S. J. Sekiluv

TM-1908/100/00

Stylistic Analysis

Report on the First Year of Research

1 March 1965

The research reported herein was conducted under SDC's independent research program and Contract Nonr-4427(00), Office of Naval Research, U. S. Navy.

TM-1908/100/00

TECHNICAL MEMORANDUM

(TM Series)

The research reported herein was conducted under SDC's independent research program and Contract Nonr-4427(00), Office of Naval Research, U. S. Navy.

Stylistic Analysis

Report on the First Year of Research

by

Sally Yeates Sedelow

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

March 1, 1965

(page 2 blank)

Stylistic Analysis

Report on the First Year of Research

by

Sally Yeates Sedelow

ABSTRACT

This document describes the progress and results of the first year of research on a computer-based system for analyzing the style and thematic content of natural language text. A comprehensive summary of the program system is provided, as well as an analysis of the results obtained with this system to date. In addition, detailed documentation of the programs and their operation is provided, including flow charts and complete program listings.

(page 4 blank)

TABLE OF CONTENTS

Section One

I. Introduction 5
II. Description of System Programs 7
A. Initial Sort and Index 7
B. Alphabetic Sort and Examination of Input Text for Words with Common Root 7
C. Thesaurus-Building Section of Stylistic Analysis Program . . . 9
III. Conclusions 11
A. VIA 11
B. MAPTEXT 17
C. A Look Ahead 18
IV. Professional Activities 20

Section Two

I. Introduction 22
II. Initial Sort and Index 23
A. Tables 23
B. FORTRAN Section 23
C. TAC Section 24
D. Subroutine NXTWD 29
E. Flow Chart 30
F. Program Listing 31
G. Output 35
III. Alphabetic Sort and Examination of Input Text for Words with Common Root 36
A. Tables 36
B. Items 38
C. Main Routines 38
D. Subroutines for SUFFIX (Root-Finding Program) 43
E. Special Note on the Letter Rule 47
F. Flow Chart 48
G. Printout of Function Word List 50
H. Printout of List of Suffixes and Exceptions 51
I. Program Listing, SORDIT and SUFFIX 59
J. Output 70
IV. Thesaurus-Building Program (THESAUR) 71
A. Tables 71
B. Items 71
C. Main Routine 72
D. Subroutines for THESAUR 78
E. Flow Chart 80
F. Program Listing 85
G. Output 105

SECTION ONE

I. Introduction

This is the first annual report on stylistic analysis research, carried out at the System Development Corporation under a grant from the Office of Naval Research, Information Systems Branch, Contract Nonr-4427(00). I want to express my appreciation to the Office of Naval Research for their generous financial assistance; thanks are also due to the System Development Corporation for administrative and financial aid, and to Saint Louis University, of whose faculty I am a member, for providing time to pursue this research.

I also want to acknowledge the fine work of Terry Ruggles, of the System Development Corporation, who is responsible for the coding and for aspects of detail in the program design. He has contributed the flow charts to Section Two of this report, as well as helpful commentary about other details of the program descriptions.

The research is directed toward the development of an automated self-adapting system for performing stylistic analysis. Stylistic analysis is the procedure of recognizing patterns formed in the process of linguistic encoding of information. Invariably there is some stylistic (structural) pattern shaping natural language in which information is encoded. Such patterns serve as clues to the methods used to transmit the information and may be used to help detect information structured to elicit certain desired responses. An intense study of pattern, or style, in natural language should contribute markedly to the efficiency with which information, both transmitted and received, can be used. The recognition of stylistic patterns is important for the automatic production of language as well as for the

analysis of information. To achieve good, idiomatic machine translations and automatic abstracts, stylistic algorithms will be necessary.

Our work thus far has been mainly concentrated on developing a procedure to aid information analysis. This procedure, a self-adapting thesaurus-construction program, will delineate word-association patterns and organizing ideas, or concepts, within a given data base. This "thematic" aspect of style is important when trying to distinguish the written records of one writer, or one country, or one military or diplomatic policy, from that of another. In any of these cases, it is extremely useful to be able to distinguish stylistic "signatures."

The remainder of this portion of the report (Section One) includes a general and comprehensive summary of the program system we have devised to discern these differences in style, a statement of the conclusions we have reached, and a list of papers and lectures given on subjects that have either grown out of this research or have helped further it. The conclusions are forward-looking as well as backward-looking; that is, they are concerned with the directions the research should take, based on what has already been achieved.

Section Two of this report contains detailed descriptions of the programs developed thus far. We have tried to make these descriptions comparatively thorough, providing verbal and diagrammatic guides as well as the program listings. Some sections of our procedure, such as that which groups words by roots (Section Two, III), may well be of value to other researchers who deal with a similar problem. We have aimed, therefore, at very explicit explanations and outlines of the programs--explicit enough, we hope, so that a reader who knows little about programming will be able to follow them.

II. Description of System Programs

For convenience of explanation, the system of stylistic analysis programs is described in three sections: "initial sort and index"; "alphabetic sort and examination of input text for words with common root"; and "thesaurus construction and printout." The programs are written for the Philco 2000, which has a 16K memory, plus eight tape drives and a drum. Our program currently uses five tape drives, in addition to the central core memory. The Philco 2000 is a fixed-word-length computer, each word having 48 bits. Since each alphameric character occupies six bits, each computer word holds eight alphameric characters. Three registers, the A, D, and Q, each one computer word in length, are used to manipulate input data and perform arithmetic operations. The programs are written in FORTRAN and, when necessary, TAC, the machine-oriented language for the Philco 2000.

A. Initial Sort and Index

The first main section of the program sorts the input text into four-computer-word entries, each consisting of one textual word and an index showing where in the text the word occurred. The input text is punched on cards in columns 1-72 (columns 73-80 are used for sequencing the deck). Words are separated from each other by blanks, and punctuation marks are separated both from words and from each other by blanks.

B. Alphabetic Sort and Examination of Input Text for Words with Common Root

The first phase, SORDIT, of this programming procedure arranges the four-computer-word textual word entries in alphabetical order and reads them out on tape.

The second phase, SUFFIX, finds words which have common roots and indicates this identity by giving them an identical MATCNT. The MATCNT is placed in the previously empty fourth computer word in the four-computer-word textual entry. It is necessary to group words according to their roots because the thesaurus is constructed on semantic principles and must work, therefore, with root meanings. As the thesaurus is currently set up, function words are not listed; when SUFFIX senses a function word, the program moves on to an examination of the next textual word.

The procedure for finding roots is an adaptation of work done by Keren McConlogue, of the System Development Corporation, and it represents a departure from the more traditional procedure of matching putative word endings against suffix lists of varying length. The technique upon which we settled is to group together content words for which the first three letters are identical. Then, additional letters are matched until the words deviate from each other; the assumption is that the point of deviation may mark the end of the root form and the beginning of the suffix. One way to check the validity of this assumption would be to compare the putative suffix with suffixes on a list; but this procedure is wasteful of time. Simply removing the deviant sections of the words would lead to too many mistakes and mispairings in the thesaurus. The procedure, therefore, is to work with two words at a time, selecting the suffix which alphabetically precedes the other, and then searching a relatively short suffix list for the suffix. If the suffix occurs, it is linked to possible "pairing" suffixes in another table. For instance, if the program were working with the words "rain" and "rainy," it would discover that the words matched through "n."

The short suffix list would then be searched for "blank," a legal suffix in this system. "Blank" would be linked to the list of possible pairing suffixes in the other table and this list would be searched for a "y." When the "y" is found, the program would switch to another linked table, which lists exceptions to the suffix pair, "blank" and "y" ("bus," for instance, would be an exception). If the word does not occur in the exception list, the "y" is removed. Thus, to remove the "y," three short lists have been examined rather than the huge lists which would otherwise need to be searched, first to find the "y" and then to look through the many exceptions to the operating rule which, in this case, would be, simply, that every final "y" is a suffix. If the first "suffix" had not occurred in the initial table, then the program would proceed on the assumption that it had not found a suffix and the operations on that word would be abandoned. Obviously, no operations at all are conducted on words for which there are no "matches" for the initial three letters.

SUFFIX prints out a list of the textual words, grouped by MATCNT, and a frequency count of the words in each root-group.

C. Thesaurus-Building Section of Stylistic Analysis Program

This section of the program produces a thesaurus tape and prints out a listing of synonymous linkings among the words in the text base. Words that are necessary for THESAUR to process a body of text are extracted from a "thesaurus." (The contents of this thesaurus differ, depending on the state of the system, and will be explained below.)

The thesaurus consists of pairs of words (primary and associated). THESAUR extracts primary words from the thesaurus, and compares them against words occurring in the text. When a match is found, the primary word and its associated

word are retained. The associated words are then compared against text words, and when a match is found, the associated word and its primary word are retained. By this sequence of operations, we reduce the data base (and thus the required processing) to those words relevant to a particular body of text.

Working with this reduced thesaurus, primary words are again compared against the text. When a match is found, the associated word is matched against other primary words. If a second match is found, the second primary word's associated word is linked to the original primary word. These linkings are saved on tape and form a rudimentary text-oriented thesaurus.

The contents of the thesaurus depend on the experience of the system. In the beginning, the thesaurus tape will contain only a paired list of "primary" words and possible-associated-words that have been selected by the researcher from context, synonym dictionaries, etc., and input by punched cards. As the system grows, the thesaurus tape will contain this manual listing augmented by all the linkings detected by THESAUR in its prior manipulations of the text. In successive processings of the text, we may add new synonyms to the computer's taped thesaurus by card input. The system will then search, as before, for possible linkings between primary and associated words from the thesaurus. In this way, cross linkings are formed and saved on tape and the total thesaurus grows.

At the end of each run, THESAUR prints out the input text in semialphabetical order, including lists of associated words linked to primary words. These, in turn, may be linked to other primary words (to a depth of five).

III. Conclusions

A. VIA

The VIA (Verbally Indexed Associations) or thesaurus-building program is in its final development phase. As initially conceived, this program would provide a conceptual or thematic outline of a given verbal data base. Even our very general, preliminary results--representing program testing, rather than attempts at textual analysis--suggest that the initial idea was a good one.

Figure A shows some excerpts from the output of a computer-run using Chapter 1 of Sokolovsky's Soviet Military Strategy (Air Force translation) as the input text. The means of achieving this output are described in detail, both verbally and in flow charts, in Section Two of this report. It should be noted that the VIA program at present has the capability of showing five levels of associations, although the examples in Figure A have only three in one case, and four in the other.

So far as the actual content of the lists is concerned, the conventional thesaurus procedure of including some antonyms as well as synonyms has been observed. For instance, in the list under "war" (which, in turn, is linked to "military"), the words "pacific" and "peace" appear on a list which includes "belligerent" and "conflict." A list may also include words which occur in the immediate context of a given primary word (the primary words in the lists referenced by the number 701 are "military," "war," and "strategy"). In Figure A, the indices for the listed words are suppressed, but VIA does have an indexing feature, which enables contextual searches. This indexing feature is exemplified in Figure B.

251 COUNTRIES 72
COUNTRY'S
COUNTRY

- CONTROL
- DICTATION
- DOMINATION
- INFLUENCE
- MASTERY
- PATRIOTIC
- POLICY
  - DESIGN
  - OUTLINE
  - PLAN
  - PROGRAM
- POWER
- SOVEREIGNTY
- STATE

701 MILITARILY 115
MILITARISTIC
MILITARIST
MILITARY

- GENERAL
- MARSHAL
- WARFARE
- WAR
  - AVIATION
  - BATTLEFIELD
  - BATTLE
  - BELLIGERENT
  - COMBAT
  - CONFLICT
  - GUNPOWDER
  - MILITARY
  - PACIFIC
  - PEACE
  - STRATEGY
    - TACTICS

FIGURE A

```
341  COUNTRIES'      (0001--0284--0001--0019)
     COUNTRIES       (0001--0228--0004--0008)(0001--0195--0002--0005)(0001--0260--0001--0030)(0001--0195--0001--0020)
                     (0001--0248--0003--0016)(0001--0224--0002--0018)(0001--0198--0002--0012)(0001--0233--0001--0010)
                     (0001--0257--0005--0014)(0001--0249--0001--0029)(0001--0228--0005--0020)(0001--0259--0002--0017)
                     (0001--0248--0002--0031)(0001--0248--0004--0025)(0001--0202--0002--0019)(0001--0260--0001--0017)
                     (0001--0248--0001--0034)(0001--0248--0002--0044)(0001--0260--0002--0050)(0001--0271--0002--0012)
                     (0001--0280--0004--0033)(0001--0265--0001--0005)(0001--0270--0002--0079)(0001--0272--0001--0021)
                     (0001--0266--0001--0025)(0001--0263--0001--0017)(0001--0273--0001--0046)(0001--0062--0003--0009)
                     (0001--0114--0002--0017)(0001--0095--0001--0012)(0001--0056--0003--0027)(0001--0107--0001--0007)
                     (0001--0115--0001--0016)(0001--0144--0003--0027)(0001--0150--0001--0014)(0001--0149--0002--0027)
                     (0001--0187--0002--0022)(0001--0142--0001--0032)(0001--0150--0001--0033)(0001--0149--0001--0029)
                     (0001--0148--0001--0006)(0001--0134--0001--0018)(0001--0148--0001--0028)(0001--0157--0001--0018)
                     (0001--0179--0001--0024)(0001--0183--0001--0070)(0001--0174--0002--0005)(0001--0150--0002--0023)
                     (0001--0141--0007--0021)(0001--0156--0002--0022)
     COUNTRY'S       (0001--0243--0002--0008)(0001--0214--0002--0003)(0001--0232--0004--0017)(0001--0189--0001--0032)
                     (0001--0186--0002--0041)(0001--0179--0003--0013)(0001--0169--0001--0025)(0001--0190--0001--0088)
                     (0001--0186--0003--0009)(0001--0140--0001--0036)(0001--0140--0001--0023)(0001--0048--0002--0039)
     COUNTRY         CONTROL         DICTATION       DOMINATION      INFLUENCE       MASTERY
                     PATRIOTIC       POLICY          POWER           REALM           SOVEREIGNTY
                     STATE
                     (0001--0220--0004--0008)(0001--0244--0002--0037)(0001--0241--0001--0005)(0001--0231--0003--0021)
                     (0001--0243--0002--0022)(0001--0239--0001--0032)(0001--0238--0001--0036)(0001--0054--0001--0034)
                     (0001--0068--0006--0024)(0001--0053--0002--0022)(0001--0051--0001--0012)(0001--0056--0003--0022)
                     (0001--0096--0004--0007)(0001--0184--0001--0023)(0001--0169--0003--0005)(0001--0175--0006--0008)
                     (0001--0190--0001--0050)(0001--0130--0002--0002)(0001--0141--0006--0011)(0001--0124--0001--0024)
                     (0001--0169--0004--0008)(0001--0175--0002--0018)(0001--0142--0004--0018)(0001--0130--0001--0036)
                     (0001--0138--0002--0016)(0001--0175--0005--0008)(0001--0141--0005--0008)(0001--0190--0001--0119)
                     (0001--0138--0003--0002)(0001--0169--0001--0005)(0001--0179--0003--0029)(0001--0180--0003--0026)
                     (0001--0050--0001--0029)(0001--0042--0001--0094)(0001--0047--0002--0028)(0001--0032--0003--0006)
CONTROL              (0001--0263--0003--0009)(0001--0175--0005--0011)
CONTROLS             (0001--0109--0002--0004)
                       .     .     .     .
                       .     .     .     .
                       .     .     .     .
                           etc.
```

FIGURE B

The number to the right of "countries" and "militarily" in Figure A indicates the number of occurrences of the words on the first level, which contains those words initially grouped together on the basis of root. The count for words appearing on lower levels appears elsewhere in the listing. The discrepancy between the count for "country" and its root group in Figure A and the count obtained by totting up index entries in Figure B shows the difference between two translations (Figure A, 72; Figure B, 102; Difference, 30). Figure A is based on the Air Force translation, published by Praeger, and Figure B on the Dinerstein, Goure, and Wolfe translation, produced for The RAND Corporation and published by Prentice-Hall, Inc.

The discrepancy between the counts in Figures A and B suggests, on an elementary level, that another of the goals of the stylistic analysis research may be realizable. This goal is to develop the capability of distinguishing between and among styles, so that (for instance) interpolations within a work by another hand might be detected. Such interpolations might well imply a "rewriting" indicative of changes or sensitivities in policy. It is, of course, obvious that VIA has a strong assist, so far as recognizing discrepancies is concerned, when dealing with two translations of the same work; because the subject matter is unchanged, differences in word choices for any given passage distinguish one translation from another. A rather more rigorous test would entail comparing the VIA output for a section or chapter from one of the translations, but not both, with the output from control sections of both translations. This test might be most successful when based upon associations involving words used esoterically and infrequently, rather than upon high-frequency, heavily content-dependent, primary words. For instance, word associations with the adjective "acid," which

occurs in Chapter 1 of the Air Force translation, but not in the RAND translation, might reveal consistent word preferences in the Air Force translation.

A glance at a full VIA listing, which includes every word--with the exception of function words*--occurring in a given input text, suggests that there may be grammatical distinctions between the translations; one translation, for instance, may have a higher incidence of adverbial "ly" forms than the other. Some grammatical forms, as well as other variables such as the incidence of certain function words, can be measured and displayed by MAPTEXT, the second part of the stylistic analysis system, which is described in B, below.

VIA's operation will be tested in a number of different ways. One method will be to continue to run and rerun VIA on the same input text until a sizeable number of associations is attained. Another method will be to examine a small section of text initially, subsequently adding sections until finally the entire chapter is being run. Yet another method will be to operate upon small sections individually, rather than cumulatively, so that comparisons can be made among sections.

The VIA programs described in detail in Section Two of this report are reasonably well stabilized, except for certain sections of the program entitled THESAUR. We are still experimenting with variant ways of maintaining the integgrity of the lists while, at the same time, preserving the associations; it may

*Function words are conventionally defined as containing word types such as prepositions, conjunctions, etc.; for a listing of words we have tentatively designated as function words, to facilitate the Sokolovsky analysis, see Section Two, III-G.

be that we will use a list processor for some parts of the THESAUR section.

Before leaving the discussion of VIA, it should be noted that VIA is distinguished from other programs, including the well-known General Inquirer, in two important ways: 1) VIA's output is based upon words which occur in a given text and includes only words appearing in that text; 2) VIA has a self-adapting capability, so that as the computer's thesaurus and cross-references grow, the need for manual selection of prospects for the thesaurus output is correspondingly reduced.

B. MAPTEXT

MAPTEXT derives its name from the fact that it will provide an abstract representation of a given variable or set of variables in an input data base. For instance, if a general look at the distribution of function words and content words in a given data base were desired, MAPTEXT might be instructed to represent each content word with a 0, and each function word with an asterisk. If a breakdown of particular function words, such as the connectives "but" and "and," were desired, MAPTEXT might represent "but" with an asterisk, "and" with a $, and everything else with a 0. The number of variables which could be "mapped" onto the abstract representation is extensive.

MAPTEXT is intended as an aid to intuition. It represents an effort to remove verbal variables from the verbal context which, because of its many associations, interferes with the perception of the patterns formed by distinct variables. A version of MAPTEXT has already been used by researchers at UCLA to study certain metrical patterns.

This representational aspect of MAPTEXT exists only in an experimental LISP

program. (For this reason, no documentation of MAPTEXT programs appears in Section Two.) Upon completion of the current VIA project, we will begin putting MAPTEXT into more permanent form.

We will also want to combine some statistical capabilities with the representational aspect of MAPTEXT. Statistical checks can be made upon either the abstract or the verbal form of the input data base. The important aspect of MAPTEXT is that it will provide both visual and statistical guides toward the detection of potent stylistic discriminators.

C. A Look Ahead

It is probably the case that adequate stylistic discrimination will entail some combination of the measures provided by VIA and MAPTEXT. Mosteller and Wallace's study of The Federalist Papers suggested that the incidence of certain function words might be a sure guide to an individual's "subconscious" stylistic mannerisms. When working on the authorship of the Junius letters, Ellegard relied solely on the incidence of content words. This procedure, which Mosteller and Wallace felt to be so unsatisfactory when working with The Federalist Papers, seemed to work very well for the Junius letters.

Presumably, none of these approaches is inevitably good or inevitably bad. We want to test the hypothesis that the combined strengths of VIA and MAPTEXT will provide a more powerful analytical approach.

As it stands, it seems that VIA alone will be of immense aid in getting at both organizing and esoteric concepts in a given data base. With the addition of MAPTEXT, the system will have the ability to explore the nonconceptual elements

in the data base, as well as to represent possible grouping of the conceptual elements, if that is desired.

For the coming year, we look toward the completion and testing of VIA and the beginning of an implementation of MAPTEXT.

IV. Professional Activities

1. Sedelow, S. Y., and Bobrow, D. G. A LISP program for use in stylistic analysis. SDC document TM-1753, Santa Monica, California, System Development Corporation, February 17, 1964.
2. Sedelow, S. Y., and Sedelow, W. A., Jr. A preface to computational stylistics. SDC document SP-1534, Santa Monica, California, System Development Corporation, February 17, 1964.
3. Sedelow, S. Y. Computational stylistics. Presented at the 2nd Annual Meeting of the Association for Machine Translation and Computational Linguistics, Indiana University, Bloomington, July 29-30, 1964.
4. Sedelow, S. Y., Sedelow, W. A., Jr., and Ruggles, T. L. Some parameters for computational stylistics: Computer aids to the use of traditional categories in stylistic analysis. Presented at the Literary Data Processing Conference, IBM Corporation, Yorktown Heights, New York, September 9-11, 1964. Published in the Proceedings of the IBM Literary Data Processing Conference, IBM, Yorktown Heights, 1964, pp. 211-229.
5. Sedelow, S. Y. Stylistic analysis and the information sciences. Presented at the Annual Meeting of the American Documentation Institute, Philadelphia, October 5-8, 1964.
6. Sedelow, S. Y. Computer skills for literary scholars. Presented at Conference 27, The Use of Computers for the Study of Language, Modern Language Association Meetings, New York, New York, December 27-29, 1964.
7. Sedelow, S. Y. Elected chairman of Conference 27 (The Use of Computers for the Study of Language), Modern Language Association, for the year 1965.

8. Sedelow, S. Y. Computers and linguistic analysis. Presented at Southern Illinois University, Carbondale, Illinois, February 19, 1965.

9. Sedelow, S. Y. Quarterly report, 1 March 1964 to 1 June 1964. SDC document TM(L)-1908/001/00, Santa Monica, California, System Development Corporation, May 27, 1964.*

10. Sedelow, S. Y. Quarterly Report, 1 June 1964 to 1 September 1964. SDC document TM(L)-1908/002/00, Santa Monica, California, System Development Corporation, August 25, 1964.

11. Sedelow, S. Y. Stylistic Analysis, Annual Summary Report (Accomplishment Summary) for Calendar Year 1964. SDC document TM(L)-1908/003/00, Santa Monica, California, System Development Corporation, October 23, 1964.

12. Sedelow, S. Y. Quarterly Report, 1 September 1964 to 1 December 1964. SDC document TM(L)-1908/004/00, Santa Monica, California, System Development Corporation, November 18, 1964.

*Documents designated "(L)" were prepared for limited circulation only. They are administrative documents required by the contractual reporting guidelines.

SECTION TWO

I. Introduction

This portion of the report provides detailed coverage of the program operation. A general explanation of the three main portions of the program system appeared in Section One. Here, under parallel headings, we expand on that general explanation, providing complete annotated lists of the tables and items in each program section of the main program routine, and of major subroutines. Following this verbal explanation, a flow chart is included to give the reader a graphic and comprehensive grasp of the program's function and operation. For those interested in even greater detail, a complete program listing is provided; and as a last item, we offer an indication of the program's output.

It may be felt that the names for tables and items are not always as mnemonic as might be ideal. This condition results from a constraint imposed by the FORTRAN language. When using FORTRAN, it is necessary to indicate fixed-point mode by using, as a first letter in a symbolic name, letters within the alphabetic range I through N.

II. Initial Sort and Index

A. Tables

KARD (10 computer words): Holds the contents of one Hollerith card. KARD is used only for the transition of the input text.

KWDS (100 computer words): The input text is transferred from KARD into KWDS, which holds the contents of ten Hollerith cards.

MATT (1024 computer words): Each entry (four computer words) in MATT holds one textual word and its index. The first two computer words contain the textual word (if the textual word is shorter than 16 characters--eight characters per computer word--the extra spaces at the end are left blank; the textual word is left-justified), and the third computer word contains the index entry, or the "count." The fourth computer word in the entry is left blank during this first section of the program.

The input text is taken from KWDS and manipulated until it is in the form appropriate for MATT. Specifically, in KWDS, several textual words may occupy one computer word, or a textual word may begin at the end of one computer word and "run over" into the next. In MATT, each textual word has a separate entry and must be left-justified in the first (and second, if it continues) computer word in that entry.

B. FORTRAN Section

This first section of the program is the input routine.

The first instruction, READ 1,KARD, reads in the "case" card. Since only the first card word in the "case" card is used by the Philco operating system, other words in the card can be used for setting up Hollerith constants (Hollerith

constants cannot be directly set up in FORTRAN II or ALTAC). Therefore MATCH is set to THEEND, the second word in the case card. THEEND signals the end of the input data deck and will appear at the end of the data deck. KARD(3), the third word in the case card, consists of blanks; this constant is used in the first DO-Loop. FORMAT statement 1 says that the input data consists of alphameric characters on cards, which consist of ten "words," each of which occupies eight columns. The first DO-Loop fills table MATT with blanks, so that the empty fourth computer word in each entry will not be filled with non-meaningful characters at the end of this section of the program. The second and third DO-Loops (the third is nested within the second) read the input cards into KARD and then into KWDS. Since KWDS holds ten KARDS, M, a counter, is incremented by ten each time KARD is emptied so that KWDS will be filled sequentially. The contents of the counter N, which advances one computer word at a time, when added to the contents of M, provide for the sequential advance through KWDS. When either THEEND(MATCH) is reached or KWDS is full, program control is advanced to the TAC section of the program. This section manipulates the textual words, counts them, places them in MATT and, when MATT is full, transfers its contents out onto tape. If THEEND has been reached, the program halts. Otherwise, more cards are read in.

C. TAC Section

The basic procedure here is to examine each computer word in KWDS, one character at a time, beginning at the left. The characters in the KWDS word are shifted, one at a time, into the left-most position of the Q register (TEMP is used for this operation) and examined. Punctuation marks are treated like

textual words. On the input cards, textual words are separated by blanks.

Index Register 1 indexes MATT.

Index Register 2 indexes KWDS.

Constants: The function of most of the constants is described by comments in the program.

Except for the first time through the program, the constant EXIT is used to return control from the subroutine NXTWD to the program instruction in the main program following a jump to NXTWD instruction.

TEMP is the storage cell which holds the KWDS word while it is being examined and shifted in the Q register. As the word is modified in the Q, it is also modified in TEMP because the Q is stored in TEMP after each shift.

WORD1, WORD2, and WORD3 are all used for temporary storage of a textual word. Three WORDs are required because up to 16 characters of the textual word are saved, and the textual word, as it appears in KWDS, very probably is not left-justified. Only after the complete textual word has been transferred into these storage constants is it then packed and left-justified into the space of two computer words or less. WORD1 is filled with blanks whenever manipulation of a textual word is about to begin. This procedure is necessary because the word is left-justified by checking for blanks (which are disposed of). If the byte (six bits) examined is not a blank, then the program assumes that a character has been reached.

At the beginning of the TAC section, the first word from KWDS is put into TEMP. A mask for the first six bits (which will contain a blank or a character), is put into the Q register, and the first six bits of TEMP are extracted. If

they comprise a blank, Q is shifted left six bits, and the next six bits are examined. This process continues until either a character is reached or all of the first computer word in KWDS has been examined. A zero signals the end of the first word in KWDS because, as the Q register is shifted left, the right is filled in with zeros. If the computer word being examined is all blank, the program branches to the subroutine NXTWD, which checks to see whether all of table KWDS has been examined. If it has not, the program proceeds through KWDS until a non-blank character is reached. When this occurs, the program branches to GOTCHA. At GOTCHA, the AQ registers are shifted six bits to the left, as a unit, so that the character which has just been found is transferred into the A register. In turn, the A register is transferred into WORD1.

The contents of the Q register are then transferred into TEMP and the first six bits in TEMP are extracted for examination. If the bits comprise a blank, the end of the textual word has been reached; if this is the case, the textual word is less than eight characters long and the program branches to SHIFTUM. If the six bits comprise a zero, the end of the computer word has been reached but the textual word may "run over" into the next computer word. If the six bits comprise an alphameric character, the character is added to the character(s) already stored in WORD1, and the next six bits in TEMP are examined. If the test for a zero is met, the program branches to subroutine NXTWD and then proceeds to the next computer word in table KWDS. Now, the storage cell, WORD2, is loaded in the same way that WORD1 was loaded. Six-bit units are examined until a blank or a zero is reached. In the former case, the program branches to SHIFTUM; in the latter case, it proceeds to the next KWDS word via NXTWD. If

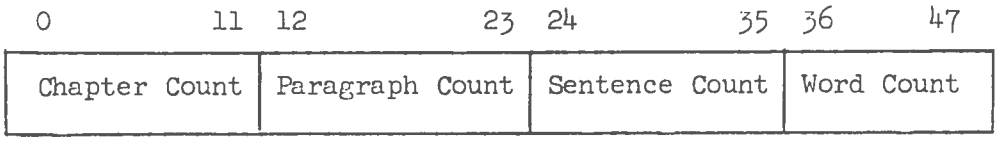
the latter case obtains, the storage cell WORD3 is filled. If a blank is reached, the program branches to SHIFTUM. If a zero is reached, the textual word contains more than the maximum 16 characters, so the rest are ignored by the program which proceeds until it reaches a blank, signalling the end of the over-long textual word and the beginning of the next one. Since the blank will often not occur at the end of the computer word, the bits following the blank--presumably containing characters--are saved in TEMP.

The section of the program labelled SHIFTUM begins with an examination of WORD3; if it contains only blanks, WORD2 is examined; if it, too, is blank, WORD1 is examined. If there are characters in WORD3, they are left-justified by shifting-circular the D register. This procedure retains any leading blanks by relocating them at the right; trailing blanks are, of course, also retained so that no zeros appear in the left-justified WORD3. Next, WORD2 is left-justified and its sequential connection with WORD3 is also retained, in the following way: if the first six bits in WORD2 are a blank, WORD2 is placed in the A register and WORD3 in the Q register, after which both registers are shifted left six bits; the newly shifted A register's contents are placed back into WORD2 (notice that WORD3, as it appears in the Q register, has had one character shifted into the A register and therefore a zero appears in the right six bits of the Q register--however the constant, WORD3, still has not been altered); next, place the contents of WORD3 in the A register, place a blank in the left six bits of the Q register, and shift the AQ registers left six bits; this shift disposes of the character which has already been placed in WORD2 and places a blank at the end of WORD3 (these blanks may ultimately be needed to fill out computer WORD2 in

the MATT entry; we know there is at least one character in WORD1, that WORD2 is filled, and that there is at least one character in WORD3; therefore there are at least ten characters, so at the most, six trailing blanks would be required and these would be supplied by the trailing blanks currently in WORD3); next, the A register is transferred into WORD3; this process is continued until WORD2 is left-justified. Next, the program disposes of leading blanks in WORD1, while keeping it sequentially connected with the contents of WORD1 and WORD2. The procedure is much like that already described. The contents of WORD1 are placed in the A register and the contents of WORD2 in the Q register; after the AQ registers are shifted left six bits, the contents of the A register are transferred into WORD1. Next, WORD2 (note that its contents have not been changed) is placed in the A register and the contents of WORD3 into the Q register. A shift left AQ of six bits disposes of the character which has just been placed in WORD1 and moves up the subsequent characters in WORDs 2 and 3. After the shift, the contents of the A register are placed in WORD2 and the contents of the Q register in WORD3. This process continues until there are no more leading blanks in WORD1. At that point, the textual word will be left-justified in WORDs 1 and 2 and the program proceeds to the indexing section.

At INDEX, the first computer word in each MATT entry is examined. If the word contains a period or a question mark, the sentence count is increased by one and the word count reset to zero, the punctuation mark itself will receive the count, zero, and the first word of the new sentence will, therefore, have a count of one. If there are two periods, the end of a paragraph has been reached; therefore the paragraph count is increased by one, the sentence count is reset

to one, and the word count to zero. If there are three periods, the end of the chapter has been reached and the chapter count must be increased by one, the paragraph and sentence counts must be reset to one, and the word count to zero. In each case, after setting the count, WORDs 1 and 2 are loaded into MATT and the counts are aligned in the following bit configuration:



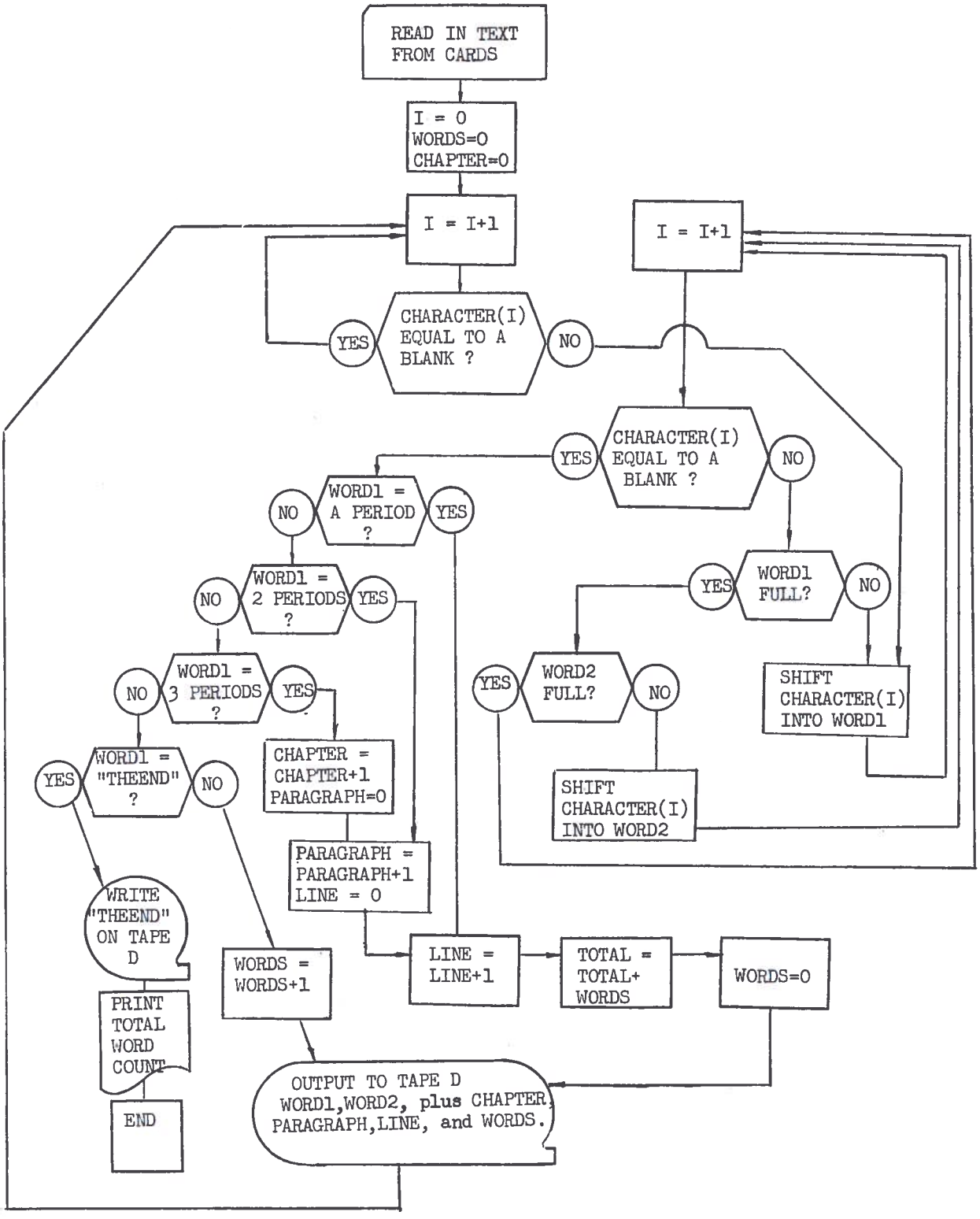
Computer Word

Then the count is loaded into MATT. If the MATT entry's first word contains THEEND, the rest of table MATT is filled with THEEND, MATT is read out on tape, and the program moves on to the second section, described in III. In every other case, a check is made to see if MATT is full. If not, the MATT counter (index register 1) is incremented by four, the next textual word is brought into the Q register (remember that the beginning of the word has been saved in TEMP), and the entire procedure is repeated. If MATT is full, it is read out on tape (N5T23 is the code for Core to Tape, Mode 2; N8T39 indicates that eight blocks are being read out on tape--table MATT is 8 × 128). Index register 1 is then reset, and the sorting and indexing process continued.

D. Subroutine NXTWD

This subroutine checks to see if all of table KWDS has been processed. If not, KWDS' counter is incremented by 1, so that it points to the next computer word in KWDS. If all of table KWDS has been processed, index register 1 (which indexes table MATT) is saved in INDEX1 and control is transferred to the read-in of more cards. Then the processing of the input data begins again.

E. Flow Chart


READ IN TEXT FROM CARDS
I = 0 WORDS=0 CHAPTER=0
I = I+1
CHARACTER(I) EQUAL TO A BLANK ?
YES
NO
I = I+1
CHARACTER(I) EQUAL TO A BLANK ?
YES
NO
WORD1 = A PERIOD ?
NO
YES
WORD1 = 2 PERIODS ?
NO
YES
WORD1 = 3 PERIODS ?
NO
YES
WORD1 = "THEEND" ?
YES
NO
WRITE "THEEND" ON TAPE D
PRINT TOTAL WORD COUNT
END
CHAPTER = CHAPTER+1 PARAGRAPH=0
PARAGRAPH = PARAGRAPH+1 LINE = 0
WORDS = WORDS+1
LINE = LINE+1
TOTAL = TOTAL+ WORDS
WORDS=0
OUTPUT TO TAPE D WORD1,WORD2, plus CHAPTER, PARAGRAPH,LINE, and WORDS.
WORD1 FULL?
YES
NO
SHIFT CHARACTER(I) INTO WORD1
WORD2 FULL?
YES
NO
SHIFT CHARACTER(I) INTO WORD2

F. Program Listing

INDEXING PROGRAM

```
      I     INDEX
            IDENTIFY F, 16K, 8X, 1000W
      DIMENSION KARD(10)                                                0001
      DIMENSION KWDS(100)                                               0002
      DIMENSION MATT(1024)                                              0003
      READ 1,KARD                                                       0004
    1 FORMAT(10A8)                                                      0005
      MATCH = KARD(2)                                                   0006
      DO 2 M = 1,1024,1                                                 0007
    2 MATT(M) = KARD(3)                                                 0008
    5 DO 3 M = 0,90,10                                                  0009
      READ 1,KARD                                                       0010
      DO 3 N = 1,10,1                                                   0011
      K=M+N                                                             0012
      KWDS(K) = KARD(N)                                                 0013
      IF(KARD(N)-MATCH)3,4,3                                            0014
    3 CONTINUE                                                          0015
    4 STARTTAC                                                          0016
            TMD     INDEX1    $                                       0017
            TDXLC   ,1        $                                       0018
            TIXZ    0,2       $   NUMBER OF CARD WORDS USED.          0019
      LO    TMQ     KWDS,2    $                                       0020
            TMA     A/        $                                       0021
      EXIT  JMP     L1        $   SUBROUTINE EXIT                     0022
      INDEX1 D/0              $   NUMBER OF WORDS FILLED IN MATTER    0023
      CCNT  D/1               $   CHAPTER COUNT                       0024
      PCNT  D/1               $   PARAGRAPH COUNT                     0025
      SCNT  D/1               $   SENTENCE COUNT                      0026
      WCNT  D/1               $   WORD COUNT                          0027
      CWCNT D/0               $   COMPLETE WORD COUNT                 0028
      TEMP  A/      $                                                 0029
      WORD1 A/      $                                                 0030
      WORD2 A/      $                                                 0031
      WORD3 A/      $                                                 0032
      L1    TAM     WORD1     $   (BBBBBBBBCARDWORD)                  0033
            TQM     TEMP      $                                       0034
            TMQ     0/77      $                                       0035
            ETA     TEMP      $   MATCH FIRST CHARACTER THAT WAS      0036
            TMD     0/60      $   WAS IN (Q) AGAINST A BLANK          0037
            JAED    L2        $                                       0038
            JAZ     (P)+2     $   THEY MUST HAVE ALL BEEN BLANK       0039
            TMA     WORD1     $                                       0040
            TMQ     TEMP      $                                       0041
            JMP     GOTCHA    $   FIRST CHARACTER THAT IS             0042
            JMP     NXTWD     $   LOAD NEXT WORD                      0043
            JMP     L1        $                                       0044
      L2    TMA     WORD1     $    GET RID OF LEADING BLANK           0045
            TMQ     TEMP      $                                       0046
            SLAQ    6         $                                       0047
            JMP     L1        $                                       0048
      GOTCHA SLAQ   6         $   (BBBBBBBCARDWORDO)                  0049
            TAM     WORD1     $                                       0050
            TQM     TEMP      $                                       0051
            TMQ     0/77      $                                       0052
            ETA     TEMP      $                                       0053
            TMD     0/60      $                                       0054
            JAED    SHIFTUM   $    LESS THAN 8 CHARACTERS            0055
            JAZ     (P)+2     $    MAYBE MORE?                        0056
            TMQ     TEMP      $                                       0057
```

INDEXING PROGRAM

```
                TMA     WORD1   $                                       0058
                JMP     GOTCHA  $                                       0059
                JMP     NXTWD   $   MORE THAN 8 CHARACTERS.             0060
        L3      TAM     WORD2   $                                       0061
                TQM     TEMP    $                                       0062
                TMQ     0/77    $                                       0063
                ETA     TEMP    $                                       0064
                TMD     0/60    $                                       0065
                JAED    SHIFTUM $    LESS THAN 16 CHARACTERS            0066
                JAZ     (P)+5H  $    MAYBE MORE?                       0067
                TMQ     TEMP    $                                       0068
                TMA     WORD2   $                                       0069
                SLAQ    6       $                                       0070
                JMP     L3      $                                       0071
                JMP     NXTWD   $   USE 3 WORDS BECAUSE THE FIRST       0072
        L4      TAM     WORD3   $   WORD (WORD1) MAY CONTAIN BLANKS      0073
                TQM     TEMP    $                                       0074
                TMQ     0/77    $                                       0075
                ETA     TEMP    $                                       0076
                TMD     0/60    $                                       0077
                JAED    SHIFTUM $   GOT MORE THAN ENOUGH                0078
                JAZ     L5      $   JUNK THE REST                       0079
                TMQ     TEMP    $                                       0080
                TMA     WORD3   $                                       0081
                SLAQ    6       $                                       0082
                JMP     L4      $                                       0083
        L5      JMP     NXTWD   $   LOAD NEXT WORD                      0084
        L6      TQM     TEMP    $   SAVE WHAT FOLLOWS THE BLANK.        0085
                ETA     0/77    $   THROW AWAY WHAT PRECEDES IT.        0086
                TMD     0/60    $                                       0087
                JAED    SHIFTUM $                                       0088
                JAZ     L5      $                                       0089
                SLQ     6       $                                       0090
                JMP     L6      $                                       0091
SHIFTUM         TMA     WORD3   $   WORD1,WORD2 AND WORD3 MAY           0092
                TMD     A/      $   CONTAIN BLANKS.                     0093
                JAED    (P)+5   $   SHIFTUM SHIFTS THE WORDS LEFT       0094
                TMQ     0/77    $   UNTIL BLANKS ARE SHIFTED OUT        0095
                ETA     WORD3   $   ON THE LEFT.                        0096
                TMD     0/60    $                                       0097
                JAED    (P)+1   $                                       0098
                JMP     (P)+5H  $                                       0099
                TMD     WORD3   $                                       0100
                SCD     42      $   TO RETAIN BLANKS ON THE RIGHT.      0101
                TDM     WORD3   $   LEFT JUSTIFY WORD3                  0102
                JMP     (P)-4   $                                       0103
                TMA     WORD2   $                                       0104
                TMD     A/      $                                       0105
                JAED    (P)+15H $                                       0106
                TMQ     0/77    $                                       0107
                ETA     WORD2   $                                       0108
                TMD     0/60    $                                       0109
                JAED    (P)+1   $                                       0110
                JMP     (P)+5   $                                       0111
                TMA     WORD2   $                                       0112
                TMQ     WORD3   $                                       0113
                SLAQ    6       $                                       0114
                TAM     WORD2   $                                       0115
                TMA     WORD3   $                                       0116
```

INDEXING PROGRAM

```
                TMQ     0/60        $                                   0117
                SLAQ    6           $   LEFT JUSTIFY WORD2              011A
                TAM     WORD3       $    AND WORD3                      0119
                JMP     (P)-13H     $                                   0120
                TMQ     0/77        $                                   0121
                ETA     WORD1       $                                   0122
                TMD     0/60        $                                   0123
                JAED    (P)+1       $                                   0124
                JMP     (P)+11H     $                                   0125
                TMQ     WORD2       $                                   0126
                TMA     WORD1       $                                   0127
                SLAQ    6           $                                   0128
                TAM     WORD1       $                                   0129
                TMA     WORD2       $                                   0130
                TMQ     WORD3       $                                   0131
                SLAQ    6           $                                   0132
                TAM     WORD2       $   LEFT JUSTIFY WORD1,             0133
                TDM     WORD3       $   WORD2 AND WORD3                 0134
                JMP     (P)-7       $                                   0135
INDEX           TMA     WORD1       $                                   0136
                TMD     A/.         $   PERIOD                          0137
                JAED    NEWSEN      $   OR                              0138
                TMD     0/36        $   QUESTION MARK                   0139
                JAED    NEWSEN      $                                   0140
                TMD     A/..        $                                   0141
                JAED    NEWPAR      $                                   0142
                TMD     A/...       $                                   0143
                JAED    NEWCHP      $                                   0144
                TMD     A/THEEND    $                                   0145
                JAED    NOMOR       $                                   0146
                JMP     LODUM       $                                   0147
NEWCHP          TMA     D/1         $   NEW CHAPTER                     0148
                TDM     SCNT        $                                   0149
                TDM     PCNT        $                                   0150
                AMS     CCNT        $                                   0151
                JMP     NEWORD      $                                   0152
NEWPAR          TMA     D/1         $   NEW PARAGRAPH                   0153
                TDM     SCNT        $                                   0154
                AMS     PCNT        $                                   0155
                JMP     NEWORD      $                                   0156
NEWSEN          TMA     D/1         $   NEW SENTENCE                    0157
                AMS     SCNT        $                                   0158
NEWORD          TMA     WCNT        $   NEW WORD                        0159
                AMS     CWCNT       $                                   0160
                CM      WCNT        $   RESET WORD COUNT                0161
LODUM           TMD     WORD1       $                                   0162
                TDM     MATT,1      $   UNLOAD WORD1, WORD2 AND WORD3   0163
                TMD     WORD2       $                                   0164
                TDM     MATT+1,1    $                                   0165
                TMA     CCNT        $                                   0166
                TMQ     PCNT        $                                   0167
                SLQ     36          $   PACK                            0168
                SLAQ    12          $   CHAPTER COUNT,                  0169
                TMQ     SCNT        $   PARAGRAPH COUNT,                0170
                SLQ     36          $   SENTENCE COUNT,                 0171
                SLAQ    12          $   AND                             0172
                TMQ     WCNT        $   WORD COUNT                      0173
                SLQ     36          $   INTO THE                        0174
                SLAQ    12          $   3RD WORD IN                     0175
```

```
INDEXING PROGRAM
                    TAM     MATT+2,1    $    EACH 4 WORD ENTRY OF MATTER          0176
                    TMD     A/          $                                          0177
                    TDM     WORD2       $                                          0178
                    TDM     WORD3       $                                          0179
                    TMA     D/1         $                                          0180
                    AMS     WCNT        $    NEW WORD                             0181
                    TMD     C/HLT,1024;C/HLT,L7$                                   0182
                    AIXJ    4,1         $    BUMP INDEX FOR MATTER BY 4            0183
               R    JMP     IOINT       $                                          0184
                    N/5T23;N/8T39       $                                          0185
                    C/HLT,TZZD;C/HLT,MATT$                                         0186
                    TIXZ    ,1          $                                          0187
              L7    TMA     A/          $                                          0188
                    TMQ     TEMP        $      RESTORE (A) AND (Q)                 0189
                    SLAQ    6           $                                          0190
                    JMP     L1          $                                          0191
            NOMOR   TMD     WORD1       $    FILL THE REST OF THE BLOCK WITH      0192
                    TDM     MATT,1      $    WHAT WAS IN THE LAST WORD1            0193
                    TMD     C/HLT,1024;C/HLT,NOMOR$              (THEEND           0194
                    AIXJ    1,1         $                                          0195
               R    JMP     IOINT       $                                          0196
                    N/5T23;N/8T39       $                                          0197
                    C/HLT,TZZD;C/HLT,MATT$                                         0198
                    TTD                 $                                          0199
                    TDA                 $                                          0200
                    JAN     (P)+1       $                                          0201
                    JMP     SYSTEM      $                                          0202
               S    BIN2ECD CWCNT       $                                          0203
               S    ZSUPP               $                                          0204
                    TAD                 $                                          0205
               R    RPTNA   8           $                                          0206
                    TDC                 $                                          0207
                    SCD     42          $                                          0208
                    JMP     SYSTEM      $                                          0209
            NXTWD   TJM     EXIT        $    GET NEXT WORD AVAILABLE              0210
                    TMD     C/HLT,100;C/HLT,L0$                                    0211
                    AIXJ    1,2         $                                          0212
                    TXDLC   ,1          $                                          0213
                    TDM     INDEX1      $                                          0214
                    ENDTAC$                                                        0215
          GO TO 5                                                                  0216
          COMMON    K,M,N,MATCH                                                  0217
          END                                                                      0218
     THEEND
```

G. Output

The output of this portion of the program is a tape consisting of textual words with their associated chapter-paragraph-line-word counts.

III. Alphabetic Sort and Examination of Input Text for Words with Common Root

A. Tables

KSUF1 (500 computer words): Input table which holds the left elements of suffix pairs as they appear on input cards.

KSUF2 (500 computer words): Input table which holds the right elements of suffix pairs as they appear on input cards.

MEXC (1000 computer words): Input table which holds the exceptions to suffix pairs. An example of the form of the input data occupying the above three tables is:

ING	Y
EXCEPT	BILLY

LEXC (500 computer words): Table which links KSUF2 to the exceptions in table MEXC. Entries in KSUF1 and KSUF2 are in one-to-one correspondence, but there may be more than one exception for each suffix pair; therefore, there is not a one-to-one correspondence between KSUF2 and MEXC. LEXC is set up as follows: each entry in LEXC corresponds to the analogous entry in KSUF2, e.g., entry #1 in LEXC corresponds to entry #1 in KSUF2, entry #2 in LEXC corresponds to entry #2 in KSUF2, etc. To find the list of exceptions for entry #1 in KSUF2 (and, therefore, for its paired suffix in KSUF1), the program looks at the contents of entry #1 in LEXC. In the initial, or first case the contents would be 1, pointing to the first computer word in MEXC as containing an exception to the suffix pair in entry #1 of KSUF1 and KSUF2; additional exceptions to this pair are contained in sequentially succeeding computer words in MEXC. The end of the exception list for this pair is signalled by placing a zero in the computer word following the last exception. If there were two exceptions for the suffix pair in KSUF1 and KSUF2, computer words 1 and 2 in MEXC would be occupied by the

exceptions and word 3 by the zero. The exceptions for the second entries in KSUF1 and KSUF2 would begin in MEXC's computer word 4 and the contents of entry #2 in LEXC would be 4. See example below:

	KSUF1
1	AL
2	ABLE

	KSUF2
1	E
2	E

	MEXC
1	CANE
2	CHORE
3	0
4	CAPE
5	0

	LEXC
1	1
2	4

MSUF1 (500 computer words): Table which contains suffixes listed in KSUF1, but duplicates have been eliminated.

MSUF2 (2000 computer words): Table which contains suffixes listed in KSUF2, but suffixes paired with eliminated duplicates in KSUF1 have been grouped with the suffix in KSUF2 which is paired with the appropriate (the one non-eliminated suffix in each set of duplicates) suffix in KSUF1. For example, all the suffixes in MSUF2 which pair with "ed" are grouped together.

LSUF2 (500 computer words): Table which links MSUF2 to MSUF1 in the same way that LEXC links KSUF2 and MEXC.

LEXL (2000 computer words): Table which links MEXC to MSUF2. The contents of LEXL are made up of entries in LEXC.

KFNC (200 computer words): Input table which contains a list of function words, so that they can be eliminated from the thesaurus if desired.

KARD (10 computer words): Table which holds the contents of one Hollerith card. This table is used for the transition of the input text.

MWORDS (128 computer words): Table which ccntains the input text, 32 textual words (each textual word occupies a four-computer-word entry) at a time.

MATCH (100 computer words): Table which contains textual words having a match (in the first three letters) with at least one other textual word.

B. Items

NFNC: Contains the count of the number of function words in table KFNC.

NSUFX: Indexes and counts number of entries in KSUF1 and KSUF2, and later, in MSUF1.

KLET: Contains word "LETTER" (item on input exception lists).

KEND: Contains word "THEEND," which indicates end of textual or input data.

KEXC: Contains word "EXCEPT," used to check for exceptions when inputing suffix pairs.

KBLNK: Contains computer word of blanks.

WORDS (floating point item): Contains count of the total number of words in the input text, including function words.

FORDS (floating point item): Contains count of the number of function words appearing in the input text.

LEMP: A fixed-point temporary storage cell.

NINP (number of inputs): Indexes table MWORDS.

NMAT: Counts the number of computer words in table MATCH, which are filled with data.

C. Main Routines

SORDIT: This routine uses an existing system program to sort the input text alphabetically.

Main SUFFIX routine (this routine will be described sequentially in terms of the comments which appear in the program listing):

1. READ FUNCTION WORDS INTO KFNC: This section of the program reads the list of function words into table KFNC and sorts them into alphabetical order. The card input form is one function word per card, with the function word occupying columns 9-17 (the second card "word," according to FORMAT statement 1) of the card. The STOP in the DO-Loop is a check against an overflow of table KFNC. Following the input DO-Loop, the alphabetical sort is performed by a shuttle-sort routine.

2. READ SUFFIX INFO INTO KSUF1, KSUF2, MEXC: This section reads suffix pairs and their exceptions into KSUF1, KSUF2, and MEXC. The first entry, set to zero, in MEXC, does not refer to a KSUF1-KSUF2 entry. The instruction appears here for coding convenience, because it will be used repeatedly to indicate the end of a given exception list. Therefore, the entry number used to exemplify tables KSUF1, KSUF2, MEXC, and LEXC in III-A are not quite accurate; they were used to avoid confusion when explaining the linking logic. Statement 5 in this section is a check against overflowing the tables. If NSUFX exceeds the table size, the program goes to statement 10 (at the end of the SUFFIX program) and halts.

3. SORT KSUF1 AND KSUF2 TO PUT SMALLER ENTRY OF EACH PAIR INTO KSUF1: As the comments suggest, this portion of the program shortens searches for suffixes when comparing textual words. If the smaller of the possible pairs is always in the first of the suffix tables, then when searching for possible matching suffixes the search is made for the smaller of the possible matching suffixes in the first suffix table (MSUF1). If this suffix does not appear, there is no profit in looking for the second suffix in the second table; therefore the search is abandoned. The test for an entry less than zero must

be made because the punches (on input cards) for some alphabetic characters set (turn on) the sign bit, thus making the numeric representation for a given textual word negative. A textual word which has a negative value is larger than a given textual word with a positive value.

4. PUT KSUF1 INTO ALPHABETICAL ORDER, AND KSUF2 AND LEXC INTO CORRESPONDING ORDER: A shuttle sort is used to alphabetize KSUF1. The logic re negative numbers is the same as in 3, above.

5. COMBINE ENDINGS IN KSUF2 WHICH HAVE A COMMON MATCHING SUFFIX IN KSUF1: This section of the program results in setting up table MSUF1 (the reduced KSUF1), MSUF2 (the rearranged KSUF2) and LSUF2 (the link from MSUF2 to MEXC). After setting up the first entry in each table, the DO-Loop processes through the rest of the data.

6. NSUFX CONTAINS NUMBER OF SUFFIXES IN MSUF1, etc.: This short section sets up items NSUFX, WORDS, FORDS, and MATCNT (all words with the same root have the same MATCNT number), and puts a final zero in tables MSUF2 and LEXL.

7. SUBROUTINE "WORDS" READS WORDS FROM INPUT TEXT TAPE INTO TABLE MWORDS: This section of the program goes first to subroutine WORDS. Upon the return from subroutine WORDS, NINP and NMAT are set and table MATCH is checked for overflow (if the input into MATCH has, through some error, been larger than the table, the program goes to statement 10 and stops). Then a check is made for the end of the input text. If the end has been reached, the program branches to statement 132 and the final routines in SUFFIX.

8. CHECK TO SEE IF WORD IS ON FUNCTION-WORD LIST: This section of the program performs a binary search through table KFNC to see whether the

textual word being examined is a function word. If it is, FORDS is incremented in 1, NINP by 4 (to look at the next textual word), and a check is made for the end of MWORDS. If all the words in MWORDS have been examined, 32 more textual words are read in by subroutine WORDS, the item WORDS is incremented by 32, NINP is reset to 1, and the processing of MWORDS begins again. If the end of MWORDS has not yet been reached, the next textual word is examined. If, on the other hand, the textual word under examination is not a function word (no match is found in table KFNC), then the program jumps to statement 108 (see 9, below).

9. BEGIN PROCEDURE FOR PROCESSING CONTENT WORDS: This section brings a textual word into table MATCH (one textual word entry is comprised of four computer words). Next, a check is made for the end of MWORDS. If the limit of MWORDS has been reached, another block is brought in and the necessary housekeeping performed. If the end of MWORDS has not been reached, the program branches to statement 110, and places the first computer word of the next textual entry into working cell M2.

10. SUBROUTINE COM3 COMPARES FIRST THREE CHARACTERS OF TWO TEXTUAL WORDS TO SEE IF THEY MATCH. IF SENSE LIGHT IS ON, THERE IS A MATCH. CONTINUE TO FILL MATCH UNTIL REACH WORD WHICH DOES NOT MATCH: This section first branches to subroutine COM3. Upon return from COM3, if sense light 7 is on (there is a MATCH), the program branches to statement 101 and examines the next textual word. If sense light 7 is not on, the program branches to statment 113 (see 11, below).

11. BEGIN SEARCH FOR TEXTUAL WORD WHICH DOES NOT HAVE A MATCNT, I. E., WHICH HAD NOT BEEN PAIRED WITH ANOTHER WORD(S). M INDEXES TABLE MATCH FROM

THE TOP AND N INDEXES MATCH FROM THE BOTTOM. IF TEXTUAL WORD (M) HAS A MATCNT, THEN SEARCH IS MADE FOR WORD (N) WHICH HAS NO MATCNT. TRY TO MATCH IT WITH FIRST WORD (M) HAVING A MATCNT. IF BOTH WORD (M) AND (N) HAVE MATCNTS, DECREASE N UNTIL IT REACHES A WORD (N) WITHOUT MATCNT AND TRY TO MATCH IT WITH WORD (M). IF N EQUALS M, AND THAT WORD HAS NO MATCNT THEN SET MATCNT TO THE VALUE IN THE ITEM MATCNT: Table MATCH contains only textual words for which the three first letters are identical. When the program begins to process the words in MATCH, the word at the beginning of the table (indexed by M) is compared with the word at the end of the table (indexed by N). If these words do not have a common root, N is decremented so that the first word and next-to-last word can be compared. If they do have a common root, this fact is indicated by giving both words a common MATCNT. Then N is decremented and the comparison of the first word and next-to-last word proceeds. This process continues until M is equal to N (in other words, the program has moved back through the table, comparing each word in turn with the first word). At this point, if M does not equal the total number of textual word entries in MATCH, M proceeds forward through MATCH until reaching a textual word which has no MATCNT (the fourth computer word in the textual entry is blank). When M begins to move forward through the table, N is reset to index the last textual word in the table. When M indexes an entry which has no MATCNT, then the comparison between words indexed by M and N proceeds as described above. The entire table has been processed when M equals the number of entries in MATCH. At this point, the entries in MATCH are transferred into buffer table DOUT, which in turn is transferred onto tape.

12. FINAL HOUSEKEEPING: When all of the input text has been processed, a final tape block is filled with the word THEEND, the total number of words and function words is printed out, and the SUFFIX program halts.

D. Subroutines for SUFFIX (Root-Finding Program)

1. WORDS: This subroutine reads in a block of textual words (128 computer words, 32 textual words) from input tape TZZE.

2. COM3: This subroutine compares the first three characters of two textual words to see if they match. Working cell M1 contains the first computer word of one textual word entry and working cell M2 contains the first computer word of a second textual word entry. The initial three letters of each entry are right-justified in M1 and M2 for comparison. If M1 = M2, sense light 7 is turned on prior to the return to the main program. If there is no match, the sense light remains off and the program returns to the main routine.

3. STEM: This subroutine checks, two words at a time, for suffixes. If they are found, sense light 7 is turned on. At the beginning of this subroutine, M1 contains the first computer word of one textual word entry and M3 contains the first computer word of a second textual word entry. M2 contains the second computer word of the first textual word entry and M4 contains the second computer word of the second textual word entry. If M1 = M3 and M2 = M4, then there are no suffixes to "remove," but the words should be grouped as a MATCH, obviously, because they are identical; therefore the sense light is turned on and control is returned to the main program. If the words are not the same, control is switched to subroutine NOS, which removes final "s," or "'s," or "s'" from the textual words.

Upon return from NOS, another check is made to see if the words are now the same; if so, the sense light is turned on and control is returned to the main program. If the words are not the same, the first half of the respective words in MAT1 and MAT2 are saved and the textual words are examined, character by character, until the place of deviation is reached. Although subroutine STEM is called only if the first three characters of the two textual words match, only two characters (SLAQ 12) are shifted out initially because, to apply the "LETTER" rule (see note under E below), the final matching letter must be saved; if only the first three characters in the two textual words match, the third character from each word will need to be saved. The coding under SL2 is required if one or both of the textual words occupies all of the computer word and part of the second; SL2 means that two computer words need to be shifted left. The coding under SL1 is used if both textual words occupy, respectively, no more than one computer word.

The coding at TEMP + 1 deals with the two textual words when the point at which they deviate from each other has been reached. MLET1 contains the last matching letter and the A register contains the letter following the last matching letter in the second textual word. The test for A = D checks for a double letter in the second textual word; if there is a double letter, it is shifted out at EM3. If the test is not met, the first textual word is checked for a double letter and, if one occurs, it is shifted out. Either after the shifting or, if no double letter occurs in either word, program control is transferred to COM (still in the TAC coding section). COM finds out which of the suffixes is the smaller (remember that tables containing suffixes have been sorted so that the smaller of each pair of suffixes

occurs in the first table) and places the smaller suffix in M1.

The FORTRAN coding following the comment SEARCH MSUF1 FOR THE SUFFIX down to the next comment consists of a binary search of MSUF1 for the smaller of the pair of suffixes which have just been found in STEM (for an explanation of the test for negative numbers, see C-3, above).

The SEARCH FOR PAIRED SUFFIX IN MSUF2 is done on the basis of LSUF2, which contains the starting address of possible pairing suffixes in MSUF2. A zero in an MSUF2 entry indicates the end of the set of possible pairing suffixes. If a matching suffix is found, then the exception list is searched. A zero in the appropriate entry in table LEXL, means that there are no exceptions, so a legal suffix pair has been found. If there are exceptions, the appropriate list in MEXC is searched; if a zero is reached, all of the exceptions have been examined, no match has been found, and therefore a legal suffix pair has been found. After testing for a zero, a test is made for the "LETTER" exception. The coding in TAC saves the actual letter in question (e.g., if the computer word contained LETTER S, the S is the actual letter) in LET1 and the word LETTER, if this is a LETTER exception, in LET6. If LET6 and item KLET match, there is a letter exception and control is transferred to statement 81. At statement 81, MLET1, which contains the last matching letter of the two textual words, is compared with the letter in the given "LETTER" entry on the exception list. If the letters are the same, the "LETTER" rule applies, the suffix pair is still legal, and the next entry on the exception list is examined. Only if an exception is found, or if the "LETTER" rule exists but there is no match between the letters, will control be returned to the main program without turning on

sense light 7; otherwise, the light is turned on to indicate a legal suffix pair, without exception. (The "LETTER" rule is explained in E, below.)

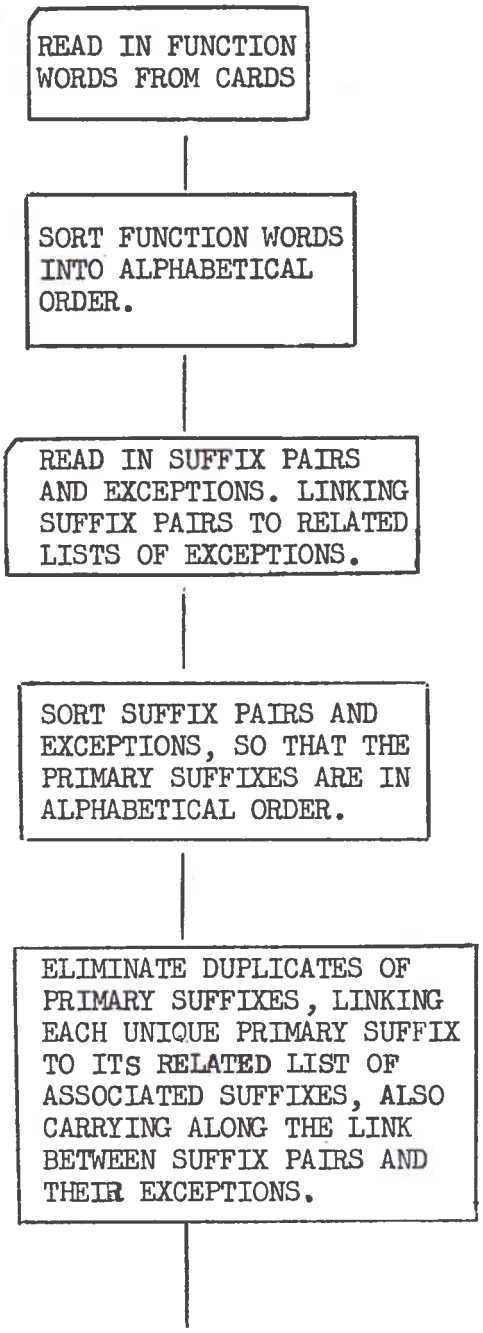
4. NOS: This subroutine removes final "s," "'s," and "s'" from textual words. The input to this subroutine consists of two textual words, upon which it operates in sequence. At least the three first characters of the textual words match. The program stores the first textual word in W1 and W2, then jumps to GAS (Get At S). The first instruction at GAS saves the program address of the instruction in the main part of the subroutine to which to return. Next, a test is made to see whether the textual word occupies one or two computer words. If the textual word does not extend into the second computer word, program control jumps to ONLY. Otherwise, the second computer word is searched from right to left until the terminal character of the textual word is found. Then, a test is made for a terminal S. If this test fails, a test is made for a terminal apostrophe. If this test also fails, control is returned to the examination of the second textual word. If the textual word ends in "S," the preceding letter is examined. If it is also an "S" the textual word ends in a double "S" and neither is removed. If the previous character is an apostrophe, the apostrophe and final "S"('s) are removed and control is returned to the examination of the next textual word. If the textual word has a terminal apostrophe, the apostrophe is removed, as is the S preceding the apostrophe. The procedure for a textual word occupying just one computer word is analogous to this one, and the examination of the second textual word is, of course, identical.

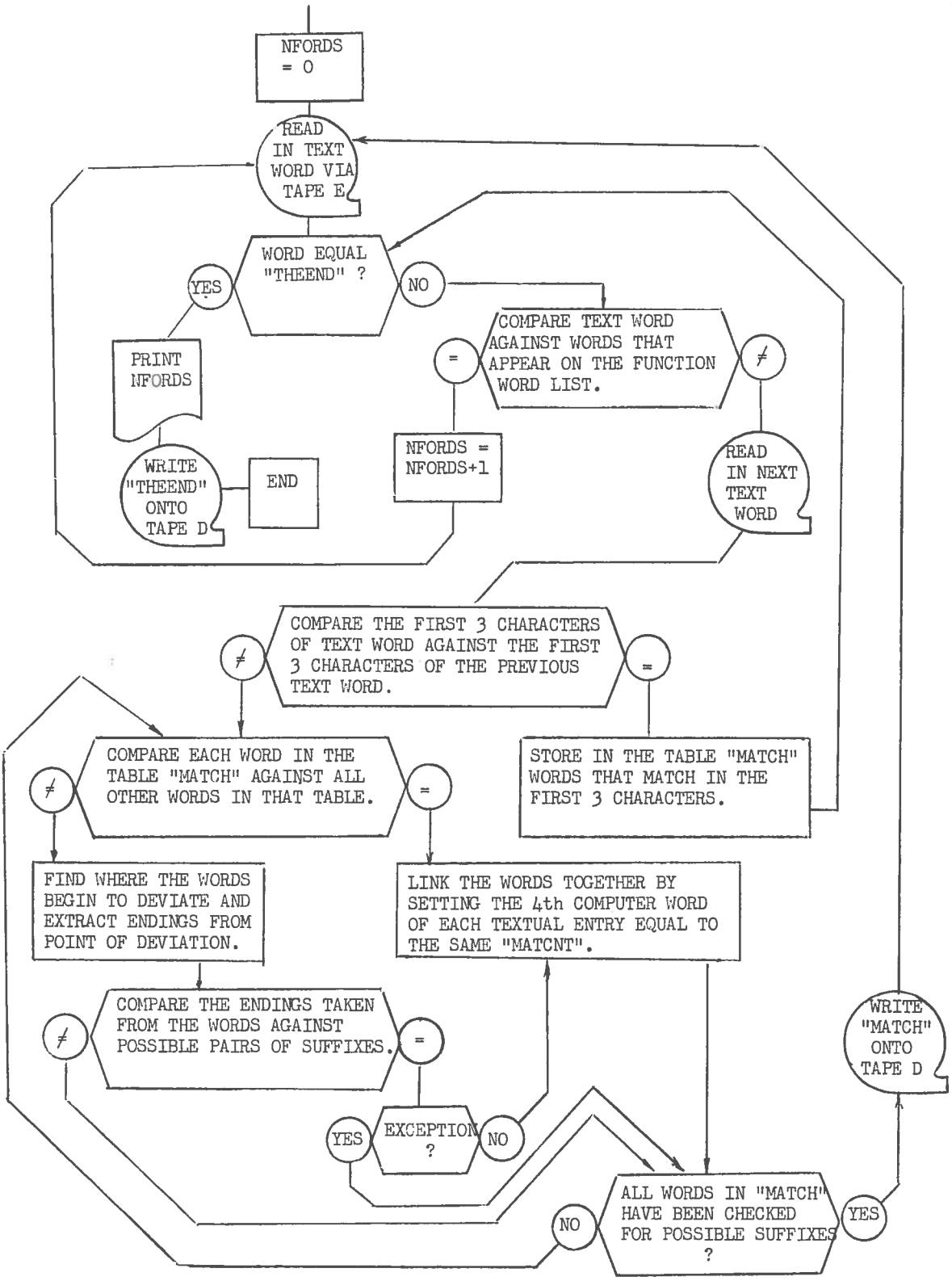
5. OUT: This subroutine fills up buffer table DOUT (holds one block--128 computer words--of data) and, when DOUT is full, writes it out on tape TZZD.

E. Special Note on the "LETTER" Rule

This rule merits a brief explanation because it somewhat misleadingly appears in the exception list. A typical example is the following: EXCEPT LETTER E. This means that if the final matching letter in the two words being examined is an E, the suffix pair is legal, unless an exception is found. If the final matching letter in the two words being examined is not an E, the program assumes it has not found a legal suffix and does not look at the rest of the exceptions.

F. Flow Chart


READ IN FUNCTION WORDS FROM CARDS
SORT FUNCTION WORDS INTO ALPHABETICAL ORDER.
READ IN SUFFIX PAIRS AND EXCEPTIONS. LINKING SUFFIX PAIRS TO RELATED LISTS OF EXCEPTIONS.
SORT SUFFIX PAIRS AND EXCEPTIONS, SO THAT THE PRIMARY SUFFIXES ARE IN ALPHABETICAL ORDER.
ELIMINATE DUPLICATES OF PRIMARY SUFFIXES, LINKING EACH UNIQUE PRIMARY SUFFIX TO ITS RELATED LIST OF ASSOCIATED SUFFIXES, ALSO CARRYING ALONG THE LINK BETWEEN SUFFIX PAIRS AND THEIR EXCEPTIONS.


NFORDS = 0
READ IN TEXT WORD VIA TAPE E
WORD EQUAL "THEEND" ?
YES
NO
PRINT NFORDS
WRITE "THEEND" ONTO TAPE D
END
COMPARE TEXT WORD AGAINST WORDS THAT APPEAR ON THE FUNCTION WORD LIST.
=
≠
NFORDS = NFORDS+1
READ IN NEXT TEXT WORD
COMPARE THE FIRST 3 CHARACTERS OF TEXT WORD AGAINST THE FIRST 3 CHARACTERS OF THE PREVIOUS TEXT WORD.
≠
=
COMPARE EACH WORD IN THE TABLE "MATCH" AGAINST ALL OTHER WORDS IN THAT TABLE.
STORE IN THE TABLE "MATCH" WORDS THAT MATCH IN THE FIRST 3 CHARACTERS.
FIND WHERE THE WORDS BEGIN TO DEVIATE AND EXTRACT ENDINGS FROM POINT OF DEVIATION.
LINK THE WORDS TOGETHER BY SETTING THE 4th COMPUTER WORD OF EACH TEXTUAL ENTRY EQUAL TO THE SAME "MATCNT".
COMPARE THE ENDINGS TAKEN FROM THE WORDS AGAINST POSSIBLE PAIRS OF SUFFIXES.
YES
EXCEPTION ?
NO
WRITE "MATCH" ONTO TAPE D
ALL WORDS IN "MATCH" HAVE BEEN CHECKED FOR POSSIBLE SUFFIXES ?
NO
YES

G. Printout of Function Word List

FUNCTION WORDS

--
.
LATTER
LIKE
MADE
MAKE
MAKING
MOREOVER
MY
NONE
NOR
NOTE
NOT
NOW
OF
ONE
ONLY
ON
ONTO
OR
OUR
PER
QUITE
RATHER
SHALL
SHE
SHOULD
SINCE
SO
STILL
SUCH
TAKE
THAN
THAT
THEIR
THEM
THEMSELVES
THEN
THEREFORE
THERE
THE
THESE
THEY
THIS
THOSE
THOUGH
THUS
TO
UNDER
UNTIL
UPON
UP
US
VERY
WAS
WERE
WE
WHATEVER
WHAT
WHEN
WHEREAS
WHERE
WHETHER
WHICH
WHILE
WHO
WHOSE
WHY
WILL
WITH
WORK
WOULD
YET
,
(
:
;
'.
ABOUT
ABOVE
AFTER
ALONG
ALSO
AMONG
AM
AND
ANOTHER
AN
ANY
ARE
A
AS
AT
BECAUSE
BEEN
BEFORE
BEING
BE
BETWEEN
BOTH
BUT
BY
CAN
CONCERNING
CONSEQUENTLY
COULD
DURING
EACH
EITHER
ETC.
EVEN
EXCEPT
FOR
FROM
HAD
HAS
HAVE
HAVING
HENCE
HER
HERSELF
HE
HIMSELF
HIS
HOWEVER
HOW
IF
INDEED
IN
INSIDE
INTO
I
IS
IT
ITSELF
ITS
...
..
.
)
?
"
THEEND

H. Printout of List of Suffixes and Exceptions

SUFFIX PAIRS

	ABILITY
	ABLE
EXCEPT	CAP
EXCEPT	CONSIDER
EXCEPT	FORT
	AIN
EXCEPT	BEG
EXCEPT	PLAIN
	AGE
	AL
EXCEPT	CAN
EXCEPT	FIN
EXCEPT	FORM
EXCEPT	INFORM
EXCEPT	JACKAL
EXCEPT	LATERAL
EXCEPT	LEG
EXCEPT	METAL
EXCEPT	MINER
EXCEPT	PERSONAL
EXCEPT	PET
EXCEPT	PHYSICAL
EXCEPT	ROY
EXCEPT	SAND
EXCEPT	SEVERAL
EXCEPT	SIGN
EXCEPT	SPIN
EXCEPT	VEST
	ALLY
EXCEPT	FIN
	AMENT
	AN
EXCEPT	LETTER E
EXCEPT	LETTER O
EXCEPT	CRIME
EXCEPT	UNCLE
	ANCE
EXCEPT	FIN
EXCEPT	IMPORT
	ANT
EXCEPT	IMPORTANT
EXCEPT	PAGE
EXCEPT	PROTESTANT
	ARD
	ARDMENT
	ARY
EXCEPT	BOUNDARY
EXCEPT	CAN
EXCEPT	ELEMENT
EXCEPT	HUNGARY
EXCEPT	LITER
EXCEPT	SECRETARY
	ARMS
	ATED
	ATIC
	ATION
EXCEPT	FOUNDATION
EXCEPT	ROT
	ATIZE
	C
	CE
EXCEPT	LETTER E
EXCEPT	AIRE
EXCEPT	FFE
EXCEPT	SUITE
	COM
	ED
EXCEPT	CARED
EXCEPT	CUB
EXCEPT	FADED
EXCEPT	FIN
EXCEPT	FIRED
EXCEPT	FOUNDED
EXCEPT	HAT
EXCEPT	MATED
EXCEPT	PAST
EXCEPT	PENN
EXCEPT	RAGGED
EXCEPT	RAT
EXCEPT	RUG
EXCEPT	SCRAPED
EXCEPT	SPARED
EXCEPT	STARED
EXCEPT	STRIPED
EXCEPT	TAMED
EXCEPT	TWINED
EXCEPT	UNITED
	EN
EXCEPT	BARREN
EXCEPT	LIST
EXCEPT	POLLEN
EXCEPT	RIP
EXCEPT	SPAN
EXCEPT	SIR
	ENCE
EXCEPT	OFF
EXCEPT	PRESS
EXCEPT	SENT
	ENED
EXCEPT	LIST
	ENT
EXCEPT	MISSENT
EXCEPT	PRESENT
EXCEPT	RIP
EXCEPT	ROD
	ER
EXCEPT	ARCHER
EXCEPT	BARBER
EXCEPT	BAT
EXCEPT	BIT
EXCEPT	BOOKER
EXCEPT	CAREER
EXCEPT	CENT
EXCEPT	COCKER
EXCEPT	CORN
EXCEPT	CUSTOM
EXCEPT	ENGINEER
EXCEPT	FIN

SUFFIX PAIRS

	EXCEPT FLOWER
	EXCEPT FORM
	EXCEPT HAM
	EXCEPT HING
	EXCEPT INN
	EXCEPT LAD
	EXCEPT LET
	EXCEPT LIT
	EXCEPT MAN
	EXCEPT MAST
	EXCEPT MATTER
	EXCEPT METER
	EXCEPT MOTH
	EXCEPT NUMB
	EXCEPT OFF
	EXCEPT PET
	EXCEPT PROP
	EXCEPT QUART
	EXCEPT RANGER
	EXCEPT RID
	EXCEPT RUB
	EXCEPT SCRAPER
	EXCEPT SETTER
	EXCEPT SHOW
	EXCEPT SHUT
	EXCEPT SLIP
	EXCEPT SOLO
	EXCEPT SPRINGER
	EXCEPT STAG
	EXCEPT SUM
	EXCEPT SWEATER
	EXCEPT TOW
	EXCEPT TWINER
	EXCEPT WICK
	ERN
	E
	EXCEPT LETTER H
	EXCEPT LETTER O
	EXCEPT LETTER S
	EXCEPT LETTER X
	EXCEPT CLOTHE
	EXCEPT MORALE
	EST
	EXCEPT DIG
	EXCEPT EARN
	EXCEPT FIN
	FALL
	FARE
	FIELD
	FUL
	HOOD
	IA
	EXCEPT GARDEN
	EXCEPT VIRGIN
	IAL
	EXCEPT BUR
	IAN
	EXCEPT PHYSIC
	IATION
	IC
	EXCEPT ANT
	EXCEPT CLASS
	EXCEPT CUB
	EXCEPT SONIC
	EXCEPT TOP
	ICAL
	EXCEPT CLASS
	EXCEPT LOG
	EXCEPT PERIOD
	ING
	EXCEPT BEAR
	EXCEPT BOOK
	EXCEPT CAR
	EXCEPT CLOTHING
	EXCEPT EARRING
	EXCEPT EVEN
	EXCEPT FAD
	EXCEPT FIR
	EXCEPT HERRING
	EXCEPT INN
	EXCEPT MAT
	EXCEPT RANG
	EXCEPT RIDING
	EXCEPT SCRAPING
	EXCEPT TAM
	EXCEPT TICK
	EXCEPT TWINING
	EXCEPT UNITING
	ION
	EXCEPT BILL
	EXCEPT LEG
	EXCEPT LOT
	EXCEPT MILL
	EXCEPT MISS
	EXCEPT PASS
	EXCEPT PORT
	EXCEPT PROCESSION
	EXCEPT STALL
	ISH
	EXCEPT FIN
	EXCEPT FLOURISH
	EXCEPT SPAN
	ISM
	IST
	EXCEPT ASS
	EXCEPT CELL
	EXCEPT PHYSIC
	ISTIC
	ITE
	ITIE
	ITION
	EXCEPT COAL
	EXCEPT PART
	ITIONAL
	ITY
	EXCEPT AUTHORITY
	EXCEPT DIVERSITY
	EXCEPT SEVERITY

SUFFIX PAIRS

	VANITY
EXCEPT	IVE
	IZATION
	IZED
	IZE
	IZING
	KING
	L
EXCEPT	LETTER A
EXCEPT	IDEAL
EXCEPT	SEA
	LEDGE
	LESS
EXCEPT	SHIFTLESS
EXCEPT	WIRE
	LIKE
	-LIKE
	LINESS
	LIZATION
	LY
EXCEPT	EARLY
EXCEPT	HARDLY
EXCEPT	HOMELY
EXCEPT	FEAR
EXCEPT	STATE
	MAKER
	MAN
EXCEPT	AIRMAN
EXCEPT	BUSH
EXCEPT	GENTLEMAN
	MEN
EXCEPT	GENTLEMEN
EXCEPT	MINUTE
	MENT
EXCEPT	APART
EXCEPT	BASE
EXCEPT	PIG
EXCEPT	STATEMENT
	MOST
	N
EXCEPT	LETTER A
EXCEPT	LETTER E
EXCEPT	LETTER W
EXCEPT	BROWN
EXCEPT	CROW
EXCEPT	DOZE
EXCEPT	FLOW
EXCEPT	LAW
EXCEPT	LINE
EXCEPT	OWE
EXCEPT	TOW
	NED
EXCEPT	BUR
EXCEPT	CROW
EXCEPT	EAR
EXCEPT	LEAVE
EXCEPT	PATTER
EXCEPT	WAR
	NESS
	CLOGY
	CR
EXCEPT	FACT
EXCEPT	MAY
EXCEPT	PASTOR
EXCEPT	ROT
EXCEPT	TAIL
EXCEPT	TRACT
EXCEPT	TENOR
	CUS
EXCEPT	GORGE
	CUT
	POWER
	R
EXCEPT	LETTER E
EXCEPT	ANGLER
EXCEPT	ARCHER
EXCEPT	BADGER
EXCEPT	BEER
EXCEPT	BROKER
EXCEPT	BUTTE
EXCEPT	CASTE
EXCEPT	CRATE
EXCEPT	COVER
EXCEPT	HOME
EXCEPT	LIVER
EXCEPT	OFFICER
EXCEPT	OLIVE
EXCEPT	PIE
EXCEPT	PRIME
EXCEPT	SKIE
	RE
EXCEPT	CENT
EXCEPT	STATU
	REN
EXCEPT	BARREN
	RENCE
	RY
EXCEPT	ARCHE
EXCEPT	COUNT
EXCEPT	HEN
EXCEPT	HUNG
EXCEPT	MAR
EXCEPT	NURSE
EXCEPT	SENT
EXCEPT	SURGERY
	SE
EXCEPT	BROW
EXCEPT	DEN
EXCEPT	TEASE
	SHIP
EXCEPT	AIRSHIP
	SIDE
EXCEPT	PRESIDE
	SLY
	SMEN
	ST
EXCEPT	LETTER E
	STORM

SUFFIX PAIRS

	T
EXCEPT	PLANT
	TENED
	TH
EXCEPT	BREADTH
EXCEPT	EARTH
EXCEPT	FIR
EXCEPT	HEAR
EXCEPT	NOR
EXCEPT	YOU
	TIME
	TOP
	TURE
EXCEPT	CAP
EXCEPT	TEMPERA
	TY
EXCEPT	CASUAL
EXCEPT	COMMUNITY
EXCEPT	PROPERTY
EXCEPT	SIX
	UAL
	URE
EXCEPT	END
EXCEPT	FEAT
EXCEPT	FIG
EXCEPT	MAN
EXCEPT	PASTURE
EXCEPT	POST
	URAL
	WARD
EXCEPT	WOOD
	WIDE
	Y
EXCEPT	ALL
EXCEPT	BATTER
EXCEPT	BILL
EXCEPT	BIG
EXCEPT	BUR
EXCEPT	BUS
EXCEPT	CARRY
EXCEPT	COOK
EXCEPT	COUNTY
EXCEPT	DOWNY
EXCEPT	EARL
EXCEPT	EVERY
EXCEPT	FACTOR
EXCEPT	FAIR
EXCEPT	FORT
EXCEPT	GULL
EXCEPT	HARD
EXCEPT	LAD
EXCEPT	MAN
EXCEPT	MAR
EXCEPT	MOB
EXCEPT	PARTY
EXCEPT	PEAR
EXCEPT	PENNY
EXCEPT	PIT
EXCEPT	PLAN

EXCEPT	POPY
EXCEPT	READY
EXCEPT	ROMANY
EXCEPT	RUB
EXCEPT	SLIM
EXCEPT	SLIPPER
EXCEPT	SPIN
EXCEPT	TINY
A	ESE
A	IAN
A	ISM
A	CN
ACTION	Y
AL	E
EXCEPT	CANE
EXCEPT	CHORE
EXCEPT	CORE
EXCEPT	FINE
EXCEPT	MORE
EXCEPT	PARTIAL
EXCEPT	PRACTICAL
EXCEPT	SEVERE
EXCEPT	SPECIAL
AL	ED
AL	IAN
AL	IST
AL	ST
AL	L
AL	Y
ABLE	E
EXCEPT	CAPE
EXCEPT	SUITE
AIN	ENANCE
AIN	ISH
AME	CME
AMENT	ED
AN	E
EXCEPT	CURAN
EXCEPT	MILAN
EXCEPT	RICE
AN	EN
AN	IN
AN	C
ANCE	ED
ANCE	CME
AND	E
EXCEPT	COME
AND	ISH
ANT	E
ARY	E
EXCEPT	CANE
EXCEPT	SALARY
ARY	ITY
AT	IT
ATE	E
EXCEPT	PRIMATE
ATE	IC
ATION	E
ATION	ED

SUFFIX PAIRS

EXCEPT	FOUNDATION
ATION	ING
AY	EGIAN
BE	PTION
BED	PTION
BILITY	TION
BY	EST
C	E
C	SM
CAL	ST
EXCEPT	TYPIST
CATION	ED
CE	T
EXCEPT	FLEET
EXCEPT	FORCE
EXCEPT	GREECE
EXCEPT	INSTANCE
EXCEPT	FEAT
EXCEPT	PRINT
EXCEPT	SPAT
EXCEPT	SPIT
CE	TIAL
CE	TIST
CE	TLY
CE	X
CH	LAND
CIE	TIC
EXCEPT	POLITIC
CITU	X
CTION	GUISH
CTION	GUISHED
CY	T
EXCEPT	CURRENCY
CY	TE
CY	TIC
D	LY
EXCEPT	CHILD
EXCEPT	CURLY
EXCEPT	FOLD
EXCEPT	HOLLY
D	MENT
EXCEPT	BASED
D	NT
EXCEPT	AGED
EXCEPT	MEAD
EXCEPT	MISSENT
EXCEPT	PAID
EXCEPT	SAID
EXCEPT	SPED
D	R
EXCEPT	LETTER E
EXCEPT	ARCHER
EXCEPT	BARBER
EXCEPT	BANNER
EXCEPT	DEER
EXCEPT	FLOWER
EXCEPT	FORMER
EXCEPT	LIVER
EXCEPT	MANNER
EXCEPT	MATTER
EXCEPT	RUBBER
EXCEPT	SHOWER
EXCEPT	TOWER
D	RAL
D	SE
D	T
EXCEPT	BEAD
EXCEPT	BOLD
EXCEPT	CARD
EXCEPT	CONTENT
EXCEPT	COLT
EXCEPT	DIET
EXCEPT	FEET
EXCEPT	FOOT
EXCEPT	FORGET
EXCEPT	FORT
EXCEPT	GRAND
EXCEPT	HATCHED
EXCEPT	HEAD
EXCEPT	HEARD
EXCEPT	HOOD
EXCEPT	HORNET
EXCEPT	MARKED
EXCEPT	MEAD
EXCEPT	MOLD
EXCEPT	MOUND
EXCEPT	PLOD
EXCEPT	TENT
EXCEPT	THREAD
EXCEPT	WARD
EXCEPT	WILT
DED	SION
EXCEPT	TENSION
DED	T
EXCEPT	CART
EXCEPT	HEAT
EXCEPT	HOOT
EXCEPT	MOLD
EXCEPT	TENDED
DING	T
E	COM
E	IAL
E	IC
EXCEPT	BASE
EXCEPT	COMIC
EXCEPT	CUBE
EXCEPT	LOGE
EXCEPT	LYRE
EXCEPT	MUSE
EXCEPT	OPERATE
EXCEPT	POLITE
EXCEPT	SLAVIC
EXCEPT	STATE
EXCEPT	TUNIC
E	ICAL
E	ICE
E	IERY
E	IFIER

SUFFIX PAIRS

E	IFY
EXCEPT	MODE
E	INAL
E	ING
EXCEPT	EARRING
EXCEPT	CASTE
EXCEPT	DAME
EXCEPT	HUGE
EXCEPT	SKIE
EXCEPT	TIDE
EXCEPT	TWINE
E	ION
EXCEPT	DEFINATE
EXCEPT	MILE
EXCEPT	MILLE
EXCEPT	NOTE
EXCEPT	PASSION
EXCEPT	STATE
EXCEPT	VERSE
E	ISH
EXCEPT	FINE
EXCEPT	MOORE
E	IT
EXCEPT	COME
EXCEPT	CUBE
EXCEPT	LIME
EXCEPT	MERE
E	ITION
E	ITY
EXCEPT	CAVE
EXCEPT	NATIVITY
E	IVE
E	IVITY
E	CH
EXCEPT	CURIOU
EXCEPT	SERIOU
E	FTICN
E	Y
EXCEPT	BUST
EXCEPT	CARE
EXCEPT	CASE
EXCEPT	CART
EXCEPT	EASE
EXCEPT	FACT
EXCEPT	HOSE
EXCEPT	LIFT
EXCEPT	MINT
EXCEPT	MOLE
EXCEPT	MUSE
EXCEPT	PACT
EXCEPT	PLANT
EXCEPT	PORE
EXCEPT	POST
EXCEPT	PRODUCT
EXCEPT	RIFT
EXCEPT	SALE
EXCEPT	SCENT
EXCEPT	SHORE
EXCEPT	SHOT
EXCEPT	SORT
EXCEPT	SPIE
EXCEPT	SPORE
EXCEPT	SQUIRE
EXCEPT	STARE
EXCEPT	SURFACT
EXCEPT	TTLE
EXCEPT	VASE
EXCEPT	WARE
E	IFN
EXCEPT	FATE
EXCEPT	MOLE
EXCEPT	SHORE
E	TH
EXCEPT	FIFE
EXCEPT	WORE
E	TION
EXCEPT	CAPE
EXCEPT	FACE
EXCEPT	PARTITION
EXCEPT	PORE
E	TRIL
E	URE
EXCEPT	CREATURE
EXCEPT	MANE
EXCEPT	PASTURE
EXCEPT	PRESSES
EXCEPT	STATE
E	Y
EXCEPT	BUSY
EAK	OKEN
ED	IHLY
ED	ING
EXCEPT	MATTER
ED	ION
EXCEPT	MILLED
EXCEPT	MISSED
EXCEPT	NOTED
EXCEPT	PASSION
EXCEPT	STATION
ED	ISON
ED	ITICN
ED	ITICNAL
ED	ITURE
ED	IVE
EXCEPT	EXECUTED
ED	MENT
ED	OR
EXCEPT	MINOR
EXCEPT	PASTOR
ED	SE
ED	STVE
ED	STVELY
ED	T
EXCEPT	CART
EXCEPT	DART
EXCEPT	FACED
EXCEPT	LEAST
EXCEPT	MINED

SUFFIX PAIRS

EXCEPT	PLANNED
EXCEPT	POST
EXCEPT	STARED
EXCEPT	TRACT
ED	TION
EXCEPT	FACED
ED	TURE
EXCEPT	POSED
ED	URE
EXCEPT	ENDED
EXCEPT	FASTED
EDED	SSFULL
ED	Y
EXCEPT	CITY
EXCEPT	COOKY
EXCEPT	COUNTY
EXCEPT	MANY
EXCEPT	PENNY
EXCEPT	SCRUBBY
EXCEPT	STORY
EXCEPT	TREATY
ELL	CLD
EN	INAL
EN	ING
EXCEPT	FASTEN
EXCEPT	LINING
EXCEPT	LISTING
ENT	Y
EXCEPT	STUDY
EP	PT
EP	PTH
ER	EST
ER	IAL
ER	ING
EXCEPT	FLOWER
EXCEPT	FORMER
EXCEPT	INNER
EXCEPT	LETTING
EXCEPT	LIVER
EXCEPT	MATTER
EXCEPT	SETTER
EXCEPT	SHOWER
EXCEPT	SHUTTER
EXCEPT	SLIPPER
EXCEPT	SPRINGING
EXCEPT	SWEATER
EXCEPT	TOWER
ER	LY
EXCEPT	HARDLY
EXCEPT	SUPPER
ER	PTION
ER	RAL
ER	RE
ER	RIC
ER	RY
EXCEPT	COUNTRY
EXCEPT	HUNGER
ER	Y
EXCEPT	COOKER
EXCEPT	COUNTY
EXCEPT	FAIRY
EXCEPT	READY
EXCEPT	SHOWER
ERY	TIC
ETIC	Y
EW	OW
EY	ISH
F	VE
EXCEPT	SERF
F	VOU
FE	VE
EXCEPT	STRIFE
FIC	ST
FIED	TED
FIED	TY
FYING	TY
GUARD	TY
I	L
IAL	IST
IAL	Y
EXCEPT	PARTY
IAN	Y
IC	ISM
IC	CE
IC	CGEN
IC	Y
EXCEPT	ITALY
EXCEPT	TERRIFY
ICAL	CLOGY
ICAL	Y
ICATION	YING
ICE	SE
ID	Y
EXCEPT	PLAY
IE	Y
IE	YING
IED	ICATIO
IED	Y
IER	Y
IES	Y
IEST	Y
IFUL	Y
IGHT	CUGHT
ILY	Y
IN	LN
INESS	Y
ING	ION
ING	MENT
EXCEPT	COMMENT
ING	TH
ING	LNG
ING	Y
EXCEPT	BILLY
EXCEPT	COOKING
EXCEPT	COUNTY
EXCEPT	KINDLY
EXCEPT	READY
EXCEPT	STORING

SUFFIX PAIRS

EXCEPT	TREATY
ION	IVE
ION	IVELY
ION	OR
EXCEPT	MENTOR
EXCEPT	MOTION
EXCEPT	VISION
ION	ORY
ION	UAL
ISM	IST
IST	C
IST	Y
ISTIC	Y
L	TING
LAND	LE
LE	ILITIE
LE	ILITY
LE	LLAR
EXCEPT	PARTICLE
M	T
EXCEPT	LETTER S
N	T
EXCEPT	BEAT
EXCEPT	CANNON
EXCEPT	CONCERT
EXCEPT	DART
EXCEPT	GREET
EXCEPT	LOON
EXCEPT	MEAT
EXCEPT	SHUN
EXCEPT	SPAN
EXCEPT	SPIN
EXCEPT	STRAIN
EXCEPT	WARN
N	TICAL
NG	TE
NT	TE
OM	SSOM
ON	RY
EXCEPT	PORRY
ON	TED
ON	VE
EXCEPT	ACTIVE
EXCEPT	CAPTION
EXCEPT	DERIVATIVE
EXCEPT	DIRECTION
EXCEPT	NATION
EXCEPT	POSITIVE
OR	RAL
OR	MENT
OR	RESS
OR	RY
OY	LOTION
R	LY
EXCEPT	HOMELY
EXCEPT	LIVER
EXCEPT	SCAR
EXCEPT	WHIRLY
R	ST
EXCEPT	BEAR
EXCEPT	BOAR
EXCEPT	FEAR
EXCEPT	ROAR
EXCEPT	YEAST
R	LR
SH	TURE
SION	T
EXCEPT	FIST
EXCEPT	MIST
EXCEPT	PAST
SIONARY	T
SIS	TIC
SITY	LS
ST	ZE
EXCEPT	BLAST
EXCEPT	ORGANIZE
TH	WARD
TION	ZE
THEEND	

I. Program Listing, SORDIT and SUFFIX

```
SORT PROGRAM
                    JOB       SORDIT
                    WRTSENT   4,ZZZZZZZZ
              ABS   IAC       MAGTAPE,LIB
              I     SORDIT
                    NAME      SORDIT$
              START NOP       $
                    SORT      INTAPE(3)
                              OUTTAPE(4)
                              WKTAPES(2,5,6)
                              INFORM(4,128)
                              INSENT(W/THEEND
                              KEY(1,A,1)
                              KEY(2,A,2)$
                    HLT       START$
                    END       START$
                    JOB       RUN
                    JOB       ENDJOB
```

SLFFIX PROGRAM

```
      I          SLFFIX
                 IDENTIFY F, 16K, 8X, 7000W
      SLEROUTINE STEM
      COMMON M1,M2,M3,M4,NSUFX
      COMMON  I,J,K,L,M,N
      COMMON MSUF1,LSUF2,MSUF2,LEXL,MEXC
      COMMON KLET,MAT1,MAT2,LET1,LET6,MLET1
      DIMENSION MSUF1(500), LSUF2(500), MSUF2(2000)
      DIMENSION LEXL(2000), MEXC(1000)
      IF (M1)NE(M3),GO TO 2
      IF (M2)E(M4),GO TO 99
    2 CALL NOS
      IF (M1)NE(M3),GO TO 3
      IF (M2)E(M4),GO TO 99
    3 MAT1 = M1
      MAT2 = M3
      STARTTAC
                TMA     M1$
                TMQ     M2$
                SLAQ    12$
                TAM     M1$
                TMA     M3$
                TMQ     M4$
                SLAQ    12$
                TAM     M3$
                TMQ     A/        $
                TMA     M2$
                SLAQ    12$
                TAM     M2$
                TMA     M4$
                SLAQ    12$
                TAM     M4$
                TMQ     A/        $
                JAEQ    (P)+1$
                JMP     SL2$
                TMA     M2$
                JAEQ    SL1$
          SL2   CA      $
                TMQ     M1$
                SLAQ    6$
                TAM     TEMP$
                CA      $
                TMQ     M3$
                SLAQ    6$
                TMQ     TEMP$
                JAEQ    (P)+1$   HAVE CHARACTERS WHICH ARE EQUAL
                JMP     TEMP+1$ CHARACTERS DO NOT MATCH
                TAM     MLET1$
                TMA     M1$ TO GET RID OF CHARACTER WHICH HAS JUST BEEN FOUND
                TMQ     M2$   EQUAL TO CHARACTER IN OTHER WORD
                SLAQ    6$
                TAM     M1$
                TMA     M3$ TO GET RID OF CHARACTER WHICH HAS JUST BEEN FOUND
                TMQ     M4$  EQUAL TO CHARACTER IN OTHER WORD
                SLAQ    6$
                TAM     M3$
                TMA     M2$
                TMQ     A/        $
                SLAQ    6$
```

SUFFIX PROGRAM

```
                    TAM     M2$
                    TMA     M4$
                    SLAQ    6$
                    TAM     M4$
                    JMP     SL2$
               SL1  CA      $
                    TMQ     M1$
                    SLAQ    6$
                    TAM     TEMP$
                    CA      $
                    TMQ     M3$
                    SLAQ    6$
                    TMQ     TEMP$
                    JAEQ    (P)+1$
                    JMP     TEMP+1$
                    TAM     MLET1$
                    TMQ     A/          $
                    TMA     M1$
                    SLAQ    6 $
                    TAM     M1$
                    TMA     M3$
                    SLAQ    6 $
                    TAM     M3$
                    JMP     SL1$
               TEMP C/D     $
                    TMD     MLET1$      WORDS DEVIATE
                    JAED    EM3$
                    TDA     $
                    JAEQ    EM1$
                    JMP     COM$
               EM1  TMA     M1$
                    TMQ     0/60$
                    SLAQ    6$
                    TAM     M1$
                    JMP     COM$
               EM3  TMA     M3$
                    TMQ     0/60$
                    SLAQ    6$
                    TAM     M3$
               COM  TMA     M3$        COMPARE AND SORT M1, M3
                    TMD     M1$
                    JAGD    OUT+1M$
                    TDM     M3$
               OUT  TAM     M1$PLACE SUFFIX WITH ALPHABETICAL PRECEDENCE IN M1
                    ENDTAC
C  SEARCH MSUF1 FOR THE SUFFIX
      J = 0
      K = NSUFX+1
   33 I = (J+K)/2
      IF (I)E(K),GO TO 4
      IF (I)E(J),GO TO 4
      IF (M1)E(MSUF1(I)),GO TO 5
      IF (M1)LT(0),GO TO 37
      IF (MSUF1(I))LT(0),GO TO 34
   36 IF (M1)GT(MSUF1(I)),GO TO 35
   34 K = I
      GO TO 33
   35 J = I
      GO TO 33
```

```
SUFFIX PROGRAM
        37  IF (MSUF1(I))GT(0),GO TO 35
            GO TO 36
         4  RETURN
      C  SEARCH FOR PAIRED SUFFIX IN MSUF2
         5  J = LSUF2(I)
         6  IF (MSUF2(J))E(0),GO TO 10
            IF (MSUF2(J))E(M3),GO TO 7
            J = J+1
            GO TO 6
      C  ZERO INDICATES THAT THERE ARE NO EXCEPTIONS; THEREFORE,HAVE FOUND A SUFFIX
         7  IF (LEXL(J))E(0),GO TO 9
            K = LEXL(J)
            J = 2
      C SEARCH EXCEPTION LIST IN MEXC; 0 INDICATES HAVE REACHED END OF LIST,
      C   THEREFORE , HAVE FOUND SUFFIX
         8  IF (MEXC(K))E(0),GO TO 9
            LET6 = MEXC(K)
            STARTTAC
                        CQ      $
                        TMA     LET6$
                        SRAQ    6$
                        SRQ     42$
                        TQM     LET1$
                        TMQ     0/60$
                        SLAQ    6$
                        TAM     LET6$
                        ENDTAC
            IF (LET6)E(KLET),GO TO 81
            IF (J)E(1),GO TO 10
            IF (MEXC(K))E(MAT1),GO TO 10
            IF (MEXC(K))E(MAT2),GO TO 10
        82  K = K+1
            GO TO 8
        81  IF (LET1)E(MLET1),GO TO 83
            IF (J)E(0),GO TO 82
            J = 1
            GO TO 82
        83  J = 0
            GO TO 82
         9  IF (J)E(1),GO TO 10
        99  SENSE LIGHT 7
        10  RETURN
            END
            SUBROUTINE COM2
            COMMON M1,M2
            CONTINUE
            STARTTAC
                        TMA     M1$
                        SRA     30$
                        TAM     M1$
                        TMA     M2$
                        SRA     30$
                        TAM     M2$
                        ENDTAC
            IF (M1)NE(M2),GO TO 90
            SENSE LIGHT 7
        90  RETURN
            END
            SUBROUTINE WORDS
```

```
SLEFIA PROGRAM
              CONTINUE
              START TAC
                        JMP     IOINT$
                        N/6T23;N/1T39%
                        HLT     TZZE$
                        HLT     0001PRO,MWORDS$
                        ENDTAC
              RETURN
              END
              SUBROUTINE OUT
              COMMON M1
              CONTINUE
              START TAC
                        TMD     XSAV$
                        TDXLC   ,1$
                        TMD     M1$
                        TDM     ,1$
                        TMD     C/HLT,DOUT+128;C/HLT,SAV$
                        AIXJ    1,1$
                        TMD     L/DOUT$
                        TDM     XSAV$
                        JMP     IOINT$
                        N/5T23;N/1T39%
                        HLT     TZZD$
                        HLT     DOUT$
                        JMP     RET$
                DOUT    ASTOR   128$
                XSAV    L/DOUT$
                SAV     TXDLC   ,1$
                        TDM     XSAV$
                RET     NOP     $
                        ENDTAC
              RETURN
              END
              SUBROUTINE NOS
              COMMON  M1,M2,M3,M4
              CONTINUE
              START TAC
                        TMD     M1      $
                        TDM     W1      $
                        TMD     M2      $
                        TDM     W2      $
                        JMP     GAS     $ TO GET AT S
                        TMD     W1      $
                        TDM     M1      $ M1 AND M2 NOW CONTAIN WORD FORM AFTER S AND
                        TMD     W2      $  APOSTROPHE S HAVE BEEN REMOVED
                        TDM     M2      $
                W34     TMD     M3      $
                        TDM     W1      $
                        TMD     M4      $
                        TDM     W2      $
                        JMP     GAS     $
                        TMD     W1      $
                        TDM     M3      $ M3 AND M4 NOW CONTAIN WORD FORM AFTER S AND
                        TMD     W2      $  APOSTROPHE S HAVE BEEN REMOVED
                        TDM     M4      $
                        ENDTAC
              RETURN
              START TAC
```

SUFFIX PROGRAM

```
GAS     TJM     EXIT        $
        TMQ     6/1T47      $
        TMA     W2          $
        TMD     A/          $
        JAED    ONLY    $ TEXTUAL WORD DOESN,T EXTEND INTO 2ND COMPUTER W
MECH    ETA     W2          $
        ES      W/          $
        JAZ     BLNK    $ NO CHARACTER IN SPACE EXAMINED
        ETA     W/SSSSSSSS$ SEARCH FOR TERMINAL S IN 2ND COMPUTER WORD
        ES      W2          $
        JAZ     SSSS    $ IF HAVE A TERMINAL S
        ETA     W/''''''''$
        ES      W2$
        JAZ     (P)+1$
EXIT    JMP     (P)     $ NO TERMINAL S, THEREFORE EXIT
        TMA     W2$
        EI      W/          $
        TAM     W2$
        JMP     GAS+1H$
BLNK    SLQ     6       $ TO PROCESS THROUGH WORD,6 BITS AT A TIME, FROM
        JMP     MECH    $ RIGHT TO LEFT, UNTIL FIND A CHARACTER
SSSS    SLQ     6       $ HAVE ALREADY LOCATED TERMINAL S, NOW WANT TO
        TMA     0/0     $ LOOK AT PRECEDING CHARACTER. FIRST,SEE
        JAEQ    FRWD    $ IF S WAS IN 1ST 6 BITS IN 2ND COMPUTER WORD. IF
        NOP         $ SO, JUMP TO FRWD TO LOOK AT END OF 1ST COMPUTER WO
        ETA     W2          $
        ES      W/SSSSSSSS$
        JAZ     EXIT    $ IF TEXTUAL WORD ENDS IN DOUBLE S, DON;T REMOVE
        ETA     W2          $
        ES      W/''''''''$ LOOK FOR APOSTROPHE S, POSSESSIVE ENDING
        JAZ     POSS        $
        TMA     W2      $ IF DON,T HAVE POSSESSIVE ENDING
        JMP     (P)+3H      $
POSS    TMA     W2      $ REMOVE APOSTROPHE
        EI      W/          $
        SRQ     6       $ REMOVE FINAL S
        EI      W/          $
        TAM     W2          $
        JMP     EXIT        $
FRWD    TMQ     6/1T47      $
        ETA     W1          $
        ES      W/SSSSSSSS$ CHECK FOR DOUBLE-S ENDING. IF FOUND, EXIT
        JAZ     EXIT        $
        ETA     W1          $
        ES      W/''''''''$
        JAZ     (P)+3   $ IF HAVE APOSTROPHE
        TMQ     6/1T5   $ TO REMOVE TERMINAL S, OCCUPYING 1ST 6 BITS IN
        TMA     W2      $  COMPUTER WORD
        EI      W/          $
        TAM     W2          $
        JMP     EXIT        $
        TMA     W1          $
        EI      W/          $ REMOVE APOSTROPHE
        TAM     W1$
        JMP     (P)-4$
ONLY    ETA     W1      $ TEXTUAL WORD CONTAINED IN 1ST COMPUTER WORD
        ES      W/          $
        JAZ     BLOP$
        ETA     W/SSSSSSSS$ CHECK FOR TERMINAL S
```

SUFFIX PROGRAM

```
                    ES      W1          $
                    JAZ     OTHER$
                    ETA     W/''''''''$
                    ES      W1$
                    JAZ     APOS$
                    JMP     EXIT        $
              RLOP  SLQ     6       $ LOOP PROVIDING FOR SEARCH, FROM RIGHT TO LEFT,
                    JMP     ONLY    $ UNTIL FIND A CHARACTER
              APOS  TMA     W1$
                    EI      W/          $
                    TAM     W1$
                    JMP     GAS+1H$
              OTHER SLQ     6       $ TO LOOK AT CHARACTER PRECEDING TERMINAL S
                    ETA     W1          $
                    ES      W/SSSSSSSS$ IF HAVE TERMINAL DOUBLE-S, DON,T REMOVE
                    JAZ     EXIT        $
                    ETA     W1          $
                    ES      W/''''''''$
                    JAZ     (P)+3H      $
                    TMA     W1          $
                    JMP     (P)+3H      $
                    TMA     W1      $ REMOVE APOSTROPHE
                    EI      W/          $
                    SRQ     6           $
                    EI      W/          $ REMOVE FINAL S
                    TAM     W1          $
                    JMP     EXIT        $
              W1    C/0                 $
              W2    C/0                 $
                    ENDTAC
          END
          COMMON  M1,M2,M3,M4,NSUFX
          COMMON  I,J,K,L,M,N
          COMMON MSUF1, LSUF2, MSUF2, LEXL, MEXC
          COMMON KLET,KARD,KFNC,NFNC
          COMMON KEND,KEXC,LEMP,KELNK
          COMMON MATCNT,NINP,NMAT
          COMMON MA1,MA2, WORDS, FORDS
            DIMENSION  MWORDS(128)
            DIMENSION  MATCH(100)
            DIMENSION  KSUF1(500), KSUF2(500), LEXC(500)
            DIMENSION  MSUF1(500), LSUF2(500), LEXL(2000)
            DIMENSION  MSUF2(2000) ,MEXC(10000)
          DIMENSION  KFNC(200)
            DIMENSION  KARD(10)
          REWIND 3
          READ 1,KARD
        1 FORMAT(10A8)
          KEND = KARD(2)
          KEXC = KARD(3)
          KLET = KARD(4)
          KELNK = KARD(5)
          L = 1
          NSUFX = 0
C READ FUNCTION WORDS INTO KFNC - NFNC WILL CONTAIN TOTAL NUMBER OF WORDS
          DO 2 NFNC = 1,200,1
          READ 1, KARD
          IF (KARD(2))E(KEND),GO TO 22
          KFNC(NFNC) = KARD(2)
```

```
SUFFIX PROGRAM
      2  CONTINUE
         STOP
C READ SUFFIX INFO INTO KSUF1,KSUF2,MEXC - LEXC(LOCATION OF EXCEPTIONS) LINKS
C       MEXC(EXCEPTION LISTS TABLE) TO KSUF2
     22 J = 2
     23 I = J+1
        IF (KFNC(J))GTE(KFNC(I)),GO TO 25
        LEMP = KFNC(J)
        KFNC(J) = KFNC(I)
        KFNC(I) = LEMP
        J = J+1
        IF (J)LT(2),GO TO 22
        GO TO 23
     25 J = J+1
        IF (J)LT(NFNC),GO TO 23
      3 READ 1,KARD
        IF (KARD(1))E(KEND),GO TO 53
        IF (KARD(1))E(KEXC),GO TO 4
        MEXC(L) = 0
        L = L+1
        NSUFX = NSUFX+1
        KSUF1(NSUFX) = KARD(1)
        KSUF2(NSUFX) = KARD(2)
        LEXC(NSUFX) = L
        GO TO 5
      4 MEXC(L) = KARD(2)
        L = L+1
      5 IF (NSUFX)GT(500),GO TO 10
        GO TO 3
     53 MEXC(L) = 0
C SORT KSUF1 AND KSUF2 TO PUT SMALLER ENTRY OF EACH PAIR INTO KSUF1, SO THAT,
C    WHEN LOOKING FOR SUFFIX OF MATCHING WORDS,IF LOWER OF PAIR ISN'T IN KSUF1,
C    DON'T BOTHER WITH KSUF2.  AN ENTRY MAY BE LESS THAN ZERO IF CHARACTER
C       SETS LEFT-MOST BIT,THE SIGN BIT, WHICH INDICATES NEGATIVE NUMBER.
     54 DO 58 I = 1,NSUFX,1
        IF (KSUF1(I))LT(0),GO TO 57
        IF (KSUF2(I))LT(0),GO TO 58
     55 IF (KSUF1(I))LTE(KSUF2(I)),GO TO 58
     56 LEMP = KSUF1(I)
        KSUF1(I) = KSUF2(I)
        KSUF2(I) = LEMP
        GO TO 58
     57 IF (KSUF2(I))GT(0),GO TO 56
        GO TO 55
     58 CONTINUE
C PUT KSUF1 INTO ALPHABETICAL ORDER, AND KSUF2 AND LEXC INTO CORRESPONDING ORDER
      6 J = 2
     69 I = J+1
        IF (KSUF1(J))LT(0),GO TO 63
        IF (KSUF1(I))LT(0),GO TO 62
     61 IF (KSUF1(J))GTE(KSUF1(I)),GO TO 7
     62 LEMP = KSUF1(I)
        KSUF1(I) = KSUF1(J)
        KSUF1(J) = LEMP
        LEMP = KSUF2(I)
        KSUF2(I) = KSUF2(J)
        KSUF2(J) = LEMP
        LEMP = LEXC(I)
        LEXC(I) = LEXC(J)
```

SUFFIX PROGRAM

```
      LEXC(J) = LEMP
      J = J-1
      IF (J)LT(2),GO TO 6
      GO TO 69
   63 IF (KSUF1(I))LT(0),GO TO 61
    7 J = J+1
      IF (J)LTE(NSUFX),GO TO 69
C COMBINE ENDINGS IN KSUF2 WHICH HAVE A COMMON MATCHING SUFFIX IN KSUF1; ERGO,
C    HAVE MSUF1,THE REDUCED KSUF1,AND MSLF2,THE REARRANGED KSUF2. LSUF2 LINKS
C    MSLF2 TO MSUF1. BY MEANS OF LEXC, LEXL LINKS MSUF2 TO MEXC
      L = 1
      M = 2
      MSLF1(1) = KSUF1(1)
      LSLF2(1) = 1
      MSLF2(1) = KSUF2(1)
      LEXL(1) = LEXC(1)
      DO  9 J = 2,NSLFX,1
      I = J-1
      IF (KSUF1(J))E(KSUF1(I)),GO TO 8
      L = L+1
      MSLF1(L) = KSUF1(J)
      MSLF2(M) = 0
      LEXL(M) = 0
      M = M+1
      LSLF2(L) = M
    8 MSLF2(M) = KSUF2(J)
      LEXL(M) = LEXC(J)
      M = M+1
    9 CONTINUE
C  NSUFX CONTAINS NUMBER OF SUFFIXES IN MSUF1; MATCNT COUNTS NUMBER OF DIFFERENT
C    ROOTS I.E. ALL WORDS WITH SAME ROOT HAVE SAME MATCNT NUMBER; NMAT COUNTS
C    NUMBER OF WORDS IN MATCH; NINP INDEXES MWORDS
        NSUFX = L
      MSLF2(M) = 0
      LEXL(M) = 0
      WORDS = 0
      FORDS = 0
      MATCNT = 1
C SUBROUTINE WORDS READS WORDS FROM INPUT TEXT TAPE INTO TABLE MWORDS
      CALL WORDS
      NINP = 1
  100 NMAT = 0
  101 NMAT = NMAT+1
      IF (NMAT)GT(1600),GO TO 10
  102 M1 = MWORDS(NINP)
      IF (M1)E(KEND),GO TO 132
C CHECK TO SEE IF WORD IS ON FUNCTION-WORD LIST
      J = 0
      K = NFNC
  103 I = (J+K)/2
      IF (I)E(K),GO TO 108
      IF (I)E(J),GO TO 108
      IF (M1)E(KFNC(I)),GO TO 105
      IF (M1)GT(KFNC(I)),GO TO 104
      K = I
      GO TO 103
  104 J = I
      GO TO 103
  105 FORDS = FORDS+1
```

23

```
SUFFIX PROGRAM
            NINP = NINP+4
            IF (NINP)LT(128),GO TO 102
            CALL WORDS
            WORDS = WORDS+32
            NINP = 1
            GO TO 102
C BEGIN PROCEDURE FOR PROCESSING CONTENT WORDS
  108 MATCH(NMAT) = MWORDS(NINP)
            NMAT = NMAT+1
            NINP = NINP+1
            MATCH(NMAT) = MWORDS(NINP)
            NMAT = NMAT+1
            NINP = NINP+1
            MATCH(NMAT) = MWORDS(NINP)
            NMAT = NMAT+1
            NINP = NINP+1
            MATCH(NMAT) = MWORDS(NINP)
            NINP = NINP+1
            IF (NINP)LT(128),GO TO 110
            CALL WORDS
            WORDS = WORDS+32
            NINP = 1
  110 M2 = MWORDS(NINP)
C SUBROUTINE COM3 COMPARES FIRST THREE CHARACTERS OF TWO TEXTUAL WORDS TO SEE
C  IF THEY MATCH; IF SENSE LIGHT IS ON,THERE IS A MATCH; CONTINUE TO FILL TABLE
C       MATCH UNTIL EACH WORD WHICH DOES NOT MATCH
            CALL COM3
            IF (SENSE LIGHT 7)101,113
C BEGIN SEARCH FOR TEXTUAL WORD WHICH DOES NOT HAVE A MATCNT I.E. WHICH HAS NOT
C  BEEN PAIRED WITH ANOTHER WORD(S); M INDEXES TABLE MATCH FROM THE TOP AND N
C  INDEXES MATCH FROM THE BOTTOM. IF TEXTUAL WORD(M) HAS A MATCNT,THEN SEARCH IS
C  MADE FOR WORD(N) WHICH HAS NO MATCNT AND THEN ATTEMPT IS MADE TO MATCH IT
C  WITH WORD(M); IF WORD(M) HAS NO MATCNT,TRY TO MATCH IT WITH FIRST WORD(N)
C HAVING A MATCNT; IF BOTH WORD(M) AND WORD(N) HAVE MATCNTS, LOOK FOR FIRST
C   WORD(N) WITHOUT MATCNT AND TRY TO MATCH IT WITH WORD(M). WHEN N EQUALS M,
C      HAVE PROCESSED THROUGH TABLE AND WILL FALL OUT.
  113 M = 0
  114 IF (M)GTE(NMAT),GO TO 130
            M = M+1
            IF (MATCH(M))E(MA1),GO TO 145
            MA1 = MATCH(M)
            M = M+1
  144 MA2 = MATCH(M)
            M = M+2
            N = NMAT
              GO TO 150
  145 M = M+1
            IF (MATCH(M))NE(MA2),GO TO 144
            M = M+2
            GO TO 114
  116 N = N-4
  150 IF (N)E(M),GO TO 121
            IF (MATCH(M))E(KBLNK),GO TO 111
            IF (MATCH(N))E(KBLNK),GO TO 111
            GO TO 116
  111 N = N-2
            M2 = MATCH(N)
            N = N-1
            M1 = MATCH(N)
```

```
SUFFIX PROGRAM
            N = N+3
            M3 = MA1
            M4 = MA2
      C  SUBROUTINE STEM CHECKS,TWO WORDS AT A TIME, FOR SUFFIXES = IF FOUND, SENSE
      C        LIGHT IS TURNED ON
            CALL STEM
            IF (SENSE LIGHT 7)117,116
        117 IF (MATCH(M))NE(KBLNK),GO TO 120
            IF (MATCH(N))E(KBLNK),GO TO 119
            MATCH(M) = MATCH(N)
            GO TO 116
        119 MATCH(M) = MATCNT
            MATCH(N) = MATCNT
            MATCNT = MATCNT+1
            GO TO 116
        120 MATCH(N)=MATCH(M)
            GO TO 116
        121 IF (MATCH(M))NE(KBLNK),GO TO 114
            MATCH(M) = MATCNT
            MATCNT = MATCNT+1
            GO TO 114
        130 DO 131 J = 1,NMAT,1
            M1 = MATCH(J)
            CALL CUT
        131 CONTINUE
            GO TO 100
        132 DO 135 I = 1,128,1
            M1 = KEND
            CALL CUT
        135 CONTINUE
            LOCKOUT 3
            WORDS = WORDS+(NINP-1)/4
            PRINT 136,WORDS,FORDS
        136 FORMAT(1H02(F8))
            STOP
         10 PAUSE
            STOP
            COMPLETE
            END$
```

J. Output

The output of this portion of the program is a tape containing the text words, arranged alphabetically, and coded with their respective MATCNTs.

At the end of the program SUFFIX, one line of printer output will show how many function words were deleted. Tape D will contain the content words from tape E. The different root forms of content words are set apart by having different MATCNTs. Like root forms will have like MATCNTs. The MATCNT is contained in the fourth word of each four-word entry. Tape D will be the input for the program THESAUR.

IV. Thesaurus-Building Program (THESAUR)

A. Tables

KARD (10 computer words): Card input table. Holds contents of one Hollerith card.

LINE (15 computer words): This table, used for printouts, holds the contents of one print line.

KRIM1 (1000 computer words): This table holds the first half (first computer word) of each of the primary words in the thesaurus.

KRIM2 (1000 computer words): This table holds the second half (second computer word) of each of the primary words in the thesaurus.

KASS1 (1000 computer words): This table holds the first half (first computer word) of each of the associated words in the thesaurus.

KASS2 (1000 computer words): This table holds the second half (second computer word) of each of the associated words in the thesaurus.

KMAT1 (1000 computer words): Contains the MATCNTs for the primary words.

KMAT2 (1000 computer words): Contains the MATCNTs for the associated words.

KTEXG, KTEXM, KTEXF, KTEXN, and KTEXT (each 128 computer words): Buffer tables which will hold, respectively, a block of textual words for input from tape or output to tape.

B. Items

I, J, K, L, M, N are items used to index various tables. They act as index registers.

M1, M2, M3, M4 serve as temporary storage cells.

NONG, NONM, NONN contain counts of the number of entries in buffer tables KTEXG, KTEXM, KTEXN.

KTEXT1, KTEXT2, KTEXT3, KTEXT4 are storage cells which contain the four computer words of a textual entry or primary and associated words.

KTEXT5, KTEXT6, KTEXT7, KTEXT8 are storage cells which contain the four computer words of the previous textual entry or primary and associated words.

KBLNK: This item contains a word of blanks.

KEND: This item contains the word, "THEEND."

INENT: This item contains a count of the number of thesaurus pairs in tables KRIM and KASS.

LEMP: This item is a fixed-point temporary storage cell.

NVAP: This item indicates whether or not to print the indexes for the textual words. If NVAP is not set to blank, the indexes are printed.

LWORD1 and LWORD2 are storage cells containing the last word read in from the input text.

NFREQ: This item holds the frequency count for each root form.

NP: This item contains a count of the number of words in a given print line.

NWDS: This item contains a count of the total number of words in the input text.

MATCNT: A unique number for each root form (1, 2, 3.....n).

KBUILD: Used to indicate an initial run and that the card input thesaurus is to be written on tape F.

C. Main Routine

1. The first short section of the program sets up Hollerith constants and the items indexing the buffer tables.

2. READ IN THESAURUS: This section of the program reads in the list of primary and associated words upon which the textual thesaurus will be based. One primary word and one associated word is listed on each input card. If a number of words are associated with one primary word, the primary word is repeated on each card until all the associated words have been listed.

3. SORT CARD INPUT IN PREPARATION FOR A MERGE WITH THE THESAURUS: A thesaurus may exist on tape from a previous computer run. If so, the new synonyms, read in from cards, have to be merged with the old synonyms on tape. If there is no thesaurus tape, the new synonyms still have to be sorted for processing of the text and forming of a thesaurus tape. The card word pairs are sorted so that the primary word is of major order, and the associated word of minor order.

4. CHECK TO SEE IF PLANNING TO MERGE OR BUILD THE THESAURUS: During the initial run of the program on a portion of the text a thesaurus tape will not exist. This fact is specified to the computer by means of card input (KBUILD). If there is not a thesaurus tape on F and KBUILD does not equal the word "BUILD" the program will halt. If KBUILD does equal "BUILD" then the primary and associated words from the card input constitute the entire present thesaurus, and it is put on tape F.

5. MERGE THESAUR TAPE (M) WITH THESAUR CARDS BACK ONTO (F): If KBUILD does not equal "BUILD" then the thesaurus formed on tape from a previous run is put onto tape M so that it can be merged with the new synonyms from card input onto tape F.

6. SAVE PRIMARY AND ASSOCIATED WORDS THAT APPEAR IN THE TEXT: A comparison is made of words on the text tape E against primary words on thesaurus

tape F. When a match is found the primary and associated words are saved in tables called KRIM1, KRIM2, KASS1, KASS2. After comparing all the textual words or primary words these tables are sorted so that the associated word is of major order. Then a comparison is made of the words on the text tape against words appearing in KASS1 and KASS2, saving only those synonym pairs where the associated word also appears in the text.

7. SORT TZZE ON MATCNT USING M, N (2 and 3) AND G (6): Initially, tape units are present; K is set to 128 preparatory to reading information into KTEXT. The reading will not proceed unless the index indicates all of KTEXT has been processed; therefore, for the first input, K must be set as though KTEXT has been processed. Subroutine INTEXT places a textual word entry in KTEXT1, 2, 3, and 4; after the second call to INTEXT, therefore, two textual words occupy KTEXT1, 2, 3, 4 and KTEXT5, 6, 7, 8, respectively. This section of the program arranges the textual words on tape TZZE on the basis of their MATCNTs; the words are sorted so that the MATCNT numbers are in sequence. The sort proceeds by examining the MATCNT of each textual word. All entries for which the MATCNTs are already in sequential order go onto tape G; those for which the MATCNTs are out of order go onto tape M or N. This section of the program also sorts and merges the out-of-order MATCNTs which have been read out on tapes M and N. Because the correctly sorted MATCNTs appear on tape N and we subsequently want to refer to it on tape M, M is set to N. Throughout the procedure described in this section and later sections, 128 word blocks which are not filled with textual words are filled up with the word, "THEEND."

8. MERGE M AND G (6) ONTO N: This section of the program merges G, which contains all textual words for which the MATCNTs were already in sequential order, and M, which contains textual words for which the MATCNTs were out of sequential order; at this point, the MATCNTs on both tapes are in sequential order and it is just a matter of fitting those on tape M into the main text. At the end of this procedure, tape N contains all textual words, arranged by MATCNTs in numerical sequence.

9. PUT FREQUENCY COUNT OF WORDS WITHIN MATCNT ON M: This section of the program totals up the number of occurrences of textual words with the same MATCNT, and stores MATCNTX, the frequency count, on tape M. An example of the output showing the results of this step and step 6 is as follows:

MATCNT	TEXTUAL WORD	FREQUENCY COUNT
3	ABSENCE ABSENT	4
4	ABSOLUTE	1
5	ACCELERATED	1
6	ACCEPTANCE ACCEPTED ACCEPT	4

10. ADD FREQUENCY COUNT TO DATA AND PUT ON G: At the beginning of this section of the program, tape N contains all textual words (i.e., content words) sorted sequentially by MATCNT and tape M contains MATCNTs and frequency counts. This section of the program merges those data by comparing MATCNTs and puts the merged material (shown in output above) on tape G; also carried along to tape G, but not shown above, are the index numbers indicating

where in the text the words occurred.

11. ADD FREQUENCY COUNT TO THESAUR (KMAT2): At the time when the text and associated words are being printed, we also want to be able to print the frequency count of associated words; to do so, it is necessary to have the frequency count immediately available. For this purpose, the program compares text words against associated words in KASS1 and KASS2; when a match is found, it stores KMAT2, the frequency count.

12. SORT THESAUR (PRIMARY WORD OF MAJOR ORDER): The program performs a sort of KRIM1, KRIM2, KASS1, KASS2, and KMAT2 so that KRIM1 and KRIM2 are of major order.

13. ADD FREQUENCY COUNT TO THESAUR (KMAT1): This section of the program operates in the same way as the section described in 11, above. Here, however, the object is to save the frequency counts of primary words, rather than associated words.

14. SORT THE THESAURUS ON PRIMARY WORDS THEN BY MATCNT FOR PRINTING: The primary words are first sorted alphabetically so that words grouped by a single MATCNT will be in alphabetical order. The associated words are shifted along with the primary words in order to maintain the correct linkings. Both the alphabetical sort and the subsequent sort on MATCNTs are of the shuttle-sort type. At the end of this section of the program, KRIM1 and KRIM2 will contain the primary words, arranged so that MATCNTs are in numerical sequence and so that, within a given MATCNT, the words are in alphabetical order. The KASS and KMAT tables are rearranged to correlate with the KRIM tables.

15. Statements 440-450: This short section, preparatory to setting up the final output format, assigns values to some of the variables to be used.

16. Statements 450-452: This section of the program places the index entries for the textual words in the print line and prints them.

17. Statements 452-470: This section of the program, which is operationally part of the preceding section, prints out the textual words and their MATCNTs and frequency counts. The MATCNT and frequency count are printed out just once for any group of words with a common MATCNT.

18. Statements 470-473: This coding checks to see whether the MATCNT of a given textual word equals that of a given primary word in the thesaurus. If the MATCNT of the textual word is less than the MATCNT of the primary word, the next textual word is brought in. If the MATCNT of the textual word is greater than the MATCNT of the primary word, the next primary word is brought in. If the MATCNTs match, the textual word matches the primary word and program control is transferred to statement 480.

19. Statements 480-500: Initially, the associated word (and its MATCNT) linked to the primary word is placed in the print line. The frequency count of the associated word is also placed in the print line and the line is printed. The first DO-Loop checks to see if the associated word is also a primary word. If it is not, the next associated word is examined, etc., until the list of associated words is exhausted. Then the next textual word is compared with the next word in the primary word list, etc. If the associated word is a primary word, then its associated words are printed out and saved on tape N; these are also checked to see if any or all are also primary words. This procedure continues down through five levels

(note the five nested DO-Loops).

20. Statements 500-542: The total number of words (content words) appearing in the text is printed.

21. (F) ONTO (M) THEN MERGE (M) AND (N) BACK ONTO (F): This is a process necessary to update the thesaurus tape F with the new linkings formed on tape M. The information on tape F is stored on tape M. Then tape M and tape N are merged onto tape F. After everything has been merged onto tape F, tape F is rewound, and the program halts.

D. Subroutines for THESAUR

CLEAR: This subroutine clears the print line by setting it equal to blanks.

CON1: This subroutine converts M1 from binary to Hollerith code.

DECON: This subroutine converts the index (chapter-paragraph-line-word) for a given textual word from binary to Hollerith code. Also in preparation for printout, the converted index entry is condensed from four to three computer words (M2, M3, M4).

FREQ: This subroutine extracts the frequency count for a given textual word, converts it from binary to Hollerith code, and saves the count, left-justified in N2, for possible FORTRAN manipulation.

INF: This subroutine brings in one textual word entry from buffer table KTEXF.

ING: This subroutine brings in one textual word entry from buffer table KTEXG.

INM: This subroutine brings in one textual word entry from buffer table KTEXM.

INM2: This subroutine brings in two computer words from buffer table KTEXM.

INN: This subroutine brings in one textual word entry from buffer table KTEXN.

INTEXT: This subroutine brings in one textual word entry from buffer table KTEXT.

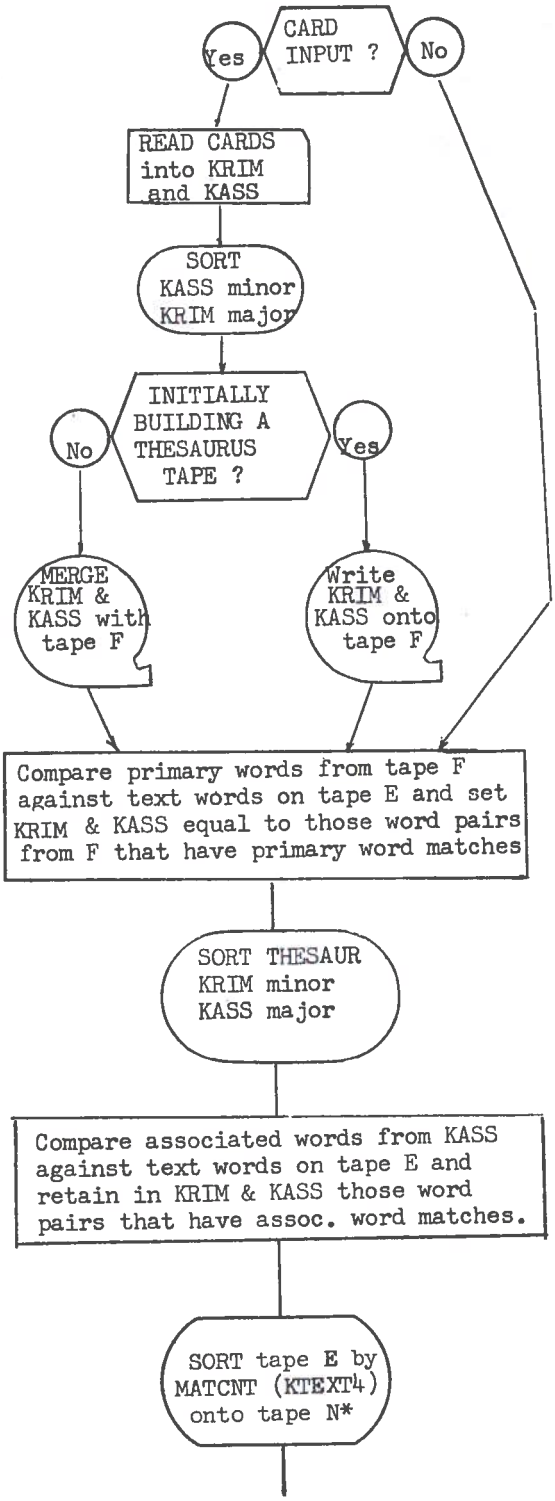
OUTG: This subroutine outputs a textual word entry to buffer table KTEXG.

OUTM: This subroutine outputs a textual word entry to buffer table KTEXM.

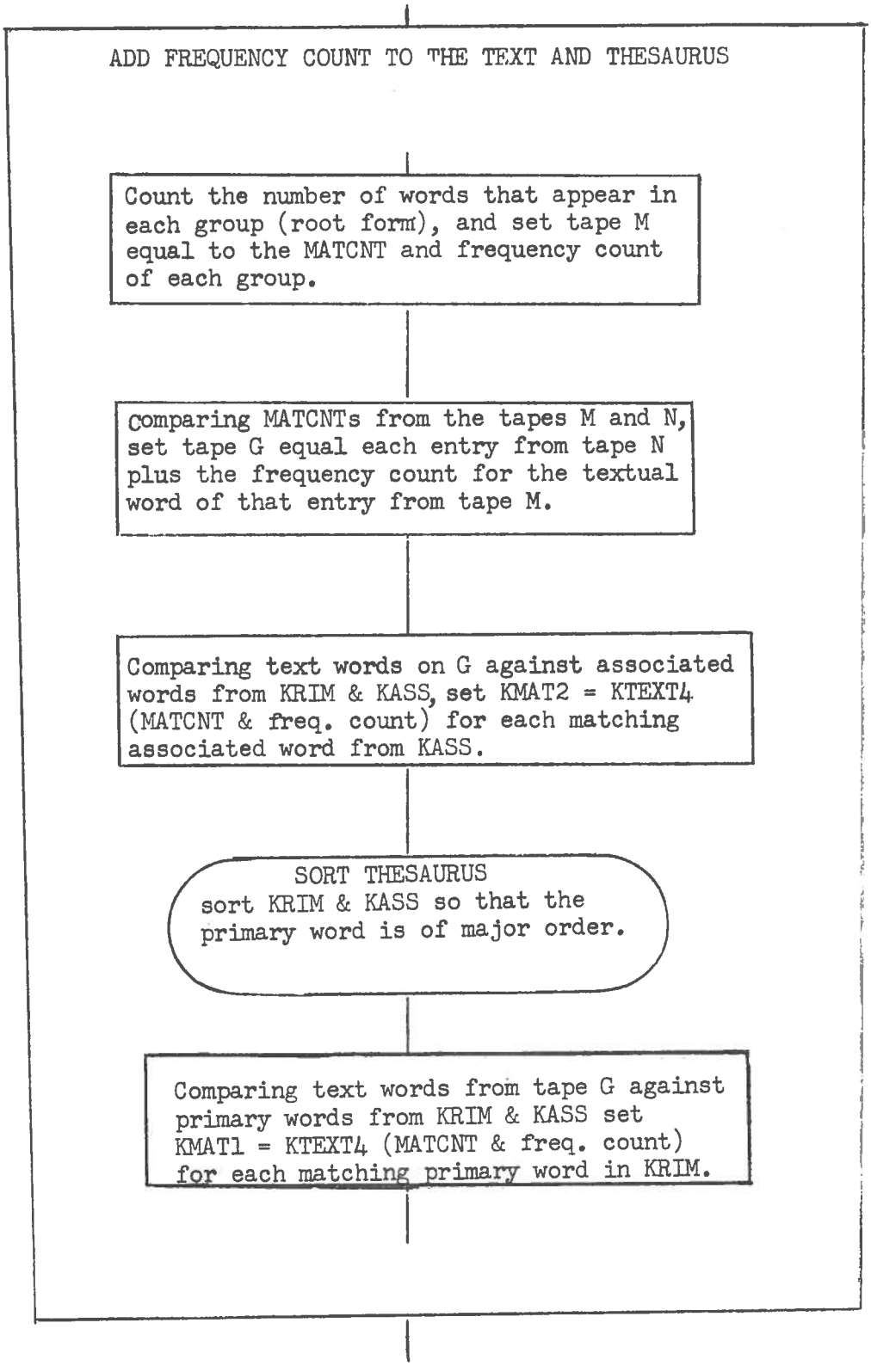
OUTM2: This subroutine outputs two computer words to buffer table KTEXM.

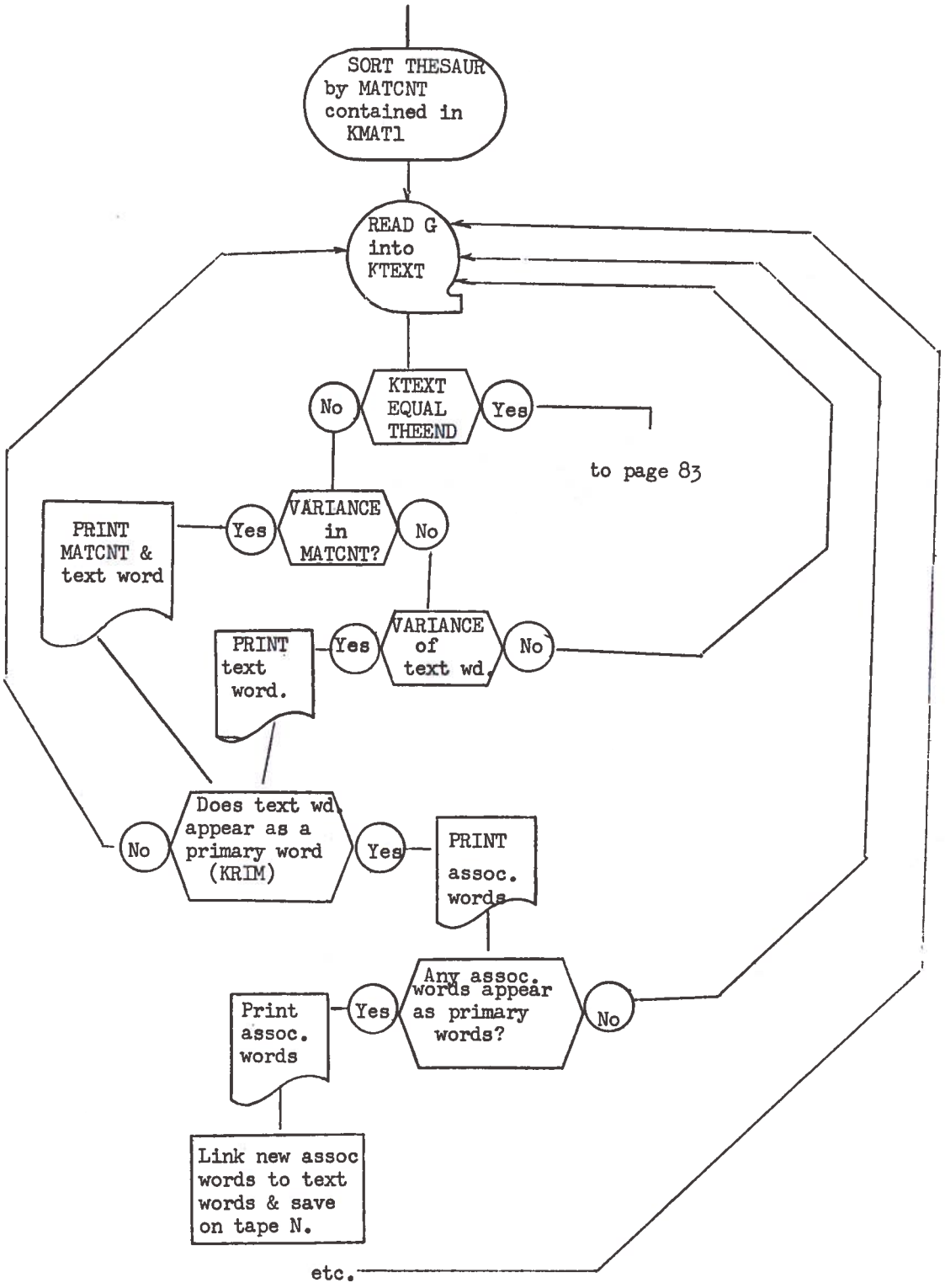
OUTN: This subroutine outputs a textual word entry to buffer table KTEXN.

E. Flow Chart

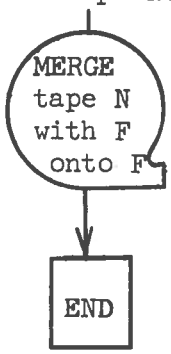


*So that textual words are grouped according to root form.


ADD FREQUENCY COUNT TO THE TEXT AND THESAURUS
Count the number of words that appear in each group (root form), and set tape M equal to the MATCNT and frequency count of each group.
Comparing MATCNTs from the tapes M and N, set tape G equal each entry from tape N plus the frequency count for the textual word of that entry from tape M.
Comparing text words on G against associated words from KRIM & KASS, set KMAT2 = KTEXT4 (MATCNT & freq. count) for each matching associated word from KASS.
SORT THESAURUS
sort KRIM & KASS so that the primary word is of major order.
Comparing text words from tape G against primary words from KRIM & KASS set KMAT1 = KTEXT4 (MATCNT & freq. count) for each matching primary word in KRIM.


SORT THESAUR by MATCNT contained in KMAT1
READ G into KTEXT
KTEXT EQUAL THEEND
No
Yes
to page 83
VARIANCE in MATCNT?
Yes
No
PRINT MATCNT & text word
VARIANCE of text wd.
Yes
No
PRINT text word.
Does text wd appear as a primary word (KRIM)
No
Yes
PRINT assoc. words
Any assoc. words appear as primary words?
Yes
No
Print assoc. words
Link new assoc words to text words & save on tape N.
etc.

The textual words have been printed with possible associated words, and new linkings of primary and associated words saved on tape N.



(to update the THESAUR tape)

The "etc." at the bottom of page 82 was meant to imply that the process of attempting to link new associated words with former primary words could be carried on indefinitely or until all possible combinations have been exhausted. In the program, the output shows up to five levels of linking. The flow chart shows only two levels.

Tape E is the text tape of textual words in alphabetical order.

Tape F contains the thesaurus words sorted alphabetically by primary words (the major order) and associated words (the minor order).

Tape G is a tape of the textual words with frequency count in numerical order of MATCNT.

Tape M is a working tape.

Tape N is a working tape.

Text tapes consist of four-word entries. The first two computer words (KTEXT1, KTEXT2) contain the textual word.

The third word (KTEXT3) is divided into four parts:

1) Chapter count;

2) Paragraph count;

3) Line count;

4) Word count.

The fourth word (KTEXT4) is divided into three parts:

1) MATCNT (a number associated with each root form, e.g., 1, 2, 3, 4.....n, where n is the total number of different root forms);

2) Frequency count (number of words that have the same root);

3) The third part as yet is unused.

F. Program Listing

```
THESAUR PROGRAM
        I           THESAUR
                    IDENTIFY F, 16K, 8X, 777W
      SUBROUTINE CLEAR                                                  0001
C        SETS THE PRINT LINE EQUAL TO BLANKS                            0002
      COMMON I,J,K,L,M,N                                                0003
      COMMON M1,M2,M3,M4                                                0004
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NORG,NONM,NONN,NORF                                        0006
      COMMON KTEXT1,KTEXT2,KTEXT3,KTEXT4                                0007
      COMMON KTEXT5,KTEXT6,KTEXT7,KTEXT8                                0008
      COMMON KBLNK,KEND,INENT,LEMP,NVAP,LWORD1,LWORD2                   0009
      COMMON KARD,LINE                                                  0010
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                              0011
      DIMENSION KTEXM(128)                                              0012
      DIMENSION KTEXN(128)                                              0013
      DIMENSION KTEXT(128)                                              0014
      DIMENSION KARD(10)                                                0015
      DIMENSION LINE(15)                                                0016
      CONTINUE                                                          0017
      STARTTAC                                                          0018
                  TMD     L/LINE$                                       0019
                  TDXLC   ,1$                                           0020
                  TMD     KBLNK$                                        0021
        L         RPTA    15$                                           0022
                  TDM     1,1$                                          0023
                  ENDTAC                                                0024
      RETURN                                                            0025
      END                                                               0026
      SUBROUTINE CON1(M1)                                               0027
C        CONVERT M1 FROM BINARY TO HOLLERITH CODE                       0028
      COMMON I,J,K,L,M,N                                                0029
      COMMON M1,M2,M3,M4                                                0030
      CONTINUE                                                          0031
      STARTTAC                                                          0032
                  TMA     M1$                                           0033
                  SRA     32$                                           0034
        S         BIN2BCD $                                             0035
        S         ZSUPP   $                                             0036
                  TMQ     0/606060$                                     0037
                  SLAQ    12$                                           0038
                  TAM     M1$                                           0039
                  ENDTAC                                                0040
      RETURN                                                            0041
      END                                                               0042
      SUBROUTINE DECON                                                  0043
C        DECODES M1 (CPLD COUNT) AND CONVERTS BINARY TO HOLLERITH       0044
      COMMON I,J,K,L,M,N                                                0045
      COMMON M1,M2,M3,M4                                                0046
      CONTINUE                                                          0047
      STARTTAC                                                          0048
                  CA        4                                           0049
                  TMQ     M1$                                           0050
                  SLAQ    12$                                           0051
                  TAM     M1$                                           0052
                  CA        4                                           0053
                  SLAQ    12$                                           0054
```

```
                TAM     M2$                                              0055
                CA        $                                              0056
                SLAQ    12$                                              0057
                TAM     M3$                                              0058
                CA        $                                              0059
                SLAQ    12$                                              0060
                TAM     M4$                                              0061
      S         BIN2BCD M1$                                              0062
                TAM     M1$                                              0063
      S         BIN2BCD M2$                                              0064
                TAM     M2$                                              0065
      S         BIN2BCD M3$                                              0066
                TAM     M3$                                              0067
      S         BIN2BCD M4$                                              0068
                TAM     M4$                                              0069
                TMQ     0/34T5$                                          0070
                SRAQ    24$                                              0071
                TMA     0/4040T47$                                       0072
                SRAQ    12$                                              0073
                TMA     M3$                                              0074
                SRAQ    6$                                               0075
                TQM     M4$                                              0076
                SRAQ    18$                                              0077
                TMA     0/4040T47$                                       0078
                SRAQ    12$                                              0079
                TMA     M2$                                              0080
                SRAQ    18$                                              0081
                TQM     M3$                                              0082
                SRAQ    6$                                               0083
                TMA     0/4040T47$                                       0084
                SRAQ    12$                                              0085
                TMA     M1$                                              0086
                SRAQ    24$                                              0087
                TMA     0/74T47$                                         0088
                SRAQ    6$                                               0089
                TQM     M2$                                              0090
                ENDTAC                                                   0091
      RETURN                                                             0092
      END                                                                0093
      SUBROUTINE  FREQ(N1,N2)                                            0094
C         EXTRACTS FREQUENCY CONST. N1 = HOLLERITH, N2 = BINARY          0095
      CONTINUE                                                           0096
      STARTTAC                                                           0097
                TMA     N1$                                              0098
                SLA     18$                                              0099
                TAM     N2$                                              0100
                SRA     N2$                                              0101
        S       BIN2BCD N                                                0102
        S       ZSUPP   N                                                0103
                TA      N1$                                              0104
                ENDTAC                                                   0105
      RETURN                                                             0106
      END                                                                0107
      SUBROUTINE INF(N1,N2,N3,N4)                                        0108
      COMMON I,J,K,L,M,N                                                 0109
      COMMON M1,M2,M3,M4                                                 0110
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN,NONE                                         0112
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                               0113
```

```
      DIMENSION KTEXM(128)                                              0114
      DIMENSION KTEXN(128)                                              0115
      DIMENSION KTEXT(128)                                              0116
      IF (NONF)GTE(128),GO TO 20                                        0117
   10 M1 = KTEXF(NONF)
      M2 = KTEXF(NONF+1)
      M3 = KTEXF(NONF+2)
      M4 = KTEXF(NONF+3)
      NONF = NONF+4                                                     0122
      RETURN                                                            0123
   20 STARTTAC                                                          0124
        R       JMP     IOINT $                                         0125
                N/6T23;N/1T39 $                                         0126
                HLT     TZZF  $   SYS 6 (FORTRAN 0)                     0127
                HLT     KTEXF $
                ENDTAC                                                  0129
      NONF = 1                                                          0130
      GO TO 10                                                          0131
      END                                                               0132
      SUBROUTINE ING(N1,N2,N3,N4)                                       0133
      COMMON I,J,K,L,M,N                                                0134
      COMMON M1,M2,M3,M4                                                0135
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN                                             0137
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                              0138
      DIMENSION KTEXM(128)                                              0139
      DIMENSION KTEXN(128)                                              0140
      DIMENSION KTEXT(128)                                              0141
      IF (NONG)GTE(128),GO TO 20                                        0142
   10 M1 = KTEXG(NONG)                                                  0143
      M2 = KTEXG(NONG+1)                                                0144
      M3 = KTEXG(NONG+2)                                                0145
      M4 = KTEXG(NONG+3)                                                0146
      NONG = NONG+4                                                     0147
      RETURN                                                            0148
   20 STARTTAC                                                          0149
        R       JMP     IOINT $                                         0150
                N/6T23;N/1T39 $     READ ONE BLOCK                      0151
                HLT     TZZG $      FROM SYS 7  (FORTRAN 6)             0152
                HLT     KTEXG $                                         0153
                ENDTAC                                                  0154
      NONG = 1                                                          0155
      GO TO 10                                                          0156
      END                                                               0157
      SUBROUTINE INM(N1,N2,N3,N4)                                       0158
      COMMON I,J,K,L,M,N                                                0159
      COMMON M1,M2,M3,M4                                                0160
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN                                             0162
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                              0163
      DIMENSION KTEXM(128)                                              0164
      DIMENSION KTEXN(128)                                              0165
      DIMENSION KTEXT(128)                                              0166
      IF (NONM)GTE(128),GO TO 20                                        0167
   10 M1 = KTEXM(NONM)                                                  0168
      M2 = KTEXM(NONM+1)                                                0169
      M3 = KTEXM(NONM+2)                                                0170
      M4 = KTEXM(NONM+3)                                                0171
```

```
      NCAM = NCAM+4                                                     0172
      RETURN                                                            0173
   20 IF (M)E(3),GO TO 30                                               0174
      STARTTAC                                                          0175
        R       JMP     IOINT $                                         0176
                N/6T23;N/1T39 $                                         0177
                HLT     TZZC  $                                         0178
                HLT     KTEXM $                                         0179
                ENDTAC                                                  0180
      GO TO 40                                                          0181
   30 STARTTAC                                                          0182
        R       JMP     IOINT $                                         0183
                N/6T23;N/1T39 $                                         0184
                HLT     TZZC  $                                         0185
                HLT     KTEXM $                                         0186
                ENDTAC                                                  0187
   40 NCAM = 1                                                          0188
      GO TO 10                                                          0189
      END                                                               0190
      SUBROUTINE INM2(N1,N2)                                            0191
      COMMON I,J,K,L,M,N                                                0192
      COMMON M1,M2,M3,M4                                                0193
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT                             0195
      COMMON NCAG,NONM,NONN
      DIMENSION KTEXF(128)                                              0196
      DIMENSION KTEXG(128)                                              0197
      DIMENSION KTEXM(128)                                              0198
      DIMENSION KTEXN(128)                                              0199
      DIMENSION KTEXT(128)                                              0200
      IF (NONM)GTE(128),GO TO 20                                        0201
   10 M1 = KTEXM(NONM)                                                  0202
      M2 = KTEXM(NONM+1)                                                0203
      NCAM = NCAM+2                                                     0204
      RETURN                                                            0205
   20 IF (M)E(3),GO TO 30                                               0206
      STARTTAC                                                          0207
        R       JMP     IOINT $                                         0208
                N/6T23;N/1T39 $                                         0209
                HLT     TZZC  $                                         0210
                HLT     KTEXM $                                         0211
                ENDTAC                                                  0212
      GO TO 40                                                          0213
   30 STARTTAC                                                          0214
        R       JMP     IOINT $                                         0215
                N/6T23;N/1T39 $                                         0216
                HLT     TZZC  $                                         0217
                HLT     KTEXM $                                         0218
                ENDTAC                                                  0219
   40 NCAM = 1                                                          0220
      GO TO 10                                                          0221
      END                                                               0222
      SUBROUTINE INN(N1,N2,N3,N4)                                       0223
      COMMON I,J,K,L,M,N                                                0224
      COMMON M1,M2,M3,M4
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT                             0226
      COMMON NCAG,NONM,NONN
      DIMENSION KTEXF(128)                                              0227
      DIMENSION KTEXG(128)                                              0228
      DIMENSION KTEXM(128)                                              0229
      DIMENSION KTEXN(128)
```

```
      DIMENSION KTEXT(128)                                              0230
      IF (NONN).GTE.(128),GO TO 20                                      0231
   10 N1 = KTEXT(NONN)                                                  0232
      N2 = KTEXT(NONN+1)                                                0233
      N3 = KTEXT(NONN+2)                                                0234
      N4 = KTEXT(NONN+3)                                                0235
      NONN = NONN+4                                                     0236
      RETURN                                                            0237
   20 IF (N).E(3),GO TO 30                                              0238
      STARTTAC                                                          0239
        R       JMP     TOINT $                                         0240
                N/6T23,N/1T39 $                                         0241
                HLT     TZZC  $                                         0242
                HLT     KTEXT $                                         0243
                ENDTAC                                                  0244
      GO TO 40                                                          0245
   30 STARTTAC                                                          0246
        R       JMP     TOINT $                                         0247
                N/6T23,N/1T39 $                                         0248
                HLT     TZZD  $                                         0249
                HLT     KTEXT $                                         0250
                ENDTAC                                                  0251
   40 NONN = 1                                                          0252
      GO TO 10                                                          0253
      END                                                               0254
      SUBROUTINE INTEXT                                                 0255
C        BUFFERS INPUT FROM TZZE TO KTEXT1,2,3,4                        0256
      COMMON I,J,K,L,M,N                                                0257
      COMMON M1,M2,M3,M4                                                0258
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN,NONF                                        0260
      COMMON KTEXT1,KTEXT2,KTEXT3,KTEXT4                                0261
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                              0262
      DIMENSION KTEXM(128)                                              0263
      DIMENSION KTEXN(128)                                              0264
      DIMENSION KTEXT(128)                                              0265
      IF (K).GTE.(128),GO TO 20                                         0266
   10 KTEXT1 = KTEXT(K)                                                 0267
      K = K+1                                                           0268
      KTEXT2 = KTEXT(K)                                                 0269
      K = K+1                                                           0270
      KTEXT3 = KTEXT(K)                                                 0271
      K = K+1                                                           0272
      KTEXT4 = KTEXT(K)                                                 0273
      K = K+1                                                           0274
      RETURN                                                            0275
   20 CONTINUE                                                          0276
      STARTTAC                                                          0277
        R       JMP     TOINT$                                          0278
                N/6T23,N/1T39$                                          0279
                HLT     TZZE$                                           0280
                HLT     KTEXT$                                          0281
                ENDTAC                                                  0282
      K = 1                                                             0283
      GO TO 10                                                          0284
      END                                                               0285
      SUBROUTINE OUTF(N1,N2,N3,N4)                                      0286
      COMMON I,J,K,L,M,N                                                0287
      COMMON M1,M2,M3,M4                                                0288
```

```
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN,NONF                                        0290
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                              0291
      DIMENSION KTEXM(128)                                              0292
      DIMENSION KTEXN(128)                                              0293
      DIMENSION KTEXT(128)                                              0294
      KTEXF(NONF) = N1
      KTEXF(NONF+1) = N2
      KTEXF(NONF+2) = N3
      KTEXF(NONF+3) = N4
      NONF = NONF+4                                                     0298
      IF (NONF).GTE.(128),GO TO 10                                      0299
      RETURN                                                            0300
   10 STARTTAC                                                          0301
        R       JMP     TOINT $                                         0302
                W/5T23;W/1T39 $                                         0303
                HLT     TZZF  $     SYS 6 (FORTRAN 0)                  0304
                HLT     KTEXF $
                ENDTAC                                                  0306
      NONF = 1                                                          0307
      RETURN                                                            0308
      END                                                               0309
      SUBROUTINE OUTG(N1,N2,N3,N4)                                      0310
      COMMON I,J,K,L,M,N                                                0311
      COMMON M1,M2,M3,M4                                                0312
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN                                             0314
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                              0315
      DIMENSION KTEXM(128)                                              0316
      DIMENSION KTEXN(128)                                              0317
      DIMENSION KTEXT(128)                                              0318
      KTEXG(NONG) = N1                                                  0319
      KTEXG(NONG+1)=N2                                                  0320
      KTEXG(NONG+2)=N3                                                  0321
      KTEXG(NONG+3)=N4                                                  0322
      NONG = NONG+4                                                     0323
      IF (NONG).GTE.(128),GO TO 10                                      0324
      RETURN                                                            0325
   10 STARTTAC                                                          0326
        R       JMP     TOINT $                                         0327
                W/5T23;W/1T39 $     WRITE ONE BLOCK                     0328
                HLT     TZZG  $     ON SYS 7 (FORTRAN 6)                0329
                HLT     KTEXG $                                         0330
                ENDTAC                                                  0331
      NONG = 1                                                          0332
      RETURN                                                            0333
      END                                                               0334
      SUBROUTINE OUTM(N1,N2,N3,N4)                                      0335
      COMMON I,J,K,L,M,N                                                0336
      COMMON M1,M2,M3,M4                                                0337
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN                                             0339
      DIMENSION KTEXF(128)
      DIMENSION KTEXG(128)                                              0340
      DIMENSION KTEXM(128)                                              0341
      DIMENSION KTEXN(128)                                              0342
      DIMENSION KTEXT(128)                                              0343
      KTEXM(NONM) = N1                                                  0344
```

```
      KTEXM(NONM+1)=N2                                                  0345
      KTEXM(NONM+2)=N3                                                  0346
      KTEXM(NONM+3)=N4                                                  0347
      NONM = NONM+4                                                     0348
      IF (NONM).GTE.(128),GO TO 10                                      0349
      RETURN                                                            0350
   10 IF (M).E.(2),GO TO 20                                             0351
      STARTTAC                                                          0352
        R       JMP     IOINT$                                          0353
                N/5T23;N/1T39$          WRITE  ONE BLOCK                0354
                HLT     TZZC$ ON 3                                      0355
                HLT     KTEXM$                                          0356
                ENDTAC                                                  0357
      GO TO 40                                                          0358
   20 STARTTAC                                                          0359
        R       JMP     IOINT$                                          0360
                N/5123;N/1T39$          WRITE ONE BLOCK                 0361
                HLT     TZZC $          ON 2                            0362
                HLT     KTEXM$                                          0363
                ENDTAC                                                  0364
   40 NONM = 1                                                          0365
      RETURN                                                            0366
      END                                                               0367
      SUBROUTINE OUTM2(N1,N2)                                           0368
      COMMON I,J,K,L,M,N                                                0369
      COMMON M1,M2,M3,M4                                                0370
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN                                             0372
      DIMENSION KTEXF(128)                                              0373
      DIMENSION KTEXG(128)                                              0374
      DIMENSION KTEXM(128)                                              0375
      DIMENSION KTEXN(128)                                              0376
      DIMENSION KTEXT(128)                                              0377
      KTEXM(NONM) = N1                                                  0378
      KTEXM(NONM+1)=N2                                                  0379
      NONM = NONM+2                                                     0380
      IF (NONM).GTE.(128),GO TO 10                                      0381
      RETURN                                                            0382
   10 IF (M).E.(2),GO TO 20                                             0383
      STARTTAC                                                          0384
        R       JMP     IOINT $                                         0385
                N/5T23;N/1T39 $                                         0386
                HLT     TZZC  $                                         0387
                HLT     KTEXM $                                         0388
                ENDTAC                                                  0389
      GO TO 40                                                          0390
   20 STARTTAC                                                          0391
        R       JMP     IOINT $                                         0392
                N/5T23;N/1T39 $                                         0393
                HLT     TZZC  $                                         0394
                HLT     KTEXM $                                         0395
                ENDTAC                                                  0396
   40 NONM = 1                                                          0397
      RETURN                                                            0398
      END                                                               0399
      SUBROUTINE OUTN(N1,N2,N3,N4)                                      0400
      COMMON I,J,K,L,M,N                                                0401
      COMMON M1,M2,M3,M4
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT
      COMMON NONG,NONM,NONN                                             0403
```

```
      DIMENSION KTEXF(128)                                              0404
      DIMENSION KTEXG(128)                                              0405
      DIMENSION KTEXM(128)                                              0406
      DIMENSION KTEXN(128)                                              0407
      DIMENSION KTEXT(128)                                              0408
       KTEXN(NONN) = M1                                                 0409
       KTEXN(NONN+1)=M2                                                 0410
       KTEXN(NONN+2)=M3                                                 0411
       KTEXN(NONN+3)=M4                                                 0412
      NONN = NONN+4                                                     0413
      IF (NONN)GTE(128),GO TO 10                                        0414
      RETURN                                                            0415
   10 IF (N)E(2),GO TO 20                                               0416
      STARTTAC                                                          0417
        R       JMP     IOINT $                                         0418
                N/5T23IN/1T39 $     WRITE ONE BLOCK                     0419
                HLT     TZZC  $     ON 3                                0420
                HLT     KTEXN $                                         0421
                ENDTAC                                                  0422
      GO TO 40                                                          0423
   20 STARTTAC                                                          0424
        R       JMP     IOINT $                                         0425
                N/5T23IN/1T39 $     WRITE ONE BLOCK                     0426
                HLT     TZZC  $     ON 2                                0427
                HLT     KTEXN $                                         0428
                ENDTAC                                                  0429
   40 NONN = 1                                                          0430
      RETURN                                                            0431
      END                                                               0432
      FUNCTION LET(X)                                                   0433
      CONTINUE                                                          0434
      STARTTAC                                                          0435
                TDXL    ,0$                                             0436
                TMA     ,0$                                             0437
                JMP     2SRS$                                           0438
                ENDTAC                                                  0439
      END                                                               0440
C        START OF THE MAIN PROGRAM                                      0441
      COMMON I,J,K,L,M,N                                                0442
      COMMON M1,M2,M3,M4
      COMMON KTEXF,KTEXG,KTEXM,KTEXN,KTEXT                              0444
      COMMON NONG,NONM,NONN,NONE                                        0445
      COMMON KTEXT1,KTEXT2,KTEXT3,KTEXT4                                0446
      COMMON KTEXT5,KTEXT6,KTEXT7,KTEXT8                                0447
      COMMON KFLNK,KEND,INEXT,LEMP,NVAP,LWORD1,LWORD2                   0448
      COMMON KARD,LINE                                                  0449
      COMMON KFREQ                                                      0450
      COMMON KTHES,KBUILD                                               0451
      COMMON NP
      DIMENSION KTEXF(128)                                              0452
      DIMENSION KTEXG(128)                                              0453
      DIMENSION KTEXM(128)                                              0454
      DIMENSION KTEXN(128)                                              0455
      DIMENSION KTEXT(128)                                              0456
      DIMENSION KARD(10)                                                0457
      DIMENSION LINE(15)                                                0458
      DIMENSION KRIM1(1000)                                             0459
      DIMENSION KRIM2(1000)                                             0460
      DIMENSION KASS1(1000)                                             0461
      DIMENSION KASS2(1000)
```

```
      DIMENSION KMAT1(1000)                                          0462
      DIMENSION KMAT2(1000)                                          0463
      COMMON N,DS                                                    0464
    1 FORMAT (1H 15A8)                                               0465
    2 FORMAT (10A8)                                                  0466
    3 FORMAT (1H 15A8/1H1)                                           0467
      I = 0                                                          0468
      ICNP = 1                                                       0469
      NCNN = 1                                                       0470
      NCNG = 1                                                       0471
      P = 2
      N = 3
      REWIND 0
      REWIND 2
      READ 2,KARD                                                    0472
      KEND = KARD(2)                                                 0473
      KSWAP = KARD(3)                                                0474
      KBLILD = KARD(4)                                               0475
      KBLNK = KARD(8)                                                0476
      KTHES = LET(8HTHESAUR )                                        0477
      CALL CLEAR                                                     0478
C        READ IN THESAURUS                                           0479
   10 READ 2,KARD                                                    0480
      IF (KARD(1))E(KEND),GO TO 20                                   0481
      I = I+1                                                        0482
      IF (I)LTE(1000),GO TO 19                                       0483
      FALSE 1                                                        0484
   19 KTM1(I) = KARD(1)                                              0485
      KTM2(I) = KARD(2)                                              0486
      KASS1(I) = KARD(3)                                             0487
      KASS2(I) = KARD(4)                                             0488
      GO TO 10                                                       0489
   20 INENT = I                                                      0490
      IF (INENT)LTE(1),GO TO 40                                      0491
C SORT CARD INPUT IN PREPARATION FOR A MERGE WITH THESAURUS          0492
   21 J = 2                                                          0493
   22 I = J-1                                                        0494
      IF (KTM1(I))E(KTM1(J)),GO TO 27                                0495
      IF (KTM1(I))LT(0),GO TO 25                                     0496
      IF (KTM1(J))LT(0),GO TO 26                                     0497
   23 IF (KTM1(J))GT(KTM1(1)),GO TO 26                               0498
   24 LEMP     = KASS1(J)                                            0499
      KASS1(J) = KASS1(I)                                            0500
      KASS1(I) = LEMP                                                0501
      LEMP     = KASS2(J)                                            0502
      KASS2(J) = KASS2(I)                                            0503
      KASS2(I) = LEMP                                                0504
      LEMP     = KTM1(J)                                             0505
      KTM1(J) = KTM1(I)                                              0506
      KTM1(I) = LEMP                                                 0507
      LEMP     = KTM2(J)                                             0508
      KTM2(J) = KTM2(I)                                              0509
      KTM2(I) = LEMP                                                 0510
      J = J-1                                                        0511
      IF (J)LT(2),GO TO 21                                           0512
      GO TO 22                                                       0513
   25 IF (KTM1(J))LT(0),GO TO 23                                     0514
      GO TO 24                                                       0515
   26 J = J+1                                                        0516
      IF (J)GT(INENT),GO TO 40                                       0517
```

```
      GO TO 22                                                          0518
   27 IF (KRIM2(I))E(KRIM2(J)),GO TO 30                                 0519
      IF (KRIM2(I))LT(0),GO TO 29                                       0520
      IF (KRIM2(J))LT(0),GO TO 26                                       0521
   28 IF (KRIM2(J))GT(KRIM2(I)),GO TO 26                                0522
      GO TO 24                                                          0523
   29 IF (KRIM2(J))LT(0),GO TO 28                                       0524
      GO TO 24                                                          0525
   30 IF (KASS1(I))E(KASS1(J)),GO TO 33                                 0526
      IF (KASS1(I))LT(0),GO TO 32                                       0527
      IF (KASS1(J))LT(0),GO TO 26                                       0528
   31 IF (KASS1(J))GT(KASS1(I)),GO TO 26                                0529
      GO TO 24                                                          0530
   32 IF (KASS1(J))LT(0),GO TO 31                                       0531
      GO TO 24                                                          0532
   33 IF (KASS2(I))E(KASS2(J)),GO TO 36                                 0533
      IF (KASS2(I))LT(0),GO TO 35                                       0534
      IF (KASS2(J))LT(0),GO TO 26                                       0535
   34 IF (KASS2(J))GT(KASS2(I)),GO TO 26                                0536
      GO TO 24                                                          0537
   35 IF (KASS2(J))LT(0),GO TO 34                                       0538
      GO TO 24                                                          0539
   36 K = J+1                                                           0540
      DO 37 K = K,INENT,1                                               0541
      KASS2(K-1) = KASS2(K)                                             0542
      KASS1(K-1) = KASS1(K)                                             0543
      KRIM2(K-1) = KRIM2(K)                                             0544
      KRIM1(K-1) = KRIM1(K)                                             0545
   37 CONTINUE                                                          0546
      INENT = INENT-1                                                   0547
      GO TO 22                                                          0548
C CHECK TO SEE IF PLANNING FOR MERGE OR BUILDING OF THESAURUS           0549
   40 NONE = 128                                                        0550
      CALL INF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0551
      IF (KTEXT1)E(KTHES),GO TO 44                                      0552
      IF (KBUILD)E(LET(BUILD   )),GO TO 45                              0553
      PAUSE                                                             0554
   44 IF (KBUILD)E(LET(BUILD   )),GO TO 45                              0555
      NONE = 1
      GO TO 47
   45 I = 0                                                             0556
      REWIND 0
      NONE = 1                                                          0559
      CALL OUTF(KTHES,KTHES,KTHES,KTHES)                                0560
   46 I = I+1                                                           0561
      IF (I)GT(INENT),GO TO 47
      KTEXT1 = KRIM1(I)                                                 0563
      KTEXT2 = KRIM2(I)                                                 0564
      KTEXT3 = KASS1(I)                                                 0565
      KTEXT4 = KASS2(I)                                                 0566
      CALL OUTF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                            0567
      GO TO 46                                                          0568
   47 IF (INENT)E(0),GO TO 60
      NONE = 128
   48 CALL INF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0570
      IF (KTEXT1)E(KEND),GO TO 49                                       0571
      CALL OUTM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                            0572
      GO TO 48                                                          0573
   49 DO 50 I = NONE,128,4                                              0574
      CALL OUTM(KEND,KEND,KEND,KEND)                                    0575
```

```
   50 CONTINUE                                                          0576
C MERGE THESAUR TAPE (M) + THESAUR CARDS BACK ONTO (F)                  0577
      REWIND 0                                                          0578
      REWIND M                                                          0579
      I = 0                                                             0580
      NCNF = 1                                                          0581
      NCNM = 128                                                        0582
      CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0583
      CALL OUTF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                            0584
      CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0585
   52 I = I+1                                                           0586
      IF (I)GT(INENT),GO TO 56                                          0587
      KTEXT5 = KRIM1(I)                                                 0588
      KTEXT6 = KRIM2(I)                                                 0589
      KTEXT7 = KASS1(I)                                                 0590
      KTEXT8 = KASS2(I)                                                 0591
   53 STARTTAC                                                          0592
                TMA     KTEXT1$                                         0593
                TMD     KTEXT5$                                         0594
                JAED    (P)+3H$                                         0595
                JAGD    LS1   $                                         0596
                JMP     LS2   $                                         0597
                TMA     KTEXT2$                                         0598
                TMD     KTEXT6$                                         0599
                JAED    (P)+3H$                                         0600
                JAGD    LS1   $                                         0601
                JMP     LS2   $                                         0602
                TMA     KTEXT3$                                         0603
                TMD     KTEXT7$                                         0604
                JAED    (P)+3H$                                         0605
                JAGD    LS1   $                                         0606
                JMP     LS2   $                                         0607
                TMA     KTEXT4$                                         0608
                TMD     KTEXT8$                                         0609
                JAED    (P)+3H$                                         0610
                JAGD    LS1   $                                         0611
                JMP     LS23                                            0612
                ENDTAC                                                  0613
      GO TO 52                                                          0614
  LS1 CALL OUTF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)
      GO TO 52                                                          0620
  LS2 CALL OUTF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)
      CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0616
      IF (KTEXT1)NE(KEND),GO TO 53
   55 CALL OUTF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                            0621
      I = I+1                                                           0622
      IF (I)GT(INENT),GO TO 57                                          0623
      KTEXT5 = KRIM1(I)                                                 0624
      KTEXT6 = KRIM2(I)                                                 0625
      KTEXT7 = KASS1(I)                                                 0626
      KTEXT8 = KASS2(I)                                                 0627
      GO TO 55                                                          0628
   56 CALL OUTF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                            0629
      CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0630
      IF (KTEXT1)NE(KEND),GO TO 56
   57 REWIND M                                                          0633
      DO 59 I = NONF,128,4                                              0634
      CALL OUTF(KEND,KEND,KEND,KEND)                                    0635
   59 CONTINUE                                                          0636
C SAVE PRIMARY AND ASSOSIATED WORDS THAT APPEAR IN THE TEXT             0637
```

```
 60 REWIND 0                                                          0638
    I = 0                                                             0639
    K = 128                                                           0640
    NCNF = 128                                                        0641
    CALL INF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                             0642
    CALL INF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                             0643
 62 CALL INTEXT                                                       0644
    IF (KTEXT1)E(KEND),GO TO 75                                       0645
    IF (LWORD1)NE(KTEXT1),GO TO 63                                    0646
    IF (LWORD2)NE(KTEXT2),GO TO 63                                    0647
    GO TO 62                                                          0648
 63 LWORD1 = KTEXT1                                                   0649
    LWORD2 = KTEXT2                                                   0650
 64 IF (KTEXT5)NE(KTEXT1),GO TO 67                                    0651
    IF (KTEXT6)NE(KTEXT2),GO TO 71                                    0652
 65 I = I+1                                                           0653
    IF (I)LTE(1000),GO TO 66                                          0654
    PAUSE                                                             0655
 66 KRIM1(I) = KTEXT5                                                 0656
    KRIM2(I) = KTEXT6                                                 0657
    KASS1(I) = KTEXT7                                                 0658
    KASS2(I) = KTEXT8                                                 0659
    CALL INF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                             0660
    IF (KTEXT5)E(KEND),GO TO 75                                       0661
    IF (KRIM1(I))NE(KTEXT5),GO TO 62                                  0662
    IF (KRIM2(I))NE(KTEXT6),GO TO 62                                  0663
    GO TO 65                                                          0664
 67 IF (KTEXT5)LT(0),GO TO 70                                         0665
    IF (KTEXT1)LT(0),GO TO 69                                         0666
 68 IF (KTEXT5)GT(KTEXT1),GO TO 62                                    0667
 69 CALL INF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                             0668
    IF (KTEXT5)E(KEND),GO TO 75                                       0669
    GO TO 64                                                          0670
 70 IF (KTEXT1)LT(0),GO TO 68                                         0671
    GO TO 62                                                          0672
 71 IF (KTEXT6)LT(0),GO TO 74                                         0673
    IF (KTEXT2)LT(0),GO TO 69                                         0674
 72 IF (KTEXT6)GT(KTEXT2),GO TO 62                                    0675
 73 GO TO 69                                                          0676
 74 IF (KTEXT2)LT(0),GO TO 72                                         0677
    GO TO 62                                                          0678
 75 REWIND 0                                                          0679
    REWIND 4                                                          0680
    INENT = I                                                         0681
    IF (INENT)LTE(1),GO TO 90                                         0682
 76 J = 2                                                             0683
 77 I = J-1                                                           0684
    IF (KASS1(I))E(KASS1(J)),GO TO 82                                 0685
    IF (KASS1(I))LT(0),GO TO 80                                       0686
    IF (KASS1(J))LT(0),GO TO 81                                       0687
 78 IF (KASS1(J))GT(KASS1(I)),GO TO 81                                0688
 79 LEMP     = KASS1(I)                                               0689
    KASS1(I) = KASS1(J)                                               0690
    KASS1(J) = LEMP                                                   0691
    LEMP     = KASS2(I)                                               0692
    KASS2(I) = KASS2(J)                                               0693
    KASS2(J) = LEMP                                                   0694
    LEMP     = KRIM1(I)                                               0695
    KRIM1(I) = KRIM1(J)                                               0696
    KRIM1(J) = LEMP                                                   0697
```

```
      LEMP     = KRIM2(I)                                              0698
      KRIM2(I) = KRIM2(J)                                              0699
      KRIM2(J) = LEMP                                                  0700
      LEMP = KMAT1(I)                                                  0701
      KMAT1(I) = KMAT1(J)                                              0702
      KMAT1(J) = LEMP                                                  0703
      J = J-1                                                          0704
      IF (J)LT(2),GO TO 76                                             0705
      GO TO 77                                                         0706
   80 IF (KASS1(J))LT(0),GO TO 78
      GO TO 79                                                         0708
   81 J = J+1                                                          0709
      IF (J)GT(INENT),GO TO 90                                         0710
      GO TO 77                                                         0711
   82 IF (KASS2(I))E(KASS2(J)),GO TO 81                                0712
      IF (KASS2(I))LT(0),GO TO 84                                      0713
      IF (KASS2(J))LT(0),GO TO 81                                      0714
   83 IF (KASS2(J))GT(KASS2(I)),GO TO 81                               0715
      GO TO 79                                                         0716
   84 IF (KASS2(J))LT(0),GO TO 83                                      0717
      GO TO 79                                                         0718
   89 I = I+1
      IF (I)GT(INENT),GO TO 100                                        0730
      GO TO 93
   90 I = 0                                                            0719
      J = 0                                                            0720
      K = 128                                                          0721
   91 CALL INTEXT                                                      0722
      IF (KTEXT1)E(KEND),GO TO 100                                     0723
      IF (LWORD1)NE(KTEXT1),GO TO 92                                   0724
      IF (LWORD2)NE(KTEXT2),GO TO 92                                   0725
      GO TO 91                                                         0726
   92 LWORD1 = KTEXT1                                                  0727
      LWORD2 = KTEXT2                                                  0728
   93 IF (KASS1(I))NE(KTEXT1),GO TO 94                                 0731
      IF (KASS2(I))NE(KTEXT2),GO TO 97                                 0732
      J = J+1                                                          0733
      KRIM1(J) = KRIM1(I)                                              0734
      KRIM2(J) = KRIM2(I)                                              0735
      KASS1(J) = KASS1(I)                                              0736
      KASS2(J) = KASS2(I)                                              0737
      GO TO 91                                                         0738
   94 IF (KASS1(I))LT(0),GO TO 96                                      0739
      IF (KTEXT1) LT (0),GO TO 89
   95 IF (KASS1(I))GT(KTEXT1),GO TO 91                                 0741
      GO TO 89
   96 IF (KTEXT1) LT (0),GO TO 95                                      0743
      GO TO 91                                                         0744
   97 IF (KASS2(I))LT(0),GO TO 99                                      0745
      IF (KTEXT2) LT (0),GO TO 89
   98 IF (KASS2(I))GT(KTEXT2),GO TO 91                                 0747
      GO TO 89
   99 IF (KTEXT2) LT (0),GO TO 98                                      0749
      GO TO 91                                                         0750
  100 REWIND 4                                                         0751
      INENT = J                                                        0752
      REWIND 6
C        SORT IZZE ON MATCNT USING M,N (2 AND 3; AND 6                 0788
      M4 = 0                                                           0789
      MCNM = 1
```

```
      K = 128                                                          0792
      CALL INTEXT                                                      0793
  162 KTEXT8 = KTEXT4                                                  0794
      KTEXT7 = KTEXT3                                                  0795
      KTEXT6 = KTEXT2                                                  0796
      KTEXT5 = KTEXT1                                                  0797
  164 CALL INTEXT                                                      0798
      IF (KTEXT1)E(KEND),GO TO 180                                     0799
      IF (KTEXT4)GTE(KTEXT8),GO TO 170                                 0800
      IF (KTEXT4)GTE(M4),GO TO 169                                     0801
C         MERGE WITH M ONTO N                                          0802
      DO 165 I = NONM,128,4                                            0803
      CALL OUTM(KEND,KEND,KEND,KEND)                                   0804
  165 CONTINUE                                                         0805
      REWIND 2                                                         0806
      REWIND 3                                                         0807
      NONN = 1                                                         0808
      NONM = 128                                                       0809
  166 CALL INM(M1,M2,M3,M4)                                            0810
      IF (M1)E(KEND),GO TO 167                                         0811
      IF (KTEXT4)GTE(M4),GO TO 168                                     0812
      CALL OUTM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                           0813
      KTEXT4 = 32000                                                   0814
      GO TO 166                                                        0815
  167 DO 163 I = 1,NONN,1
      KTEXM(I) = KTEXN(I)
  163 CONTINUE                                                         0818
      LEMP = M                                                         0819
      M = N                                                            0820
      N = LEMP                                                         0821
      NONM = NONN
      REWIND N                                                         0822
      GO TO 164                                                        0823
  168 CALL OUTM(M1,M2,M3,M4)                                           0824
      GO TO 166                                                        0825
  169 CALL OUTM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                           0826
      M4 = KTEXT4                                                      0827
      GO TO 164                                                        0828
  170 CALL OUTG(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                           0829
      GO TO 162                                                        0830
  180 CALL OUTG(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                           0831
      DO 190 I = NONM,128,4                                            0832
      CALL OUTM(KEND,KEND,KEND,KEND)                                   0833
  190 CONTINUE                                                         0834
      DO 200 I = NONG,128,4                                            0835
      CALL OUTG(KEND,KEND,KEND,KEND)                                   0836
  200 CONTINUE                                                         0837
C         MERGE M AND G ONTO N                                         0838
      REWIND 2                                                         0839
      REWIND 3                                                         0840
      REWIND 4                                                         0841
      REWIND 6                                                         0842
      NONG = 128                                                       0843
      NONM = 128                                                       0844
      NONN = 1                                                         0845
      CALL ING(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                            0846
  220 CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                            0847
      IF (KTEXT1)E(KEND),GO TO 250                                     0848
  230 IF (KTEXT4)GTE(KTEXT8),GO TO 240                                 0849
      CALL OUTM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                           0850
```

```
      GC TC 220                                                          0851
  240 CALL OUTN(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                             0852
      CALL ING(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                              0853
      IF (KTEXT5)E(KEND),GC TO 260                                       0854
      GC TC 230                                                          0855
  250 CALL OUTN(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                             0856
      CALL ING(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                              0857
      IF (KTEXT5)NE(KEND),GC TC 250                                      0858
      GC TC 270                                                          0859
  260 CALL OUTN(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0860
      CALL INN(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                              0861
      IF (KTEXT1)NE(KEND),GC TO 260                                      0862
  270 DC 271 I = NONN,128,4                                              0863
      CALL OUTN(KEND,KEND,KEND,KEND)                                     0864
  271 CONTINUE                                                           0865
      REWIND 2                                                           0866
      REWIND 3                                                           0867
      REWIND 6                                                           0868
C        PUT FREQUENCY COUNT OF WORDS WITHIN MATCNT ON M                 0869
      NONM = 1                                                           0870
      NONN = 128                                                         0871
      CALL INN(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                              0872
      GC TC 273                                                          0873
  272 CALL INN(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                              0874
      IF (KTEXT4)E(KTEXT8),GO TO 274                                     0875
      CALL OUTM2(KTEXT8,KFREQ)                                           0876
  273 KTEXT8 = KTEXT4                                                    0877
      KFREQ = 0                                                          0878
      IF (KTEXT1)E(KEND),GC TO 275                                       0879
  274 STARTTAC                                                           0880
                TMA     1/1131$                                          0881
                AMS     KFREQ$                                           0882
                ENDTAC                                                   0883
      GC TC 272                                                          0884
  275 DC 276 I = NONM,128,2                                              0885
      CALL OUTM2(KEND,KEND)                                              0886
  276 CONTINUE                                                           0887
      REWIND 2                                                           0888
      REWIND 3                                                           0889
C        ADD FREQUENCY COUNT TC DATA AND PUT ON 6                        0890
      NONG = 1                                                           0891
      NONM = 128                                                         0892
      NONN = 128                                                         0893
      CALL INM2(KTEXT8,KFREQ)                                            0894
  277 CALL INN(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                              0895
      IF (KTEXT1)E(KEND),GC TO 280                                       0896
  278 IF (KTEXT8)E(KTEXT4),GO TO 279                                     0897
      CALL INM2(KTEXT8,KFREQ)                                            0898
      IF (KTEXT8)E(KTEXT4),GO TO 279                                     0899
      PAUSE                                                              0900
  279 KTEXT4 = KTEXT4+ KFREQ                                             0901
      CALL OUTG(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                             0902
      GC TC 277                                                          0903
  280 DC 290 I = NONG,128,4                                              0904
      CALL OUTG(KEND,KEND,KEND,KEND)                                     0905
  290 CONTINUE                                                           0906
      REWIND 2                                                           0907
      REWIND 3                                                           0908
      REWIND 6                                                           0909
C   ADD FREQUNCY COUNT TO THESAUR (KMAT2)
```

```
      I = 0
      KCNG = 128
  301 I = I+1                                                          0913
      IF (I)GT(INENT),GO TO 312
  303 CALL ING(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                            0915
      IF (KTEXT1)E(KEND),GO TO 310
      IF (KTEXT1)NE(LWORD1),GO TO 309
      IF (KTEXT2)NE(LWORD2),GO TO 309
      GO TO 303
  309 LWORD1 = KTEXT1
      LWORD2 = KTEXT2
      IF (KTEXT1)NE(KASS1(I)),GO TO 303
      IF (KTEXT2)NE(KASS2(I)),GO TO 303
      KMAT2(I) = KTEXT4
      GO TO 301                                                        0924
  310 REWIND 6
      KCNG = 128
      GO TO 303
  312 REWIND 6
      IF (INENT)LTE(1),GO TO 320
C SORT THEAURUS, (PRIMARY KEYD OF MAJOR ORDER)                         0754
  321 J = 2
  322 I = J-1
      IF (KRIM1(I))E(KRIM1(J)),GO TO 327
      IF (KRIM1(I))LT(0),GO TO 325
      IF (KRIM1(J))LT(0),GO TO 326
  323 IF (KRIM1(J))GT(KRIM1(I)),GO TO 326
  324 LEMP     = KASS1(J)                                              0762
      KASS1(J) = KASS1(I)                                              0763
      KASS1(I) = LEMP                                                  0764
      LEMP     = KASS2(J)                                              0765
      KASS2(J) = KASS2(I)                                              0766
      KASS2(I) = LEMP                                                  0767
      LEMP     = KRIM1(J)                                              0768
      KRIM1(J) = KRIM1(I)                                              0769
      KRIM1(I) = LEMP                                                  0770
      LEMP     = KRIM2(J)                                              0771
      KRIM2(J) = KRIM2(I)                                              0772
      KRIM2(I) = LEMP
      LEMP     = KMAT2(J)
      KMAT2(J) = KMAT2(I)
      KMAT2(I) = LEMP                                                  0773
      J = J-1
      IF (J)LT(2),GO TO 321
      GO TO 322
  325 IF (KRIM1(J))LT(0),GO TO 323
      GO TO 324
  326 J = J+1
      IF (J)GT(INENT),GO TO 320
      GO TO 322
  327 IF (KRIM2(I))LT(0),GO TO 329
      IF (KRIM2(J))LT(0),GO TO 326
  328 IF (KRIM2(J))GTE(KRIM2(I)),GO TO 326
      GO TO 324
  329 IF (KRIM2(J))LT(0),GO TO 328
      GO TO 324
C   ADD FREQUENCY COUNT TO THESAUR (KMAT1)                             0910
  320 I = 0                                                            0911
      KCNG = 128                                                       0912
  331 I = I+1
```

```
      IF (I)GT(INENT),GO TC 399
  333 CALL ING(KTEXT1,KTEXT2,KTEXT3,KTEXT4)
      IF (KTEXT1)E(KEND),GC TO 340
      IF (KTEXT1)NE(LWORD1),GO TO 339
      IF (KTEXT2)NE(LWORD2),GO TO 339
      GC TC 333
  339 LWORD1 = KTEXT1
      LWORD2 = KTEXT2                                              0920
      IF (KTEXT1)NE(KRIM1(I)),GO TO 333
      IF (KTEXT2)NE(KRIM2(I)),GO TO 333
      KMAT1(I) = KTEXT4                                            0923
      GC TC 331
  340 REWIND 6
      NCNG = 128
      GC TC 333
  399 REWIND 6
C  SORT THESAURUS ON KMAT1                                         0926
  400 J = 2                                                        0927
  401 I = J-1                                                      0928
      IF (KMAT1(J))GTE(KMAT1(I)),GO TO 406                         0929
  404 LEMP = KMAT1(I)                                              0930
      KMAT1(I) = KMAT1(J)                                          0931
      KMAT1(J) = LEMP                                              0932
      LEMP = KMAT2(I)                                              0933
      KMAT2(I) = KMAT2(J)                                          0934
      KMAT2(J) = LEMP                                              0935
      LEMP = KRIM1(I)                                              0936
      KRIM1(I) = KRIM1(J)                                          0937
      KRIM1(J) = LEMP                                              0938
      LEMP = KRIM2(I)                                              0939
      KRIM2(I) = KRIM2(J)                                          0940
      KRIM2(J) = LEMP                                              0941
      LEMP = KASS1(I)                                              0942
      KASS1(I) = KASS1(J)                                          0943
      KASS1(J) = LEMP                                              0944
      LEMP = KASS2(I)                                              0945
      KASS2(I) = KASS2(J)                                          0946
      KASS2(J) = LEMP                                              0947
      J = J-1                                                      0948
      IF (J)LT(2),GO TO 400                                        0949
      GC TC 401                                                    0950
  406 J = J+1                                                      0951
      IF (J)GT(INENT),GO TC 440                                    0952
      GC TC 401                                                    0953
C   OUTPUT TO PRINT TAPE AND SAVE NEW SYNONYMUS FORMS ON (N)       0955
  440 LWORD1 = 0
      LWORD2 = 0                                                   0957
      KTEXT8 = 0                                                   0958
      NWDS = 0                                                     0959
      M2 = 0                                                       0960
      I = 1                                                        0961
      M = 2
      N = 3
      NCNN = 1
      NF = 7                                                       0962
      NCNG = 128                                                   0963
      GC TC 452                                                    0964
  450 IF (NVAP)NE(KBLNK),GC TO 452                                 0965
      M1 = KTEXT3                                                  0966
      CALL DECOM                                                   0967
```

```
      IF (NP)LT(15),GO TO 451                                          0968
      PRINT 1,LINE                                                     0969
      CALL CLEAR                                                       0970
      NP = 7                                                           0971
  451 LINE(NP) = M2                                                    0972
      LINE(NP+1) = M3                                                  0973
      LINE(NP+2) = M4                                                  0974
      NP = NP+3                                                        0975
  452 CALL ING(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                            0976
      IF (KTEXT1)E(KEND),GO TO 540                                     0977
      IF (KTEXT4)NE(KTEXT8),GO TO 455                                  0978
      IF (LWORD1)NE(KTEXT1),GO TO 460                                  0979
      IF (LWORD2)NE(KTEXT2),GO TO 460                                  0980
      GO TO 450                                                        0981
  455 IF (NP)E(7),GO TO 456                                            0982
      PRINT 1,LINE                                                     0983
      CALL CLEAR                                                       0984
      NP = 7                                                           0985
  456 KTEXT8 = KTEXT4                                                  0986
      CALL CON1(KTEXT4)                                                0987
      LINE(1) = KTEXT4                                                 0988
      KTEXT4 = KTEXT8                                                  0989
      CALL FREQ(KTEXT4,KFREQ)                                          0990
      NWDS = NWDS+KFREQ                                                0991
      LINE(4) = KTEXT4                                                 0992
  460 IF (NP)E(7),GO TO 461                                            0993
      PRINT 1,LINE                                                     0994
      CALL CLEAR                                                       0995
      NP = 7                                                           0996
  461 LINE(2) = KTEXT1                                                 0997
      LINE(3) = KTEXT2                                                 0998
      LWORD1  = KTEXT1                                                 0999
      LWORD2  = KTEXT2                                                 1000
      PRINT 1,LINE                                                     1001
      CALL CLEAR                                                       1002
  470 IF (I)GT(INENT),GO TO 450                                        1003
      IF (KMAT1(I))GT(KTEXT8),GO TO 450                                1004
      IF (KMAT1(I))LT(KTEXT8),GO TO 473                                1005
      IF (KRIM1(I))NE(KTEXT1),GO TO 450                                1006
      IF (KRIM2(I))NE(KTEXT2),GO TO 450                                1007
      GO TO 480                                                        1008
  473 I = I+1                                                          1009
      GO TO 470                                                        1010
  480 LINE(5) = KASS1(I)                                               1011
      LINE(6) = KASS2(I)                                               1012
      M1      = KMAT2(I)                                               1013
      CALL FREQ(M1,KFREQ)                                              1014
      LINE(14) = M1                                                    1015
      PRINT 1,LINE                                                     1016
      CALL CLEAR                                                       1017
      DO 500 J = 1,INENT,1                                             1018
      IF (KASS1(I))NE(KRIM1(J)),GO TO 500                              1019
      IF (KASS2(I))NE(KRIM2(J)),GO TO 500                              1020
      LINE(6) = KASS1(J)                                               1021
      LINE(7) = KASS2(J)                                               1022
      PRINT 1,LINE                                                     1023
      CALL CLEAR                                                       1024
      IF (KTEXT1)NE(KASS1(J)),GO TO 485
      IF (KTEXT2)E(KASS2(J)),GO TO 486
  485 M1 = KTEXT1                                                      1025
```

```
      M4 = KTEXT2                                                        1026
      M3 = KASS1(J)                                                      1027
      M4 = KASS2(J)                                                      1028
      CALL CUTA(M1,M2,M3,M4)                                             1029
  486 DO 499 K = 1,INENT,1                                               1030
      IF (KASS1(J))NE(KRIM1(K)),GO TO 499                                1031
      IF (KASS2(J))NE(KRIM2(K)),GO TO 499                                1032
      IF (KRIM1(K))NE(KRIM1(I)),GO TO 491                                1033
      IF (KRIM2(K))E(KRIM2(I)),GO TO 495                                 1034
  491 LINE(7) = KASS1(K)                                                 1035
      LINE(8) = KASS2(K)                                                 1036
      PRINT 1,LINE                                                       1037
      CALL CLEAR                                                         1038
      DO 498 L = 1,INENT,1                                               1044
      IF (KASS1(K))NE(KRIM1(L)),GO TO 498                                1045
      IF (KASS2(K))NE(KRIM2(L)),GO TO 498                                1046
      IF (KRIM1(L))NE(KRIM1(J)),GO TO 492                                1047
      IF (KRIM2(L))E(KRIM2(L)),GO TO 498                                 1048
  492 LINE(8) = KASS1(L)                                                 1049
      LINE(9) = KASS2(L)                                                 1050
      PRINT 1,LINE                                                       1051
      CALL CLEAR                                                         1052
      DO 497 M = 1,INENT,1                                               1058
      IF (KASS1(L))NE(KRIM1(M)),GO TO 497                                1059
      IF (KASS2(L))NE(KRIM2(M)),GO TO 497                                1060
      IF (KRIM1(M))NE(KRIM1(K)),GO TO 493                                1061
      IF (KRIM2(M))E(KRIM2(K)),GO TO 497                                 1062
  493 LINE(9) = KASS1(M)                                                 1063
      LINE(10)= KASS2(M)                                                 1064
      PRINT 1,LINE                                                       1065
      CALL CLEAR                                                         1066
      DO 496 N = 1,INENT,1                                               1072
      IF (KASS1(M))NE(KRIM1(N)),GO TO 496                                1073
      IF (KASS2(M))NE(KRIM2(N)),GO TO 496                                1074
      IF (KRIM1(N))NE(KRIM1(I)),GO TO 494                                1075
      IF (KRIM2(N))E(KRIM2(L)),GO TO 496                                 1076
  494 LINE(10)= KASS1(N)                                                 1077
      LINE(11)= KASS2(N)                                                 1078
      PRINT 1,LINE                                                       1079
      CALL CLEAR                                                         1080
  496 CONTINUE                                                           1086
      N = 3
  497 CONTINUE                                                           1087
      M = 2
  498 CONTINUE                                                           1088
  499 CONTINUE                                                           1089
  500 CONTINUE                                                           1090
      GO TO 473                                                          1091
  540 IF (NP)E(7),GO TO 541                                              1092
      PRINT 1,LINE                                                       1093
      CALL CLEAR                                                         1094.
  541 CALL CON1(NWDS)                                                    1095
      LINE(4) = NWDS                                                     1096
      PRINT 3,LINE                                                       1097
      DO 542 I = NONN,128,4                                              1098
      CALL OUTA(KEND,KEND,KEND,KEND)                                     1099
  542 CONTINUE                                                           1100
C        (F) ONTO (M) THEN MERGE (M) AND (N) BACK ONTO (F)               1101
      NCNM = 1
      NCNF = 128
```

```
      CALL INF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                          1102
  551 CALL INF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                          1103
      IF (KTEXT1)E(KEND),GC TO 553
      CALL OUTM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                         1105
      GC TC 551                                                      1106
  553 DC 554 I = NONM,128,4
      CALL OUTM(KEND,KEND,KEND,KEND)
  554 CONTINUE
      REWIND 0
      REWIND 2
      REWIND 3
      NONF = 1                                                       1109
      CALL OUTF(KTHES,KTHES,KTHES,KTHES)                             1110
      NONM = 128                                                     1112
      NONN = 128                                                     1113
      CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                          1114
  602 CALL INN(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                          1115
      IF (KTEXT5)E(KEND),GC TO 640                                   1116
  603 IF (KTEXT1)E(KTEXT5),GO TO 610                                 1117
      IF (KTEXT1)LT(0),GO TC 606                                     1118
      IF (KTEXT5)LT(0),GO TC 605                                     1119
  604 IF (KTEXT1)GT(KTEXT5),GO TO 607                                1120
  605 CALL OUTF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                         1121
      CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                          1122
      IF (KTEXT1)E(KEND),GC TO 650                                   1123
      GC TC 603                                                      1124
  606 IF (KTEXT5)LT(0),GO TC 604                                     1125
  607 CALL OUTF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                         1126
      GC TC 602                                                      1127
  610 IF (KTEXT2)E(KTEXT6),GO TO 620                                 1128
      IF (KTEXT2)LT(0),GO TC 612                                     1129
      IF (KTEXT6)LT(0),GO TC 605                                     1130
  611 IF (KTEXT2)GT(KTEXT6),GO TO 607                                1131
      GC TC 605                                                      1132
  612 IF (KTEXT6)LT(0),GO TC 611                                     1133
      GC TC 607                                                      1134
  620 IF (KTEXT3)E(KTEXT7),GO TO 630                                 1135
      IF (KTEXT3)LT(0),GO TC 622                                     1136
      IF (KTEXT7)LT(0),GO TC 605                                     1137
  621 IF (KTEXT3)GT(KTEXT7),GO TO 607                                1138
      GC TC 605                                                      1139
  622 IF (KTEXT7)LT(0),GO TC 621                                     1140
       GO TO 607                                                     1141
  630 IF (KTEXT4)E(KTEXT8),GO TO 602                                 1142
      IF (KTEXT4)LT(0),GO TC 632                                     1143
      IF (KTEXT8)LT(0),GO TC 605                                     1144
  631 IF (KTEXT4)GT(KTEXT8),GO TO 607                                1145
      GC TC 605                                                      1146
  632 IF (KTEXT8)LT(0),GO TC 631                                     1147
      GC TC 607                                                      1148
  640 CALL OUTF(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                         1149
      CALL INM(KTEXT1,KTEXT2,KTEXT3,KTEXT4)                          1150
      IF (KTEXT1)E(KEND),GC TO 660                                   1151
      GC TC 640                                                      1152
  650 CALL OUTF(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                         1153
      CALL INN(KTEXT5,KTEXT6,KTEXT7,KTEXT8)                          1154
      IF (KTEXT5)E(KEND),GC TO 660                                   1155
      GC TC 650                                                      1156
  660 DC 661 I = NONF,128,4                                          1157
      CALL OUTF(KEND,KEND,KEND,KEND)                                 1158
  661 CONTINUE                                                       1159
      REWIND 0                                                       1160
      STOP                                                           1161
      COMPLETE                                                       1162
      END$                                                           1163
        THEEND
```

G. Output

Examples of the output of this section of the program are given under "Conclusions" (Section One, III), Figures A and B.

System Development Corporation / 2500 Colorado Avenue / Santa Monica, California