 00086200
....   G                                                 00086300
 . ..  AR                                                00086400
.. ..  N                                                 00086500
 ....  T                                                 00086600
..... O                                                  00086700
     . ¬                                                 00086800
.     .CH                                                00086900
  .   .EN                                                00087000
. .   .GH                                                00087100
    ..-                                                  00087200
.   ..U                                                  00087300
  . ..  ?                                                00087400
. . ..V                                                  00087500
```

```
 .   . _                                                                  00087600
..   .SH                                                                  00087700
 ..  .OW                                                                  00087800
...  .ED                                                                  00087900
 .  ..ING                                                                 00088000
..  .. X                                                                 00088100
 .. ..THE                                                                 00088200
... ..AND                                                                 00088300
   . . =                                                                  00088400
.  . .WH                                                                 00088500
  .. ..                                                                  00088600
. .. .DL                                                                 00088700
   ...'                                                                  00088800
.  ...7                                                                  00088900
  ....)(                                                                 00089000
. ....OF                                                                 00089100
 . . .456                                                                00089200
.. . .TH                                                                 00089300
 ... .W                                                                  00089400
.... .ER                                                                 00089500
 . ... #                                                                 00089600
.. ...Y                                                                  00089700
 .....WTH                                                                00089800
......FOR                                                                00089900
                  LAST SIGN TABLE CARD                                   00090000
PROOF                                                                    00100100
NO BRAILLE                                                               00100200
NO PPQ                                                                   00100300
NO PUNCHED OUTPUT                                                        00100400
SIGNS/LINE       38                                                      00100500
LINES/PAGE       09                                                      00100600
PAGINATION                        0001                                   00100700
$TLS ¬SAMPLE ¬¬OCTSYS ¬RUN $TLE $PG $P ¬THIS TEST HAS THREE PARTS: $P
  (1)  REALISTIC RUNNING TEXT;       $P
  (2)  SPECIALLY CONCOCTED EXERCISES OF THE CONTROL SYMBOLS AND OTHER FE
ATURES; AND  $P
  (3)  THE FIRST PAGE OF THE APPROXIMATELY 5800 PROBLEM WORDS FROM THE
¬TRANSCRIBERS' ¬GUIDE, TRANSLATED AND AUTOMATICALLY FLAGGED, IF WRONG,
BY THE SELF-CHECKING FEATURE.
$TLS ¬¬BUSINESS ¬¬DATA ¬¬PROCESSING$TLE
 $PG $HDS ¬EMBOSSED IN ¬BRAILLE $HDE $L $HDS ON AN ¬¬IBM ¬¬SYSTEM/360
COMPUTER $HDE $HDSAT THE ¬ATLANTA ¬BOARD OF ¬EDUCATION $HDE $PG $P $P
¬FRONT ¬COVER $P __¬WHAT IS DATA _PROCESSING? $P __¬WHY HAS IT CAUSED
A VIRTUAL REVOLUTION IN THE ADMINISTRATION OF MODERN _BUSINESS? $P
__¬WHAT ARE THE TWO MAJOR MODERN PROCESSING _SYSTEMS? $P __¬WHAT
PROBLEMS DO BUSINESSES FACE IN INITIATING MODERN DATA PROCESSING INTO
THE BUSINESS _COMPLEX? $P ¬¬BUSINESS ¬¬DATA ¬¬PROCESSING, ¬SECOND
¬EDITION BY ¬ELIAS =¬M. ¬AWAD, ANSWERS ALL THESE QUESTIONS SIMPLY AND
COMPLETELY.  ¬IF YOUR POSITION REQUIRES YOU TO KNOW THE FUNDAMENTALS OF
MODERN DATA PROCESSING (ITS PRINCIPLES AND ITS EQUIPMENT), THIS
UP-TO-DATE BOOK IS ESSENTIAL READING FOR YOU.  ¬IT OFFERS SO
COMPREHENSIVE A TREATMENT OF
THE FIELD THAT YOU CAN EASILY ACQUIRE A FULL NONTECHNICAL MASTERY OF
```

```
MODERN DATA PROCESSING PRINCIPLES AND METHDOS, WITH NO PRIOR BACKGROUND
$TLS SAMPLE TITLE$TLE                                                   00090800
$PG                                                                     00090900
THE 36 BOYS MEASURED OFF 24 INCHES AND 36 MILES WITH THEIR 12 SLIDE     00091000
RULES.  THE 2 BOYS LIFTED 53 POUNDS 1 FOOT 13 METERS IN THE AIR.        00091100
$PG $TAB06 ¬SAMPLE ¬RUN $P ¬ALL OF THE SPECIAL SYMBOLS,                 00091200
AS DESCRIBED IN __¬¬DOTSYS ¬¬II ¬USER'S _¬GUIDE <¬KEYPUNCH ¬INSTRUCTIONS00091300
> ARE USED.  ¬HE SAID, "I SAID, $'THE SIGN FOR =K IS GIVEN THE PROOF    00091400
SYMBOL =K EVEN WHEN IT REPRESENTS THE WORD $"$GKNOWLEDGE$G. $"R$'R"     00091500
¬THE EDITOR MAY WISH TO USE THE TERMI$/NATOR$T.                         00091600
$TLS $TLE                                                               00091700
      (A)  'B'  "C"  (1)  'THIS'  "WHY NOT" (K)                         00091800
 (_=¬ A) '= C' "_HS" _= 0 _¬K '=¬ K'     (1)                            00091900
$OCT1152536372527000036721021707012
$TAB14                                                                  00092100
$CPBEASTIOLRNCOPMU¬.BFVG                                                00092200
$STB1R14                                                                00092300
$STB2L05                                                                00092400
$STB3D09                                                                00092500
$#3 456.503                                                             00092600
$#3 4560                                                                00092700
$#2 TESTLEFT                                                            00092800
$#1 TESTRIGHT                                                           00092900
$STB4R16$STB5R15$STB6R14$#4TESTING
$SL14                                                                   00093000
$TLS SECOND SAMPLE TITLE$TLE                                            00093100
$HDSSAMPLE HEADING$HDE                                                  00093200
$HDS  SAMPLE HEADING $HDE
$PTYS
THIS IS A TEST OF THE POETRY LINE INDENT. $L                            00093400
EACH SENTENCE THAT EXTENDS INTO A NEW LINE SHOULD BE INDENTED TWO SPACES00093500
. $L  THE POETRY SYMBOLS TEST WILL FOLLOW THIS LINE. $L                 00093600
POETRY SYMBOLS $FT $CS $SV $LV
$PTYE
$TLS 3RD SAMPLE TITLE $TLE$PG$HDSTHIS BOOK WEIGHS 3 LBS. 11 OZ. $HDE
$HDS96 LB. TEST RUN$HDE   $TLS SAMPLE TITLE 9 MILLIMETERS LONG $TLE $PG
 3 YD, 2 FT, 4 IN;   3 YD., 2 FT., 4 IN.
 NOW, CANCEL THE TITLE: $TLS $TLE $PG
$HDSTHIS IS A SAMPLE TABLE$HDE                                          00093900
 $STB1L02 $STB2R15 $STB3025 $STB4D32                                    00094000
 $#1 ITEM $#2 MAN'R $#3 RATING $#4 PRICE                                00094100
 $#1 SOAP $#2 ACME $#3 42.3 $#4 5.22                                    00094200
 $#1 SUDS $#2 MILLER $#3 100 $#4 .89                                    00094300
 $#1 QUACKS $#2 DUCKS $#3 0 $#4 1000000.1                               00094400
 $#1 OVERFLOWFIELD $#2 WAG $#3 9999.99 $#4 CHEAP                        00094500
/_ABOUT_/ /_EE_//_KNOWLEDGE_/
$PG TRY COD INPUT SYMBOLS:
@   &
 $PG
 $TLS 5000 TYPICAL AND PROBLEM WORDS $TLE
$PG THE FOLLOWING ARE TYPICAL AND PROBLEM WORDS LISTED IN THE TRANSCRIBE
RS' GUIDE.    THE SELF-CHECKING FEATURE WILL CAUSE THOSE INCORRECTLY
TRANSLATED BY DOTSYS TO BE FLAGGED.  SOME CORRECTLY TRANSLATED WORDS MAY
```

```
 ALSO BE FLAGGED; THESE SPURIOUS ERRORS ARE EXPLAINED IN THE USERS' GUID
E.  NOTE ALSO THAT THE =AB OF =AB INITIO, THE =AL OF =AL FINE, AND
THE =X OF =X-RAY SHOULD BE PRECEDED BY A LITERAL SIGN.  THE SELF-CHECKIN
G FEATURE DOES NOT PRESENTLY CHECK FOR THIS.
 $PG $SCON$/$/$/$/$/
¬A|AR|E   ¬A|AR|CA   ¬AB   ¬ABA|CO|CA   ABALONE   AB|ANO|ON|EO|   ABA|SHOOOOOIOO
||EO|   A|BE|ACY   A|EE| E   A|BE|EY   ¬A|BB|OTT   A|BB|REVI|ATION   ABOOOJOO2JO
ON|IN||ALLY|   ABOOM|IN||OUS   AB|EA|M   ABECEO|AR||AN   AB|EO|   ¬AB|EOOOOO3OO
RIOE|EN|   AB|ER|R|ANCE|   ABEY|ANCE|   _AB   _|IN|ITIO   A|BLE|-BOOI|EOOOOOO4OO
|   ABLEGATE   AEC|AR|D   ABOM|IN|A|BLE||NESS|   ABOM|IN||ATION|   _A   OOOOO5OO
_BON   _M|AR||CH| E   ABCR|TICA||IST|   AB|OUGHT|   AB|CUNO||ING|   |ABOOOOOO6OO
UT|-FACE   |ABOVE|EO|AR|O   |ABOVE|GR|OUNO|   |ABOVE|-M|EN||TION||EO|   OOOOO7OO
|ABOVE|-|SAID|   ABRA|SION|   AEREAC|TICN|   ABR|EA||ST|   ABS|ENCE|   AOOOOOBOO
 $SOFF $PG
 THIS IS THE END OF THE RUN.
```

APPENDIX II

PROOF OUTPUT FROM STANDARD SAMPLE

NOECHO

RIGHT CONTEXT TABLE

NON-TERMINAL	INPUT CLASSES			
P	14	03	02	13
L	01	13	13	13

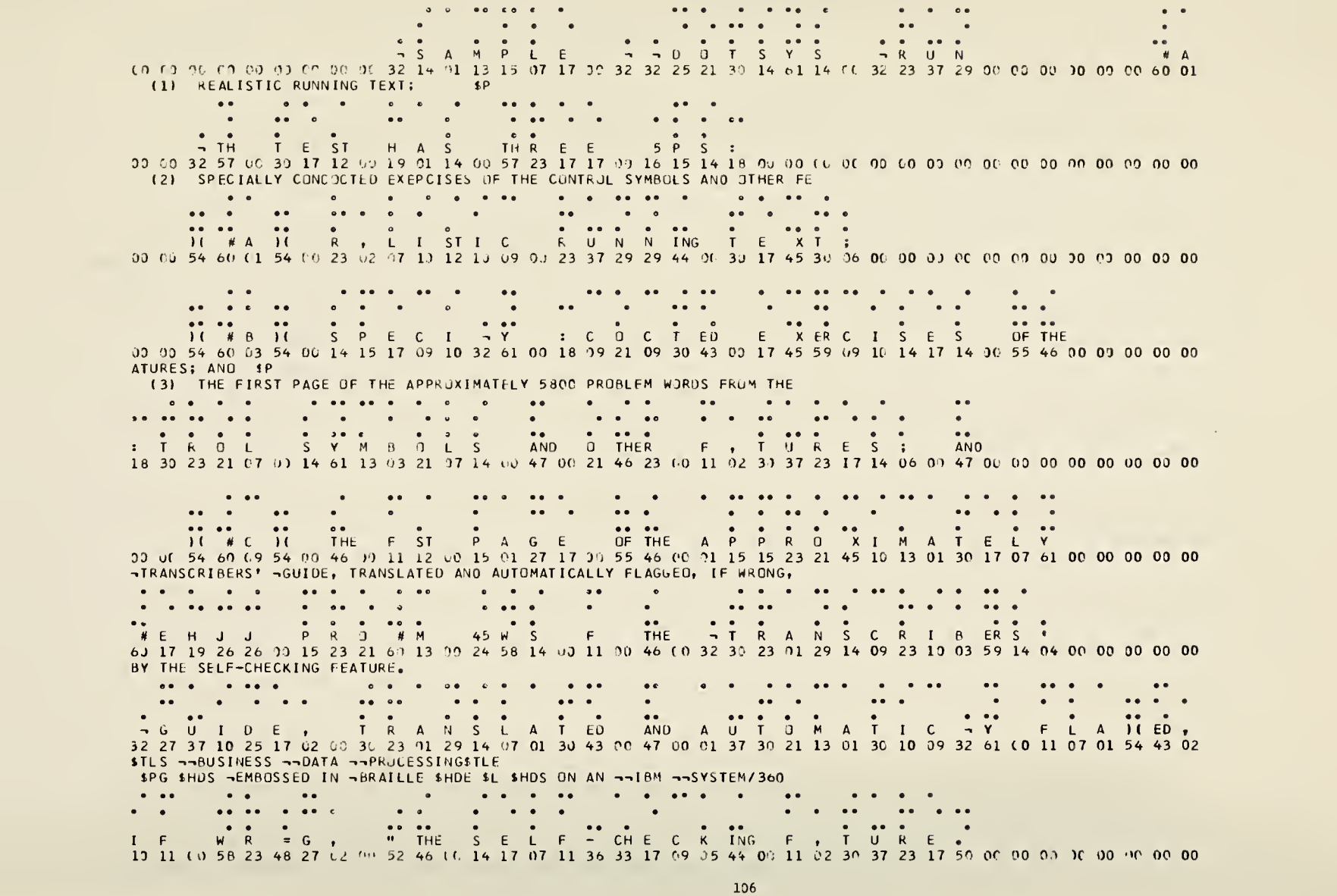
DECISION TABLE

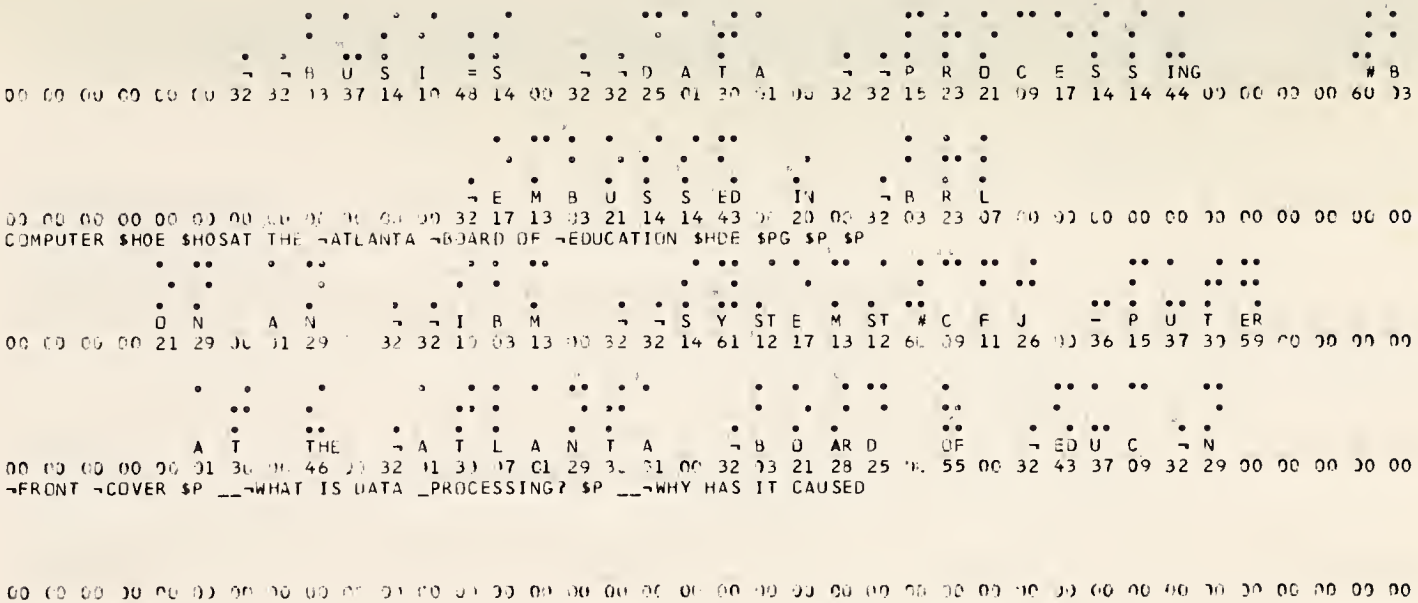
COLUMN		01	02	03	04	05	06	07	08	09	10	11
INPUT CLASS	01	Y	F	-	-	-	-	-	-	-	-	-
	02	-	-	Y	F	-	-	-	-	-	-	-
	03	G	-	-	-	-	-	-	-	-	-	-
	04	-	Y	F	-	-	-	-	-	-	-	-
	05	G	-	-	-	-	-	-	-	-	-	-
	06	-	-	-	Y	F	-	-	-	-	-	-
	07	-	-	-	Y	F	-	-	-	-	-	-
	08	-	-	-	-	-	Y	F	-	-	-	-
	09	-	-	-	-	-	-	Y	F	-	-	-
	10	-	-	-	-	-	-	-	Y	F	-	-
	11	-	-	-	-	-	-	-	-	Y	F	-
	12	G	-	-	-	-	-	-	-	-	-	-
	13	G	-	-	-	-	-	-	-	-	-	-
	14	G	-	-	-	-	-	-	-	-	-	-
	15	-	-	N	G	-	-	-	-	-	-	-
	16	Y	F	-	-	-	-	-	-	-	-	-
	17	-	-	-	-	Y	F	-	-	-	-	-
	18	-	-	-	-	Y	F	-	-	-	-	-
	19	-	-	-	Y	F	-	-	-	-	-	-
	20	-	-	-	-	-	-	-	-	-	Y	F
	21	G	-	-	-	-	-	-	-	-	-	-
	22	-	-	-	-	-	-	-	-	-	-	Y
STATE VARIABLE	01	-	-	N	-	-	-	-	-	-	-	-
	02	-	-	-	N	-	-	-	-	-	-	-
	03	N	N	-	N	N	-	-	-	-	-	N
	04	-	-	-	-	-	Y	N	N	Y	-	-
	05	-	-	-	-	-	N	Y	-	-	-	-
	06	-	Y	-	-	N	-	-	-	-	-	-
	07	N	N	-	N	N	-	-	-	-	Y	N
	08	-	-	-	-	-	-	-	-	-	-	Y

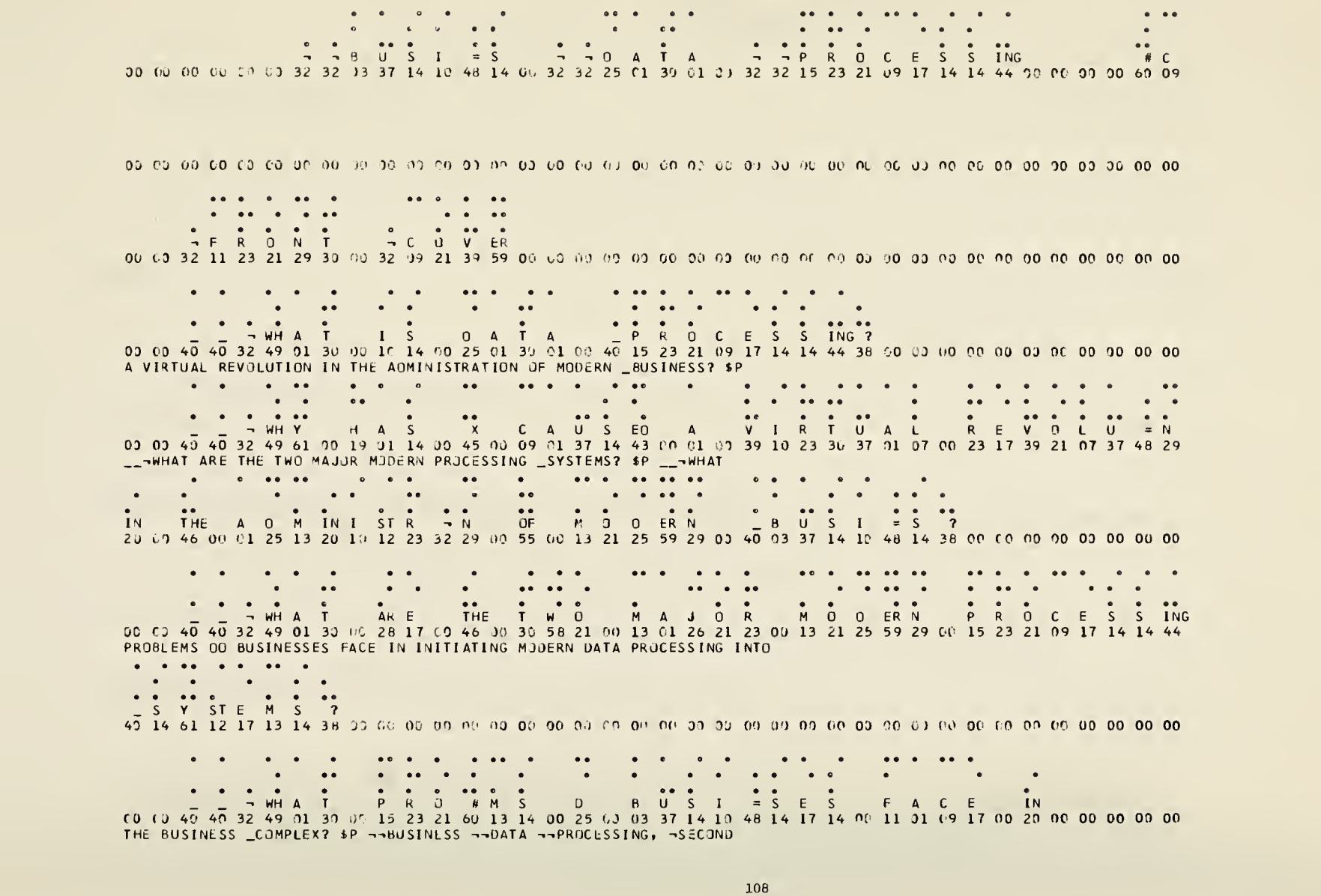
TRANSITION TABLE

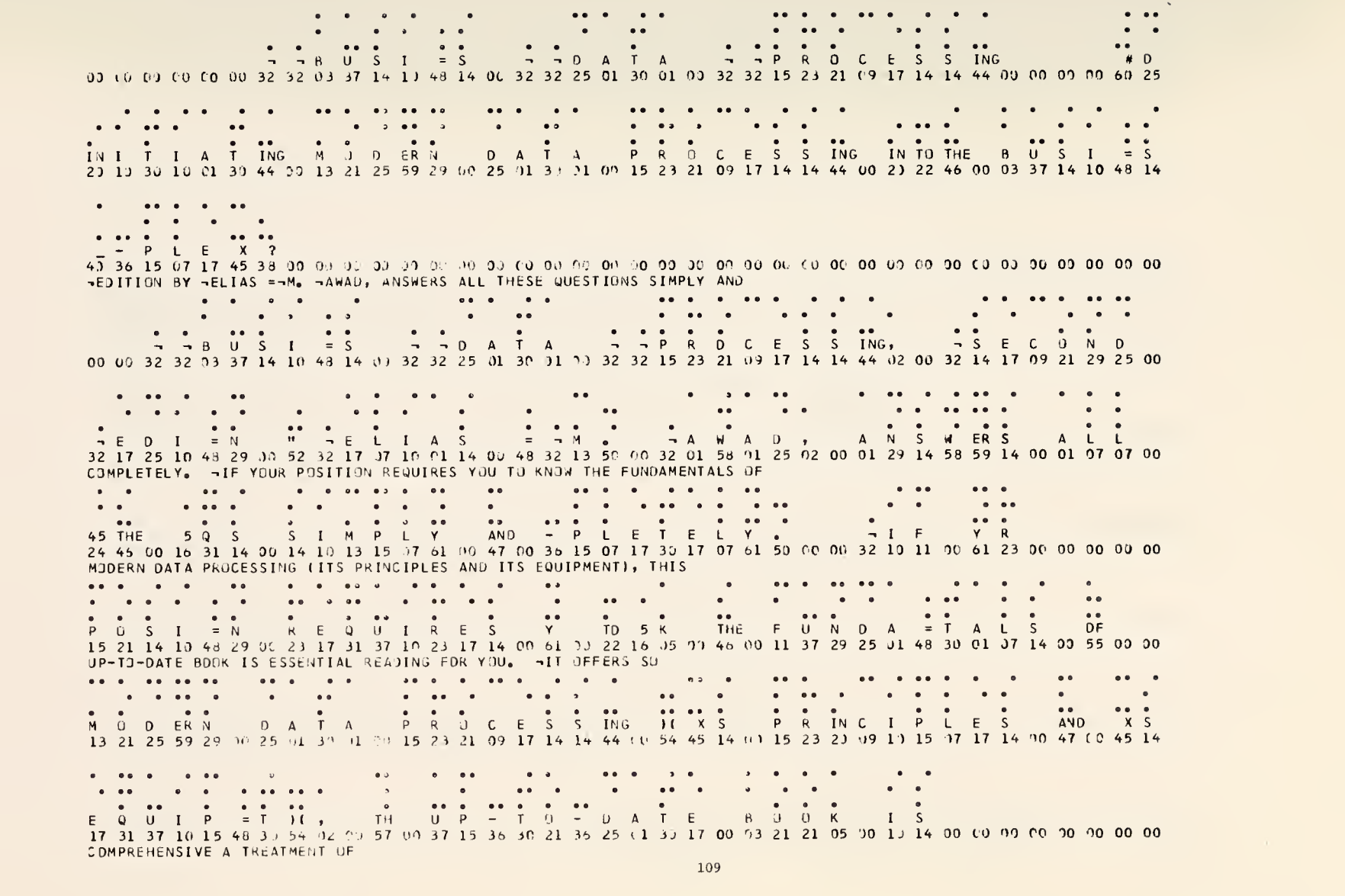
STATE VARIABLE		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
INPUT CLASS	01	R	S	-	-	-	S	R	R
	02	S	R	-	-	-	R	R	R
	03	-	R	-	-	-	R	R	R
	04	R	S	-	-	-	S	R	R
	05	-	-	T	-	-	-	R	R
	06	R	S	-	-	-	S	R	R
	07	R	R	-	-	-	R	R	R
	08	R	R	-	-	-	R	R	R
	09	P	R	-	-	-	R	R	R
	10	R	R	-	S	-	R	R	R
	11	R	R	-	R	-	R	R	R
	12	-	R	-	-	S	R	R	R
	13	-	R	-	-	R	R	R	R
	14	R	R	-	-	-	R	R	R
	15	R	P	-	-	-	R	R	S
	16	R	R	-	-	-	R	R	R
	17	R	S	-	-	-	R	R	R
	18	R	S	-	-	-	S	R	R
	19	R	S	-	-	-	-	S	R
	20	R	S	-	-	-	S	R	R
	21	R	R	-	-	-	S	R	R
	22	R	S	-	-	-	S	R	R

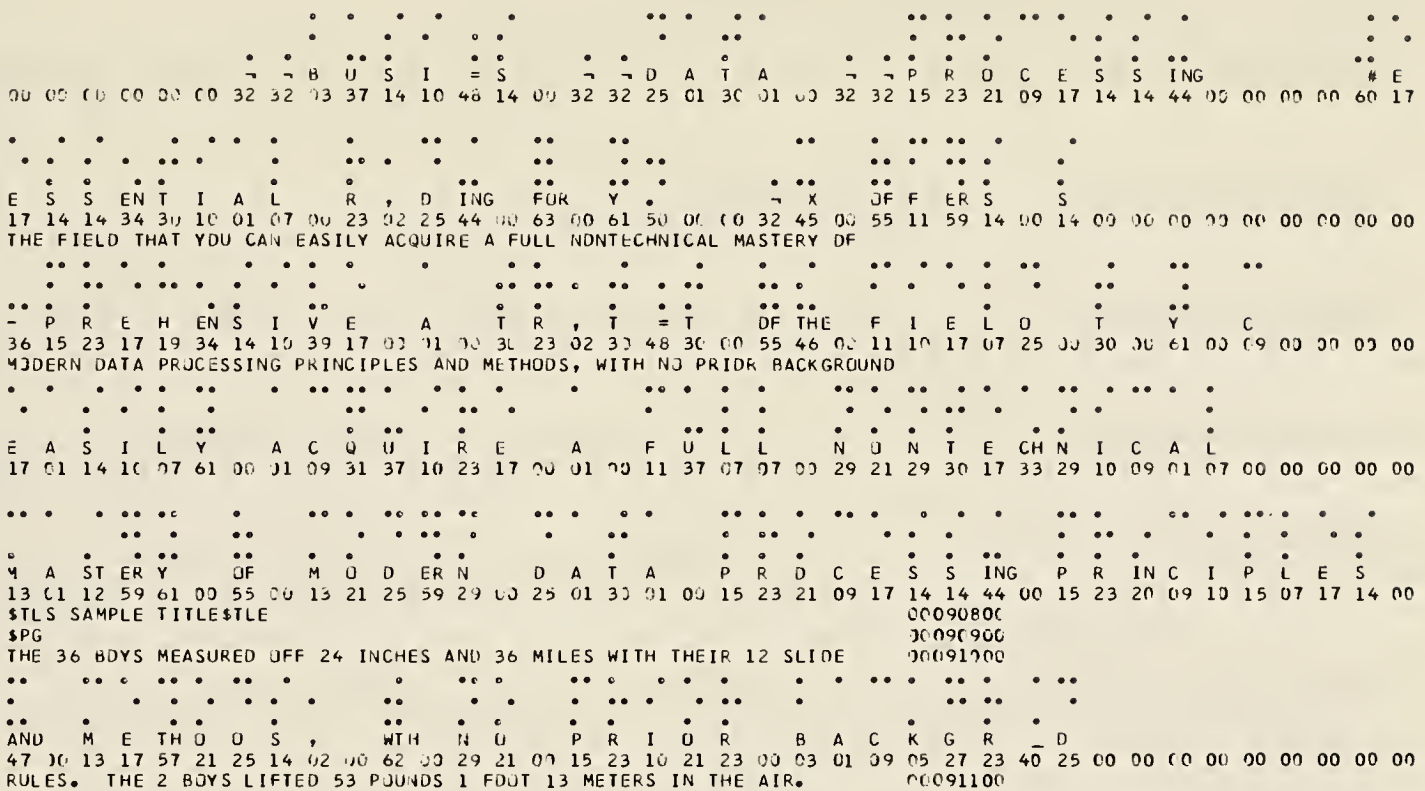
```
$TLS ¬SAMPLE ¬¬DOTSYS ¬RUN $TLE $PG $P ¬THIS TEST HAS THREE PARTS: $P

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

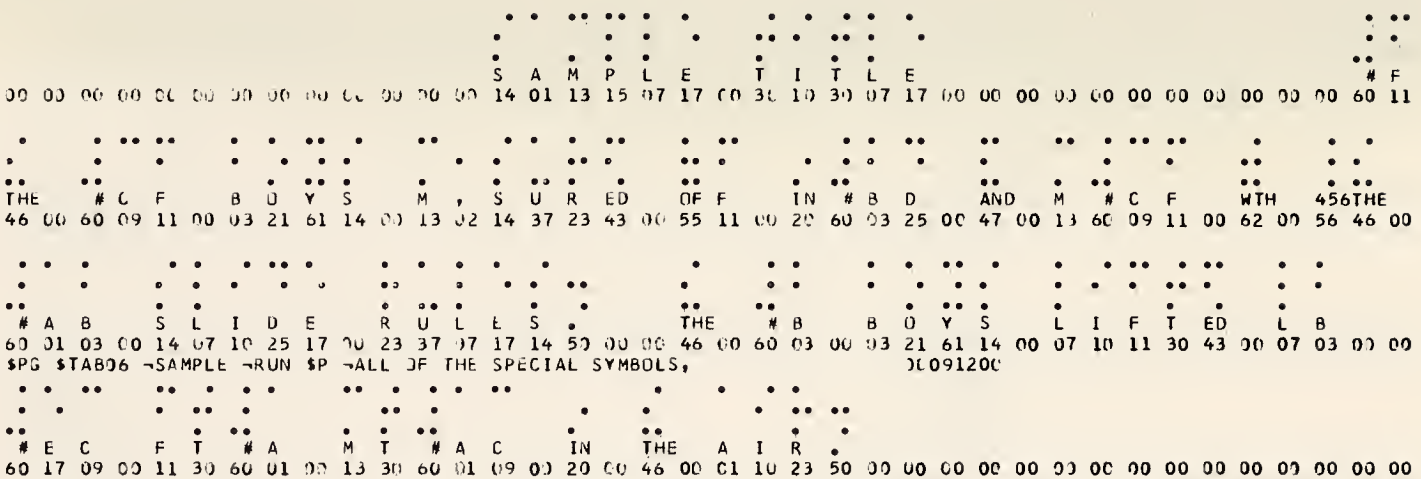

¬ S A M P L E ¬ ¬ D O T S Y S ¬ R U N # A
(1) REALISTIC RUNNING TEXT; $P
¬ TH T E ST H A S TH R E E 5 P S :
(2) SPECIALLY CONCOCTED EXEPCISES OF THE CONTROL SYMBOLS AND OTHER FE
)(# A)(R , L I ST I C R U N N ING T E X T ;
)(# B)(S P E C I ¬ Y : C O C T ED E X ER C I S E S OF THE
ATURES; AND $P
(3) THE FIRST PAGE OF THE APPROXIMATELY 5800 PROBLEM WORDS FROM THE
: T R O L S Y M B O L S AND O THER F , T U R E S ; ANO
)(# C)(THE F ST P A G E OF THE A P P R O X I M A T E L Y
¬TRANSCRIBERS' ¬GUIDE, TRANSLATED AND AUTOMATICALLY FLAGGED, IF WRONG,
E H J J P R O # M 45 W S F THE ¬ T R A N S C R I B ER S '
BY THE SELF-CHECKING FEATURE.
¬ G U I D E , T R A N S L A T ED AND A U T O M A T I C ¬ Y F L A)(ED ,
$TLS ¬¬BUSINESS ¬¬DATA ¬¬PROCESSING$TLE
$PG $HDS ¬EMBOSSED IN ¬BRAILLE $HDE $L $HDS ON AN ¬¬IBM ¬¬SYSTEM/360
I F W R = G , " THE S E L F - CH E C K ING F , T U R E ,

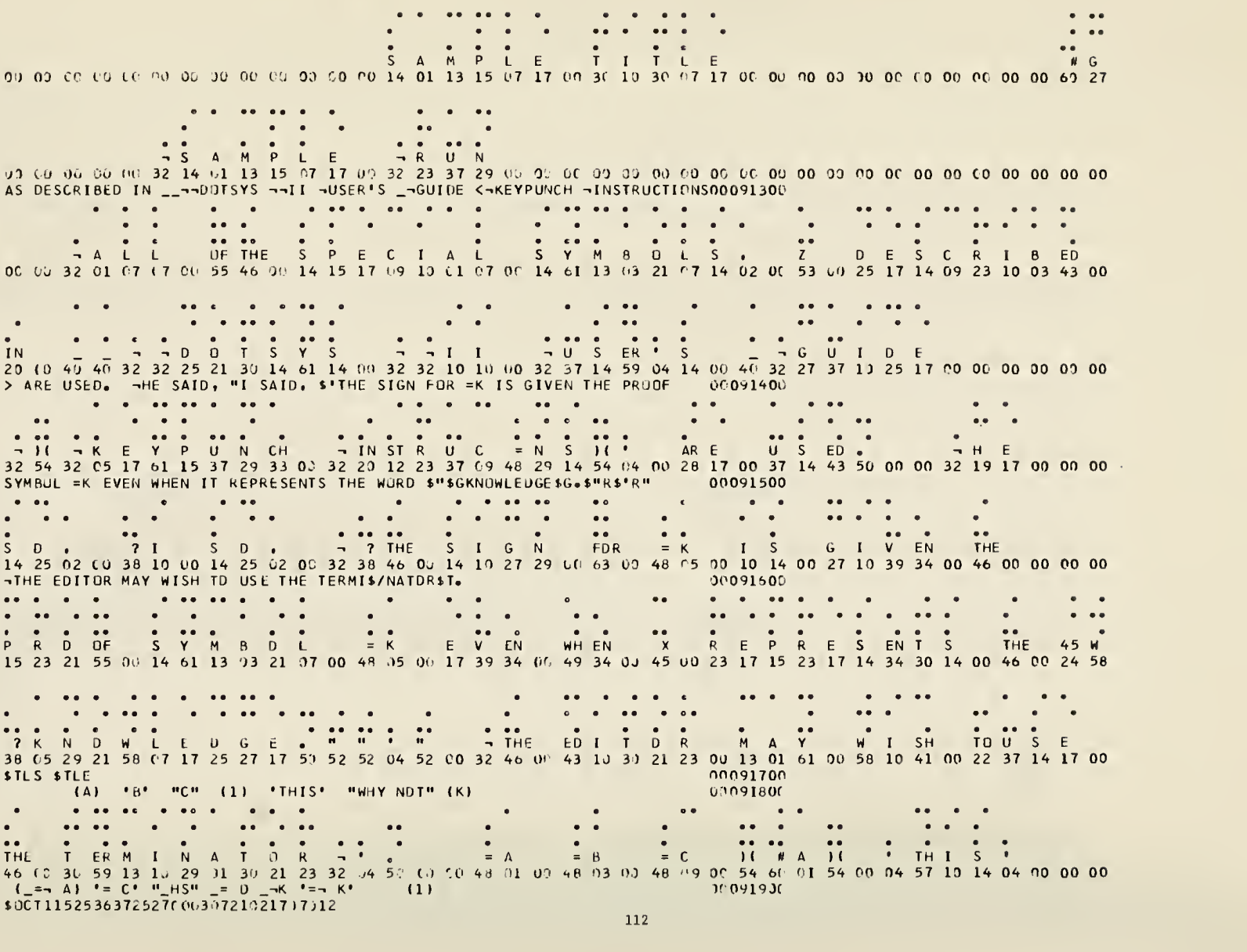

¬ ¬ B U S I = S ¬ ¬ D A T A ¬ ¬ P R O C E S S ING # B
00 00 00 00 00 00 32 32 13 37 14 10 48 14 00 32 32 25 01 30 01 00 32 32 15 23 21 09 17 14 14 44 00 00 00 00 60 03
¬ E M B U S S ED IN ¬ B R L
00 00 00 00 00 00 00 00 00 00 00 00 32 17 13 03 21 14 14 43 09 20 00 32 03 23 07 00 00 00 00 00 00 00 00 00 00 00
COMPUTER $HOE $HOSAT THE ¬ATLANTA ¬BOARD OF ¬EDUCATION $HOE $PG $P $P
D N A N ¬ ¬ I B M ¬ ¬ S Y ST E M ST # C F J — P U T ER
00 00 00 00 21 29 00 01 29 32 32 10 03 13 00 32 32 14 61 12 17 13 12 60 09 11 26 00 36 15 37 30 59 00 00 00 00
A T THE ¬ A T L A N T A ¬ B O AR D OF ¬ ED U C ¬ N
00 00 00 00 00 01 30 00 46 00 32 01 30 07 01 29 30 01 00 32 03 21 28 25 00 55 00 32 43 37 09 32 29 00 00 00 00 00
¬FRONT ¬COVER $P __¬WHAT IS DATA _PROCESSING? $P __¬WHY HAS IT CAUSED
00 00 00 00 00 00 00 00 00 00 01 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 00 00 00 00 00 00 00 00

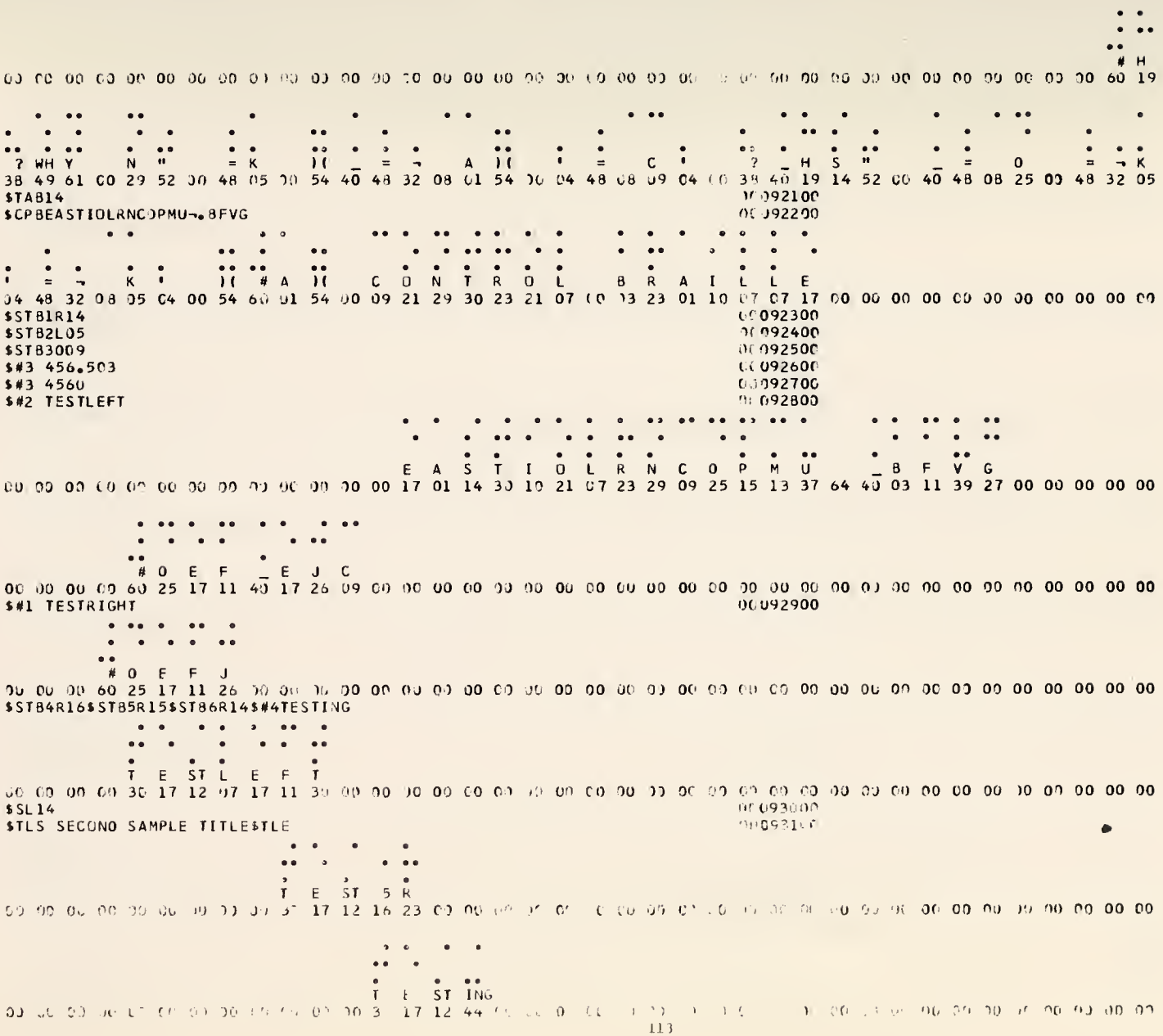

¬ ¬ B U S I = S ¬ ¬ O A T A ¬ ¬ P R O C E S S ING # C
00 00 00 00 00 00 32 32 03 37 14 10 48 14 00 32 32 25 01 30 01 20 32 32 15 23 21 09 17 14 14 44 00 00 00 00 60 09
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
¬ F R O N T ¬ C U V ER
00 00 32 11 23 21 29 30 00 32 09 21 39 59 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
_ _ ¬ WH A T I S O A T A _ P R O C E S S ING ?
00 00 40 40 32 49 01 30 00 10 14 00 25 01 30 01 00 40 15 23 21 09 17 14 14 44 38 00 00 00 00 00 00 00 00 00 00
A VIRTUAL REVOLUTION IN THE AOMINISTRATION OF MODERN _8USINESS? $P
_ _ ¬ WH Y H A S X C A U S EO A V I R T U A L R E V O L U = N
00 00 40 40 32 49 61 00 19 01 14 00 45 00 09 01 37 14 43 00 01 00 39 10 23 30 37 01 07 00 23 17 39 21 07 37 48 29
__¬WHAT ARE THE TWO MAJOR MODERN PROCESSING _SYSTEMS? $P __¬WHAT
IN THE A O M IN I ST R ¬ N OF M O O ER N _ B U S I = S ?
20 00 46 00 01 25 13 20 10 12 23 32 29 00 55 00 13 21 25 59 29 00 40 03 37 14 10 48 14 38 00 00 00 00 00 00 00 00
_ _ ¬ WH A T AR E THE T W O M A J O R M O O ER N P R O C E S S ING
00 00 40 40 32 49 01 30 00 28 17 00 46 00 30 58 21 00 13 01 26 21 23 00 13 21 25 59 29 00 15 23 21 09 17 14 14 44
PROBLEMS 00 BUSINESSES FACE IN INITIATING MODERN DATA PROCESSING INTO
_ S Y ST E M S ?
40 14 61 12 17 13 14 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
_ _ ¬ WH A T P R O # M S D B U S I = S E S F A C E IN
00 00 40 40 32 49 01 30 00 15 23 21 60 13 14 00 25 00 03 37 14 10 48 14 17 14 00 11 01 09 17 00 20 00 00 00 00 00
THE BUSINESS _COMPLEX? $P ¬¬BUSINLSS ¬¬DATA ¬¬PROCESSING, ¬SECOND


¬ ¬ B U S I = S ¬ ¬ D A T A ¬ ¬ P R O C E S S ING # D
IN I T I A T ING M U D ER N D A T A P R O C E S S ING IN TO THE B U S I = S
¬ E P L E X ?
¬EDITION BY ¬ELIAS =¬M. ¬AWAD, ANSWERS ALL THESE QUESTIONS SIMPLY AND
¬ ¬ B U S I = S ¬ ¬ D A T A ¬ ¬ P R O C E S S ING, ¬ S E C O N D
¬ E D I = N " ¬ E L I A S = ¬ M . ¬ A W A D , A N S W ER S A L L
COMPLETELY. ¬IF YOUR POSITION REQUIRES YOU TO KNOW THE FUNDAMENTALS OF
THE 5 Q S S I M P L Y AND - P L E T E L Y . ¬ I F Y R
MODERN DATA PROCESSING (ITS PRINCIPLES AND ITS EQUIPMENT), THIS
P U S I = N R E Q U I R E S Y TO 5 K THE F U N D A = T A L S DF
UP-TO-DATE BOOK IS ESSENTIAL READING FOR YOU. ¬IT OFFERS SU
M O D ER N D A T A P R O C E S S ING)(X S P R IN C I P L E S AND X S
E Q U I P = T)(, TH U P - T O - D A T E B U O K I S
COMPREHENSIVE A TREATMENT OF

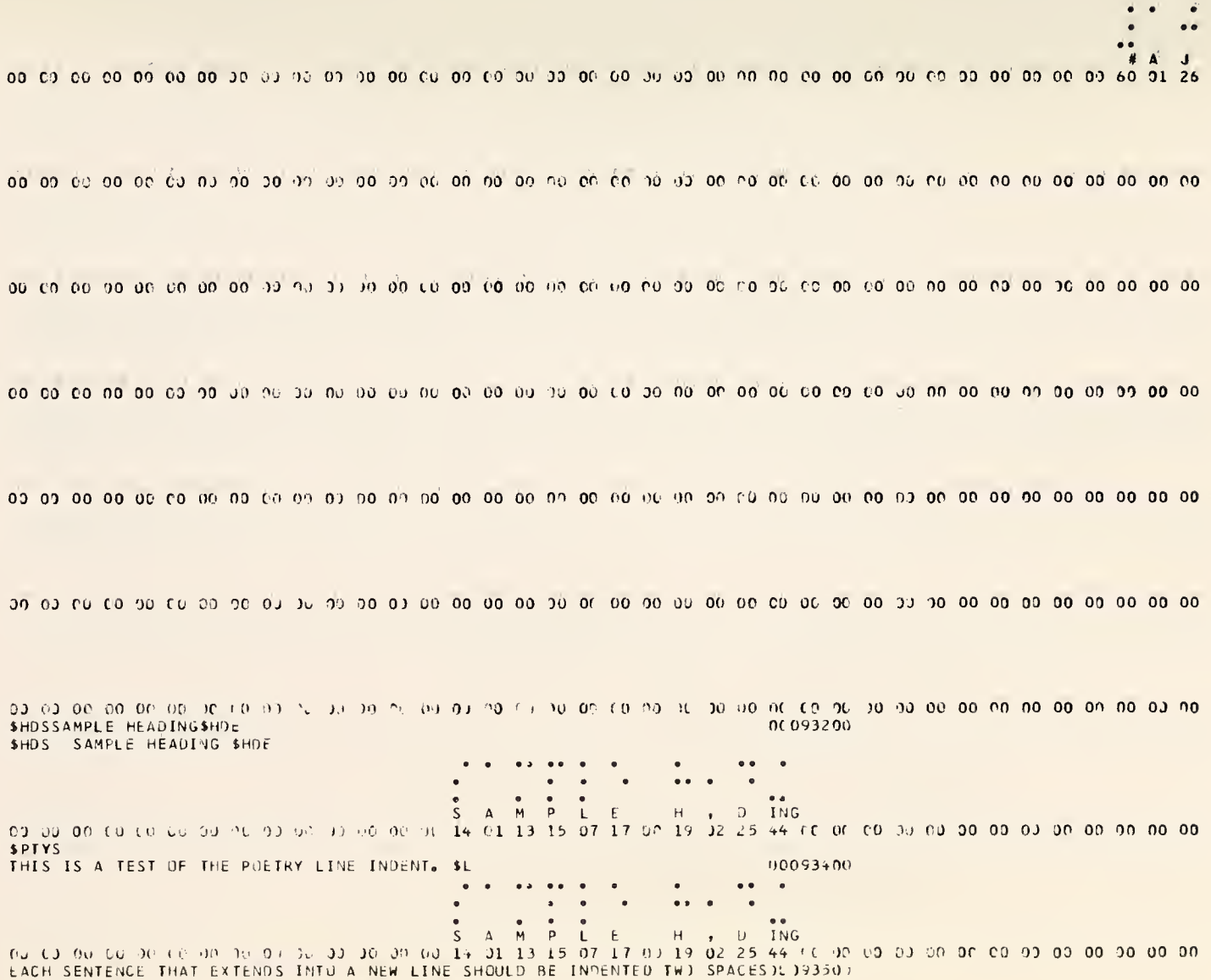

¬ ¬ B U S I = S ¬ ¬ D A T A ¬ ¬ P R O C E S S ING # E
00 00 00 00 00 00 32 32 03 37 14 10 46 14 00 32 32 25 01 30 01 00 32 32 15 23 21 09 17 14 14 44 00 00 00 00 60 17
E S S EN T I A L R , D ING FOR Y . ¬ X OF F ER S S
17 14 14 34 30 10 01 07 00 23 02 25 44 00 63 00 61 50 00 00 32 45 00 55 11 59 14 00 14 00 00 00 00 00 00 00 00 00
THE FIELD THAT YDU CAN EASILY ACQUIRE A FULL NDNTECHNICAL MASTERY DF
- P R E H EN S I V E A T R , T = T OF THE F I E L D T Y C
36 15 23 17 19 34 14 10 39 17 00 01 00 30 23 02 30 48 30 00 55 46 00 11 10 17 07 25 00 30 00 61 00 09 00 00 00 00
MODERN DATA PROCESSING PRINCIPLES AND METHODS, WITH NO PRIDR BACKGROUND
E A S I L Y A C Q U I R E A F U L L N O N T E CH N I C A L
17 01 14 10 07 61 00 01 09 31 37 10 23 17 00 01 00 11 37 07 07 00 29 21 29 30 17 33 29 10 09 01 07 00 00 00 00 00
M A ST ER Y OF M O D ER N D A T A P R O C E S S ING P R IN C I P L E S
13 01 12 59 61 00 55 00 13 21 25 59 29 00 25 01 30 01 00 15 23 21 09 17 14 14 44 00 15 23 20 09 10 15 07 17 14 00
$TLS SAMPLE TITLE$TLE 00090800
$PG 00090900
THE 36 BOYS MEASURED OFF 24 INCHES AND 36 MILES WITH THEIR 12 SLIDE 00091000
AND M E TH O O S , WITH N O P R I O R B A C K G R _ D
47 00 13 17 57 21 25 14 02 00 62 00 29 21 00 15 23 10 21 23 00 03 01 09 05 27 23 40 25 00 00 00 00 00 00 00 00 00
RULES. THE 2 BOYS LIFTED 53 POUNDS 1 FOOT 13 METERS IN THE AIR. 00091100

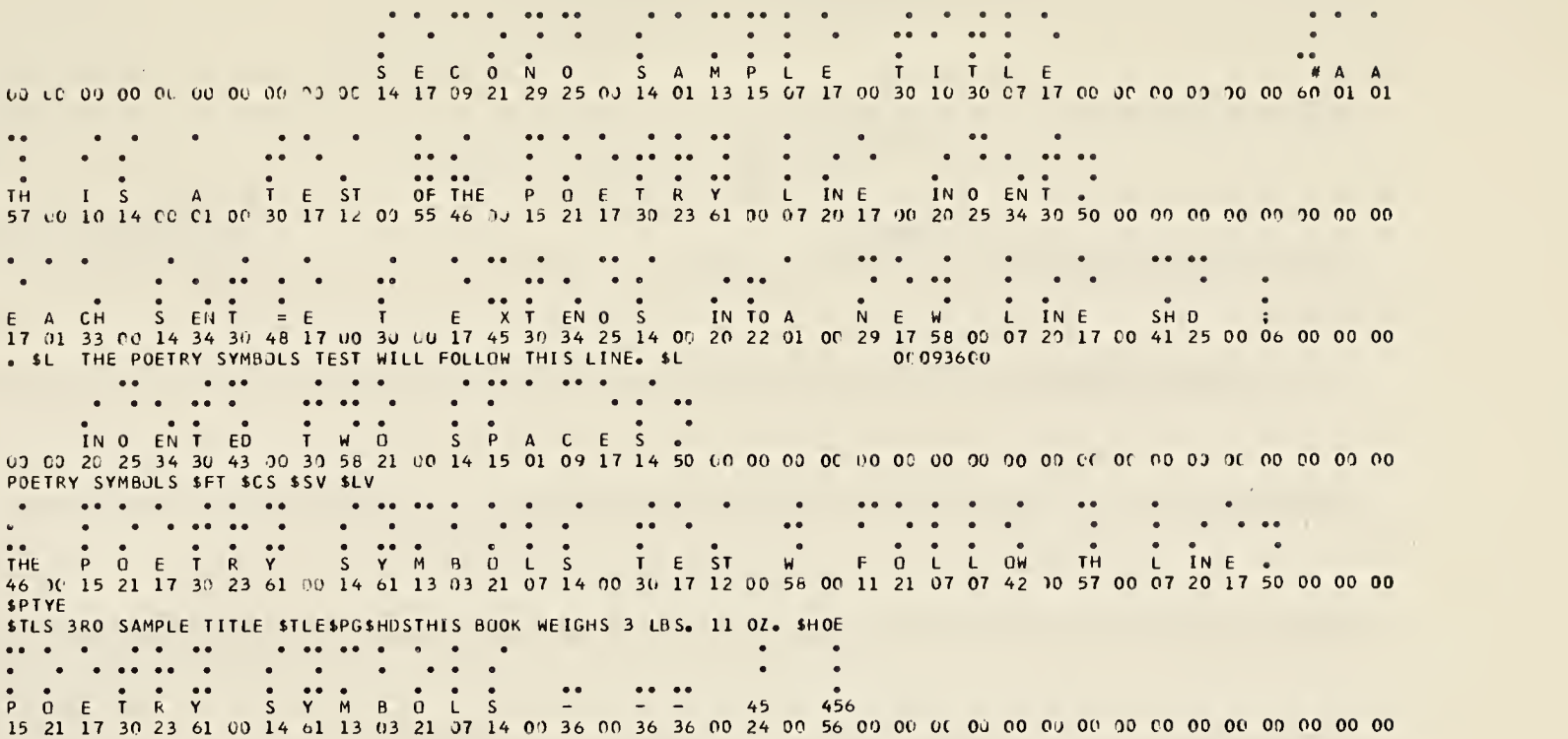

SAMPLE TITLE
00 00 00 00 00 00 00 00 00 00 00 00 14 01 13 15 07 17 00 30 10 30 07 17 00 00 00 00 00 00 00 00 00 00 00 60 11
THE # C F B O Y S M , S U R ED OF F IN # B D AND M # C F WTH 456THE
46 00 60 09 11 00 03 21 61 14 00 13 02 14 37 23 43 00 55 11 00 20 60 03 25 00 47 00 13 60 09 11 00 62 00 56 46 00
A B S L I D E R U L E S . THE # B B O Y S L I F T ED L B
60 01 03 00 14 07 10 25 17 00 23 37 07 17 14 50 00 00 46 00 60 03 00 03 21 61 14 00 07 10 11 30 43 00 07 03 00 00
$PG $TAB06 ¬SAMPLE ¬RUN $P ¬ALL OF THE SPECIAL SYMBOLS,
OC09120C
E C F T # A M T # A C IN THE A I R .
60 17 09 00 11 30 60 01 00 13 30 60 01 09 00 20 00 46 00 01 10 23 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

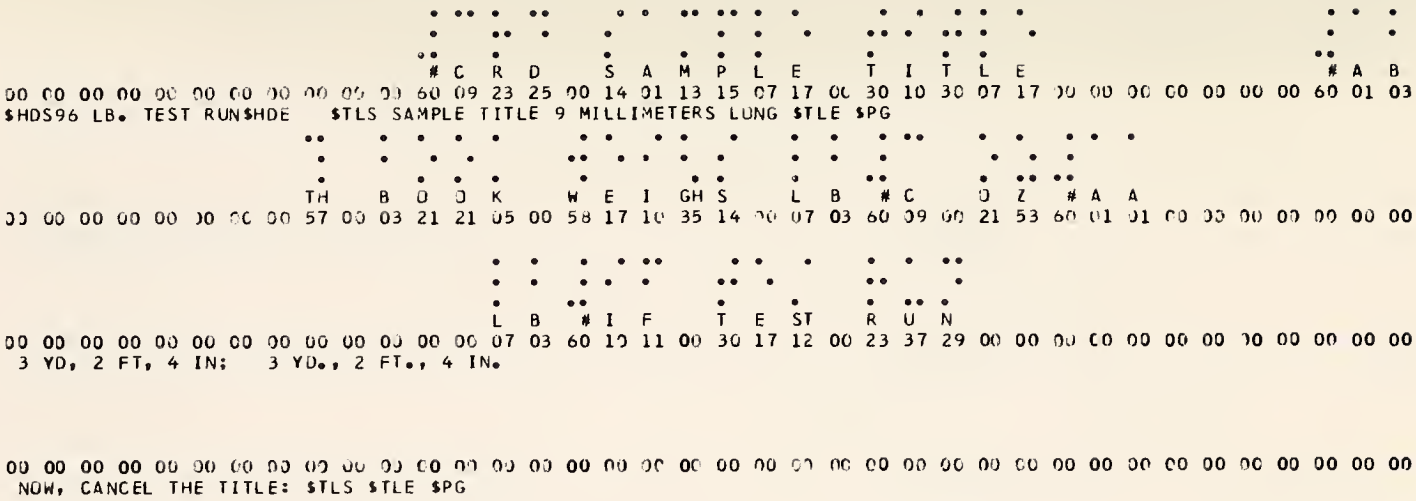

SAMPLE TITLE
¬SAMPLE RUN
AS DESCRIBED IN __¬¬DDTSYS ¬¬II ¬USER'S _¬GUIDE <¬KEYPUNCH ¬INSTRUCTIONS00091300
¬ALL OF THE SPECIAL SYMBOLS ARE USED. ¬HE SAID, "I SAID, $'THE SIGN FOR =K IS GIVEN THE PROOF 00091400
¬)(¬KEYPUNCH ¬INSTRUCTIONS 00091500
SYMBOL =K EVEN WHEN IT REPRESENTS THE WORD $"$GKNOWLEDGE$G.$"R$'R"
S D , ? I S D , ¬ ? THE SIGN FOR =K IS GIVEN 00091600
¬THE EDITOR MAY WISH TO USE THE TERMI$/NATOR$T.
PRODUCE WHEN X REPRESENTS THE 45 W
? KNOWLEDGE $TLS $TLE 00091700
(A) 'B' "C" (1) 'THIS' "WHY NOT" (K) 00091800
THE TERMINATOR =K 00091900
(_=¬ A) '= C' "_HS" _= D _¬K '=¬ K' (1)
$DCT1152536372527(0030721021717)12

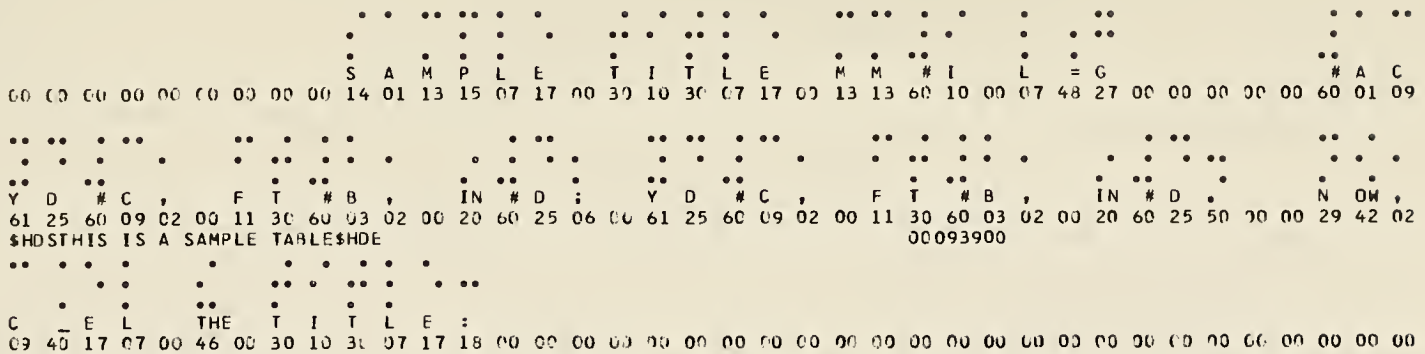

$TAB14
$CPBEASTIOLRNCDPMU¬.8FVG
$STB1R14
$STB2L05
$STB3009
$#3 456.503
$#3 4560
$#2 TESTLEFT
$#1 TESTRIGHT
$ST84R16$ST85R15$ST86R14$#4TESTING
$SL14
$TLS SECOND SAMPLE TITLE$TLE

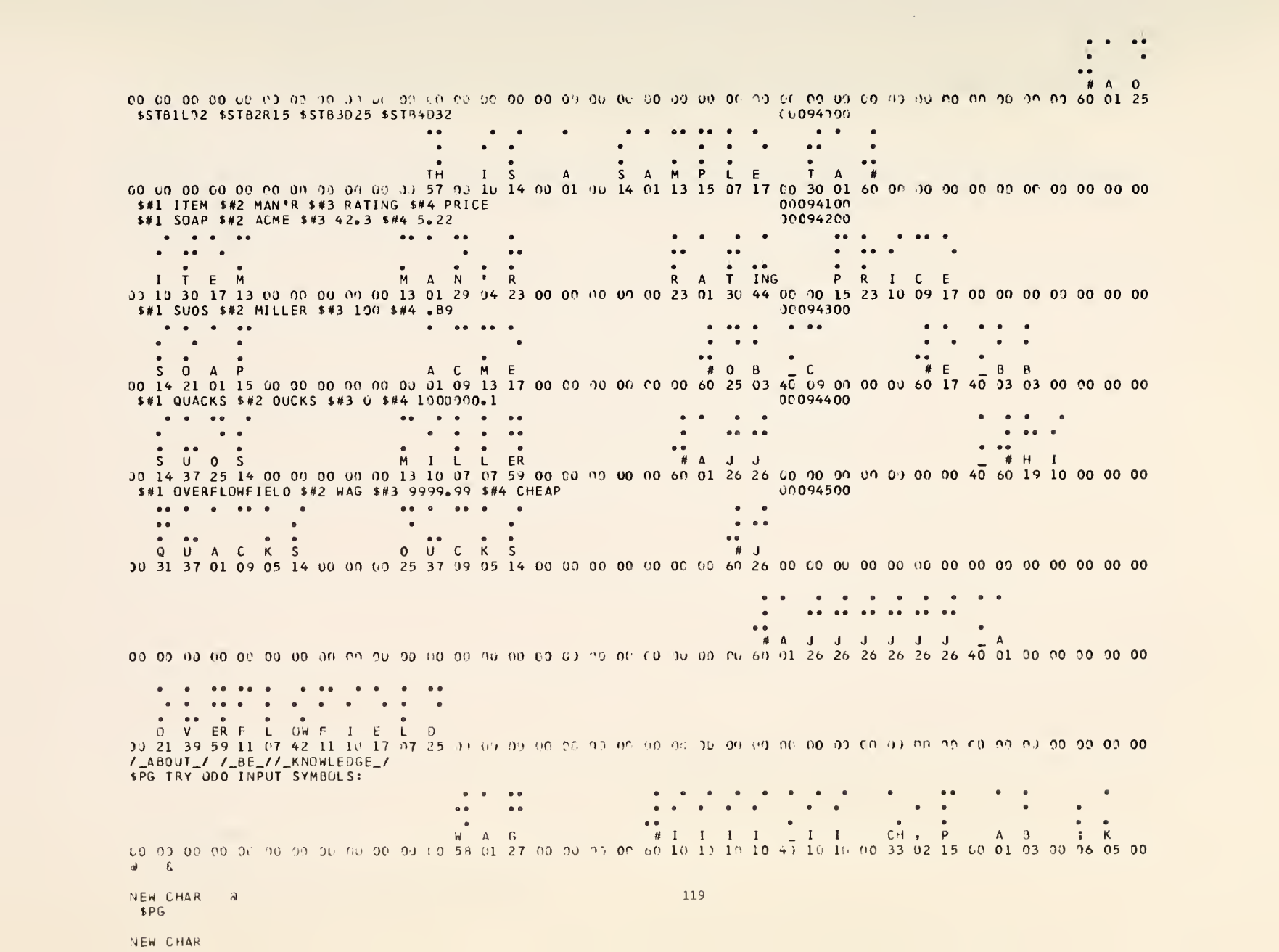
```
                                                                                                           . .
                                                                                                           : .
                                                                                                           ..
                                                                                                           # I
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 60 10

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```


$HDSSAMPLE HEADING$HDE
$HDS SAMPLE HEADING $HDE
SAMPLE HEADING
$PTYS
THIS IS A TEST OF THE POETRY LINE INDENT. $L
SAMPLE HEADING
EACH SENTENCE THAT EXTENDS INTO A NEW LINE SHOULD BE INDENTED TWO SPACES $L

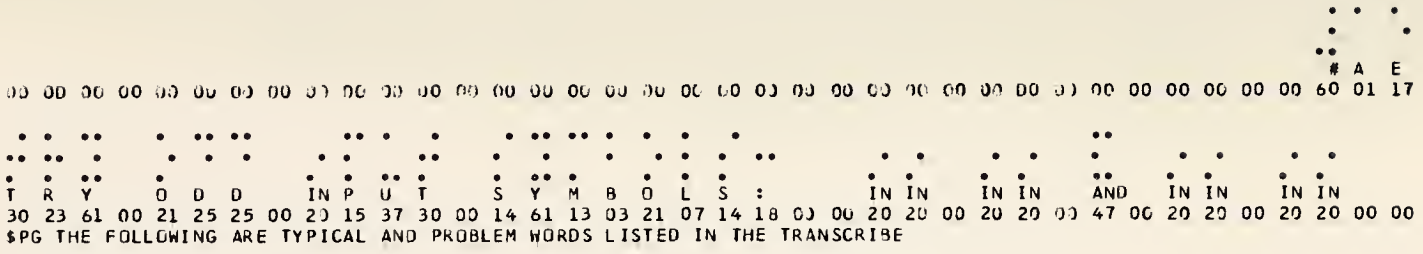

SECOND SAMPLE
THIS IS A TEST OF THE POETRY SYMBOLS
E A CH S EN T = E T E X T EN O S IN TO A N E W L IN E SH D ;
. $L THE POETRY SYMBOLS TEST WILL FOLLOW THIS LINE. $L
IN O EN T ED T W O S P A C E S .
POETRY SYMBOLS $FT $CS $SV $LV
THE P O E T R Y S Y M B O L S T E ST W F O L L OW TH L IN E .
$PTYE
$TLS 3RO SAMPLE TITLE TLEPG$HDSTHIS BOOK WEIGHS 3 LBS. 11 OZ. $HOE
P O E T R Y S Y M B O L S 45 456

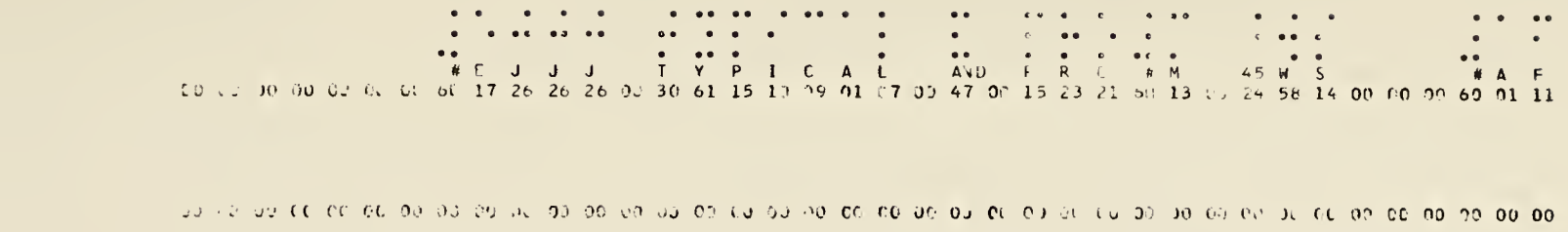

00 00 00 00 00 00 00 00 00 00 00 60 09 23 25 00 14 01 13 15 07 17 00 30 10 30 07 17 00 00 00 00 00 00 00 60 01 03
$HDS96 LB. TEST RUN$HDE $TLS SAMPLE TITLE 9 MILLIMETERS LUNG $TLE $PG
00 00 00 00 00 00 00 00 57 00 03 21 21 05 00 58 17 10 35 14 00 07 03 60 09 00 21 53 60 01 01 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 07 03 60 10 11 00 30 17 12 00 23 37 29 00 00 00 00 00 00 00 00 00 00 00 00
3 YD, 2 FT, 4 IN; 3 YD., 2 FT., 4 IN.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
NOW, CANCEL THE TITLE: $TLS $TLE $PG

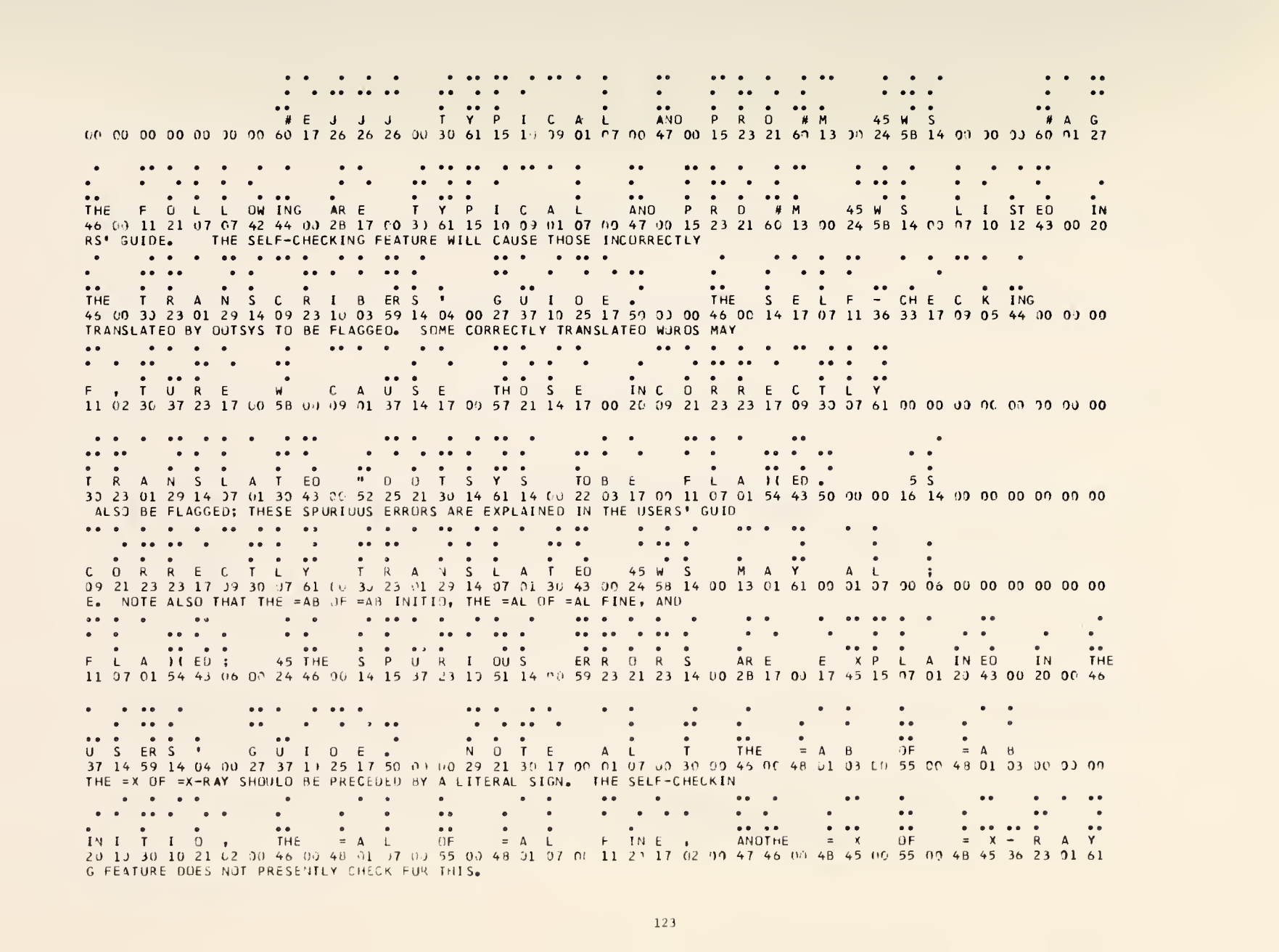

S A M P L E T I T L E M M # I L = G # A C
00 00 00 00 00 00 00 00 14 01 13 15 07 17 00 30 10 30 07 17 00 13 13 60 10 00 07 48 27 00 00 00 00 00 60 01 09
Y D # C , F T # B , IN # D ; Y D # C , F T # B , IN # D . N OW ,
61 25 60 09 02 00 11 30 60 03 02 00 20 60 25 06 00 61 25 60 09 02 00 11 30 60 03 02 00 20 60 25 50 00 00 29 42 02
$HDSTHIS IS A SAMPLE TABLE$HDE
00093900
C _ E L THE T I T L E :
09 40 17 07 00 46 00 30 10 30 07 17 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

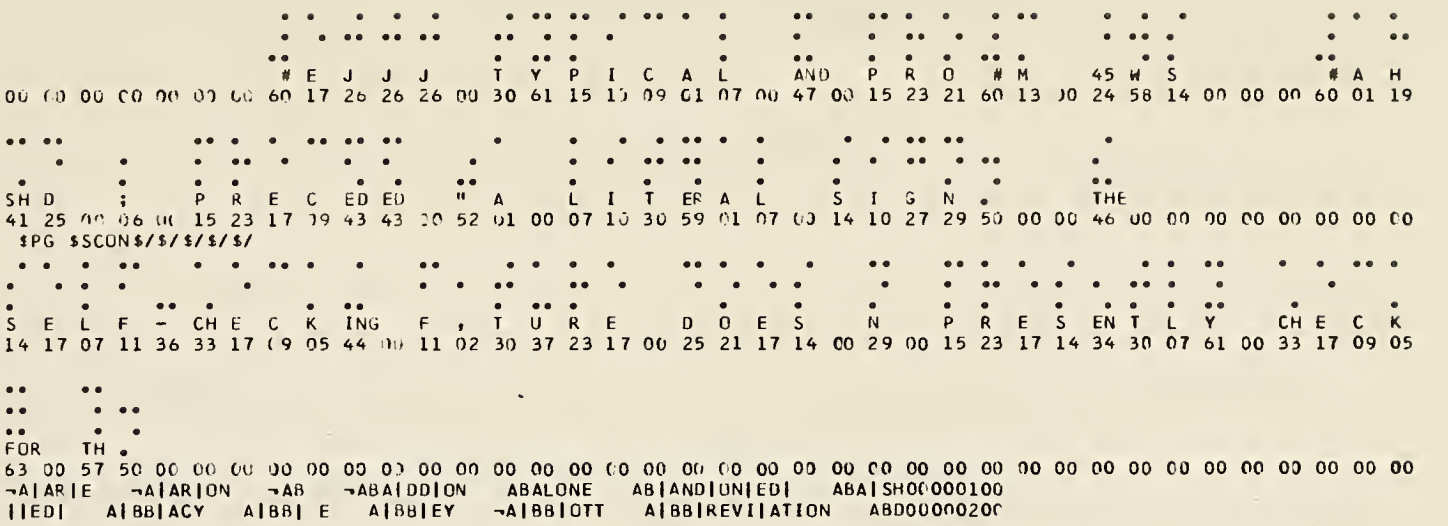

$STB1L02 $STB2R15 $STB3D25 $STB4D32
$#1 ITEM $#2 MAN'R $#3 RATING $#4 PRICE
$#1 SOAP $#2 ACME $#3 42.3 $#4 5.22
$#1 SUOS $#2 MILLER $#3 100 $#4 .89
$#1 QUACKS $#2 OUCKS $#3 0 $#4 1000000.1
$#1 OVERFLOWFIELD $#2 WAG $#3 9999.99 $#4 CHEAP
/_ABOUT_/ /_BE_//_KNOWLEDGE_/
$PG TRY ODD INPUT SYMBOLS:
@ &
NEW CHAR @
$PG
NEW CHAR

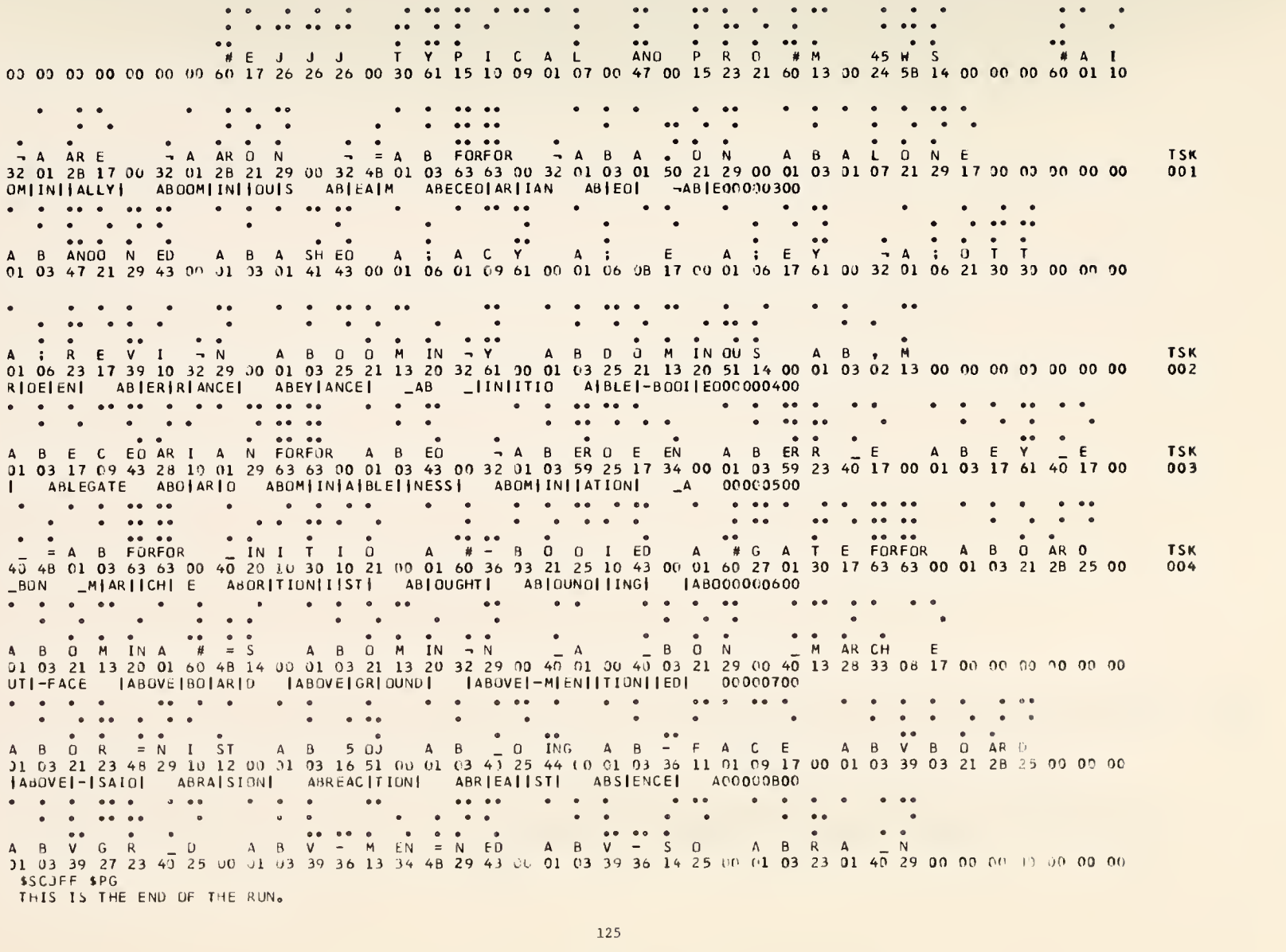
```
 $TLS SUCD TYPICAL AND PROBLEM WORDS $TLE
NEW CHAR
NEW CHAR
```

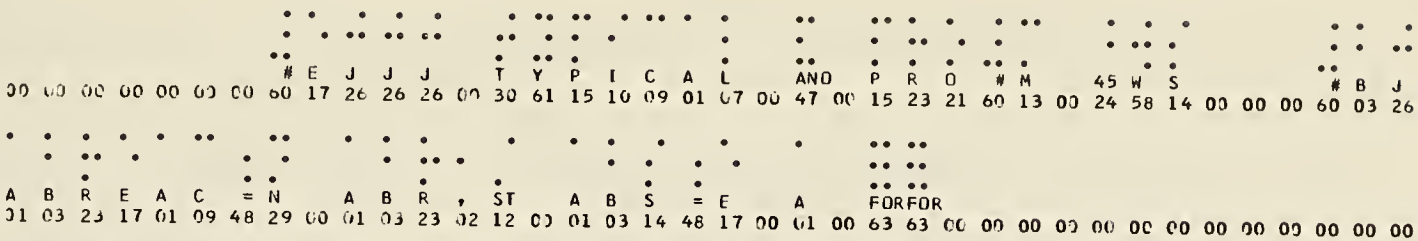

T R Y O D D IN P U T S Y M B O L S : IN IN IN IN AND IN IN IN IN
30 23 61 00 21 25 25 00 20 15 37 30 00 14 61 13 03 21 07 14 18 00 00 20 20 00 20 20 00 47 00 20 20 00 20 20 00 00
$PG THE FOLLOWING ARE TYPICAL AND PROBLEM WORDS LISTED IN THE TRANSCRIBE

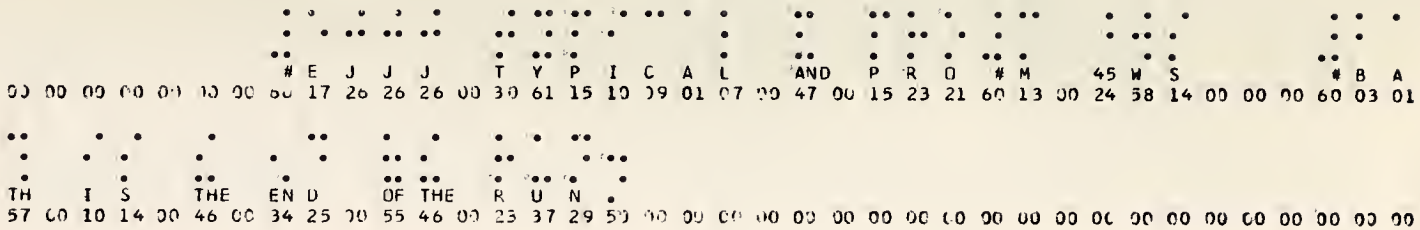

E J J J T Y P I C A L AND F R C M W S A F
17 26 26 26 30 61 15 13 09 01 07 47 15 23 21 13 24 58 14 60 01 11


#EJJJ TYPICAL AND PROD #M 45WS #AG
00 00 00 00 00 00 00 60 17 26 26 26 00 30 61 15 10 09 01 07 00 47 00 15 23 21 60 13 00 24 5B 14 00 00 00 60 01 27
THE FOLLOWING ARE TYPICAL AND PROD #M 45WS LIST EDIN
46 00 11 21 07 07 42 44 00 2B 17 00 30 61 15 10 09 01 07 00 47 00 15 23 21 60 13 00 24 5B 14 00 07 10 12 43 00 20
RS' GUIDE. THE SELF-CHECKING FEATURE WILL CAUSE THOSE INCORRECTLY
THE TRANSLATED BY OUTSYS TO BE FLAGGED. SOME CORRECTLY TRANSLATED WORDS MAY
46 00 30 23 01 29 14 09 23 10 03 59 14 04 00 27 37 10 25 17 50 00 00 46 00 14 17 07 11 36 33 17 09 05 44 00 00 00
F, TURE WOULD CAUSE THOSE CORRECTLY TRANSLATED WORDS
11 02 30 37 23 17 00 5B 00 09 01 37 14 17 00 57 21 14 17 00 20 09 21 23 23 17 09 30 07 61 00 00 00 00 00 00 00 00
TRANSLATED; ALSO BE FLAGGED; THESE SPURIOUS ERRORS ARE EXPLAINED IN THE USERS' GUID
30 23 01 29 14 07 01 30 43 00 52 25 21 30 14 61 14 00 22 03 17 00 11 07 01 54 43 50 00 00 16 14 00 00 00 00 00 00
CORRECTLY TRANSLATED; ALSO BE FLAGGED; THESE SPURIOUS ERRORS ARE EXPLAINED IN THE USERS' GUID
09 21 23 23 17 09 30 07 61 00 30 23 01 29 14 07 01 30 43 00 24 58 14 00 13 01 61 00 01 07 00 06 00 00 00 00 00 00
E. NOTE ALSO THAT THE =AB OF =AB INITIO, THE =AL OF =AL FINE, AND
FLAGGED; 45 THE SPURIOUS ERRORS ARE EXPLAINED IN THE
11 07 01 54 43 06 00 24 46 00 14 15 37 23 10 51 14 00 59 23 21 23 14 00 2B 17 00 17 45 15 07 01 20 43 00 20 00 46
USER'S GUIDE. NOTE ALSO THAT THE =AB OF =AB
37 14 59 14 04 00 27 37 10 25 17 50 00 00 29 21 30 17 00 01 07 00 30 00 46 00 48 01 03 00 55 00 48 01 03 00 00 00
THE =X OF =X-RAY SHOULD BE PRECEDED BY A LITERAL SIGN. THE SELF-CHECKIN
IN I T I O , THE =A L OF =A L F IN E , AND THE =X OF =X-RAY
20 10 30 10 21 02 00 46 00 48 01 07 00 55 00 48 01 07 00 11 20 17 02 00 47 46 00 48 45 00 55 00 48 45 36 23 01 61
G FEATURE DOES NOT PRESENTLY CHECK FOR THIS.


#EJJJ TYPICAL AND PROO#M 45W S #AH
00 00 00 00 00 00 00 60 17 26 26 26 00 30 61 15 15 09 01 07 00 47 00 15 23 21 60 13 00 24 58 14 00 00 00 60 01 19
SHD ; PRECEDED " ALITERAL SIGN THE
41 25 00 06 00 15 23 17 09 43 43 20 52 01 00 07 10 30 59 01 07 00 14 10 27 29 50 00 00 46 00 00 00 00 00 00 00 00
$PG $SCON$/$/$/$/$/
SELF-CHECKING F,TURE DOES NOT PROVIDE CHECK
14 17 07 11 36 33 17 09 05 44 00 11 02 30 37 23 17 00 25 21 17 14 00 29 00 15 23 17 14 34 30 07 61 00 33 17 09 05
FOR TH
63 00 57 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
¬A|AR|E ¬A|AR|ON ¬AB ¬ABA|DD|ON ABALONE AB|AND|ON|ED| ABA|SH00000100
||ED| A|BB|ACY A|BB| E A|BB|EY ¬A|BB|OTT A|BB|REVI|ATION ABD00000200


TSK
001
TSK
002
TSK
003
TSK
004
$SCOFF $PG
THIS IS THE END OF THE RUN.


00 00 00 00 00 00 00 60 17 26 26 26 00 30 61 15 10 09 01 07 00 47 00 15 23 21 60 13 00 24 58 14 00 00 00 60 03 26
E J J J T Y P I C A L AND P R O # M 45 W S # B J
01 03 23 17 01 09 48 29 00 01 03 23 02 12 00 01 03 14 48 17 00 01 00 63 63 00 00 00 00 00 00 00 00 00 00 00 00 00
A B R E A C = N A B R , ST A B S = E A FORFOR
TSK
005


#EJJJ TYPICAL AND PROM#45WS #BA
00 00 00 00 00 00 00 60 17 26 26 26 00 30 61 15 10 09 01 07 00 47 00 15 23 21 60 13 00 24 58 14 00 00 00 60 03 01
THIS THE END OF THE RUN
57 00 10 14 00 46 00 34 25 00 55 46 00 23 37 29 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

APPENDIX III

WORDS INCORRECTLY TRANSLATED IN 5808 PROBLEM WORDS

DOTSYS III (TABLES 10/70)	TRANSCRIBERS' GUIDE
abeced/arian	abecedarian
ablegate	ablegate
acreage	acreage
ACTH	ACTH
fine	fine
althorn	althorn
Antigone	Antigone
areaway	ar/eaway
backsword	backsword
bandog	bandog
Beelzebub	Beelzebub
Beguine	Beguine
benefic	benefic
Benes	Benes
Benét	Benét
Ber/ing	Bering
Bethesda	Be/thesda
bezique	bezique
binucleate	binucleate
Blindheim	Blindheim
Boer	Boer
brougham	brou/gham
Bundestag	Bundestag
cadenced	cadenced
Caen	Caen
canzone	canzone
Castlerea/gh	Castlerea/gh
castlery	castlery
cen/time	centime
chalone	chalone
Chisholm	Chisholm
cicer/one	cicerone
conamed	conamed
conenose	conenose
congealed	congealed
congee	congee
congen/er	congen/er
congenial	congenial
congenital	congenital
conger	conger
dar/edevil	daredevil
deaminate	deaminate
dedication	dedication
denudative	denudative
disacch/aride	disacch/aride
dis/ease	disease (ill at ease)
distingue	distingue
do (musical note)	do
Doolittle	Doolittle
dumbbell	dumbbell
ed/entulous	edentulous
Ein/thoven	Einthoven
Faenza	Faenza
Fer/inghee	Ferin/ghee
firedrake	firedrake
Fotheringhay	Fotherin/ghay
froghopper	froghopper
furlong	furlong
furth/er/er	furtherer
gallinipper	gallinipper
garderobe	garderobe
gastight	gastight
genitourin/ary	genitourin/ary
Gingold	Gingold
goatherd	goatherd
Goer/ing	Goering
gooseneck	gooseneck
Grantham	Grantham
Greatheart	Greatheart
Hadassah	Hadassah
Hadrian	Hadrian
Hapgood	Hapgood
hartshorn	hartshorn
heartsease	heartsease
hedger/ow	hedgerow
Here (goddess)	Here
Hergesheimer	Hergesheimer
Himalayas	Himalayas

APPENDIX III (Continued)

DOTSYS III (TABLES 10/70)	TRANSCRIBERS' GUIDE
Holin/sh/ed	Holinshed
hornblende	hornblende
hyaena	hyaena
in/en/arrable	inenarrable
inglenook	inglenook
insof/ar	insofar
Ione	Ione
Iredell	Iredell
isinglass	isinglass
jibboom	jibboom
krone	krone
Letter/er	Letter/er
Letterman	Letterman
Lever (bros.)	Lever
lin/eage (alignment)	lineage
ling/erie	lingerie
Littleton	Littleton
locoweed	locoweed
Loffler	Loffler
maenad	maenad
Maugham	Maugham
men/ingeal	men/ingeal
mesitylene	mesitylene
Micronesian	Micronesian
midwifery	midwifery
minestrone	minestrone
mistitled	mistitled
More's (name)	More's
muraena	muraena
nea/therd	neatherd
newsletter	newsletter
nuthatch	nuthatch
Oenone	Oenone
oer/st/ed	oerst/ed
optime	optime
Osgood	Osgood
oughtlins	ou/ghtlins
ou/thaul	outhaul
padrone	padrone
paleaceous	paleaceous
pandemonism	pandemonism
pedometer	pedometer
Persephone	Persephone
persever/ance	persever/ance
pigh/eaded	pigheaded
pokeroot	pokeroot
potherb	potherb
praenomen	praenomen
pronephros	pronephros
pros and cons	pros and cons
protonema	protonema
rar/eripe	rareripe
rawhide	rawhide
Reno	Reno
reredos	rer/edos
retroflex	retroflex
riboflavin	riboflavin
roped/ancer	ropedancer
saintonge	saintonge
sawhorse	sawhorse
Sheean	Sheean
shorthand	shorthand
shorthorn	shorthorn
Shosh/one	Shoshone
skedaddle	skedaddle
snakeroot	snakeroot
so (musical note)	so
Somes	Somes
Soong	Soong
sou'ea/st/er	sou'east/er
spathose	spathose
speakeasy	speakeasy
spikenard	spikenard
spumone	spumone
stagh/ound	staghound
st/ereoisomer	st/ereoisomer
stirabout	stirabout
stringendo	string/endo
suede	suede
super/erogatory	supererogatory

APPENDIX III (Concluded)

DOTSYS III (TABLES 10/70)	TRANSCRIBERS' GUIDE
suprar/enal	suprarenal
swastikaed	swastikaed
taenia	taenia
tearoom	tearoom
tearose	tearose
teleran	teleran
Theresa	Theresa
thiourea	thiourea
toadeater	toadeater
tonelada	tonelada
transmental	transmental
treenail	treenail
trenal	trenal
treponema	treponema
tuberose	tuberose
tufthunter	tufthunter
tweedledee	tweedledee
tweedledum	tweedledum
'twouldn't	'twouldn't
undenomin/ational	undenomin/ational
underogating	underogating
us'n	us'n
vainglorious	vainglorious
Vandyke	Vandyke
violone	violone
wakerife	wakerife
Wenceslaus	Wenceslaus
whaddaya	whaddaya
Wingate	Wingate
wired/ancer	wiredancer
wiredrawn	wiredrawn
wiseacre	wiseacre

APPENDIX IV

DEFINITION OF STATE VARIABLES AND INPUT CLASSES

State Variable	1	after the start of a number
	2	after the start of a word
	3	grade 1 translation
	4	in a quotation
	5	in italicized text
	6	not at the start of a prefix or stem
	7	part way through a word or phrase too long for one entry
	8	just after a space (or A-J), following a number
Input Class	1	contractions always used in grade 2
	2	digits
	3	most punctuation
	4	contractions used after the start of a word
	5	$G (grade switch)
	6	contractions used after the start of a word
	7	isolated full-word contractions
	8	$P" (start paragraph in quotation)
	9	$P (start paragraph in italics)
	10	" (left quote)
	11	" (right quote)
	12	_ _ (begin italics)

Input Class	13	_ (last word of italics)
	14	(space)
	15	A to J or space occurring in a number
	16	contractions always used in grade 2 containing terminal punctuation
	17	prefix or first word of compound word
	18	non-prefix beginning of word
	19	first entry of double entry
	20	second entry of double entry
	21	"forced" contraction begin sign (/_)
	22	units of measure and numbers following numbers

APPENDIX V

SUMMARY OF SPECIAL SYMBOLS

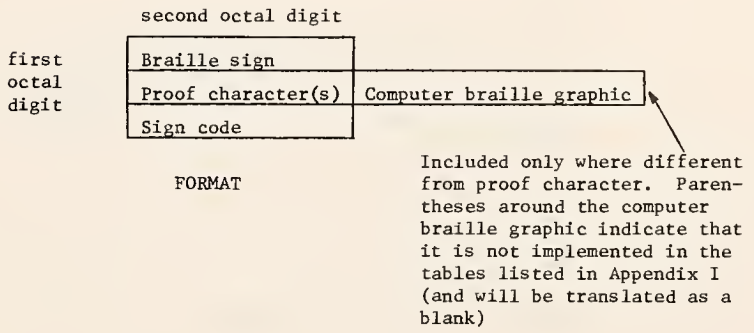
Symbol	Input Representation	Text Reference Page
capitalize letter	¬	
capitalize word	¬¬	
italicize word	_	
italics start	_ _	
left single quote	$'	
right single quote	$'R	
left double quote	$"	
right double quote	$"R	
accent marks	¢	
left bracket ([)	<	
right bracket ([)	>	
start paragraph	$P	
begin new line	$L	
begin new page	$PG	
skip to column nn	$TABnn	
letter sign	=	
change grade	$G	
divide word	$/	
termination sign	$T	

Symbol	Input Representation	Text Reference Page
long vowel sign	$LV	
start poetry	$PTYS	
end poetry	$PTYE	
start paragraph within quotation	$P"	
start title	$TLS	
end title	$TLE	
forced blank	$B	
start heading	$HDS	
end heading	$HDE	
skip lines	$SLnn	
short vowel sign	$SV	
set tab t to col. nn	$STBt[L,R or D]nn	
turn self-checking on	$SCON$/$/$/$/$/	
turn self-checking off	$SCOFF	
end of poetry foot sign	$FT	
computer braille	$CPB	
caesura sign	$CS	
call (permanent) tab t	$#t	
octal braille	$OCT	
start forced contraction	/_	
end forced contraction	_/	

APPENDIX VI

EQUIVALENT SIGNS, SYMBOLS AND CODES

	0	1	2	3	4	5	6	7	
0		¢	(@) 5	(") 45	¬	_	. =	; 456	_
	00	08	16	24	32	40	48	56	
1	A	C	E	D	CH	* SH	% WH	: TH	?
	01	09	17	25	33	41	49	57	
2	,	1 I	:	3 J	EN	5 OW	, .	4 W	
	02	10	18	26	34	42	50	58	
3	B	F	H	G	GH	< ED	$ OU	(\) ER	(])
	03	11	19	27	35	43	51	59	
4	'	ST	/ IN	9 AR	> -	ING	(+) "	0 # (zero)	
	04	12	20	28	36	44	52	60	
5	K	M	0	N	U	X	Z	Y	
	05	13	21	29	37	45	53	61	
6	;	2 S	TO	6 T	?	8 THE	!)(	7 WTH	(
	06	14	22	30	38	46	54	62	
7	L	P	R	Q	V	AND	& OF	) FOR	=
	07	15	23	31	39	47	55	63	



APPENDIX VII

TRANSLATOR - STACKER SIGN CODES

Code	Meaning
00	required blank
01-63	braille sign codes
64	end of braille word (blank or end of line)
65	new line
66	unused
67	paragraph start
68	one-time tab (skip to col. nn)
69	new page
70	unused
71	skip (multiple) lines
72	tab (skip according to permanent tab)
73-80	unused
81	start heading input
82	end heading input
83	unused
84	set continuous text stacking mode
85	idle (reserved for: reset continuous text stacking mode)
86	unit of measure
87	unused
88	start running title input

Code	Meaning
89	end of running title input
90	unused
91	self-checking mode off
92	self-checking mode on
93	set tab
94	start octal braille
95	start poetry mode
96	end poetry mode
97	start computer-braille
98	end or run
99	(filler in contraction table)

APPENDIX VIII

MISCELLANEOUS CODED VARIABLES

Variable	Code	Meaning
STACK-INDICATOR	2	Normal text; clear stack after each entry
	4	Continuous stacking of title or heading
	5	Continuous stacking of normal text
CURRENT-TYPE	1	Normal text or heading stack
	2	Title stack

Note: logical indicator variables are generally coded according to the convention 0 = "off" = "no" = "false," 1 = "on" = "yes" = "true."

REFERENCES

1. English Braille, American Edition, 1959 (Revised 1966), American Printing House for the Blind, Louisville, Kentucky.

2. J. K. Millen, DOTSYS II: Finite-State Syntax-Directed Braille Translation, MTR-1829, The MITRE Corporation, (1970).

3. J. K. Millen, DOTSYS II: User's Guide and Transfer and Maintenance Manual, MTR-1853, The MITRE Corporation, (1970).

4. U.S.A. Standard COBOL, American National Standards Institute X3.23-1968.

5. R. L. Haynes, "Computer Translation of Grade II Braille," Proceedings Conference on New Processes for Braille Manufacture, 1968, American Printing House for the Blind, Louisville, Kentucky, pp. 1-4.

6. B. M. Krebs, Transcribers' Guide to English Braille, The Jewish Guide for the Blind, New York, 1967.

DISTRIBUTION LIST

INTERNAL

C-01
C. W. Farr

D-06
P. R. Vance

D-07
J. H. Burrows
J. J. Croke

D-11
C. E. Duke
J. F. Jacobs

D-12
C. A. Zraket

D-63
R. S. Nielsen

D-72
T. L. Connors
C. G. Crothers
F. Engel
M. T. Gattozzi
W. R. Gerhart (5)
J. Mitchell
L. M. Thomas

D-73
W. Amory
N. W. Anschuetz
E. H. Bensley
J. A. Clapp
R. A. J. Gildea (20)
J. B. Glore
E. L. Lafferty (7)
J. F. Jacobs
J. K. Millen (20)
C. M. Sheehan
J. E. Sullivan (5)
N. B. Sutherland
E. W. Williamson

PROJECT

G. Dalrymple, M.I.T.
M. Leonard, M.I.T.
Prof. R. W. Mann, M.I.T.
V. A. Proscia, M.I.T. (50)

EXTERNAL

H. Bassler
Colonial Penn Insurance Co.
112 South 16th Street
Philadelphia, Pennsylvania 19102

Dr. M. P. Boyles (10)
Director, Computer-Braille Project
Instructional Services Center
Atlanta Public Schools
2930 Forrest Hill Drive, S. W.
Atlanta, Georgia 30315

L. L. Clark (2)
Director, IRIS
American Foundation for the Blind
15 West 16th Street
New York, New York 10011

E. L. Glaser
Computation Center
Case Western Reserve University
University Circle
Cleveland, Ohio 44106

Dr. C. E. Hallenbeck
Department of Psychology
University of Kansas
Lawrence, Kansas 66044

R. Haynes
American Printing House for
the Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

EXTERNAL (Concluded)

Dr. K. R. Ingham
Room 20-B-207
Massachusetts Institute
of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts 02139

R. E. LaGrone
IBM Corporation
Federal Systems Division
Department PC4, Room 2P25
18100 Frederick Pike
Gaithersburg, Maryland 20760

Dr. L. Leffler
Applied Mathematics Division
Argonne National Laboratories
9700 South Cass Avenue
Argonne, Illinois 60440

R. J. McNaughton
RCA Aerospace Systems Division
Burlington, Massachusetts 01801

Prof. A. Nemeth
Mathematics Department
University of Detroit
4001 W. McNichols
Detroit, Michigan 48821

Dr. B. Perella
Department of Defense
Fort George Meade, Maryland

A. Schack
Schack Associates
127 West 12th Street
New York, New York 10011

J. Siems
American Printing House for the
Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

Dr. E. J. Waterhouse
Director
Perkins School for the Blind
175 North Beacon Street
Watertown, Massachusetts 02172

V. Zickle
American Printing House for the
Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

FOREIGN

P. W. F. Coleman
4 George Street
Eastleight, Hants
S05 4 BU, United Kingdom

C. W. Garland
Royal National Institute for
the Blind
224-6 Great Portland Street
London, W. 1, England

R. House
c/o Frank Giannone
1510 E. 11th Avenue
Vancouver 13, B.C.
Canada

FOREIGN (Concluded)

Hans Karlgren
Research Group for Quantitative Linguistics
Sodermalmstorg 8,
11645 Stockholm
Sweden

Dr. G. Lamprecht
Wistfalische Wilhelms Universitat
44 Munster, Germany
Roxeler Strasse 64

B. Lindqvist
De Blindas Forening
Utvechlingsavdelningen
Gotlandsgatan 46
116 65 Stockholm
Sweden

J. Lindstrom
Handikappinstitutet
Pa Iach, 161 03 Bromma 3
Sweden

V. Mokleby
Husby OFF Skole for Blinde
Hovseterveien 3
Oslo 7
Norway

W. Sorke
Deutsche Blindenstudienanstalt
355 Marburg/Lahn
Am Schlag 8
West Germany

J. Vinding
Statens Institut for Blinde og Svagsynede
Rymarksvej 1
2900 Hellerup
Denmark

Prof. H. Werner
Institute fur Numerische und Instrumentelle Mathematik
Universitat Munster
Schlossplatz 5
44 Munster, West Germany