

# SERIES 200

## TAPE SORT C AND COLLATE C

GENERAL SYSTEM:

SERIES 200/OPERATING SYSTEM - MOD 1

SUBJECT:

Programming and Operating Procedures for  
the Programs Tape Sort C and Collate C.

SPECIAL  
INSTRUCTIONS:

This software manual completely supersedes  
the bulletin Tape Sort C and Collate C, Order  
No. 018, dated December 8, 1965, and Addendum  
#1 thereto, dated November 24, 1965.

DATE: May 25, 1966

FILE NO.: 123.6005.021C.2-018\*

8891

8566

Printed in U. S. A.

\*When ordering this publication please specify  
Title and Underscored portion of File Number.

## FOREWORD

Tape Sort C and Collate C are programs which function in the SERIES 200/OPERATING SYSTEM - MOD 1. Both programs are described in this manual because of the likelihood that they will be run consecutively. For instance, when two or more sorted output files have been produced, it may be desirable to collate them into a single ordered file.

Tape Sort C and Collate C can be stored on a binary run tape (BRT) and run under control of the Tape Loader-Monitor C or Floating Tape Loader-Monitor C program, or the specific program could be punched on cards for a binary run deck (BRD) and loaded via the Card Loader-Monitor B program or the card loading option of the Floating Tape Loader-Monitor C program. In any case, the sort or collate program is loaded and used independently. For more extensive information on the Loader-Monitor programs, refer to the manuals entitled PLUS - Tape Loader-Monitor (DSI-327), Floating Tape Loader-Monitor C and Interrupt Control D, Order Number 005, or Card Loader-Monitor B, Order Number 154, as applicable.

The information presented in this manual completely supersedes that given in the Tape Sort C and Collate C Software Bulletin, Order Number 018 and Addendum #1 thereto, and reflects the latest revisions of Tape Sort C and Collate C. The major additional, yet optional, capabilities of these revisions provide for incorporation of a console typewriter and for program execution in the four-character address mode. Several other features have been included and are described in their appropriate context.

This manual has been arranged in sections for direct reference according to the reader's familiarity with Honeywell sorting and collating programs. An explanation of the Tape Sort C is contained in Sections I through V: (I) general sorting procedures and sort program characteristics, (II) parameter information for specialization of the sort, (III) descriptions of optional own-coding routines, (IV) considerations and methods for running the sort, and (V) detailed operating procedures. Collate C program information is contained in Sections VI through IX: (VI) collate program characteristics and general collating procedures, (VII) parameter information for specializing Collate C, (VIII) considerations and methods for running the collate, and (IX) detailed operating procedures. Information concerning program timing and optional user-coded routines is contained in Appendices A and B, respectively.

Copyright 1966  
Honeywell Inc.  
Electronic Data Processing Division  
Wellesley Hills, Massachusetts 02181

The experienced programmer and operator need only refer to Sections IV and V to run Tape Sort C or to Sections VIII and IX to run Collate C.

The reader should be familiar with the information presented in the Honeywell Series 200 Programmers' Reference Manual, Models 200/1200/2200, Order Number 139, and the general operating characteristics of the Series 200 equipment to be used for any specific application, as explained in the Honeywell Series 200 Equipment Operators' Manual (Model 200), Order Number 040.

## TABLE OF CONTENTS

		Page
Section I	Basic Concepts of Tape Sort C.....	1-1
	Introduction.....	1-1
	Characteristics of Tape Sort C.....	1-1
	Check and Checkpoint Terminology for Tape Sort C.....	1-2
	Machine Requirements.....	1-2
	Summary Description of Tape Sort C .....	1-3
	The Presort.....	1-3
	The Merge.....	1-4
	The Last Pass .....	1-5
	Data Requirements .....	1-5
	Sort Keys and Key Fields.....	1-7
	Magnetic Tape Units.....	1-7
	Unreadable Input Records .....	1-7
	Program History.....	1-7
	Restart Capabilities .....	1-8
	Section II	Tape Sort C Program Specialization .....
Parameter Designation.....		2-1
Tape Unit Specifications.....		2-1
Item and Record Sizes .....		2-3
Record Size Determination.....		2-3
Padding.....		2-4
Tape Labels.....		2-4
Banner Characters.....		2-5
Parity and Record Gap .....		2-6
Error Options.....		2-6
Printer Control.....		2-6
Last-Pass Check Option.....		2-7
Highest Memory Address.....		2-8
Collating Sequence and Character Code.....		2-8
Ascending or Descending Sequenced Output .....		2-8
Own-Coding.....		2-8
Presort Own-Coding.....		2-9
Last-Pass Own-Coding .....		2-9
Parameter Card Identification.....		2-9
Key Fields.....		2-9
Input File Name Verification.....		2-10
Output File Name Change.....		2-10
Last-Pass Own-Coding Program Name.....		2-10
Last-Pass Own-Coding Segment Name .....		2-10
Search Direction for Last-Pass Own-Coding.....		2-10
Next Program Name .....		2-10
Next Program Segment Name .....		2-11
Search Direction for Next Program.....		2-11
Tape Positioning .....		2-11
Beginning of Sort .....		2-11
End of Sort.....	2-11	
Effects of Characters 15, 16, and 19.....	2-13	

TABLE OF CONTENTS (cont)

	Page
Section III	
Tape Sort C Own-Coding.....	3-1
Presort Own-Coding.....	3-1
Presort Header-Trailer Own-Coding .....	3-1
Multi-Reel Input.....	3-2
Read Backwards Input.....	3-2
Presort Item-by-Item Own-Coding .....	3-2
Presort Process Determination .....	3-4
Inspecting or Modifying an Item .....	3-4
Deleting an Item.....	3-4
Adding an Item .....	3-4
Terminating Presort Own-Coding .....	3-5
Own-Coding Input.....	3-6
Last-Pass Own-Coding .....	3-6
Last-Pass Header-Trailer Own-Coding.....	3-6
Last-Pass Item-by-Item Own-Coding.....	3-7
Last-Pass Process Determination.....	3-7
Inspecting or Modifying an Item .....	3-7
Deleting an Item.....	3-7
Adding an Item .....	3-8
Terminating Last-Pass Own-Coding.....	3-8
Multi-Reel Output .....	3-9
Hash Totals.....	3-9
Considerations for Programming Own-Coding.....	3-9
Section IV	
Tape Sort C Operating Considerations.....	4-1
Introduction .....	4-1
Tape Sort C Program Format .....	4-1
Calling and Loading Tape Sort C with a Tape Loader-	
Monitor.....	4-2
Programmed Call for Tape Sort C on a BRT.....	4-2
Card Call for Tape Sort C on a BRT with Parameters	
on Cards.....	4-3
Calling and Loading Tape Sort C from Cards .....	4-3
Programmed Call for Tape Sort C on Cards.....	4-3
Card Call for Tape Sort C Program and Parameters on	
Cards.....	4-4
Calling and Loading Tape Sort C Own-Coding.....	4-5
Entering the Input to be Sorted.....	4-5
Tape Sort C Parameter Summary.....	4-6
Section V	
Tape Sort C Operating Procedures.....	5-1
Preliminary Considerations.....	5-1
Loading Tape Sort C from a BRT .....	5-1
Loading Tape Sort C from a BRT with Parameters on	
Cards.....	5-1
Loading Tape Sort C from a BRT by Programmed Call ...	5-1
Loading Tape Sort C from an Object Program Deck.....	5-2
Loading Tape Sort C Parameters and Program from	
Cards.....	5-2

TABLE OF CONTENTS (cont)

	Page
Section V (cont)	
Loading Tape Sort C from an Object Program Deck by Programmed Call.....	5-4
Check and Checkpoint Procedures .....	5-4
Check Message Coding.....	5-6
Restart Procedures.....	5-6
Restart Precautions.....	5-9
Tape Positioning for Restarting.....	5-9
Positioning the Input Tape.....	5-10
Positioning of Work Tape 1.....	5-9
Search Direction for Restart Segment .....	5-10
Restart Message.....	5-10
Restart Failure.....	5-10
Section VI	
Collate C Basic Concepts.....	6-1
Introduction.....	6-1
Characteristics of Collate C .....	6-1
Check and Checkpoint Terminology for Collate C .....	6-2
Machine Requirements .....	6-2
Data Requirements.....	6-3
Collate Keys and Key Fields .....	6-3
Magnetic Tape Units .....	6-4
Unreadable Input Records.....	6-4
Optional Printouts .....	6-4
Collate Process .....	6-4
Section VII	
Collate C Program Specialization .....	7-1
Parameter Preparation.....	7-1
Tape Unit Specifications .....	7-1
Item and Record Sizes.....	7-2
Padding.....	7-2
Tape Labels.....	7-3
Output Label Verification .....	7-3
Banner Characters.....	7-4
Parity and Record Gap.....	7-4
Error Options .....	7-4
Printer Control.....	7-4
Highest Memory Address .....	7-5
Collating Sequence Translation .....	7-5
Ascending or Descending Sequenced Output.....	7-5
Rewind Option .....	7-6
Own-Coding .....	7-6
Number of Data Records Per Output Reel.....	7-6
Parameter Card Identification.....	7-6
Key Fields .....	7-6
Input File Name and Reel Number Verification .....	7-7
Output File Name Change .....	7-7
Next Program Name.....	7-7
Next Segment Name .....	7-7
Search Direction for Next Program .....	7-8

TABLE OF CONTENTS (cont)

	Page
Section VII (cont)	
Tape Positioning.....	7-8
Beginning of Collate.....	7-8
End of Collate .....	7-8
Collate C Own-Coding.....	7-8
How to Use Collate C Own-Coding .....	7-9
Section VIII	
Collate C Operating Considerations.....	8-1
Introduction.....	8-1
Collate C Program Format .....	8-1
Calling and Loading Collate C with a Tape Loader-Monitor ...	8-1
Programmed Call for Collate C on a BRT.....	8-2
Card Call for Collate C on a BRT with Parameters on	
Cards .....	8-2
Calling and Loading Collate C from Cards.....	8-2
Programmed Call for Collate C Program on Cards .....	8-3
Card Call for Collate C Program and Parameters on	
Cards .....	8-4
Calling and Loading Own-Coding.....	8-4
Collate C Parameter Summary.....	8-4
Section IX	
Collate C Operating Procedures.....	9-1
Preliminary Considerations.....	9-1
Loading Collate C from a BRT .....	9-1
Loading Collate C from a BRT with Parameters on	
Cards .....	9-1
Loading Collate C from a BRT by Programmed Call .....	9-1
Loading Collate C from an Object Program Deck.....	9-2
Loading Collate C Program and Parameters from Cards ..	9-2
Loading Collate C from an Object Program Deck by	
Programmed Call.....	9-2
Check and Checkpoint Procedures .....	9-4
Check Message Coding .....	9-4
Appendix A	
Timing Formulas for Tape Sort C and Collate C .....	A-1
Tape Sort C Timing.....	A-1
Collate C Timing .....	A-4
Appendix B	
Programming Aids for Own-Coding.....	B-1
Sample Own-Coding Routines.....	B-1

## LIST OF ILLUSTRATIONS

		Page
Figure 1-1.	Summary of Presort, Merge, and Last Pass.....	1-6
Figure 2-1.	Tape Positioning in the Tape Sort C Program.....	2-12
Figure 2-2.	Tape Positioning for Effects of Characters 15, 16, and 19.....	2-14
Figure 3-1.	Summary Flow Chart of Presort "One-Shot" Branch.....	3-3
Figure 3-2.	Word Mark Locations in a Sort Item .....	3-5
Figure 4-1.	Parameter Card Input for Loading Tape Sort C from Tape.....	4-4
Figure 5-1.	Card Input to Tape Sort C.....	5-2
Figure 5-2.	Suggested Object Program Deck Format for Loading Tape Sort C with Parameter Cards .....	5-3
Figure 5-3.	Suggested Object Program Deck Format for Loading Tape Sort C without Parameter Cards.....	5-5
Figure 6-1.	Sample Collate Process .....	6-2
Figure 6-2.	Collate Processing of Two Input Files.....	6-5
Figure 7-1.	Collate C Tape Positioning.....	7-9
Figure 8-1.	Parameter Card Input for Loading Collate C Parameters.....	8-3
Figure 9-1.	Suggested Object Program Deck Format for Loading Collate C with Parameter Cards .....	9-3
Figure 9-2.	Suggested Object Program Deck Format for Loading Collate C without Parameter Cards.....	9-3

## LIST OF TABLES

Table 2-1.	Tape Sort C Label Options for Tape Input.....	2-5
Table 2-2.	Sort Program History.....	2-7
Table 2-3.	Parameter Characters for Key Fields 1 through 10.....	2-10
Table 4-1.	Tape Sort C Program Format.....	4-1
Table 4-2.	Tape Sort C Label Options for Card Input.....	4-6
Table 4-3.	Tape Sort C Parameter Specification.....	4-6
Table 5-1.	Tape Sort C Checkpoint Identification and Corrective Action.....	5-7
Table 5-2.	Restart Entries.....	5-9
Table 7-1.	Collate C Label Options .....	7-3
Table 7-2.	Parameter Characters for Collate Key Fields 1 through 10.....	7-7
Table 8-1.	Collate C Program Format .....	8-1
Table 8-2.	Collate C Parameter Specification.....	8-4
Table 9-1.	Collate C Checkpoints.....	9-5
Table A-1.	Timing Formulas for Tape Sort C .....	A-1
Table A-2.	Timing Formulas for Collate C.....	A-4

## SECTION I

### BASIC CONCEPTS OF TAPE SORT C

#### INTRODUCTION

Sorting is an integral function of most data processing applications, in that sorting processes rearrange randomly ordered input data to produce a sequentially ordered output data file. For many applications, it is necessary to sort files consisting of fixed-length units of information; consequently, Honeywell has further developed Tape Sort C, its general-purpose Series 200 program for performing fixed-length sorting functions using 1/2-inch magnetic tape.

As a general-purpose program, Tape Sort C can be adapted to different data formats and equipment configurations, enabling it to handle a wide variety of sorting applications. The information used to specialize Tape Sort C for a particular application is entered at the beginning of sort processing by means of parameters specified by the user.

Input to Tape Sort C is processed through the program's many operational segments which may be grouped into three logical segments: presort, merge, and last pass. The presort segment accepts the input data in the form of fixed-length units of information called "items" and distributes ordered groups of items called "strings" on from two to five tape reels. The merge segment combines the presorted strings into fewer and longer strings during a series of merge phases, resulting in only one long string on each work tape except one. The last-pass segment further combines these single strings into a contiguous sequence — the sorted file.

The activities performed during the presort and last-pass segments can be augmented by routines written by the user. These routines, which can be easily written using EasyCoder C symbolic language, are referred to as "own-coding." Own-coding can be used to provide all the input to be processed, or to supplement sort processing by modifying header and trailer labels and inspecting, modifying, adding or deleting items.

#### CHARACTERISTICS OF TAPE SORT C

Tape Sort C is characterized by the following activities:

1. Sorts fixed-length items which are blocked to a specified number per record;
2. Allows a maximum record size dependent upon the main memory capacity and the number of tapes used in the merge;
3. Performs read-backward, polyphase merging using from three to six tapes;
4. Allows execution in three- or four-character address mode;

5. Sorts according to control information contained in up to ten sort key fields in each item;
6. Sorts at least one full reel of records (input or output data can be contained on any number of reels, but successful completion of the sort is not guaranteed if the input data exceeds one full reel);
7. Operates with either control panel or console typewriter (Type 220-2 or -3);
8. Labels output tapes as specified by the user;
9. Provides remedial activities for handling unreadable records;
10. Provides for the inclusion of own-coding elements;
11. Provides the final output tape on a specified drive (work tape 1);
12. Provides a program history on the printer and/or typewriter, as designated; and
13. Offers the capability of immediate or delayed restarting.

#### Check and Checkpoint Terminology for Tape Sort C

Throughout Sections I to V of this manual, the following terms should be interpreted as defined here. A "checkpoint" is a routine programmed to check or delay processing when Tape Sort C requires an operator decision. A "check" is a term used to denote the occurrence of a checkpoint and is relayed to the operator through either the control panel (via a machine halt) or the console typewriter (via a typed message). All checkpoints associated with Tape Sort C are listed in Table 5-1 (on page 5-6) and numbered to correspond to the appropriate operator action.

#### MACHINE REQUIREMENTS

Tape Sort C requires the following equipment configuration:

1. A minimum main memory capacity of 8,192 character locations is required if the program is assembled and run in three-character address mode, but up to 32K locations can be used to advantage; similarly, with four-character address mode, a minimum of 12,288 locations are required, but up to 65K locations can be used to advantage.
2. A minimum of three tape drives (Type 204B) using 1/2-inch magnetic tape. Up to five additional drives can be used for any or all of the following purposes:
  - a. To provide an alternate reel for multi-reel input;
  - b. To increase the sort power to three-, four-, or five-way merging; or
  - c. To eliminate tape changing for the merge segment.
3. An additional tape drive, or a card reader, for program loading.
4. The Advanced Programming Instructions feature (011).
5. Read-write channels two and three (RWC2 and RWC3); used for reading and writing, respectively.
6. Index registers one through four.

NOTE: A printer (Type 222 or 206) may be used for printouts of unreadable records, parameters, and a history of the program run.

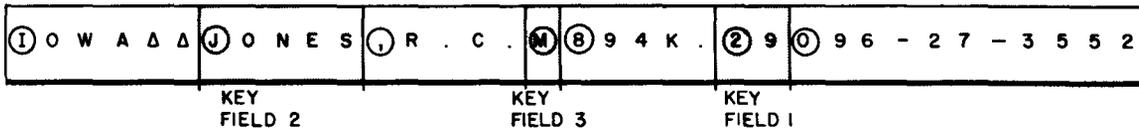
### SUMMARY DESCRIPTION OF TAPE SORT C

As explained previously, Tape Sort C may be logically considered as a tri-segmented program. The following paragraphs briefly describe the activities performed during each of the three segments.<sup>1</sup>

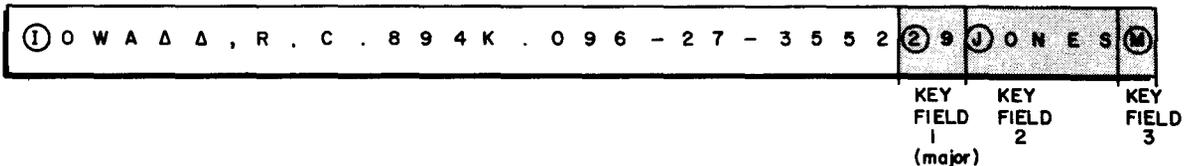
#### The Presort

The sorting technique applied in the presort is a replacement selection process. This process is advantageous because it maximizes the length of each string, therefore, minimizing both the total number of strings and, consequently, the time required to complete the merge segment of the program. These features are especially evident if any preordering exists in the input file (i. e., if some groups of input items appear in ascending or descending sequence).

Input items are rearranged by the presort into a format determined by the number and locations of the key fields in the items. The key fields are placed in the last (rightmost) locations of each item according to their relative importance (the first, or major, key field precedes the second, etc.). The remaining non-key field information is positioned to the left.



In the example above, the input item consists of a person's resident state, the person's surname (key field 2), specific code data, sex (key field 3), code data, age (key field 1), and social security number. The items are to be sorted by surname and sex within the age group. The presort segment rearranges the item into the following format.



<sup>1</sup> For more detailed information concerning the sorting techniques applied in Tape Sort C, refer to Honeywell Technical Bulletin No. 118 entitled Sorting Capabilities Report, File Number 127.6105.0200.00.00.

The following outline presents a simplified description of the activities performed during the presort.

1. A record consisting of one or more items is read from an input file into an input buffer area in memory.
2. The key fields of each item in the input buffer are rearranged (see figures above) as the item is moved to an item storage area.
3. The addresses of the items in the storage area are then stored in pairs in another area in memory.
4. The key fields of the items (referenced by the paired addresses) are compared. (The group of key fields in an item that constitutes a unit of comparison with all other items to be sorted may be termed the sort key for the item.)
5. From each pair of addresses, that address which corresponds to the item having the lower sort key is selected and paired with another address which was similarly selected.
6. Pairing of selected addresses continues until the address of the item having the lowest sort key is determined. The item associated with this sort key becomes the initial item of a string.
7. The item with the lowest sort key is moved to an output area, and the item storage area in which it was initially stored receives a new item from the input buffer.
8. The sort key of the new item is first compared to the sort key of the item it replaced. If the new key is equal or higher, the new item is considered as an eventual entry in the string currently being formed. If the new sort key is lower, the new item remains in storage, but its key will not be considered until the creation of the next string.
9. Formation of the current string is terminated when all sort keys in storage are of lower value than the sort key of the last item moved to the output area.
10. A new string is then created by repeating steps 2 through 9 above. Double-buffering is used by Tape Sort C, so that, as soon as an input buffer is depleted, the contents of a second buffer are moved into item storage as needed, while the depleted buffer receives a new record from the input file.

NOTE: The above description, steps 5 through 9, is concerned with the creation of an ascending string, but the presort creates ascending and descending strings alternately on each work tape. Therefore, as a descending string is being created, the presort will compare the items in storage to find the highest rather than the lowest sort key.

### The Merge

The merge applies a technique developed by Honeywell — read-backward, polyphase merging. This method permits merging with an odd or even number of tapes, while eliminating all tape rewind time. A polyphase merge minimizes the number of passes over the data during the merge because, for any configuration of work tapes, only one tape is used as the merge output tape; therefore, if six work tapes are used, each merge phase is a five-way merge (five tapes for input and one for output). The read-backward feature in conjunction with the presort

creation of alternating ascending and descending strings eliminates all rewind time for the merge segment; for example, once the presorted data have been written forward onto the specified work tapes, the work tapes are immediately read backward for the first merge phase.

The read-backward, polyphase merge may be summarized as follows:

1. The presort has distributed alternating ascending and descending strings in a prescribed ratio on all but one of the work tapes (see A and B in Figure 1-1).

NOTE: The creation of ascending and descending strings in a unique ratio insures that strings of like sequence will be combined when reading backwards for the merge.

2. The merge now combines strings taken from these input work tapes (A and B) to produce longer strings on the work tape left available, i. e., the output tape (C).
3. Because the presort deliberately writes a different number of strings on each work tape, one tape (A) will be read backwards to its beginning before the others. When this happens, the depleted tape becomes the merge work tape to be written forward in the next merge phase. Note that a "phase" is completed each time a work tape is depleted.
4. The output tape of the previous phase (C) will now act as an input work tape to be read backward immediately for the next merge phase. The data on this new input work tape will be sequentially combined with the data remaining on the input work tape (B) which was not depleted in the previous merge phase.
5. The merge continues through as many phases as necessary to produce a single string on all but one of the work tapes. At this point in the sort processing, another phase is required which is termed the "last pass," as it is the only phase that passes over all the data.

#### The Last Pass

The last pass performs the last merge phase of the data. It combines the single string remaining on each work tape into one continuous output file.

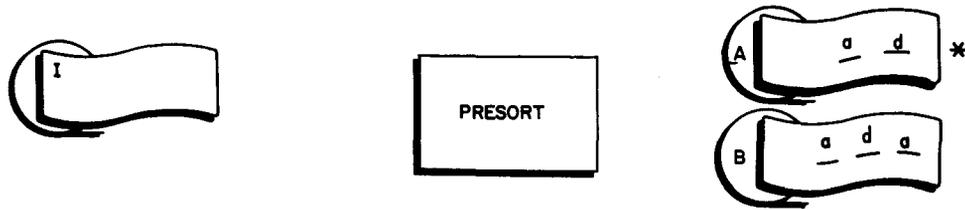
A summary of the presort, merge, and last-pass segments is shown in Figure 1-1. In this illustration, the sort program uses a minimum of three tape drives ("I" is the primary input tape), and the application requires only three merging phases to produce a sorted output file in ascending sequence. After the presort segment, the primary input tape is either rewound and replaced with another reel to be used as the merge work tape (C) or written forward, if there is more tape available on the input reel, with merge output data being written after the end-of-file record of the primary input data.

#### DATA REQUIREMENTS

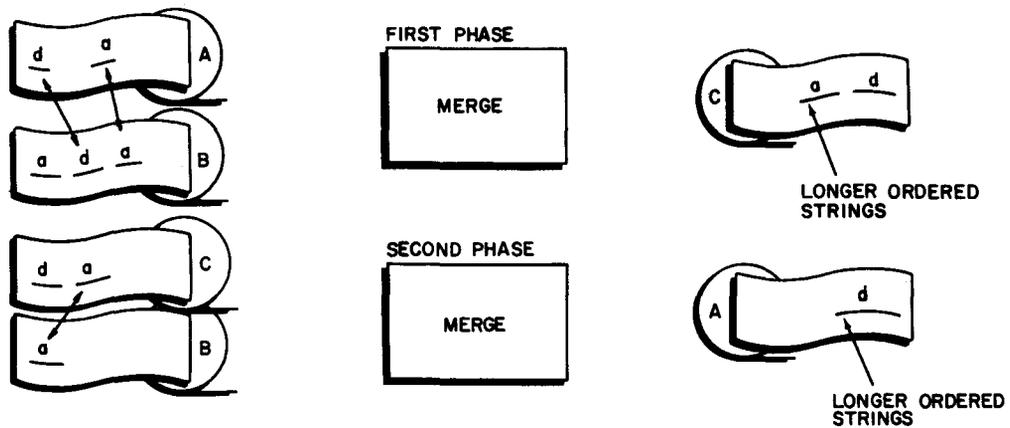
The sort program processes fixed-length items; and Tape Sort C input and output files may be blocked, that is, several items are combined to form one tape record. The formulae

SECTION I. BASIC CONCEPTS OF TAPE SORT C

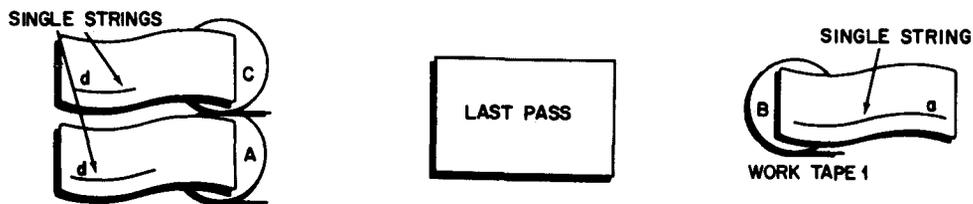
PRESORT ARRANGES THE RANDOM INPUT DATA FROM TAPE OR CARDS INTO ALTERNATING ASCENDING AND DESCENDING GROUPS OF ORDERED DATA (STRINGS) ON ALL BUT ONE OF THE WORK TAPES.



MERGE READS BACKWARDS AND COMBINES STRINGS FROM BOTH WORK TAPES A AND B (OUTPUT OF PRESORT), PRODUCING LONGER ORDERED STRINGS ON THE EMPTY WORK TAPE (C) UNTIL ONE WORK TAPE (A) IS DEPLETED (A PHASE). TAPE C THEN BECOMES AN INPUT TAPE, AND THE DEPLETED TAPE (A) IS MADE THE OUTPUT TAPE, ETC., UNTIL SINGLE STRINGS ARE PRODUCED ON ALL BUT ONE OF THE WORK TAPES.



LAST PASS COMBINES THE SINGLE STRINGS OF EACH WORK TAPE (OUTPUT OF MERGE) INTO A SINGLE STRING ON ONE OUTPUT TAPE (WORK TAPE 1). THIS FINAL OUTPUT TAPE IS THE SORTED FILE.



\* IN THE FIGURE, "a" DENOTES AN ASCENDING STRING AND "d" DENOTES A DESCENDING STRING

Figure 1-1. Summary of Presort, Merge, and Last Pass

given in Section II can be used to determine the maximum and optimum record sizes for both input and output. For a given sorting application, the user establishes the record size according to these formulae. It should be noted that the sort will operate most efficiently when the record does not exceed the optimum size for the application. At least one input item is required for sorting; the sort program will not handle a "no-input" situation.

### SORT KEYS AND KEY FIELDS

Data items are identified and sequentially ordered by means of sort keys. A sort key consists of those fields within an item which are designated as the basis for developing a sequential relationship among all the items to be sorted. Each sort key may contain up to ten key fields with up to 99 characters per field. These key fields can be located anywhere within an item, providing that they are in the same position within each item.

The location of each key field is specified by the user. If more than one key field is used, all fields must be specified in decreasing order of importance; i. e., the key field of major importance is specified first.

Sort key fields can contain any octal configuration. Standard Honeywell collating sequence is normally used in the sort, but the user may produce his output in IBM collating sequence by specifying the appropriate parameters.

### MAGNETIC TAPE UNITS

Tape drives for 1/2-inch tape are used by the Tape Sort C program. In systems equipped with a recording density option, the higher density offers higher sorting efficiency. The user may elect to stop the program before the last pass to change the density of the final output tape.

### UNREADABLE INPUT RECORDS

In the event that an input record is unreadable (the read-error indicator remains on after several attempts at rereading), the sort will proceed according to the error response specified in parameters.

### PROGRAM HISTORY

The user may follow a history of the sort processing by designating specific parameters to activate the printer, the typewriter, or both devices. If only a printer is to be used and is designated, the sort program prints out a listing of sort parameters (if loaded by Tape Sort C), the input tape header label (if standard), merge phase counts, and sort progress statements, giving item and record counts where applicable. When only a typewriter is used, only the sort progress

statements are listed. Using both devices yields a combined printing and typing of the history, where the typewriter lists the sort progress statements and the printer lists its information, as mentioned above, excluding the sort progress statements.

#### RESTART CAPABILITIES

A method of restarting has been incorporated into the Tape Sort C program to allow the operator to resume execution of the sort if it has been stopped for either of two reasons. One occasion for restarting is in a situation where sort processing has a low priority and it is desired to stop the sort and run another program in its place. In this case, the sort is stopped and all tapes are demounted; at a later time the sort will continue through a restart action. The second occasion for restarting occurs if the sort cannot proceed because of a malfunction. In this instance, if the malfunction is correctable, the restart is activated to resume execution immediately.

## SECTION II

### TAPE SORT C PROGRAM SPECIALIZATION

#### PARAMETER DESIGNATION

The information which specializes Tape Sort C for a particular application is made available to the program by means of parameters designated by the user. Parameters may be supplied to the sort program in a variety of ways, depending upon the method by which the program is being run. For instance, if Tape Sort C is to be executed as a single run, the parameter characters can be punched on three parameter cards, as described in Section IV, and read in from the card reader under control of Tape Sort C segment 00. Another method of parameter entry, which is convenient when Tape Sort C is to be run as one of a series of programs, is for the user to include coding in the program executed immediately before the sort which will set up in memory the specific parameters to be used during sort processing and call in the first segment of Tape Sort C.

If Tape Sort C parameters are entered in memory by the preceding program, the need for parameter card loading is eliminated, as the parameters will be in memory when the preceding program calls the Loader-Monitor to load the sort program. If the parameters are set up by the preceding program (and are therefore essentially fixed), any minor modifications needed for a particular run can be entered manually from the control panel or console when Tape Sort C is loaded.<sup>1</sup>

Parameter information is summarized in Table 4-3, page 4-6, and the parameter characters are loaded into memory starting at location 2,477 (octal). The following paragraphs describe each parameter field.

NOTE: When numeric parameters are specified, leading zeros must be designated; for example, the field specifying the maximum number of characters per item (parameters 20-23) is designated 0080 if items do not consist of more than 80 characters.

#### Tape Unit Specifications (Parameter Characters 1-19)

Tape Sort C requires a minimum of three tape drives. Up to five additional drives may be used for multi-reel input and for work tapes to reduce the total sort time. The user specifies the address and function of each tape drive with parameter characters 1 through 16. Two

---

<sup>1</sup>Parameters may be loaded into memory by any method available to the user at any time after the Loader-Monitor is loaded, but the parameters must be in correct format in the specified memory locations before the sort program is loaded.

characters are required to address a tape drive: the first character specifies the tape control, and the second character designates a particular tape drive attached to that control.

Characters 1 and 2 specify the address (tape control designation followed by drive number) of the primary input tape. In the event that the input to Tape Sort C is on punched cards, character 1 is a "C" and character 2 is blank. Characters 1 and 2 are blank if all of the input data are to be provided through own-coding. Characters 3 and 4 designate the address of the alternate input reel if multi-reel input is used. If card input is to be used, character 3 specifies the address of the card reader and character 4 is blank. Characters 3 and 4 are blank if neither multi-reel input nor card input is used.

Characters 5 and 6 designate the address of the first work tape. The tape drive at this address will contain the final sorted output file. Characters 7 and 8 designate the address of the second work tape. Characters 9 and 10 specify the address of an optional third work tape; these characters are blank if work tapes 3, 4, and 5 are not used. Characters 11 and 12 specify the address of an optional fourth work tape; these characters are blank if work tapes 4 and 5 are not used. Characters 13 and 14 designate the address of an optional fifth work tape; these characters are blank if no fifth work tape is used.

Characters 15 and 16 specify the merge work tape. For a merge work tape, the user may designate the primary input tape, the alternate input tape (assuming one is used), or any other available tape not previously specified as a work tape in characters 5 through 14 (see "Effects of Parameter Characters 15, 16, and 19" on page 2-13).

If parameter characters 15 and 16 are "ΔR", the primary input tape is rewound and locked at the end of the presort, and the merge uses the newly mounted reel on that drive for the merge work tape. In the event of multi-reel input with an alternate input drive specified and "ΔR" designated, the merge work tape will be on the drive from which the next to last input reel was removed. If a tape drive which was not previously designated is specified as the merge work tape, the merge will use that portion of the tape following the read/write head, thereby preserving the data preceding the read/write head.

Characters 17 and 18 designate the total number of input reels to be read by the sort. If the number of input reels is greater than one, each input reel is rewound and locked regardless of parameter characters 15 and 16.

Character 19 specifies input tape direction: a blank indicates that the tape is to be read forward, and a "B" indicates that the tape is to be read backwards. To have the input tape

read backwards, standard labels without tape marks must be used, the tape must be positioned after the end-of-file record, and the number of input reels must be one.

If the input is read backwards, the trailer label accepted by the presort is transferred to the last pass as a header label (the 1EOF is replaced by a 1HDR); the header label accepted by the presort is transferred to the last pass as a trailer label (the 1HDR is replaced by a 1EOF). Presort header-trailer own-coding cannot be used to modify the labels when the input is read backwards.

NOTE: If the drive specified for the merge work tape is the same as that designated for the primary or alternate input tape, the input is not preserved and is destroyed during the merge; but, if a new work tape is to be mounted on the input drive as indicated by "R" for parameter character 16, the action of the presort is the same as that for read forward input.

#### Item and Record Sizes (Parameter Characters 20-31)

Characters 20 through 23 specify the number of characters per item. Note that item size must remain fixed throughout the sort and cannot exceed the maximum record length derived from formula 2 below. Note that the item size can never exceed four thousand characters.

Characters 24 through 27 designate the number of items per input record (characters / input record ÷ characters/item = items/input record). Characters 28 through 31 designate the number of items per output record. The record size limits for output records are the same as for input records, but input and output blocking need not be the same.

#### RECORD SIZE DETERMINATION

The user should solve the record size formulas given below to determine whether his application is efficiently using the available memory and tape drives. Tape Sort C processing is most efficient when double-buffering can be applied during the merge, that is, when two read/write buffers are provided for each merge input and output tape. To achieve double-buffering, the user's record size (the largest size specified in parameters for input or output) cannot exceed the optimum record size calculated from formula 1, below. If the user's record length is larger than the optimum record length, double-buffering cannot occur without raising the highest address available to the sort program or decreasing the number of tapes used in the merge.

$$1. \quad \text{Optimum record length} = \frac{\text{HMA} - \text{MON} - \text{PROG}}{2\text{NT}_m} - 13.$$

The variables used in formulas 1 and 2 are defined as follows:

HMA	Highest memory address available to the sort (parameter characters 45-50) <sup>f</sup>
MON	Monitor requirements (1,340 characters)
PROG	Sort program requirements (3,860 + TYP + ADM)

TYP	Typewriter availability (520 with three-character address mode; 590 with four-character address mode)
ADM	Address mode used (no additional locations required for three-character mode; 1,000 for four-character mode)
NT <sub>m</sub>	Number of tapes used by the merge

If the user's record size exceeds the optimum record size, he must then determine the maximum permissible record size according to formula 2. The maximum record length calculated must be larger than the user's record length, or the sort program cannot process the file without record reblocking.<sup>1</sup> If the user's record size is larger than the maximum record size but his item size is less than the maximum record size, he can allow the sort program to block the records down to a permissible length and may still process his file.

$$2. \quad \text{Maximum Record Length} = \frac{\text{HMA} - \text{MON} - \text{PROG}}{\text{NT}_m + 2} - 13.$$

NOTE: If the user's record length is greater than the maximum record length determined from formula 2, blocking down is necessary; and, in order to process a file when blocking down is required, the items of the file cannot be greater in size than the maximum record length calculated.

If the Sort program does block down, it will create fewer items per record until the reduced record will fit within the maximum record length calculated. Therefore, blocking down generates more records for the file internally, also resulting in the formation of additional interrecord gaps on tape. The user with a large volume of input should use caution when blocking down, as the internal expansion of the file could cause a work tape to overflow during the merge.

#### Padding (Parameter Character 32)

If input and output blocking are different, an incomplete record may result at the end of the file being sorted. Using the padding character specified in parameter 32, the sort program pads this record (adds the proper number of padding items, consisting of padding characters, to fill the record). It is suggested that the padding character be high in the collating sequence in order to preserve the sequencing of the file; e. g., "9" for a numeric file, "Z" for an alphanumeric file.

NOTE: In all cases where padding items are added to an even parity file, the last character of each padding item is an IBM record mark (72<sub>8</sub>).

#### Tape Labels (Parameter Characters 33-34)

Characters 33 and 34 specify labeling options for tape input, as indicated in Table 2-1.

<sup>1</sup>If a run is attempted, checkpoint 10 occurs, allowing the user to set SENSE switch 1 ON and press RUN or type "G" to have the sort program attempt the blocking-down processes. If the check recurs immediately, the file cannot be processed for this machine configuration.

If standard labels and trailers are specified, the output tape header is identical to the header of the first input reel except that the serial number of the output tape will be retained. The trailer record, if any, and tape mark on the last input reel are copied onto the final output tape with an 80-character end-of-recorded-information record (IERI) written after the tape mark.

Table 2-1. Tape Sort C Label Options for Tape Input

Characters 33 and 34	Beginning of File	End of File
$\Delta\Delta$	no label	tape mark
$\Delta T$	no label	tape mark, trailer, and tape mark
SS*	standard label (with or without following tape mark)	standard trailer (with or without preceding tape mark) and tape mark
N $\Delta$	non-standard label	tape mark
NT	non-standard label	tape mark, trailer, and tape mark
LL	standard (120-character) IBM 1410/7010 label and tape mark	tape mark, standard (120-character) IBM 1410/7010 trailer, and tape mark

\*If a tape mark appears at the beginning of file, one must precede the trailer at the end of a file; if a tape mark precedes a trailer, the final output will always have a tape mark following the trailer.

For any other label combinations, the final output tape will have the same type of labeling as the input tape, except that the sort program will append a tape mark (17<sub>g</sub>) to a trailer preceded by a tape mark if a tape mark did not follow the trailer on input. Note that the user may formulate any label changes through own-coding.

When "SS" or "LL" is designated, the sort program references the five-character data field of the Loader-Monitor communication area (octal locations 216-222) and writes the data field configuration in the final output header label. The data is written in character positions 31-35 of an 80-character label (SS) or in character positions 11-15 of a 120-character label (LL).

When using card input, it is necessary to define the label combinations for the final output tape in parameter characters 33 and 34 (see page 4-5). If last-pass header-trailer own-coding was specified, the user can modify the output tape labels without regard to the labeling option designated in parameters.

#### Banner Characters (Parameter Characters 36-37)

Characters 36-37 are used to specify the presence or absence of banner characters in input and/or output records. Character 36 is blank if the first character in each input record is any type of banner character or a "1" if banner characters are not present. Character 37 is blank if the first character of each output record is to contain the Honeywell standard banner character (56<sub>g</sub>) or a "1" if output records are not to have banner characters. These conditions are summarized in the following table.

If characters 36-37 specify	each input data record has	each output data record is to have
Δ Δ	a banner character	a banner character
Δ 1	a banner character	no banner character
1 Δ	no banner character	a banner character
1 1	no banner character	no banner character

#### Parity and Record Gap (Parameter Characters 38-39)

Character 38 is blank for odd-parity input or an "E" for even-parity input. Character 39 is blank if the output is to be written with odd parity and short interrecord gaps, an "L" if the output is to be written with odd parity and 3/4-inch interrecord gaps, or an "E" if the output is to be written with even parity and 3/4-inch interrecord gaps.

#### Error Options (Parameter Characters 40-41)

In the event of an uncorrectable read error, Tape Sort C proceeds according to the error option indicated by characters 40 and 41. These characters are blank if it is desired to cause a check when an uncorrectable read error is encountered (checkpoint 15); specifying "PΔ" causes the record to be printed and checkpoint 15 to occur. "PD" can be designated to have the record printed and deleted from the file (during presort and last pass only). Note that records causing uncorrectable read errors appear on the printer only.

Should either check option be specified ("ΔΔ" or "PΔ"), checkpoint 15 offers procedures enabling the operator to attempt to reread the record or delete the record from the file (see Table 5-1 on page 5-6).

#### Printer Control (Parameter Character 42)

If a printer is to be used during the sort program run, character 42 must designate the address of the printer control unit. This character is blank if a printer is not used.

Note that the sort program is prepared to list a history of the program if a printer, a typewriter, or both devices are available. The chart below is designed to show the actual messages that appear on the device(s) used. A checkmark in a column to the left of the chart indicates that the information given in the corresponding row is listed by the device symbolized at the head of that column. When both devices are used, a "T" (typewriter) or "P" (printer) or "T+P" is designated to indicate which device lists the appropriate information given in the corresponding row.

Table 2-2. Sort Program History

P	T	T/P	Information List
✓		P	Parameter card images
	✓	T	Program signal: : TAPE SORT C ON □
✓	✓	P	Input header label image
		T	Input reel item count: as each input reel is processed, an item count is listed which includes items added through own-coding but excludes items deleted through own-coding or dropped due to read errors. The following is an example of a listing from a sort requiring three input reels: : REEL #01 ITEMS 034260 DEL 000000 : REEL #02 ITEMS 041980 DEL 000000 : FINAL REEL ITEMS 036300 DEL 000004
✓	✓	T	End of presort notation : END OF PRESORT. RESTART MSG 04M
✓		P	Merge phase counts: the phase number is listed at the completion of each merge phase, e. g., : END OF PHASE 04 : END OF PHASE 03 : END OF PHASE 02
✓	✓	T+P	Last pass phase completion with record count per output reel and output tape and drive designation, e. g., 001741 RECORDS REEL 01 001691 RECORDS REEL 02 : END OF PHASE 01 001732 RECORDS FINAL TAPE TCU 00 DRV 2
✓	✓	T+P	End of sort notation, e. g., : END OF SORT 000112540 ITEMS 0000 PADDING 000000 DELETED  (The item count above consists of the items processed and the items added by own-coding; it excludes the items deleted by own-coding, the padding items, and items dropped due to read errors.)
	✓	T	End of job notation with tape control designation followed by the number of the drive holding the final output tape, e. g., : 00 2 END OF JOB

Last-Pass Check Option (Parameter Character 44)

A "1" for character 44 causes a check (checkpoint 13) after the merge so that the operator can change the density of work tape 1 (the final output tape) or reassign the logical address of this tape to an unused drive. Character 44 is blank if it is not desired to stop the sort except for tape changing or error options.

Highest Memory Address (Parameter Characters 45-50)

The highest memory address available to the sort program is specified by characters 45 through 50. The address is expressed as either a decimal number with leading zeros (e.g., 012287) or a number of 4K modules with leading blanks (e.g.,  $\Delta\Delta\Delta\Delta 3$  for 12,287) and must be lower than the origin of presort own-coding (if any).

If Floating Tape Loader-Monitor C is used to load the sort program, the base address of the Loader-Monitor is compared to the highest address specified for the sort, and the sort program uses the lower address of the comparison as the highest address available for program execution.

Collating Sequence and Character Code (Parameter Characters 51-52)

Tape Sort C may be used with any collating sequence, and, if desired, the character code on the final output tape may be different from that on the input tape. Collating sequence translation is exercised through tables in the presort and last-pass segments. The translation tables provided with the sort program permit the user to sort in IBM sequence. Use of the tables to produce a sorted output in IBM collating sequence implies that the IBM input tapes are recorded in binary coded decimal (even parity) and that the tape control is equipped with the 051 code compatibility feature.

Characters 51 and 52 are blank when no sequence translation is required in either the presort or the last pass. This parameter configuration specifies either a standard Honeywell sort or, with the 051 feature, a sort involving IBM-coded input that will produce output in Honeywell collating sequence and character code. Characters 51 and 52 are "TT" when the translation tables are to be used in the presort and last-pass segments (the 051 feature is required), the input is in either Honeywell or IBM character code, and the sorted output is to be in IBM collating sequence and character code.

Ascending or Descending Sequenced Output (Parameter Character 53)

If the final output is to be in ascending sequence, character 53 is blank. If the final output tape is to be in descending sequence, character 53 is a "D".

Own-Coding (Parameter Characters 54-77)

Own-coding may be used in conjunction with the presort and last-pass segments of Tape Sort C to perform additional data processing. Two own-coding exits can be specified for both the presort and the last pass. One exit offers the user the opportunity to inspect and modify header and trailer labels, while the other exit allows the user to perform additional processing

on an item-by-item basis. When the item-by-item exit is specified, program control first exits to own-coding after the sort segment involved has been specialized, permitting the user to modify the segment coding and to prepare for item processing. Control subsequently exits to own-coding for each item of the file, allowing items to be inspected and modified and enabling items to be added to and deleted from the file. Own-coding features are discussed further in Section III.

All own-coding addresses are expressed as decimal values with leading zeros. When designating own-coding addresses, note that presort own-coding must be located above the highest memory address (parameter characters 45-50), but last-pass own-coding may be below that address at the expense of restricting the memory area to be used by the last-pass segment.

#### **PRESORT OWN-CODING (Parameter Characters 54-65)**

Characters 54 through 59 designate the own-coding address to which the presort branches whenever a header or trailer label has been read. These characters are blank if this option is not used. Characters 60 through 65 designate the own-coding address to which the presort branches (1) after the presort has been specialized and (2) before processing each item. Characters 60 through 65 are blank if this option is not used.

#### **LAST-PASS OWN-CODING (Parameter Characters 66-77)**

Characters 66 through 71 designate the own-coding address to which the last pass branches before writing a header or trailer label. These characters are blank if this option is not used. Characters 72 through 77 designate the address to which the last pass branches (1) after the last pass has been specialized and (2) after each item has been placed in the output buffer. Characters 72 through 77 are blank if this option is not used.

#### **Parameter Card Identification (Parameter Characters 79-80)**

If parameters are to be entered through the card reader by Tape Sort C segment 00, columns 79 and 80 of parameter card 1 must contain the configuration "S2" to identify the parameter cards being read. If parameters are entered by any other method, these characters are not specified.

#### **Key Fields (Parameter Characters 81-140)**

The specification of each sort key field requires six characters: four to specify the position of the high-order (leftmost) character of the key field in the item, and two to specify the number of characters in the field (see Table 2-3). Decimal values with leading zeros are used in designating the six characters. The position within the item of the high-order character of each

SECTION II. TAPE SORT C PROGRAM SPECIALIZATION

key field is determined by counting the leftmost character in the item as 0001 and adding the number of positions required to reach the high-order character of the specific key field. If fewer than ten sort key fields are specified, the unused parameter characters are blank.

Table 2-3. Parameter Characters for Key Fields 1 Through 10

Key Field Number	1	2	3	4	5	6	7	8	9	10
Key Field Position	81-84	87-90	93-96	99-102	105-108	111-114	117-120	123-126	129-132	135-138
Number of Characters in the Key Field	85-86	91-92	97-98	103-104	109-110	115-116	121-122	127-128	133-134	139-140

Input File Name Verification (Parameter Characters 141-150)

If standard labels are used (characters 33 and 34), the file name of the first input reel can be verified without own-coding, provided that characters 141 through 150 specify the file name. These characters are blank if the file name is not to be verified.

Output File Name Change (Parameter Characters 151-160)

If standard labels are used, the file name of the output file may be changed without own-coding. The new output file name is specified by characters 151 through 160, and the sort program makes the change. These characters are blank if no change is desired.

Last-Pass Own-Coding Program Name (Parameter Characters 161-166)

Characters 161 through 166 designate the name of the last-pass own-coding program if last-pass own-coding is to be used and has not already been loaded into memory. The last pass automatically branches to the Loader-Monitor program to initiate a search for the own-coding. Characters 161 through 169 are blank if there is no last-pass own-coding or if it was already loaded into memory before the sort began.

Last-Pass Own-Coding Segment Name (Parameter Characters 167-168)

Characters 167 and 168 designate the name of the last-pass own-coding segment.

Search Direction for Last-Pass Own-Coding (Parameter Character 169)

Character 169 specifies the direction in which the Loader-Monitor should search for the last-pass own-coding. Character 169 is a "B" for a search forward or a "C" for a search backward.

Next Program Name (Parameter Characters 170-175)

Characters 170 through 175 designate the name of the program (if any) that is to follow

Tape Sort C. If these characters are specified, there is an automatic branch to the Loader-Monitor program to initiate a search for the requested program and segment. Characters 170 through 178 are blank if Tape Sort C is not to be followed by automatic loading of another program; in this case, once the sort is completed, the operator can press RUN or type "M" to revert control to the Loader-Monitor general return address.

NOTE: Parameters 170-178 are to be specified for the communication area of the Loader-Monitor; search mode and visibility are not changed by the sort program.

#### Next Program Segment Name (Parameter Characters 176-177)

Characters 176 and 177 designate the name of the first segment within the next program to be loaded.

#### Search Direction for the Next Program (Parameter Character 178)

Character 178 specifies the direction in which the Loader-Monitor should search for the next specified program on the program tape. A "B" designates a search forward, and a "C" designates a search backward.

#### TAPE POSITIONING

NOTE: The letters used to label the following paragraphs are also used to designate the corresponding diagrams in Figure 2-1.

#### Beginning of Sort

##### Primary Input Tape

- A. If the primary input tape(s) is to be read forward, it must be positioned immediately before the header label or, if there is no header label, before the first data record.
- B. If the primary input tape is to be read backward, standard labels without tape marks are used ("SS" in parameter characters 33-34), the tape is positioned immediately after the end-of-file record, and the number of input reels must be one.

##### Work Tapes

- C. Work tapes may be positioned at any point on the tape. Beyond the read-write head, there must be two valid records and enough tape to prevent reaching the end of reel during the merge. All data preceding the read-write head are preserved.

#### End of Sort

##### Primary Input Tape

- D. If the primary input tape has been read forward, it is positioned after the trailer record or tape mark.

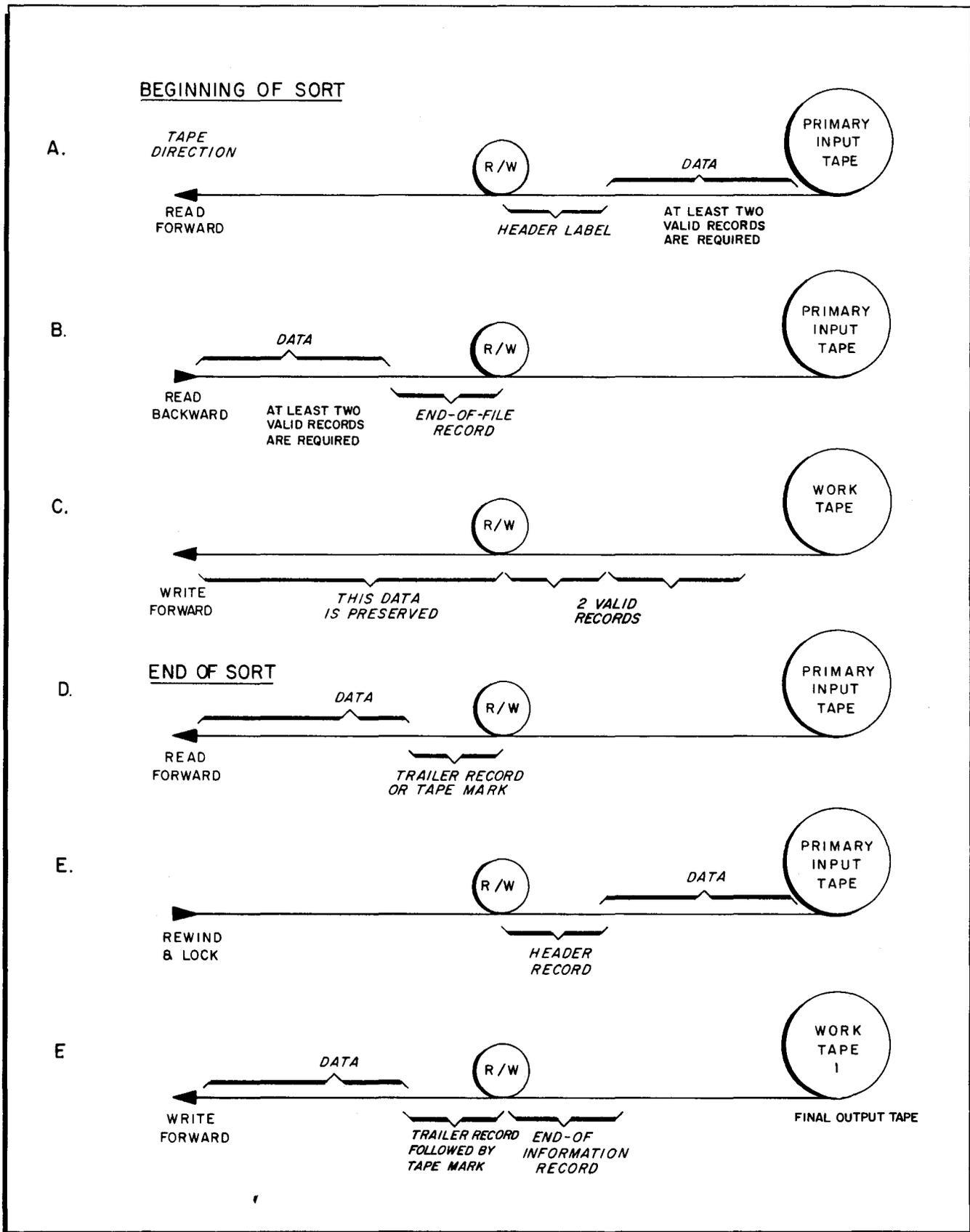


Figure 2-1. Tape Positioning in the Tape Sort C Program

- E. If the rewind and lock option has been used on the primary tape, the tape is rewound.
- F. If the primary input tape has been read backward, it is positioned before the header record, as in Figure 2-1E.

#### Work Tapes

- G. Work tape 1, the final output tape, is positioned between the trailer record and an end-of-recorded-information record.
- H. If any work tapes were rewound before the sort began, the read-write heads are positioned immediately after the first record.
- I. All other work tapes, excepting those mentioned in G and H, are repositioned to the same point at which they were located when the sort began.

#### Effects of Characters 15, 16, and 19

NOTE: The letters used to label the following paragraphs are also used to designate the corresponding diagrams in Figure 2-2.

#### Characters 15, 16, and 19

- |        |                                |  |
|--------|--------------------------------|--|
| A.     | $\Delta$ , R, and $\Delta$     | The primary tape is read forward, rewound, and locked at the end of the presort.   |
| B.     | $\Delta$ , R, and B            | The primary input tape is positioned at the beginning of the sort with the end-of-file record preceding the read-write head. The tape is read backward, and at the end of the presort, the tape is rewound and locked. |
| C.     | (t, t), and $\Delta$           | The primary input tape is read forward; then the merge output data is written forward on the primary input tape after the end-of-file record of the input data.  |
| t, t = | Address of primary input tape. |  |
| D.     | (t, t), and B                  | The primary input tape is read backward so that its header label lies beyond the read-write head. The merge output data is then written forward on the primary input tape overlaying the input data.                   |
| t, t = | Address of primary input tape. |  |
| E.     | (t, t), and $\Delta$ or B      | A work tape may be positioned at any point on the tape. The merge uses that portion of the tape beyond the read-write head, while preserving the data preceding the read-write head.                                   |
| t, t = | Address of work tape.          |  |

CHARACTERS 15, 16, and 19

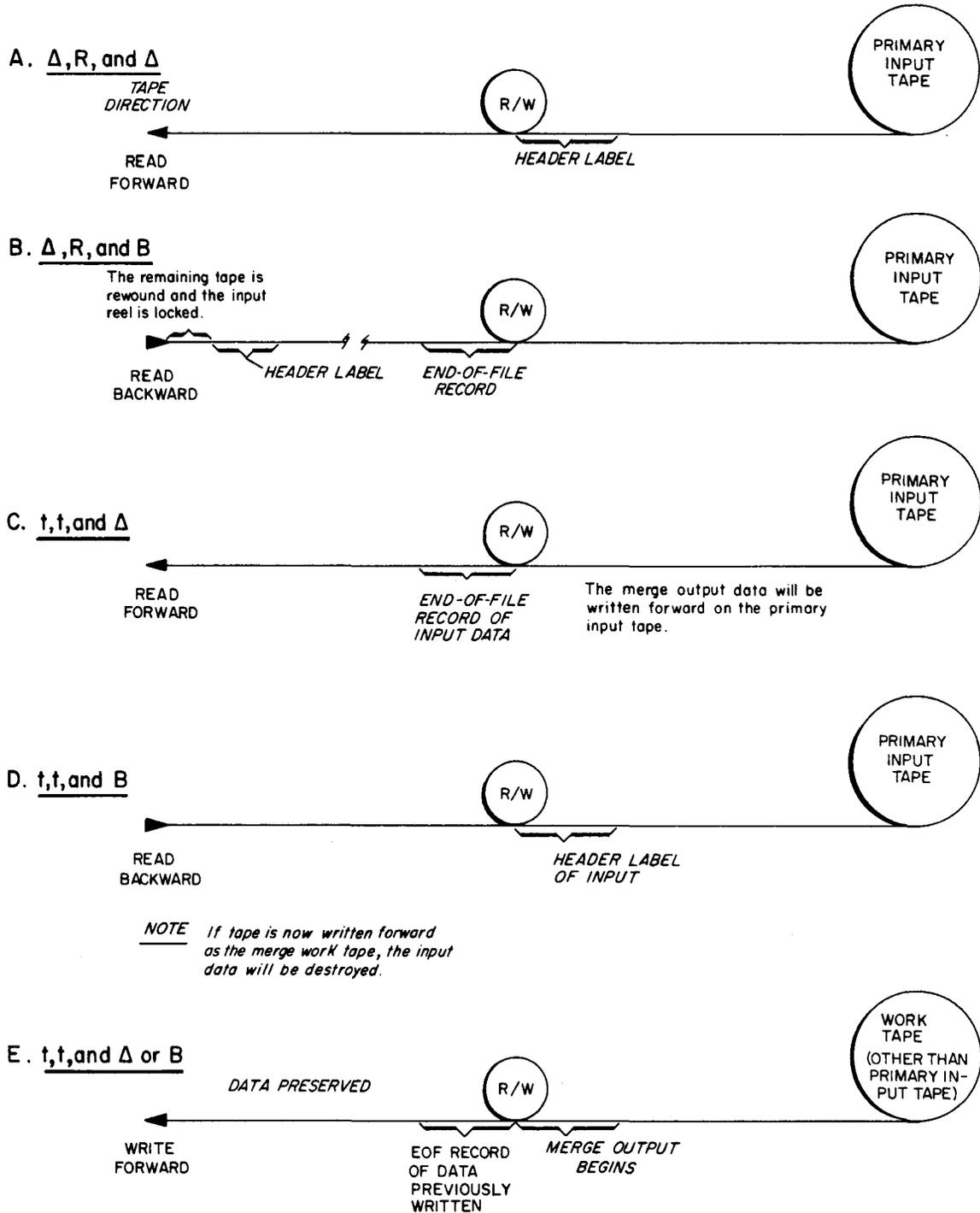


Figure 2-2. Tape Positioning for Effects of Characters 15, 16, and 19

## SECTION III

### TAPE SORT C OWN-CODING

Own-coding routines may be written by the user in EasyCoder C symbolic language, assembled, and used in the presort and the last-pass segments of the Tape Sort C program. The presence of these routines is indicated to the sort program by the specification of parameter characters 54 through 77. The user may designate a branch to header-trailer own-coding to inspect, modify, or replace the tape labels. A branch to item-by-item own-coding offers the user the opportunity to inspect, modify, add, or delete items, but these routines cannot change the size of an item.

This section contains explanations of the possible own-coding exits and their corresponding functions as well as the considerations to be noted when programming the actual own-coding routines. While studying the text, the reader can visualize the methods of linking own-coding to the sort program by juxtaposing the written information with the appropriate lines of a sample own-coding routine in Appendix B. Once the reader has an understanding of the own-coding functions, he can reference the linkage routines in Appendix B when programming the desired operations.

#### PRESORT OWN-CODING

##### Presort Header-Trailer Own-Coding

Presort header-trailer own-coding is specified by designating parameter characters 54 through 59 as the initial address of the user's own-coding label routine. Header-trailer own-coding in the presort enables the user to access a header or trailer before it is processed, since designation of characters 54 through 59 specializes the sort program to first load index register 1 with the address of the first (leftmost) character of any label that is read. If a header label is read, the presort sets an item mark at the initial own-coding address designated; an item mark is not set if a trailer label is read. Program control then branches to the initial label own-coding address, where the first instruction of the user's own-coding routine should be an SCR (Store Control Register) of the B-address register (see sample, lines 4-6). Label processing may then consist of the following:

1. inspection and modification
2. modification without changing label length
3. modification with changes in label length

The user must clear and set record marks in the event of a change in the label length, so that the record mark defining the original label length is cleared and a new record mark is set one character beyond the last (rightmost) character of the modified label. The modified label cannot exceed in length a data input record or 80 characters, whichever is greater; if parameter characters 33 and 34 specify "LL" to designate that standard IBM 120-character labels are to be used, the modified label cannot exceed in length a data input record or 120 characters.

Own-coding may also be used to replace the existing header or trailer label with a different label by loading index register 1 with the address of the first (leftmost) character of the new label and setting a record mark one character beyond the last (rightmost) character of the new label (see sample, lines 8 and 9). In the presort, own-coding may not be used to insert such a label where no label exists on the input; last-pass header-trailer own-coding may be used to add a label when none exists on the input (see page 3-7).

NOTE: For program control to branch to presort header-trailer own-coding, parameter characters 33 and 34 must specify that there is a header or trailer record on the input. For example, if a tape mark followed by a trailer is specified in these characters, the presort will branch to own-coding after reading that trailer; but if all the input is to come from own-coding, there will not be a branch to the header-trailer own-coding address in characters 54 through 59, regardless of whether header-trailer own-coding was specified.

#### Multi-Reel Input

If multi-reel input is used, the presort branches to own-coding for each header or trailer that is read; but the user is able to modify only the first header and the final trailer. For each intermediate tape, the header and trailer are available only for examination.

#### Read Backwards Input

If the input is read backwards, header-trailer own-coding cannot be used. Item-by-item own-coding options are available whether the input is read backward or forward, as the last character of an item is the character occupying the numerically highest memory location when the input was read backward or forward.

#### Presort Item-by-Item Own-Coding

If item-by-item own-coding is to be used during the presort segment of the Tape Sort C program, the initial address of the user's own-coding routine must be designated in parameter characters 60 through 65. If these parameters are specified, the presort automatically performs a "one-shot" branch to the designated own-coding location before any input records are read. The "one-shot" branch offers the user his only opportunity to modify the program and to store the contents of index registers 1, 2, and 3 for further own-coding reference (see sample,

lines 14 and 40-50). The user must store the contents of the B-address register upon entering his own-coding routine. The contents of these registers are stored for the following reasons:

1. The address contained in the B-address register will designate the program re-entry location to which the own-coding routine branches to re-enter the presort from this "one-shot" routine.
2. The address contained in index register 1 designates the program re-entry location to which own-coding branches when adding an item. Location 2 (in index register 1) contains a wordmark to terminate the retrieval of this address.
3. The address contained in index register 2 designates the program re-entry location to which own-coding branches when deleting an item. Location 6 (in index register 2) contains a word mark.
4. The address contained in index register 3 designates the program re-entry location to which presort own-coding branches when terminating. Location 10 (in index register 3) contains a word mark.

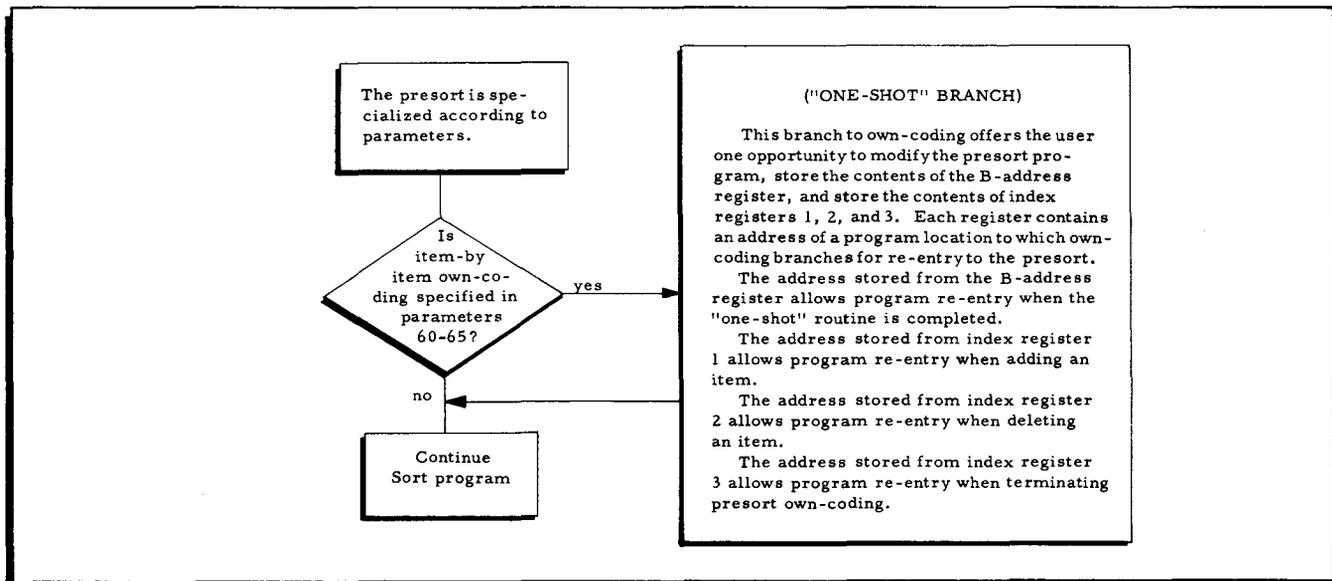


Figure 3-1. Summary Flow Chart of Presort "One-Shot" Branch

The "one-shot" branch enables the user to store the contents of index registers 1, 2, and 3 in own-coding locations that he can access. The address stored from each register is referenced for a particular own-coding process, i. e., adding an item, deleting an item, etc. Each address references a location at which the program is re-entered to complete the processing of a specific item (to add the item, delete it, or use it to terminate own-coding). If these addresses were not stored from the index registers at the time of the "one-shot" branch, the specific program re-entry locations could not be accessed.

Note that after the user has stored the contents of index registers 1, 2, and 3 during the "one-shot" branch, the settings of these registers change as the program is processed. If the

user desires to apply index registers 1, 2, or 3 for a particular function within an own-coding routine, e. g., for data manipulation, his coding must store the contents of the register used each time own-coding is entered and restore the original contents of the register before re-entering the Sort program. This ensures that proper program sequence is retained when the program is re-entered.

#### PRESORT PROCESS DETERMINATION

After the "one-shot" branch is executed, the user is given access to each input item through the presort which loads index register 1 with the address of the last (rightmost) character of the input item and branches to the own-coding starting address. The user should again store the contents of the B-address register upon entering the own-coding routine, as the stored address can now be used to provide a normal re-entry to the sort program whenever desired. It then becomes necessary to determine the action to be performed on the particular item accessed (e. g., to inspect or modify it, to delete it, etc.) and to branch to the routine for performing the appropriate action (see sample, lines 16-23).

#### INSPECTING OR MODIFYING AN ITEM

If the user wishes to inspect and possibly to modify an item, he can reference index register 1 for the address of the last character of the current item to be processed. Note that item length and punctuation must not be changed during modification. After the desired processing is performed, the sort is re-entered by branching to the address stored from the B-address register upon entering the own-coding routine (see sample, lines 30-32).

#### DELETING AN ITEM

If it has been determined that the current input item is to be deleted from the file, control can simply branch to the delete exit address stored from index register 2 during the "one-shot" branch, and the item is deleted upon re-entry to the sort (see sample, lines 27-29).

#### ADDING AN ITEM

Own-coding provides the user with the capability to add an item (s) to the file before any specific input item to be processed. If it has been determined that the current input item is to have an own-coding item added before it, instructions in the user's add routine must load index register 1 with the address of the last (rightmost) character of an item to be added and branch to the address stored from index register 1 during the "one-shot" branch (see sample, lines 24-26). The presort is thereby re-entered and performs the addition.

After adding an item from own-coding, the current item in the input buffer is again presented for processing, program control again branches to own-coding; and, if there is another

item to be added, the procedure given above must be executed. If no more items are to be added at this point, re-enter the sort by branching to the address stored from the B-address register; the current item in the input buffer will then be processed.

An item to be added must be in the input format specified by the sort parameters (i. e., maximum item length, key field locations, and key field lengths must be the same). The item to be added must contain word marks on the following characters:

1. The first character of the item.
2. The first character of each key field.
3. The character immediately following each key field.

There must not be any other word marks in the item.

Figure 3-3 shows an item to be added to the input data of a sort program run. This item is in the same format as the input items to be sorted. The locations of word marks are indicated by circles around the characters with which they are associated.

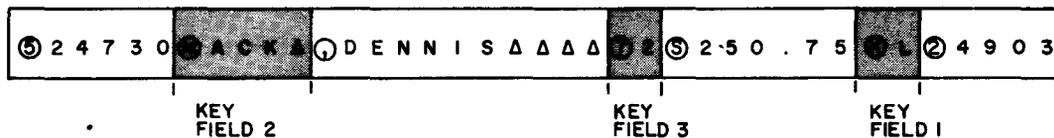


Figure 3-2. Word Mark Locations in a Sort Item

#### Terminating Presort Own-Coding

If it is determined that own-coding is to be terminated, the sort program is re-entered at the program location specified by the address stored from index register 3 during the "one-shot" branch.

The branch to the terminating location may be made before or after all input items have been processed. When the presort segment has processed all of the input data, an item mark is automatically set at the presort item-by-item own-coding address, and own-coding may either terminate or add more items (see sample, lines 15 and 33-38). If there are no more items to be added, there must be a branch from own-coding to the address stored from index register 3 during the "one-shot" branch, as stated above, and the presort will execute its end-of-job routine.

If there are more items to be added after all input data have been processed, own-coding must load index register 1 with the address of the last character of the item to be added and re-enter the sort program by branching to the address stored from index register 1 during the "one-shot" branch. The item is added by the sort program and control returns to the own-coding routine to add another item (see sample, lines 36-38). A cycle begins with the address of the last character of the item to be added being loaded into index register 1, followed by a branch from own-coding to the address stored from index register 1 during the "one-shot" branch. This cycle continues until control is returned to own-coding and there are no more items to be added. At this time, a branch must be made to the address stored from index register 3 during the "one-shot" branch in order to terminate presort own-coding.

NOTE: The address of a program location stored from an index register during the "one-shot" branch is hereafter referred to as the exit address for the specified routine (e.g., the address stored from index register 1 during the "one-shot" branch is hereafter referred to as the add exit address).

#### Own-Coding Input

When all the input for sort processing is entered by own-coding (parameters 1 through 4 are blank), the presort is initialized as having processed all of its normal input. The user must have instructions in his own-coding to enter items from an input device, and he must code routines, as described above, to add each input item for processing.

#### LAST-PASS OWN-CODING

##### Last-Pass Header-Trailer Own-Coding

Last-pass header-trailer own-coding is specified by designating parameter characters 66 through 71 as the initial address of the user's label routine. Header-trailer own-coding in the last pass enables the user to access a header or trailer after it has been processed and placed in the output buffer. After placing the label in the output buffer, the sort program loads index register 1 with the address of the first (leftmost) character of the label; if the label is a header, the last pass sets an item mark on the header-trailer own-coding location before branching to it. The last pass then branches to the initial label own-coding address, where the first instruction should store the contents of the B-address register. Upon completion of the header-trailer own-coding routine, the program is re-entered by branching to the address stored from the B-address register (see sample, lines 60-68).

Inspection, modification, or replacement of labels in last-pass header-trailer own-coding is performed in the same manner as in the presort, with one exception: in the last pass, if a change is made to the length of a header or trailer, the user must set up the entire label in his own-coding area. A record mark must be set one character beyond the desired length, and index

register 1 must be loaded with the address of the first character of the header or trailer as set up in own-coding. Upon completion of the header-trailer own-coding routine, the program is re-entered by branching to the address stored from the B-address register.

The user may add a header or trailer during the last pass if one was not present on the input. If own-coding was specified in parameter characters 66 through 71, a branch to own-coding will occur whether or not a header or trailer was present on the input. After executing the branch, own-coding may be used to add a header or trailer by writing the label on the output tape, and the program is then re-entered by branching to the address stored from the B-address register.

#### Last-Pass Item-by-Item Own-Coding

If item-by-item own-coding is to be used during the last-pass segment of the Tape Sort C program, parameter characters 72 through 77 must designate the initial address of the user's item-by-item own-coding routine. After specialization but before any item processing, the last pass performs a "one-shot" branch to the designated own-coding location (see sample, lines 71 and 95-101). This branch is similar to the "one-shot" branch of presort own-coding. The registers perform the same functions in the last pass as in the presort (see page 3-3).

#### LAST-PASS PROCESS DETERMINATION

After the "one-shot" branch is executed, the user is given access to each processed item placed in the output buffer, as the last pass loads index register 1 with the address of the last (rightmost) character of the item just placed in the output buffer and branches to the own-coding starting address. Upon entering the own-coding routine, the user should store the contents of the B-address register to provide an address for normal re-entry to the sort. The action to be performed on the current item accessed should now be determined (e.g., delete it, add an item after it, etc.) and then followed by a branch to the appropriate routine for performing the desired action (see sample, lines 74-80).

#### INSPECTING OR MODIFYING AN ITEM

If the user wishes to inspect and possibly to modify an item, he can reference index register 1 for the address of the last character of the current item in the output buffer; note that item length and punctuation must not be changed during modification. After the desired processing is performed, the sort is re-entered by branching to the address stored from the B-address register upon entering the own-coding routine (see sample, lines 90-92).

#### DELETING AN ITEM

If it has been determined that the current input item is to be deleted from the output file,

control can simply branch to the delete exit address stored from index register 2 during the "one-shot" branch, and the item is deleted upon re-entry to the sort (see sample, lines 76 and 77).

#### ADDING AN ITEM

Own-coding provides the user with the capability to add an item(s) to the output file after any specific output item. If it has been determined that the item just placed in the output buffer is to have an own-coding item added after it, instructions in the user's add routine must set an item mark at the location specified by the add exit address (stored from index register 1 during the "one-shot" branch). Control may then either return to the address stored from the B-address register or, at this time, control could be given to the delete exit address for deletion of the item just placed in the output buffer (see sample, lines 81-84).

Before placing another processed item in the output buffer, the last pass interrogates the location designated by the add exit address for the presence of an item mark. If an item mark is present, the last pass does not place the next processed item in the output buffer but branches to own-coding, expecting an own-coding item to be added beginning at the location designated by the address currently in index register 1. To complete the addition, own-coding must then branch to the add exit address to re-enter the sort program (see sample, lines 85-89).

As long as the item mark remains in the location specified by the add return address, the last pass will expect items to be added and will not merge items from its input tapes. When the item, or series of items, has been added, the item mark must be cleared before returning to the add return address (see sample, line 88). When adding an item, the word marks in the output buffer must not be changed (see Figure 3-2 on page 3-5).

#### Terminating Last-Pass Own-Coding

Own-coding must branch to the terminate exit address (stored from index register 3 during the "one-shot" branch) when all own-coding processing is completed. This branch may be made before or after the last pass has processed all of its normal input.

When the last pass has processed all of its input, an item mark is set at the own-coding location specified in parameter characters 72 through 77 and must not be disturbed by the own-coding program. At this time, own-coding may be used either to add more items or to terminate the last-pass segment. Own-coding may add an item without setting an item mark at the location specified by the add exit address. If there are no items to be added, the sort program is re-entered at the terminate exit address, and the last pass will execute its end-of-job routine (see sample, line 73).

If an item is to be added after all tape input has been processed, the address of the last (rightmost) character of the own-coding item must be loaded into index register 1 and a branch must be executed to the add exit address. This cycle continues, loading index register 1 with the address of the last character of the item to be added and branching to the add exit address, until control is returned to the own-coding routine and there are no more items to be added. Own-coding then branches to the terminate exit address, and the last-pass segment terminates.

#### Multi-Reel Output

If multi-reel output occurs, each output tape is rewound when filled and must be replaced with another tape on that drive address; the user continues to have access to header-trailer and item-by-item own-coding.

#### Hash Totals

Tape Sort C does not generate hash totals. Should the user wish to sort even-parity files and desire to include hash totals on the final output, methods can be adopted to preserve or generate hash totals according to the number of input reels used and output reels anticipated. For instance, if a single reel of input is to be sorted and one reel of output is anticipated, the hash total in characters 21-30 of the standard IBM (even parity) input trailer will automatically be copied onto the final output trailer. When multi-reel input is used and a single reel of output is expected, the user can apply presort header-trailer own-coding in order to accumulate the hash totals from each input trailer and use last-pass header-trailer own-coding to include the accumulated total on the final output trailer. If the user anticipates multi-reel output and desires hash totals, he must use last-pass own-coding to generate the hash totals.

#### CONSIDERATIONS FOR PROGRAMMING OWN-CODING

The starting location of presort own-coding must be above the highest address available, as given in parameters 45 through 50. The starting location of last-pass own-coding may be specified below the highest address available at a cost of restricting the memory area to be used by the merge and last-pass segments.

If header-trailer own-coding is used, the address which references a label references the first (leftmost) character of that label. When item-by-item own-coding is being used, the address which references an item references the last (rightmost) character of that item. These rules apply to addresses supplied to own-coding, as well as to addresses supplied to the sort program by own-coding.

During the presort segment, the branch to item-by-item own-coding is executed before each item is moved from the input buffer for processing by the sort program. The branch to

last-pass item-by-item own-coding is executed after the item has been processed by the last-pass segment and moved to the output buffer.

When writing own-coding instructions, the user should be especially aware of the following:

1. Tape Sort C uses read/write channels two and three (RWC2 and RWC3) for reading and writing, respectively. The user must specify RWC1 and RWC1' in his instructions - RWC2 and RWC3 must not be used for own-coding.
2. Tape Sort C uses index registers 1 through 4. If these registers are used during an own-coding routine, the user must store the contents of the registers to be used each time own-coding is entered and restore the original contents of these registers before re-entering the sort program.<sup>1</sup>

---

<sup>1</sup>The one exception to this rule occurs when adding an item from own-coding, as index register 1 is not restored with its original contents (see page 3-4).

SECTION IV  
TAPE SORT C OPERATING CONSIDERATIONS

INTRODUCTION

Tape Sort C may be run as one program within a series, as the initial program in a series, or as a single program. Regardless of the method of running Tape Sort C, the program will be loaded in the same manner as other programs functioning in the Mod 1 Operating System, i. e., by a call to a Loader-Monitor program. Tape Sort C may be stored on tape to be run under control of Tape Loader-Monitor C(3), Tape Loader-Monitor C(4), or Floating Tape Loader-Monitor C; or it could be stored on cards in binary run format to be run under control of Card Loader-Monitor B or the card loading option of Floating Tape Loader-Monitor C. Suggested in this section are various methods of running Tape Sort C according to the type of Loader-Monitor used, including the considerations in preparing for each method of operation.

TAPE SORT C PROGRAM FORMAT

Essential to discussing the mode of operation of Tape Sort C is the sequence of the program's operational segments, which must be executed in the order listed below. The segments are logically grouped as specializing segments and as operating segments for the presort, merge, and last pass. The specializing segments are optional and can be used as follows: 00 for calling parameter cards from the card reader and loading the information into the proper memory locations; T1 for typing out any errors occurring during the parameter card read operation; RS for providing the restart function of the sort program; and T2 for typing out information used in the restart action. The typewriter segments (T1 through T6) are only used when a typewriter is available.

Table 4-1. Tape Sort C Program Format

<u>Program Name*</u>	<u>Segment Name</u>	
AADS2Δ	00	} Specializing Segments
AADS2Δ	T1	
AADS2Δ	RS	
AADS2Δ	T2	
AADS2Δ	01	} Presort
AADS2Δ	T3	
AADS2Δ	A1	
AADS2Δ	T4	
AADS2Δ	02	
AADS2Δ	03	
AADS2Δ	04	
AADS2Δ	05	

Table 4-1 (cont). Tape Sort C Program Format

<u>Program Name*</u>	<u>Segment Name</u>	
AADS2Δ	06	} Presort
AADS2Δ	07	
AADS2Δ	08	
AADS2Δ	M1	} Merge
AADS2Δ	T5	
AADS2Δ	M2	
AADS2Δ	M3	
AADS2Δ	G1	} Last pass
AADS2Δ	T6	
AADS2Δ	G2	
AADS2Δ	G3	
AADS2Δ	G4	

\*When the Sort program is assembled in four-character addressing mode, the program name is AADS2F.

#### CALLING AND LOADING TAPE SORT C WITH A TAPE LOADER-MONITOR

A Tape Loader-Monitor is used when Tape Sort C is stored on a binary run tape (BRT). The calling and loading considerations for the sort program will depend upon the manner in which it is to be run, i. e., whether or not the sort parameters are placed in memory by the program executed before Tape Sort C.

A general consideration is that a Tape or Card Loader-Monitor program assembled in three-character mode should not be used to load a version of Tape Sort C assembled in four-character mode with typewriter options. A three-character mode Loader-Monitor cannot load program segments above 32K; and, since Tape Sort C loads its typewriter routines into upper memory according to the highest address available, as designated by parameter characters 45-50, it is possible that the typewriter routines for this configuration will not be loaded if the highest address specified is above 32K.

#### Programmed Call for Tape Sort C on a BRT

When Tape Sort C is run as a program within a series, it is useful to have the sort parameters left in memory by the program executed before Tape Sort C and to include coding in that program which will initiate the call for the sort program. The Tape Sort C program name is AADS2Δ and first segment name 01 should be placed in the appropriate fields of the Loader-Monitor communication area to call in the program. Specification of segment 01 assumes the presence of parameters in memory.

NOTE: If Tape Sort C has been assembled in four-character address mode, the program name becomes AADS2F, but the methods of running the

program are the same as those explained for the three-character version, AADS2Δ.

#### Card Call for Tape Sort C on a BRT with Parameters on Cards

If Tape Sort C is to be run as the initial program in a series or as a single program, the sort parameters can be loaded from the card reader under control of segment 00. Once the Loader-Monitor program has been loaded into memory, a Console Call card can be used to call in sort segment 00 to enter the parameters from the card reader and initiate loading of the sort program. The card reader should contain the necessary cards positioned in their proper order as shown in Figure 4-1 on page 4-4.

#### CALLING AND LOADING TAPE SORT C FROM CARDS

When the Tape Sort C program is punched on cards in BRT format, Card Loader-Monitor B or the card loading option of Floating Tape Loader-Monitor C is used to call and load the program. The considerations for calling and loading the sort program will vary according to the manner in which the program is run, i. e., whether sort parameters can be loaded by the program executed before Tape Sort C.<sup>1</sup> A general consideration is that a three-character mode Loader-Monitor should not be used to load a version of Tape Sort C on four-character mode with typewriter options.

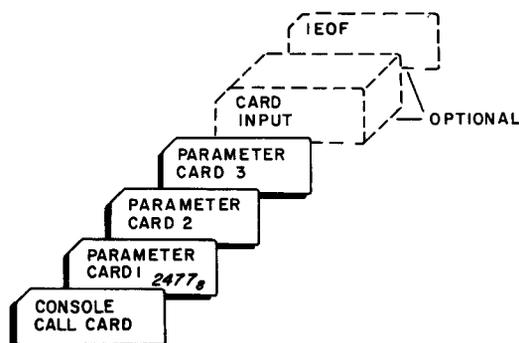
#### Programmed Call for Tape Sort C on Cards

If Tape Sort C is loaded within a series of programs, it is recommended that the preceding program enter the sort parameters in their proper memory locations (the locations are specified in Table 4-3 on page 4-6). The program which precedes Tape Sort C is used to initiate a call by entering the sort program name (AADS2Δ) and first segment name (01) in the appropriate fields of the Loader-Monitor communication area.

The sort program deck is made up of segments, as shown in Figure 5-3 on page 5-5, each of which is identified by a segment header card, a card with an X8 (Q) or an X84 (\*) punch in column 1. After the preceding program has entered the parameters in memory and initiated a call for segment 01 of Tape Sort C, the Loader-Monitor searches forward, reads the 01 segment header card (with the sort program name in columns 11 through 16 and the segment name in columns 17 and 18), and loads the Tape Sort C program.

---

<sup>1</sup> Unless otherwise specified, Card Loader-Monitor B loads punched-card program segments in the order in which they are placed in the card reader input hopper; but the card loading option of Floating Tape Loader-Monitor C requires a Console Call card before any segment not loaded by its preceding segment.



1. The Console Call card. The Console Call card is punched as follows:
 

Columns 1-6:	contain the characters AADS2Δ (or AADS2F if four-character) to indicate the program name.
Columns 7-8:	contain the characters 00 to indicate the segment name.
Column 9:	contains the character 0 to indicate the address of the program run tape containing the Sort program.
Columns 10-17:	contain either blanks to specify program execution after program loading or a specific program and segment name to designate a program halt after loading the specified segment.
Column 18:	contains an asterisk (*) to identify the Console Call card.
2. Parameter Card 1 (parameter characters 1-80 in Table 4-3 on page 4-6).
3. Parameter Card 2 (parameter characters 81-160 in Table 4-3).
4. Parameter Card 3 (parameter characters 161-178 in Table 4-3).

\*If no Loader-Monitor information is required, parameter card 3 is blank; however, the blank card must be placed after parameter card 2.

Figure 4-1. Parameter Card Input for Loading Tape Sort C from Tape

#### Card Call for Tape Sort C Program and Parameters on Cards

When Tape Sort C is to be run as the initial program in a series or as a single program, the sort parameters are punched on parameter cards and placed after the end-of-segment card for 00, which contains an X4 (M) punch in column 1 for easy identification. When the typewriter segment T1 is used, the parameter cards should be placed after the end-of-segment card for T1. The program run deck should be in the card reader in the order shown in Figure 5-2 on page

5-3. Once the Loader-Monitor has been loaded, it will search forward for a loading unit, read the 00 segment header card, and load the sort program (see footnote on page 4-3).

#### CALLING AND LOADING TAPE SORT C OWN-CODING

If presort and/or last-pass own-coding are used with Tape Sort C on a BRT, presort own-coding can be positioned before or after any sort segment on the BRT, but last-pass own-coding must be positioned before sort segment G2. If Tape Sort C is stored on cards, last-pass own-coding must be positioned before sort segment G2.

If presort own-coding is applied with Tape Sort C on tape or cards, the user must provide a method of calling and loading the own-coding. Presort own-coding can be called and loaded in the same manner as any other segment of a loading unit, but it must be loaded before the first sort segment is loaded.

NOTE: If the user has the memory locations available, he can load both presort and last-pass own-coding before the sort program is loaded; in this case, parameter characters 161 through 169 are blank.

#### ENTERING THE INPUT TO BE SORTED

Parameters 1 and 2 must designate the method used to enter the input to be sorted. The input must be in an acceptable format; the sort program will not handle a "no-input" situation. The three possible methods for entering input are given below.

1. If parameters 1 and 2 specify the tape control address and drive number of a tape unit, the input is expected to be on magnetic tape, and the input tape must be mounted before the sort program is loaded.
2. If parameters 1 and 2 (3 and 4 also) are blank, all the input is to be provided by own-coding, and the user must include instructions in his own-coding routine to read the input from a designated Series 200 input device.

NOTE: When own-coding is used to enter all the input, the presort segment is initialized as having processed all of its normal tape input, and own-coding is used to add one item at a time for sort processing.

3. If parameter characters 1 through 4 are specified as "CΔxΔ", where "x" represents the card reader address, the input is expected in card format and must be placed in the card reader input hopper. The first card of the input deck must be a data card and the card following the last data card must contain 1EOF in columns 1 through 4. With card input, certain label combinations are permissible for the final output tape(s); parameter characters 33 and 34 must specify one of the label combinations given in Table 4-2, unless last-pass own-coding is used to provide the labels.

SECTION IV. TAPE SORT C OPERATING CONSIDERATIONS

Table 4-2. Tape Sort C Label Options for Card Input

Characters 33-34	Beginning of File	End of File
ΔΔ	no label	tape mark
ΔT	no label	tape mark, trailer, and tape mark trailer format: characters 1-4 are 1EOF characters 5-76 are blank characters 77-80 are EOF1
SS	header header format: characters 1-4 are 1HDR characters 5-16 are blank characters 17-19 contain reel sequence number character 20 is blank characters 21-30 contain file name, if specified characters 31-35 contain date characters 36-76 are blank characters 77-80 are HDR1	trailer trailer format: characters 1-4 are 1EOF character 5 is blank characters 6-10 contain number of data records, in decimal characters 11-20 contain the num- ber of items processed, in decimal characters 21-76 are blank characters 77-80 are EOF1
NΔ	header header format: characters 1-4 are 1HDR characters 5-76 are blank characters 77-80 are HDR1	tape mark
NT	header header format: characters 1-4 are 1HDR characters 5-76 are blank characters 77-80 are HDR1	tape mark, trailer, and tape mark trailer format: characters 1-4 are 1EOF characters 5-76 are blank characters 77-80 are EOF1
LL	NOT TO BE USED	

TAPE SORT C PARAMETER SUMMARY

The table below lists and defines the parameters for Tape Sort C, gives the octal locations for the first character of each parameter field, and describes the specific parameter configurations (giving options where possible) to be entered in memory.

NOTE: In Table 4-3, "T" is used to designate a tape control, "t" specifies a tape drive attached to the control unit, and "x", unless otherwise specified, represents a decimal digit.

Table 4-3. Tape Sort C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration
Tape Unit Specifications	1-2	2477	Address of primary input tape Address of alternate input tape (optional)	Tt, CΔ, or ΔΔ* Tt, xΔ, or ΔΔ
	3-4	2501		

## SECTION IV. TAPE SORT C OPERATING CONSIDERATIONS

Table 4-3 (cont). Tape Sort C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration				
Tape Unit Specifications (cont)	5-6	2503	Address of first work tape (final output tape)	Tt				
	7-8	2505	Address of second work tape	Tt				
	9-10	2507	Address of third work tape (optional)	Tt or ΔΔ				
	11-12	2511	Address of fourth work tape (optional)	Tt or ΔΔ				
	13-14	2513	Address of fifth work tape (optional)	Tt or ΔΔ				
	15-16	2515	Address of merge work tape Tt - the address of any tape not previously specified in characters 5-14 ΔR - the primary input tape is rewound and demounted, and the newly mounted tape on that drive acts as the merge work tape	Tt or Δ R				
	17-18	2517	Total number of input reels	xx				
	19	2521	Input tape direction B - read input tape backward Δ - read input forward	B or Δ				
Item and Record Sizes	20-23	2522	Number of characters per item	xxxx				
	24-27	2526	Number of items per input record	xxxx				
	28-31	2532	Number of items per output record	xxxx				
	32	2536	Padding character (If used with even parity files, an IBM record mark "72 <sub>g</sub> " is inserted as the last character of each padding item added by the sort program.)	Any alphanumeric character				
Label Indicators	33-34	2537	Input label indicators (for card input label options see page 4-5):  <table style="margin-left: 40px; border: none;"> <tr> <td style="text-align: center;"><u>beginning</u></td> <td style="text-align: center;"><u>end of file</u></td> </tr> <tr> <td style="text-align: center;">of file</td> <td style="text-align: center;">end of file</td> </tr> </table> ΔΔ - no label tape mark Δ T - no label tape mark, trailer, and tape mark SS - std. label std. trailer SS - std. label tape mark, and tape mark SS - std. label tape mark, std. trailer, and tape mark SS - std. label tape mark, std. trailer, and tape mark	<u>beginning</u>	<u>end of file</u>	of file	end of file	ΔΔ, Δ T, SS, LL, NΔ, or NT (When specifying SS, the user should have one of the proper file label combinations given under "Label Indicators"; the labels in the given combinations can not be interchanged.)
<u>beginning</u>	<u>end of file</u>							
of file	end of file							

## SECTION IV. TAPE SORT C OPERATING CONSIDERATIONS

Table 4-3 (cont). Tape Sort C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration
Label Indicators (cont)	33-34	2537	beginning of file            end of file LL - std. (120- tape mark, std. char.) (120-char.) IBM IBM (1410/ (1410/7010) 7010) label trailer, and tape and tape mark mark NΔ - non-std. tape mark label NT - non-std. tape mark, label trailer, and tape mark	
	35		Not Used	
Banner characters	36	2542	Banner characters: <u>input</u>	Δ or 1
	37	2543	<u>output</u> Δ - banner character 1 - no banner character	Δ or 1
Parity and Gap Size Indicators	38	2544	Input parity indicator: E - even parity Δ - odd parity	E or Δ
	39	2545	Output parity and gap-size indicator: L - odd parity, 3/4" gaps E - even parity, 3/4" gaps Δ - odd parity, short gaps	L, E, or Δ
Error Options	40-41	2546	Error options ΔΔ - check PΔ - print the record and check PD - print and delete the record (presort and last-pass only)	ΔΔ, PΔ, or PD
Printer Control Specifications	42	2550	Printer control unit: x - address of the printer control unit Δ - no printouts	x or Δ
	43	2551	Not used	
Merge Segment Halt	44	2552	Option to stop after the merge segment: 1 - stop after the merge to change the density setting for the final output tape Δ - do not stop for the density change	1 or Δ

## SECTION IV. TAPE SORT C OPERATING CONSIDERATIONS

Table 4-3 (cont). Tape Sort C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration
Highest Presort Address	45-50	2553	Highest address available to the presort. (If the address is specified in decimal, leading zeros must be included when necessary.)	xxxxxxx or an x for character 50, where x represents a number of 4K modules (leading blanks must be included)
Collating Sequence Translation	51-52	2561	Collating Sequence: TT - output in IBM collating sequence $\Delta\Delta$ - output in Honeywell collating sequence	TT or $\Delta\Delta$
	53	2563	Ascending or descending sequenced output: $\Delta$ - final output in ascending sequence D - final output in descending sequence	$\Delta$ or D
Own-Coding	54-59	2564	Address of presort header-trailer own-coding (leading zeros must be specified)	xxxxxxx or blank if not used
	60-65	2572	Address of presort item-by-item own-coding (leading zeros must be specified)	xxxxxxx or blank if not used
	66-71	2600	Address of last-pass header-trailer own-coding (leading zeros must be specified)	xxxxxxx or blank if not used
	72-77	2606	Address of last-pass item-by-item own-coding (leading zeros must be specified)	xxxxxxx or blank if not used
	78	2614	Not used	
Parameter Card Identification	79-80	2615	Parameter card identification: S2 - used only if parameters are entered from the card reader	S2 or not used
Key Field Specifications	81-84	2617	1st key field position (position of the leftmost, high-order character in the major key field)	xxxx
	85-86	2623	Number of characters in first key field	xx
	87-134		Similar configurations for key fields 2-9 (see page 2-9)	
	135-138	2705	10th key field position (If less than ten key fields are used, the remaining characters in the key field area must be blank.)	xxxx or blank if not used
	139-140	2711	Number of characters in the 10th key field	xx or blank if not used

## SECTION IV. TAPE SORT C OPERATING CONSIDERATIONS

Table 4-3 (cont). Tape Sort C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration
File Name Check	141-150	2713	Input file name verification	An alphanumeric name of up to ten characters, or a blank field if the verification is not desired
	151-160	2725	Output file name change option	An alphanumeric name of up to ten characters, or a blank field if the option is not desired
Loader-Monitor Information	161-166	2737	Last-pass own-coding program name	xxxxxx or blank if not used
	167-168	2745	Last-pass own-coding segment name	xx or blank if not used
	169	2747	Search direction for last-pass own-coding program: B - search forward C - search backward Δ - last-pass own-coding has already been loaded or is not used	B, C, or Δ
	170-175	2750	Next program name (these characters are blank if Tape Sort C is not followed by automatic loading of another program.)	xxxxxx or blank if not used
	176-177	2756	Next program segment name	xx or blank if not used
	178	2760	Search direction for next program: B - search forward C - search backward Δ - Tape Sort C not followed by another program	B, C, or Δ

\*To designate the input media, parameter characters 1 through 4 are specified by one of the following combinations.

<u>characters</u>	<u>tape input</u>	<u>card input</u>	<u>own-coding input</u>
1-2	Tt	CΔ	ΔΔ
3-4	Tt or ΔΔ	xΔ	ΔΔ

(x represents the card reader control address)

SECTION V  
TAPE SORT C OPERATING PROCEDURES

PRELIMINARY CONSIDERATIONS

The operator should be familiar with the general operating characteristics of the Series 200 equipment to be used, as explained in the Honeywell Series 200 Equipment Operators' Manual (Model 200), Order Number 040. Depending upon how the Tape Sort C program is stored, the operator should be familiar with the necessary loading procedures according to the Loader-Monitor to be used. Information concerning the operating procedures for the Loader-Monitor to be used can be found in the appropriate Honeywell Information Bulletin mentioned in the foreword of this manual.

LOADING TAPE SORT C FROM A BRT

Suggested methods for calling and loading Tape Sort C are given below, and in each instance given, the Tape Loader-Monitor used is considered to have been loaded previously.

Loading Tape Sort C from a BRT with Parameters on Cards

If the program executed before the sort did not include instructions for loading the parameters and calling Tape Sort C, the call to load the parameters and program can be issued from the card reader via a Console Call card. The Console Call card and the parameter cards are placed in the card reader input hopper as shown in Figure 4-1 on page 4-4. The operator should then perform the steps listed below.

1. Verify the following:
  - a. The Loader-Monitor is ready to accept a console call.
  - b. All peripheral devices to be used are cycled-up.
  - c. Sort input and work tapes are mounted on the appropriate drives according to parameters (see page 4-6). If the input to be sorted is on cards, place the cards in the card reader input hopper (see Figure 4-1 on page 4-4) and cycle-up the card reader.
2. Press RUN on the control panel or console. (The Loader-Monitor reads the Console Call card which calls sort segment 00 to read in the parameter cards and begin sort program loading and execution. If a check should occur during the loading or running of the program, it can be detected and corrected as explained on page 5-6).

Loading Tape Sort C from a BRT by Programmed Call

If there are programs preceding Tape Sort C on the BRT, the program executed before the sort program can enter the parameters and initiate a call for the Loader-Monitor to load Tape Sort C. If this method is used, no operator action is required to load the sort program.

The operator should have the sort input and work tapes mounted on their appropriate drives when the programmed call is executed. If card input is used, the input deck must be placed in the card reader input hopper (see Figure 5-1), and the reader must be cycled-up.

NOTE: If a check should occur during the loading or running of the Tape Sort C program, see "Tape Sort C Checkpoints" starting on page 5-6 to determine the cause and necessary corrective action.

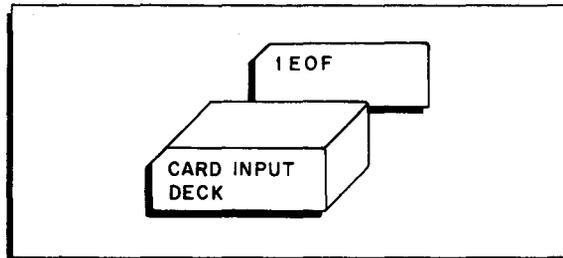


Figure 5-1. Card Input to Tape Sort C

#### LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK (BRD)

Either Card Loader-Monitor B or the card loading option of Floating Tape Loader-Monitor C may be used to call and load Tape Sort C; the suggested methods given below are used for calling and loading any object program deck. Use of the card loading option of Floating Tape Loader-Monitor C requires the insertion of an appropriately designated Console Call card before segment 00 in the Tape Sort C program deck; a Console Call card is not required when using Card Loader-Monitor B.

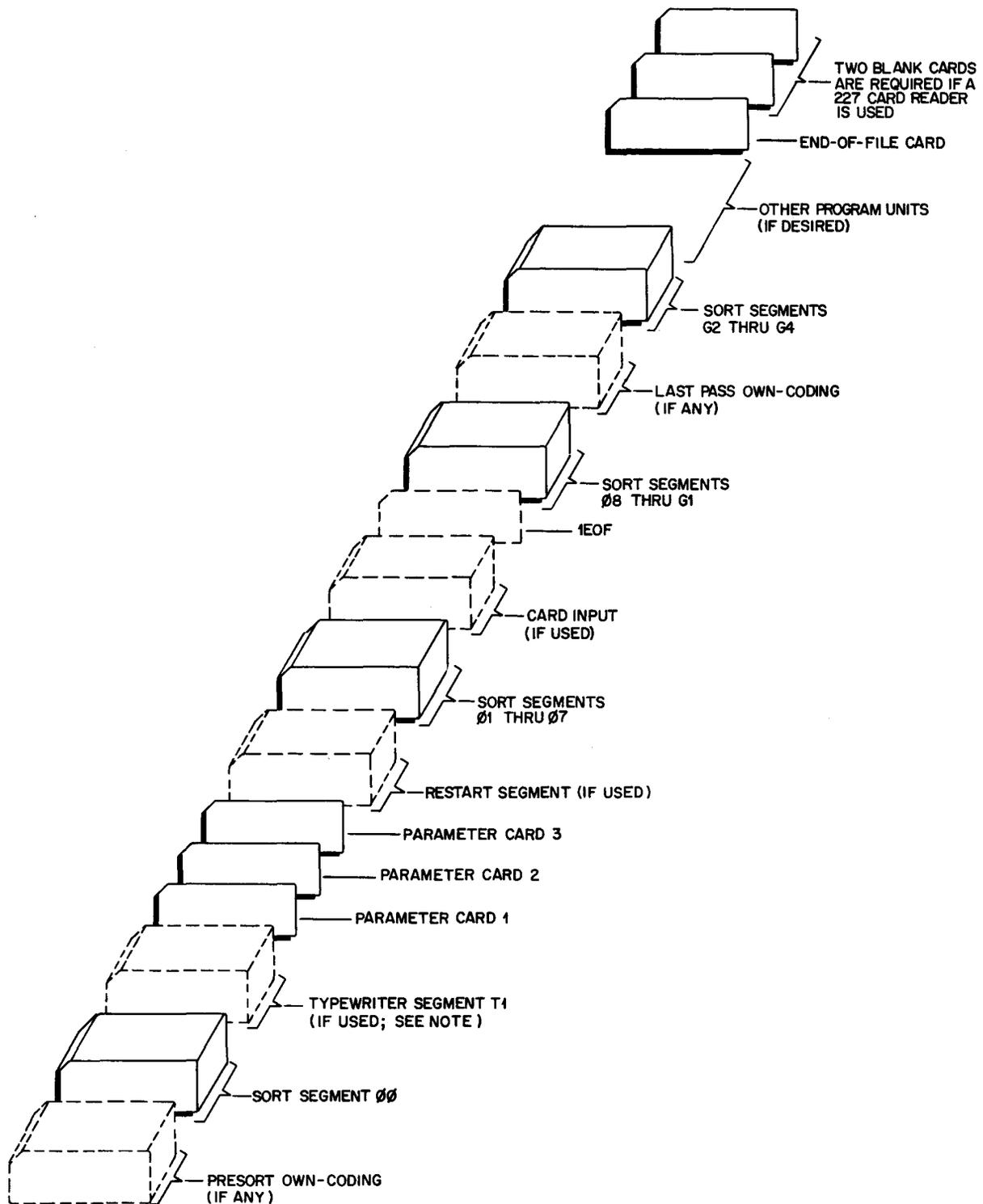
#### Loading Tape Sort C Parameters and Program from Cards

If the program executed before Tape Sort C did not enter the sort parameters, the parameters are punched on three cards and inserted between segment 00 (or T1 if included) and the next program segment in the object deck.<sup>1</sup> A suggested format for the object program deck with the above configuration is shown in Figure 5-2.

Once the applicable Loader-Monitor is loaded and awaiting a console call, the operator should perform the steps listed below.

1. Verify the following:
  - a. The Loader-Monitor is ready to accept a console call. (A Console Call card is the first card in the deck if the Floating Tape Loader-Monitor is used.)

<sup>1</sup>The parameters are placed after the end-of-segment card for 00, which contains an X4 (M) punch in column 1, or after the end-of-segment card for T1.



NOTE: Only when a typewriter is used must the typewriter routines be included in the object program deck, as prescribed in Table 4-1 on page 4-1.

Figure 5-2. Suggested Object Program Deck Format for Loading Tape Sort C with Parameter Cards

- b. All peripheral devices to be used are cycled-up.
  - c. Sort input and work tapes are mounted on the appropriate drives according to parameters (see parameters on page 4-6).
2. Press RUN. (The Loader-Monitor reads the 00 segment header card when called and proceeds with loading and starting of the Tape Sort C program.)

#### Loading Tape Sort C from an Object Program Deck by Programmed Call

When there are programs preceding Tape Sort C in the object run deck, it is assumed that the program executed before the sort program will enter parameters and initiate a call for the Loader Monitor to load Tape Sort C. If a check should occur during the loading or running of the sort program, see "Tape Sort C Checkpoint Identification and Corrective Action," starting on page 5-6.

If the programmed call method is used to call Tape Sort C, a suggested object program deck format is shown in Figure 5-3.

#### CHECK AND CHECKPOINT PROCEDURES

As explained previously, a check can be either a machine halt or a program loop caused by a "busy" test of the console typewriter. A check is accompanied by a descriptive message to the operator notifying him that the program is in a particular status, requesting specific entries, or designating an error condition to be corrected if possible. The message medium may be either a control panel or a console typewriter (Type 220-2 or -3), whichever is used at the installation. Therefore, the sort program must determine whether a typed message or a machine halt is to be coincidental with a check. Tape Sort C decides the method of presenting check messages by interrogating octal location 233 of the Loader-Monitor communication area for the presence of a word mark. If the location is word-marked, messages are typed out; if a word mark is not found, messages are coded in the A- and B-address registers.

For each check message, the operator can determine the cause and necessary corrective action by relating the message to its corresponding checkpoint in Table 5-1. If the checkpoint indicates that the sort processing can be continued, the continuation procedure is dependent upon the method used to present the check message, i. e., if the message was typed, the program loops on a "busy" test until the operator corrects the situation and types "G"; or, if the message was coded in the A- and B-address registers, the program halts until the situation is corrected and the operator presses RUN.

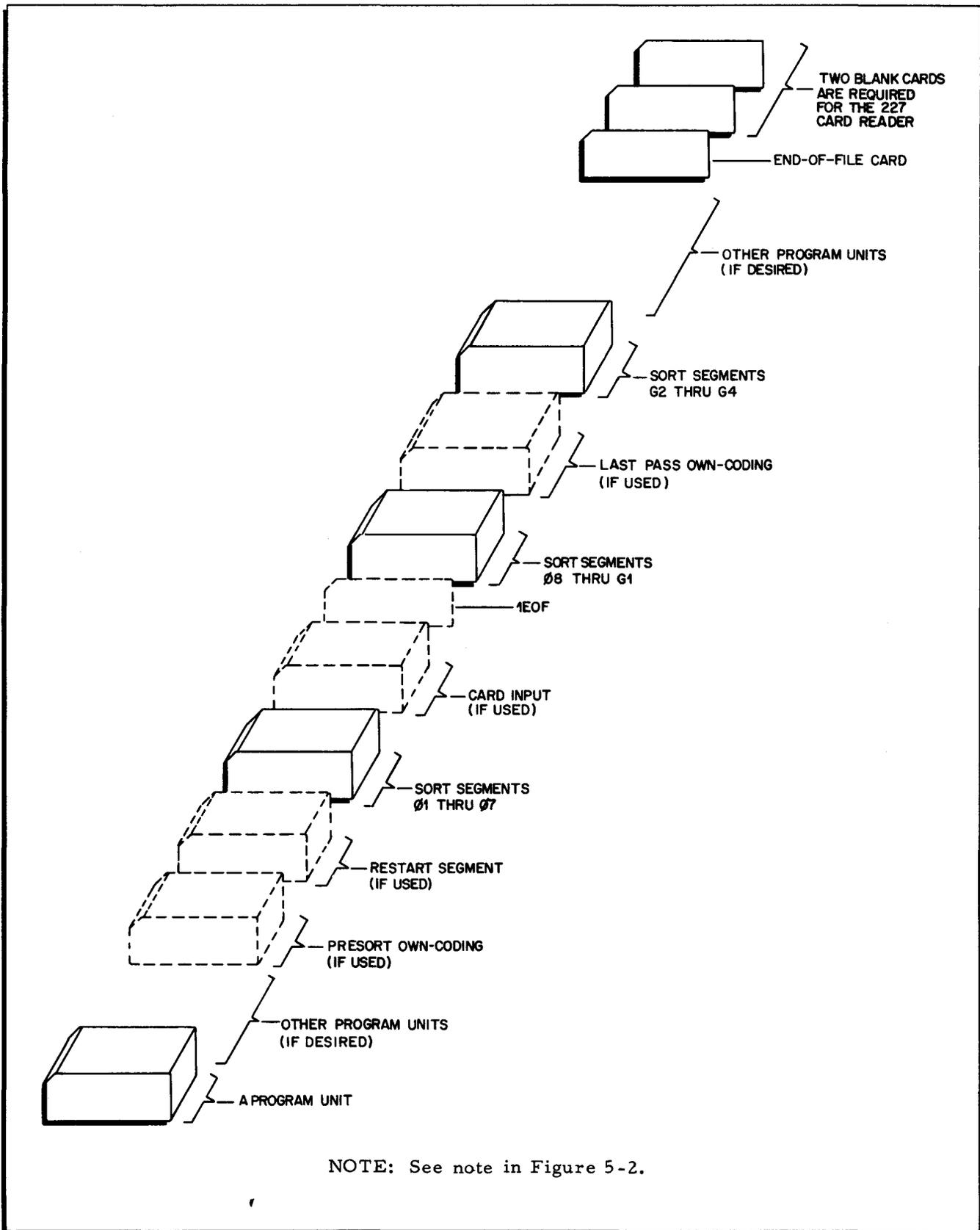


Figure 5-3. Suggested Object Program Deck Format for Loading Tape Sort C without Parameter Cards

The checkpoint may designate that the sort cannot be continued and must rerun. In this case, the operator can either attempt the restart procedure if a peripheral malfunction has occurred, as explained on page 5-8, or prepare the Loader-Monitor for a console call and rerun the program. If a typed message indicates that the sort program must be rerun, typing a response entry of "M" automatically returns control to the Loader-Monitor preparing it for a console call; if any other response is typed, the message "TERMINATE RUN" is typed, indicating the request for an "M" entry. If the rerun message occurred with a machine halt, the operator can enter octal 126 into the memory address register and press RUN to prepare the Loader-Monitor for a console call.

### Check Message Coding

Table 5-1 lists the checkpoints by number and indicates the check message whether contained in the A- and B-address registers (e.g., A: 00000 B: 04000) or typed out (e.g., :SEQUENCE ER). The specific segment(s) involved are listed beside the probable cause of the check and its possible corrective action.

Check messages in most instances contain coded characters to make them more concise and meaningful. When messages are typed, they may include some of the following code characters: RD - read, WR - write, ER - error, TP - tape, CD - card, TT - control unit address, and D - tape drive number. The check messages as represented in the A- and B-address registers are divided into two categories for easy identification: (1) peripheral errors and (2) non-peripheral errors. When a particular check specifies a peripheral error, the B-address register contains a "0" in the fourth octal digit (from the right). With a non-peripheral error, the B-address register contains a "4" in the fourth octal digit. These configurations may include the following: "cu" for control unit address, "d" for tape drive number, and "bbbb" for the buffer address. The buffer address is the address of the first (leftmost) location of the input buffer.

### RESTART PROCEDURES

The restart capability of Tape Sort C allows the operator to stop the sort at any time after presort specialization and to continue the processing sequence whenever desired by calling and executing the restart segment.

When a restart situation occurs, the restart is initiated in either of two ways. One method is to enter octal 142 (referencing Loader-Monitor fixed-start 3) into the memory address register and activate processing, causing a branch to a routine in the sort program which will call, load, and begin executing the restart segment, AADS2ΔRS.<sup>1</sup> Another method requires entering

---

<sup>1</sup>If assembled in four-character address mode, the sort program and segment name is AADS2FRS.

SECTION V. TAPE SORT C OPERATING PROCEDURES

---

octal 126 (referencing Loader-Monitor fixed-start 0) into the memory address register and activating processing to prepare the Loader-Monitor for a console call. A console call can now be entered to load and begin executing the restart segment. With either method, the three-character address mode indicator must be set and the sector bits (bits 16 through 18) in the A-address register must be cleared to zeros before the RUN button is pressed to activate processing.

**NOTE:** Normally, to effect a restart when the sort program is on punched cards, place the restart segment RS directly in front of sort segment 01 and use the fixed-start 0 restart action, as described above.

Once a restart has been initiated, a check occurs (checkpoint 27), requesting the entry of the necessary characters for Table 5-2. If a control panel is used, the appropriate characters are entered from left to right, beginning at the address contained in the A-address register. Once the necessary characters have been entered, the RUN button is pressed to resume processing. If a console typewriter is being used, simply type the necessary characters; typing the fourth character activates the restart.

Table 5-1. Tape Sort C Checkpoint Identification and Corrective Action

Checkpoint No.	Check Message	Segment(s) Involved	Cause	Corrective Action
1	A: 05555 B: 05555 : S2 ID ER, CARD 1, COLS 79, 80	Segment 00	S2 not found in column 79-80 of parameter card 1	Correct parameter card 1 and rerun the sort; or set SENSE switch 1 ON and press RUN or type "G" to ignore the error and continue processing.
2	A: 00000 B: 04000 : SEQUENCE ER	Merge and Last Pass	Record out of sequence in merge; or record or data out of sequence in last pass	Rerun the sort.
3	A: 00001 B: 04000 : PARAM ER, 1-80	Presort	Invalid parameters 1-80	Correct the parameters and rerun the sort.
4	A: 00002 B: 04000 : FILE NAME ER	Presort	Incorrect input file name: standard header	Rerun sort with correct tape or parameter; or mount the correct tape and press RUN or type "G."
5	A: 00003 B: 04000 : DUP TAPE ADDRESS ER	Presort	Duplication of tape drive assignments	Correct parameters and rerun the sort.
6	A: 00004 B: 04000 : KEY ER	Presort	Overlap in key field	Correct parameters and rerun the sort.
7	A: 00005 B: 04000 : PARAM ER, 81-160	Presort	Invalid parameters 81-160	Correct parameters and rerun the sort.
8	A: 00006 B: 04000 : OWN CODING ER	Presort	Own-coding lies within the sort program	Correct parameters and rerun the sort.
9	A: 00007 B: 04000 : INSUFFICIENT MEMORY	Presort	Insufficient memory to sort	Correct parameters and rerun the sort.
10	A: 00010 B: 04000 : BLOCKING DOWN NECESSARY	Presort	Blocking down required	To accept reduced blocking, set SENSE switch 1 ON and press RUN or type "G." If the check recurs, the record size is too large for the memory available.
11	A: 00001 B: 0cu0d : TT D HEADER NOT FOUND	Presort	1HDR record missing	Reposition the input tape and press RUN or type "G"; or rerun the sort.
12	A: 00002 B: 00t0d : MISPOSITIONED TAPE (merge) : PARAM REC NOT FOUND (last pass)	Merge and Last Pass  Merge and Last Pass	Tapes are mispositioned	Rerun the sort.
13	A: 00004 B: 0cu0d : TT D CHANGE OUTPUT DENSITY/ ADDRESS	Last Pass	To change tape density or to reassign the logical address of work tape 1 to an unused drive	When output tape is ready, press RUN or type "G."

SECTION V. TAPE SORT C OPERATING PROCEDURES

Table 5-1 (cont). Tape Sort C Checkpoint Identification and Corrective Action

Checkpoint No.	Check Message	Segment(s) Involved	Cause	Corrective Action
14	A: 00000 B: 0cu1l : TT CD RD ER□	Segment 00 and Presort	Hole-count error or illegal punch error on parameter card or card input	Correct the card, refeed it, and press RUN or type "G"; or rerun the sort.
15	A: bbbbb B: 0cu1d : TT D RD ER□	All segments	Read error: data record (64 attempts have been made to reread the record)	To try to reread the record, set SENSE switches 1 and 2 OFF and press RUN or type "G."  To delete the record, during presort and last pass only, set SENSE switch 2 ON and press RUN or type "G."  If the error recurs and the record should not be deleted, rerun the sort.
16	A: bbbbb B: 0cu2d : TT D WR ER□	All segments	Write error (a routine to backspace, erase, and re-write has been executed nine times)	For nine more attempts to write, press RUN or type "G."
17	A: 00000 B: 0cu3d : TT D END TP□	Presort and Merge	End of tape	Rerun the sort.
18	A: 00001 B: 0cu3d : TT D MOUNT NEXT OUTPUT TP□	Last Pass	End of tape	Mount a new output reel and press RUN or type "G."
19	A: bbbbb B: 0cu4d : RECORD LENGTH ER□	Presort	Data record of incorrect length	Press RUN or type "G." If check recurs, the data record is of the wrong length. To drop the record, set SENSE switch 2 ON and press RUN or type "G"; otherwise, the sort must be rerun.
20	A: bbbbb B: 0cu5d : TT D HDR RD ER□	Presort and Last Pass	Read error: header record	To reread the record, press RUN or type "G."  To process the record, set SENSE switch 1 ON and press RUN or type "G."
21	A: bbbbb B: 0cu7d : TT D RD ER LABEL□	Presort	Read error: trailer record	To reread the record, press RUN or type "G."  To process the record, set SENSE switch 1 ON and press RUN or type "G."
22	A: bbbbb B: 0cu6d : TT D LABEL LENGTH ER□	Presort	Header or trailer record of incorrect length (standard header or trailer ≠ 80 or 120 characters, or non-standard header or trailer exceeds record length)	To process the record, set SENSE switch 1 ON and press RUN or type "G." (This action supplies an 80 or 120 character label. If a non-standard label was specified, this action supplies a record equal to the size of an input buffer.)  If the record is not to be processed, the sort must be rerun.
23	A: ddddd B: 040xl : ITEM COUNT ER□	Last Pass	Number of items in presort differs from number of items in the last pass	Rerun the sort; press RUN or type "G" to effect "end-of-job." Presort is high if x = 0; and presort is low if x = 1. ddddd = amount of discrepancy (decimal).
24	A: 01111 B: 0cu0d : TT D MOUNT MERGE WORK TAPE	Presort	Check for input tape rewind	When new tape is mounted or dialed as the merge work tape, press RUN or type "G."
25	A: 02222 B: 0cu0d : TT D MOUNT NEXT INPUT TAPE□	Presort	Check for multi-reel input	Mount the new input tape on the drive specified, "d", and press RUN or type "G." (This check will occur whenever there is more than one input reel specified and no alternate input drive is designated. If an alternate input drive is designated, the program will not cause a check but will stall until the next reel is mounted.)
26	A: 07777 B: 0cu0d : END OF SORT xxxxxxxx ITEMS xxx : PADDING xxxxx DELETED□ : TT D END OF JOB□	Last Pass	End of job	With machine halt: the B-address register contains the address of the final output tape (work tape 1) press RUN to revert control to Loader-Monitor general return address.  With typed message: "D" is the address of the final output tape (work tape 1); type "G" or "M" to revert control to the Loader-Monitor general return address.
27	A: aaaaa B: 07700 : ENTER TCU/DR/MCODE/PCODE□	Restart	The following parameters must be entered: TCU - control unit address of work tape 2; DR - drive number of work tape 2; MCODE - restart point, M or Δ (M if sort is currently in the merge); PCODE - positioning of work tape 1, P or Δ (P if the operator is to go position work tape 1).	Enter the required parameters in the following order: TCU, DR, MCODE, and PCODE. If entries are made at the control panel the leftmost character is entered in location aaaaa, and the remaining parameters are entered in successive locations. Once the parameters are entered, press RUN. If a typewriter is used, simply type the four parameter values; typing the fourth value initiates the restart.
28	A: 04444 B: 07700 : RESTART FAILED□	Restart	Restart failed	Rerun the sort.
29	: TERMINATE RUN□	All segments	When a typewriter is used and a response to an uncorrectable error has been made.	No corrective action. Type "M" to revert control to the Loader-Monitor general return address.

Restart Precautions

Restarts do not restore any own-coding other than that which is called in by the last pass. Presort or last-pass own-coding loaded prior to the Tape Sort C program is not restored.

Table 5-2. Restart Entries

Character	Content	Explanation
1-2	pd	Peripheral control unit address (p) and drive number (d) for work tape 2.
3	M or $\Delta$	If the sort is stopped during the merge, "M" should be specified to prevent restarting at the beginning of the presort rather than the beginning of the current merge phase. "M" must not be entered if the sort was stopped during the presort or last pass.
4	P or $\Delta$	"P" is designated if work tape 1 (the final output tape) was not rewound at the beginning of the sort; otherwise, this character is blank. When "P" is specified, the user must reposition work tape 1 as it was when the sort began.

When the input to the sort is such that no string is written on work tape one, the presort prepares to bypass merge processing, and only last-pass processing is required to complete sorting the file. In this case, a restart during the merge will produce unspecified results and should be avoided.

Tape Positioning for Restarting

Tape Sort C will always resume execution at the beginning of the phase in which the halt or interruption occurred. When restarting during the presort, the restart point is at the beginning of the presort. Restarting in the merge causes execution to resume at the start of the current phase of the merge; if the print option was specified, a printout will appear when merge processing has begun (see page 5-10). A restart during the last pass resumes execution at the beginning of the last pass.

NOTE: A restart cannot be effected if the sort is stopped between segment M3 of the merge and segment G1 of the last pass.

If the sort is to be interrupted for a considerable time and all work tapes are to be de-mounted, the tapes must be mounted (before the restart is initiated) on the same logical drives from which they were removed. If locations 142 through 145 of the Loader-Monitor communication area were modified during the delay, such that fixed-start 3 could not be used for restarting, a console call can be entered to initiate the restart. The console call must be entered for

segment AADS2ΔRS.<sup>1</sup> Work tape 2 must be mounted on the proper logical drive, for work tape 2 contains the information necessary for restarting.

#### POSITIONING THE INPUT TAPE

If the restart is returning to the beginning of the presort, the input tape must be mounted and positioned as it was at the beginning of the sort. Normally, positioning requires rewinding the tape; however, if the input was positioned beyond the beginning of the tape, it is necessary for the operator to reposition it to the actual starting point.

#### POSITIONING OF WORK TAPE 1

If work tape 1 (the final output tape) was positioned at the beginning of tape when the sort run began, the operator should enter a blank as the fourth control character (see Table 5-2) to cause the restart to rewind the tape from the beginning. If work tape 1 was not positioned at the beginning of tape when the sort began, the operator must enter "P" as the fourth control character. The "P" entry inhibits the rewinding of work tape 1 and assumes that the operator has repositioned the tape exactly as it was at the beginning of the sort.

#### Search Direction for Restart Segment

The sort program normally searches forward for its next segment; but, when the restart is called using the address in Loader-Monitor fixed-start 3, the program searches backwards for the restart segment. The restart segment, once loaded, signals the sort program to return to a forward search direction. Note that searching backward is not possible when using Tape Sort C in binary run card format, therefore, fixed-start 3 can only be used when Tape Sort C is on a BRT.

#### Restart Message

If a typewriter or printer is available and has been specified, the following message is listed at the end of the presort:

END OF PRESORT, RESTART MSG pdM<sup>2</sup>

After this message appears, the value "M" should always be entered as the third control character when restarting in the merge (see Table 5-2).

#### Restart Failure

If the restart was attempted before the sort had established any restart information, it will fail; checkpoint 29 occurs, and the sort must be rerun.

---

<sup>1</sup>See footnote on page 5-7.

<sup>2</sup>Where p = peripheral control unit address, d = drive number, and "M" indicates that merge processing has begun.

## SECTION VI

### COLLATE C BASIC CONCEPTS

#### INTRODUCTION

Collate C is a general-purpose program which functions in the SERIES 200/OPERATING SYSTEM - MOD 1. The Collate C program is designed to combine from two to five previously sequenced (sorted) files consisting of fixed-length items of information in identical format, thereby producing a single sequenced output file - the collated file.

The Collate C program performs a single-pass routine to compare items from each input file and to write the items in sequence (according to comparisons) on 1/2-inch magnetic tape. Each input file may be contained on one or more magnetic tape reels; however, all reels containing a given file are processed sequentially from a single tape unit. If it is desired to collate more than five input files, two or more collate runs are required.

Because Collate C is a general-purpose program, it can be adapted to a variety of data formats and equipment configurations. The information used to specialize the program for a particular collating application is entered at the beginning of the collate by means of parameters specified by the user. The prescribed functions to be performed by Collate C can be augmented by own-coding routines written by the user. These routines, which can be easily written using the Easycoder C symbolic language, are explained on page 7-6. Own-coding can be used to modify the header and trailer labels and to inspect, modify, add, and delete items during the collate process.

#### CHARACTERISTICS OF COLLATE C

Collate C is characterized by the following activities:

1. Combines two to five previously ordered files into a single ordered file;
2. Processes fixed-length items blocked a specified number per record;
3. Collates according to control information contained in up to ten collate key fields in each item;
4. Allows manual correction or removal of unreadable records;
5. Provides for collating sequence translation;
6. Allows label record changes;
7. Collates in ascending or descending sequence;
8. Provides for the inclusion of own-coding;
9. Allows execution in three- or four-character address mode; and
10. Operates with either control panel or console typewriter (Type 220-1, -2, or -3).

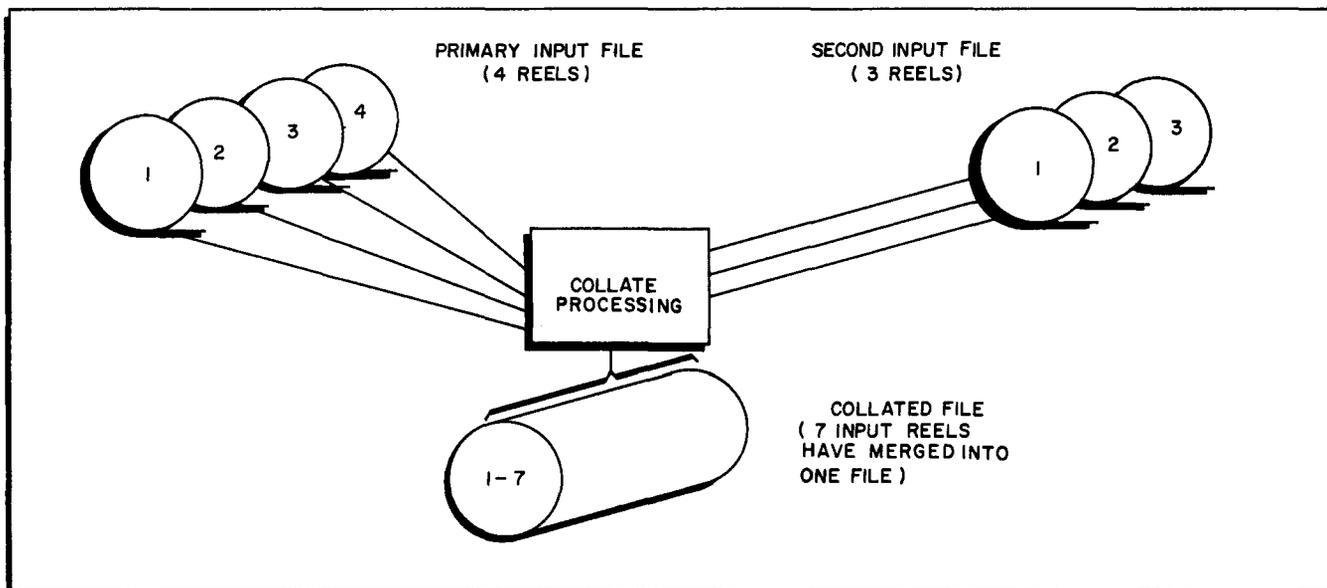


Figure 6-1. Sample Collate Process

#### Check and Checkpoint Terminology for Collate C

As defined for Tape Sort C (on page 5-6), a "checkpoint" in Collate C is a routine programmed to check or delay processing when an operator decision is required. A "check" is a term used to denote the occurrence of a checkpoint and is relayed to the operator through either the control panel (via a machine halt) or the console typewriter (via a typed message). All checkpoints associated with Collate C are listed in Table 9-1 (on page 9-6) and numbered to correspond to the appropriate operator action.

#### MACHINE REQUIREMENTS

Collate C requires the following equipment configuration:

1. A minimum main memory capacity of 8,192 character locations is required if the program is assembled and run in three-character address mode, but up to 32K locations can be used to advantage; similarly, with four-character address mode, a minimum of 12K locations are required, but up to 65K locations can be used to advantage.
2. A minimum of three tape drives (Type 204B) using 1/2-inch magnetic tape. Another drive may be specified as an alternate for multi-reel output; and up to three additional drives may be specified to increase the number of files to be combined.
3. An additional tape drive, or a card reader, for program loading.
4. The Advanced Programming Instructions feature (011).
5. Read-write channels two and three (RWC2 and RWC3); used for reading and writing, respectively.
6. Index registers one through four.

NOTE: A printer (Type 222 or 206) may be used for printouts of unreadable records, parameters, etc.

DATA REQUIREMENTS

The Collate C program processes fixed-length items which can be blocked a specified number per record. Note that the number of items per record designated for output records need not be the same as the number specified for input records. The following formulas can be used to determine the maximum and optimum record sizes for both input and output:

$$\text{Maximum record length} = \frac{\text{HMA} - \text{MON} - \text{PROG} - (\text{NTC}) (\text{NKC})}{\text{NTC} + 2}$$

$$\text{Optimum record length} = \frac{\text{HMA} - \text{MON} - \text{PROG} - (\text{NT}) (2\text{NKC}) - \text{NKC}}{2\text{NTC}}$$

NT	Number of input files.
HMA	Highest memory address available to collate (parameter characters 49-54).
TRANS	Translation requirements are 100 locations if parameter character 55 is a "T"; if character 55 is blank, no locations are required.
MON	Monitor requirements are 1,340 locations.
ALT	If an alternate output tape is specified, 150 locations are required; if not, the 150 locations are not needed.
PROG	Collate program requirements are 3,550 + TRANS + ALT + ADM + TYP locations.
NTC	Number of input files plus one.
NKC	Number of key field characters in an item. (Total collate key length.)
ADM	3-character addressing (0 locations) 4-character addressing (1,000 locations)
TYP	3-character addressing (455 locations) 4-character addressing (520 locations)

COLLATE KEYS AND KEY FIELDS

The files to be collated are made up of items that are identified and sequentially ordered by means of collate keys. A collate key consists of those key fields within an item which are designated as the basis for developing a sequential relationship among the items from each file to be collated. Each input item may contain up to ten key fields with up to 99 characters per field. These key fields can be located anywhere within an item, provided that they are in the same position within each item.

The location of each key field is specified by the user in parameter characters 81 through 140. If more than one key field is used, all fields must be specified in decreasing order of importance; i. e., the key field of major importance is specified first.

Collate key fields can contain any octal configuration. Standard Honeywell collating sequence is normally used in the collate, but the user may produce his output in standard IBM collating sequence by specifying the appropriate parameter.

### MAGNETIC TAPE UNITS

Tape drives for 1/2-inch tape are used by the Collate C program. In systems equipped with a recording density option, the higher density offers higher collating efficiency.

### UNREADABLE INPUT RECORDS

In the event that an unreadable record occurs (the read-error check recurs after several attempts at rereading), the collate program will proceed according to the error option specified in the appropriate parameters.

### OPTIONAL PRINTOUTS

If a printer is to be used and is specified in parameters, the collate program automatically prints out a list of collate parameters (if loaded by the program), the input header label (if standard), and an end-of-collate message giving a count of the items processed, padding items, and items deleted.

### COLLATE PROCESS

The Collate C program compares the key fields of items from each input file and moves the item with the lowest key field configuration to an output buffer area where a specified number of collated items will be written in record form on an output tape.<sup>1</sup> The steps given below in conjunction with Figure 6-2 provide a summary of the Collate C process.

1. A record consisting of a number of items is read from each input file into an input buffer area in memory (see A of Figure 6-2).
2. The key fields of the leading item in each input record are rearranged (according to parameters) as they are moved to a key field storage area. (The groups of rearranged key fields are referred to as the collate keys for the items.)
3. The collate keys of the items are compared.
4. The item found to have the lowest collate key configuration is selected and moved to an output buffer area.
5. The Collate C program now compares the next item to be processed in the same record from which the item was moved to the output buffer. (Note the positioning of the arrow in B of Figure 6-2.)
6. Comparisons continue in the above manner for all items of the input files, until the files have been collated into one file.

---

<sup>1</sup>If input files are in ascending order, items will be collated according to the lowest key field configuration; input files in descending order are collated according to the highest key field configuration.

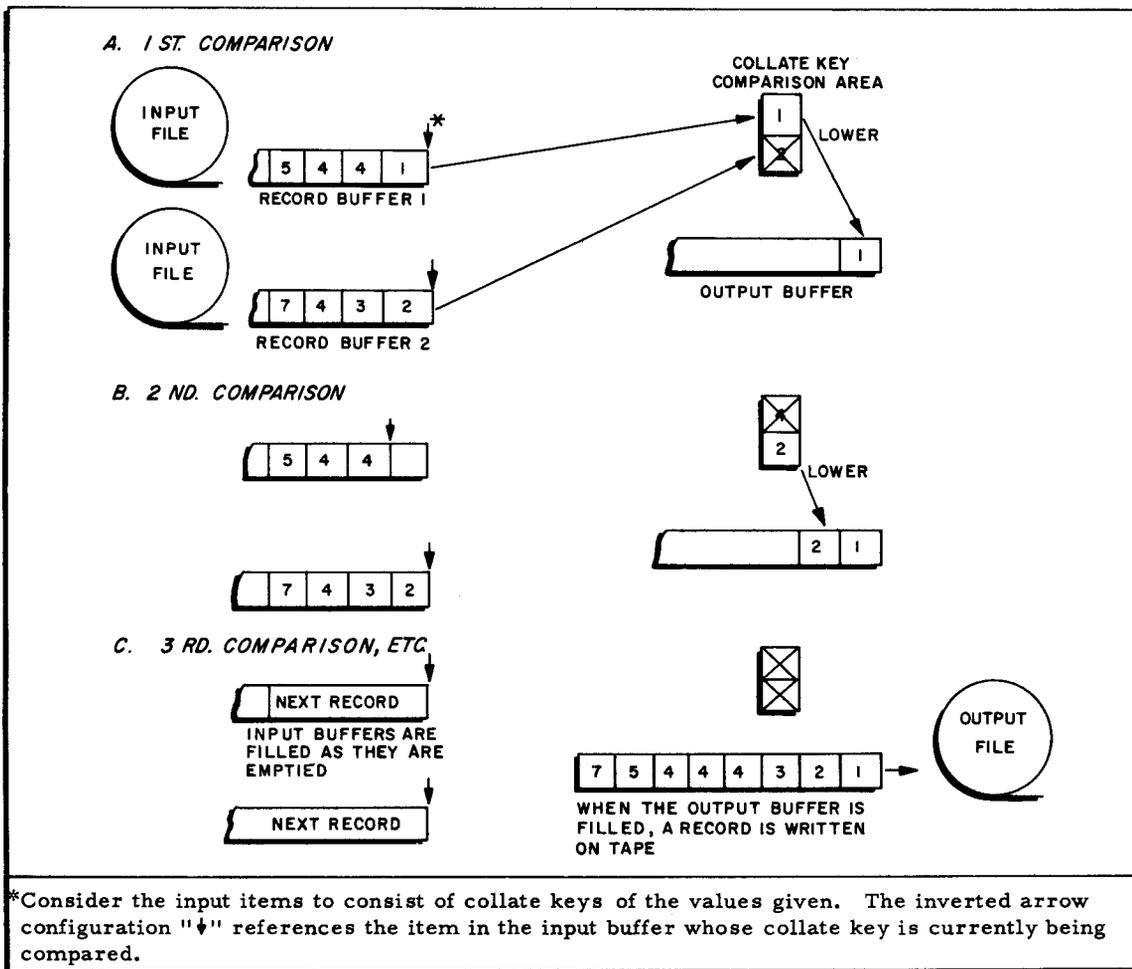
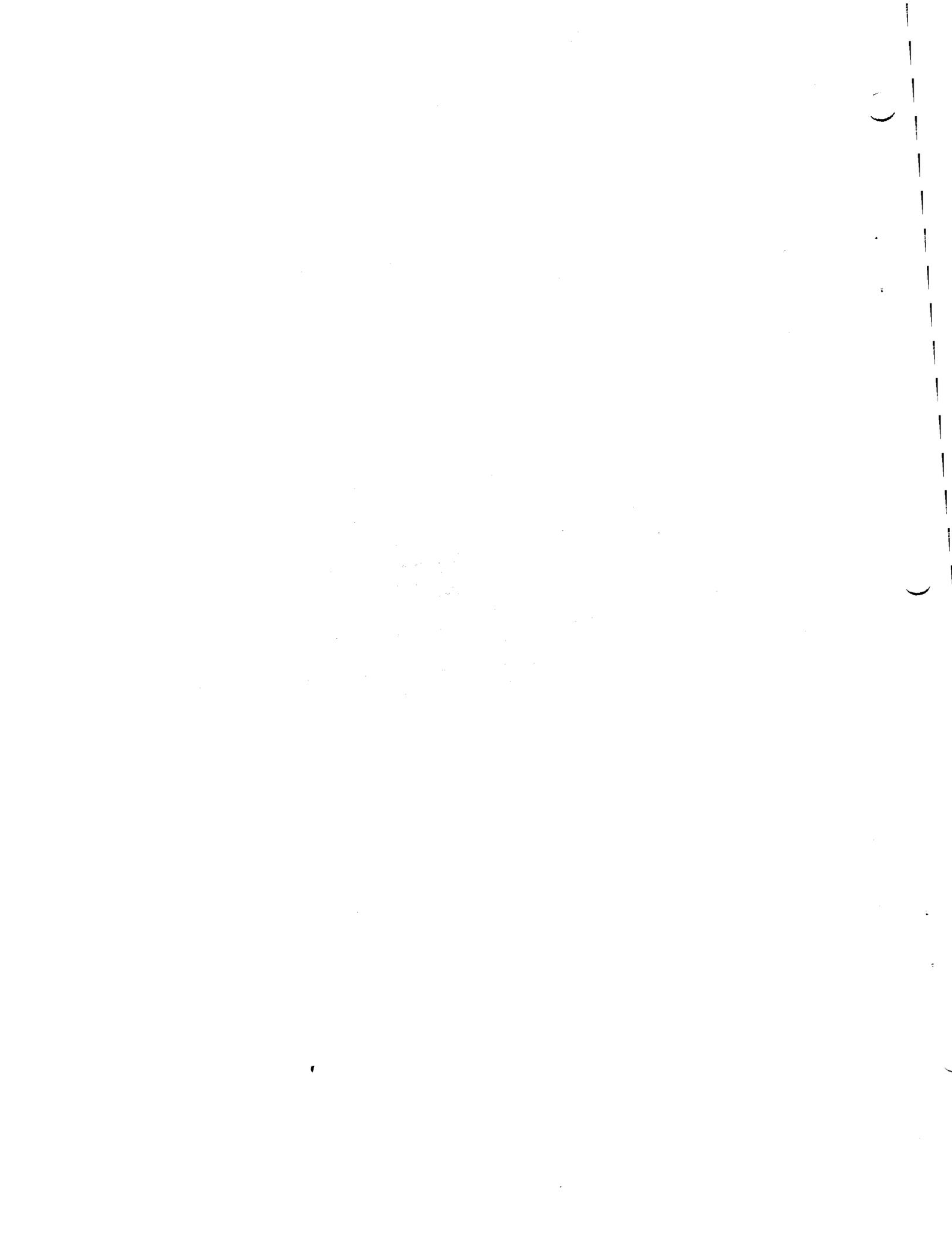


Figure 6-2. Collate Processing of Two Input Files



SECTION VII  
COLLATE C PROGRAM SPECIALIZATION

PARAMETER PREPARATION

The information which specializes Collate C for a particular application is described to the program by means of parameters designated by the user. Parameters may be entered in a variety of ways depending upon the method in which the collate program is being run. For instance, if the program is to be executed as a single run, the parameters can be punched on two parameter cards, as described on page 8-2, and read in from the card reader under control of collate segment 00. Another method of parameter entry, which is convenient when Collate C is to be run as one of a series of programs, is for the user to include coding in the program executed immediately before the collate program which will set up in memory the specified parameters to be used during collate processing and call in the first collate segment.

If collate parameters are entered in memory by the preceding program, the need for parameter card loading is eliminated, as the parameters will be in memory when the preceding program calls the Loader-Monitor to load Collate C. If the parameters are set up by the preceding program (and are therefore essentially fixed), any minor modifications needed for a particular run can be entered manually from the control panel when Collate C is loaded.<sup>1</sup>

Parameter information is summarized in Table 8-2 on page 8-4, and the parameter characters are loaded into memory starting at location 2, 477 (octal). The following paragraphs describe each parameter field.

NOTE: When numeric parameters are specified, leading zeros must be designated; for example, the field specifying the maximum number of characters per item (parameters 25 through 28) is designated 0100 if there is to be a maximum item size of 100 characters.

Tape Unit Specifications (Parameter Characters 1-24)

The Collate C program requires a minimum of three tape drives. Up to four additional drives may be used to increase the number of input files being collated and for multi-reel output. The user specifies the address and function of each tape drive in characters 1 through 24. Two characters are required to address a tape drive: the first specifies the tape control, and the second designates a particular drive attached to that control.

---

<sup>1</sup>Parameters may be loaded by any method available to the user at any time after the Loader-Monitor is loaded, but the parameters must be in correct format in the specified memory locations before the Collate C program is loaded.

Characters 1 and 2 specify the address (tape control designation followed by drive number) of the primary input file. Note that header and trailer labels for the output file will be taken from this file. Characters 3 and 4 designate the number of reels in the primary input file. Characters 5 and 6 designate the address of the second input file. Characters 7 and 8 designate the numbers of reels in the second input file.

Characters 9 and 10 designate the address of an optional third input file, or they are blank if a third input file is not used. Characters 11 and 12 designate the number of reels in the third file, or they are blank if a third input file is not used. Characters 13 and 14 specify the address of an optional fourth input file; these characters are blank if only two or three files are to be collated. Characters 15 and 16 designate the number of reels in the fourth file; these characters are blank if there is no fourth file. Characters 17 and 18 specify the address of an optional fifth input file; these characters are blank if only two, three, or four input files are to be collated. Characters 19 and 20 designate the number of input reels in the fifth file; these characters are blank if there is no fifth input file.

Characters 21 and 22 designate the address of the output tape drive. Characters 23 and 24 designate the address of an alternate tape drive; if an alternate output tape is not used, these characters are blank.

#### Item and Record Sizes (Parameter Characters 25-36)

Characters 25 through 28 specify the number of characters per item. The designated item size must remain fixed throughout the collate and cannot exceed the maximum record length as derived from the formula on page 6-3.

Characters 29 through 32 designate the number of items per input record (characters/input record ÷ characters/item = items/input record). All input files must have the same number of items per record. Characters 33 through 36 designate the number of items per output record. The record size limits for output records are the same as for input records, but input and output blocking need not be the same.

#### Padding (Parameter Character 37)

If the last data record of the output reel is lacking one or more items because of reblocking or own-coding, the Collate C program automatically uses the character specified in parameter 37 to pad this record, i. e., to add the proper number of padding items, consisting of padding characters, to fill the record.

NOTE: When padding items are added to an even parity, the last character of each padding item is an IBM record mark (octal 72).

Tape Labels (Parameter Characters 38-39)

Characters 38 and 39 specify tape labeling options as indicated in Table 7-1.

Table 7-1. Collate C Label Options

Characters 38 and 39	Beginning of File	End of File
ΔΔ	no label	tape mark
ΔT	no label	tape mark, trailer, and tape mark
ST	standard label	tape mark, standard trailer, and tape mark
SS*	standard label (with or without following tape mark)	standard trailer (with or without preceding tape mark) and tape mark
NΔ	non-standard label	tape mark
NT	non-standard label	tape mark, trailer, and tape mark
LL	standard (120-character) IBM 1410/7010 label and tape mark	tape mark, standard (120-character) IBM 1410/7010 trailer, and tape mark

\*If a tape mark appears at the beginning of file, one must precede the trailer at the end of a file; if a tape mark precedes a trailer, the final output will always have a tape mark following the trailer.

The output file will have the same type of labeling as the primary input file, unless the user formulates a change through own-coding. If standard labels and trailers are specified, the output tape header label will be identical to the header label of the primary input tape, except that the serial number of the output tape will be retained. The output trailer will be identical to the trailer of the last reel of the primary input file, except that record and item counts for the combined file will be established by the Collate C program. If non-standard labels are specified, the header label of the primary input file and the trailer record, if any, on the last reel of this file will be copied onto the final output reel.

Output Label Verification (Parameter Character 40)

Character 40 is a "1" if multi-reel output is expected and it is desired to have the collate program verify the date and retention cycle on all final output reels except the first (it is the user's responsibility to assure that the first final output tape can be written on).

For each output reel after the first, the tape is read forward, so that one record is read into memory. If the first five characters of the record are "1HDRA", the date and retention cycle are compared to those prescribed by the Loader-Monitor, and a check occurs if the label is in error (checkpoint 9).

Character 40 is blank if output label verification is not desired.

Banner Characters (Parameter Characters 41-42)

Characters 41 and 42 are used to specify the presence or absence of banner characters in input and/or output records. Character 41 is blank if the first character in each input record is any type of banner character, or specifies a "1" if banner characters are not present. Character 42 is blank if the first character of each output record is to contain the Honeywell standard banner character ( $56_8$ ), or specifies a "1" if output records are not to have banner characters.

Parity and Record Gap (Parameter Characters 43-44)

Character 43 is blank for odd parity input or specifies an "E" for even parity input. Character 44 is blank if the output is to be written with odd parity and short interrecord gaps, specifies an "L" if the output is to be written with odd parity and 3/4-inch interrecord gaps, or specifies an "E" if the output is to be written with even parity and 3/4-inch interrecord gaps.

Error Options (Parameter Characters 45-46)

In the event of an uncorrectable read error, Collate C proceeds according to the error option designated in characters 45 and 46. These characters are blank if it is desired to cause a check when an uncorrectable read error is encountered (checkpoint 17); specifying "PA" causes the record to be printed and checkpoint 17 to occur. "PD" can be designated to have the record printed and deleted from the file.

Should either check option be specified ("AA" or "PA"), the action taken by the operator at checkpoint 17 is explained in Table 9-1 on page 9-5.

Printer Control (Parameter Character 47)

Character 47 designates the address of the printer control unit if a printer is to be used during the collate program run. If a printer is not to be used, this character is blank.

If a printer is specified, Collate C will print out the following information during processing.

1. Parameter card images
2. Input header label image
3. Output reel item and record count: as an output reel is processed, a count is printed out excluding those items and records added or deleted through own-coding or dropped due to read errors. The following is an example of a printout from a collate requiring one output reel.

END OF REEL 1 0000000100 ITEMS 00100 RECORDS

4. End of collate message: when the collate is completed, an end of collate message is printed giving a count of the items processed and listing the number of padding items and items deleted or dropped due to read errors.

END OF COLLATE 000000100 ITEMS 000 PADDED 00000 DELETED

(The count of items above consists of the items processed and the items added by own-coding; it excludes the items deleted by own-coding, padding items, and items dropped due to read errors.)

NOTE: If a typewriter is used during the collate process, the following end-of-collate message is typed giving a count of the items processed and listing the address of the tape and drive containing the final output at the end of job.

```
END OF COLLATE 000000100 ITEMS 000 PADDED 00000
DELETED 00 6 OUTPUT TAPE E.O.J.
```

#### Highest Memory Address (Parameter Characters 49-54)

The highest memory address available to Collate C is specified in characters 49 through 54. The address is expressed as a decimal number with leading zeros (e. g., 012287), or a number of 4K memory modules with leading blanks (e. g.,  $\Delta\Delta\Delta\Delta\Delta 3$  for 12,287).

If Floating Tape Loader-Monitor C is used to load the Collate C program, the base address of the Loader-Monitor is compared to the highest address specified for the collate, and the collate program uses the lower address of the comparison as the highest address available for program execution.

#### Collating Sequence Translation (Parameter Character 55)

Collate C may be used with any collating sequence; and, if desired, the character code on the final output file may be different from that on the input files. The translation table provided with the collate program permits collating in IBM sequence, but by overlaying this 64-character translation table with a table of another desired code, the user can obtain another desired collating sequence. Use of the table to collate in IBM sequence implies that the input files are recorded in binary coded decimal (even parity) and that the Series 200 tape control is equipped with the 051 code compatibility feature.

Character 55 is blank when no sequence translation is required. This parameter configuration specifies the standard Honeywell collating sequence. Parameter 55 is a "T" when the translation table is used, and it is assumed that the input files are in IBM code and are to be collated in IBM sequence.

#### Ascending or Descending Sequenced Output (Parameter Character 56)

Character 56 is blank if all input files are in ascending sequence and the final output file is to be in ascending sequence. Character 56 is a "D" if all input files are in descending sequence and the final output is to be in descending sequence.

Rewind Option (Parameter Character 57)

Character 57 is blank if all tapes are to be rewound at the beginning of the collate or a "1" if the tapes are not to be rewound.

Own-Coding (Parameter Characters 58-63)

Own-coding can be used in conjunction with Collate C to perform additional data processing. Two own-coding exits can be specified in the collate program: one exit allows inspection and modification of header and trailer labels, while the other exit provides for additional processing on an item-by-item basis. When the item-by-item exit has been designated, program control first exits to own-coding after the collate program has been specialized, permitting the user to modify the program coding and to prepare for item processing; control subsequently exits to own-coding after each item of the file is processed, allowing items to be inspected and modified and enabling items to be added to and deleted from the file. Own-coding is further discussed on page 7-8.

Collate C own-coding may be loaded into any available locations of either higher or lower order than those containing the Collate C and Loader-Monitor programs. All own-coding addresses are expressed as decimal values. Characters 58 through 63 designate the own-coding address to which the Collate program will branch after reading a header or trailer label. These characters are blank if this option is not used. Characters 64 through 69 designate the address to which the program will branch (1) after Collate C has been specialized and (2) after each item has been placed in the output buffer. These characters are blank if this option is not used.

Number of Data Records Per Output Reel (Parameter Characters 70-74)

Characters 70 through 74 are blank if it is desired to fill an output reel with data records until the end of tape is sensed. If it is desired to write only a specified number of data records on each output reel, characters 70 through 74 specify this decimal number with leading zeros.

Parameter Card Identification (Parameter Characters 79-80)

If parameters are being entered through the card reader by collate segment 00, columns 79 and 80 of parameter card 1 must contain the configuration "C2" to identify the parameter cards being read. If parameters are entered by any other method, these characters are not specified.

Key Fields (Parameter Characters 81-140)

The specification of each collate key field requires six characters: four to specify the position of the high-order (leftmost) character of the key field in the item, and two to specify the number of characters in the field (see Table 7-2). Decimal values with leading zeros are used

in specifying the six characters. The position within the item of the high-order character of each key field is determined by counting the leftmost character in the item as 0001 and adding the number of positions needed to reach the high-order character of the specific key field. If fewer than ten collate key fields are specified, the unused characters must be blank.

Table 7-2. Parameter Characters for Collate Key Fields 1-10

Key Field Number	1	2	3	4	5	6	7	8	9	10
Key Field Position	81-84	87-90	93-96	99-102	105-108	111-114	117-120	123-126	129-132	135-138
Number of Characters in the Key Field	85-86	91-92	97-98	103-104	109-110	115-116	121-122	127-128	133-134	139-140

#### Input File Name and Reel Number Verification (Parameter Character 141)

If standard labels are used, the file name of each input reel within a file can be compared to the file name on the first reel. The reel number can also be verified to insure that all reels of a file are mounted in the correct sequence.

Character 141 is blank if it is not desired to verify the file name and reel number, or specifies an "F" if verification is desired.

#### Output File Name Change (Parameter Characters 142-151)

If standard labels are used, the file name of the output file may be changed without own-coding. The new output file name is specified in characters 142 through 151, and the collate program will make the change. These characters are blank if no change is desired.

#### Next Program Name (Parameter Characters 152-157)

Characters 152 through 157 designate the name of the program (if any) which is to follow Collate C. If these characters are specified, there is an automatic branch to the Loader-Monitor program to initiate a search for the requested program and segment. Characters 152 through 160 are blank if Collate C is not followed by automatic loading of another program.

NOTE: If these fields are blank, the collate program causes a check upon completion (checkpoint 25), where program control can be reverted to the Loader-Monitor for a console call by pressing RUN or typing "M."

#### Next Segment Name (Parameter Characters 158-159)

Characters 158 and 159 designate the name of the first segment within the next program to be loaded.

Search Direction for Next Program (Parameter Character 160)

Character 160 specifies the direction in which the Loader-Monitor should search for the next specified program on the program tape. A "B" designates a search forward, and a "C" designates a search backward.

TAPE POSITIONING

NOTE: The letters used to label the following paragraphs are also used to designate the corresponding diagrams in Figure 7-1 on page 7-8

Beginning of Collate

Input Files

- A. If all tapes are to be rewound at the beginning of the collate (parameter 57), each input tape may be positioned at any point. If the tapes are not to be rewound, they must be positioned before the header label or, if there is no header label, before the first data record.

Output Files

- B. The output tape may be positioned at any point on the tape. If rewind is not specified, the collate will begin writing at the point where the output tape is positioned. If rewind is specified, the collate will rewind the tape and write from the beginning of tape. There must be one valid record at the beginning of tape if the tape is rewound.

End of Collate

Input Files

- C. At the end of the collate, the last reel (or the only reel) of each input file will be positioned after the trailer record (after the tape mark if there is no trailer).

Output File

- D. The last reel of the output will be positioned between the trailer record (or tape mark if there is no trailer) and an end-of-recorded-information record (IERI).

COLLATE C OWN-CODING

Own-coding routines are written by the user for inclusion in the Collate C program. The presence of these routines is indicated to the Collate C program by means of parameter characters. Characters 58 through 63 designate a branch to a header-trailer own-coding address, where the user may inspect, modify, or replace the tape labels. Characters 64 through 69 designate a branch to an item-by-item own-coding address, where the user may inspect, modify, add, or delete items. These routines cannot change an item's size.

How to Use Collate C Own-Coding

The user should review the information presented in Section III for Tape Sort C own-coding, especially last-pass own-coding, for the preparation of Collate C own-coding routines and the registers referenced by such routines are identical to those described for Tape Sort C last-pass own-coding beginning on page 3-6.

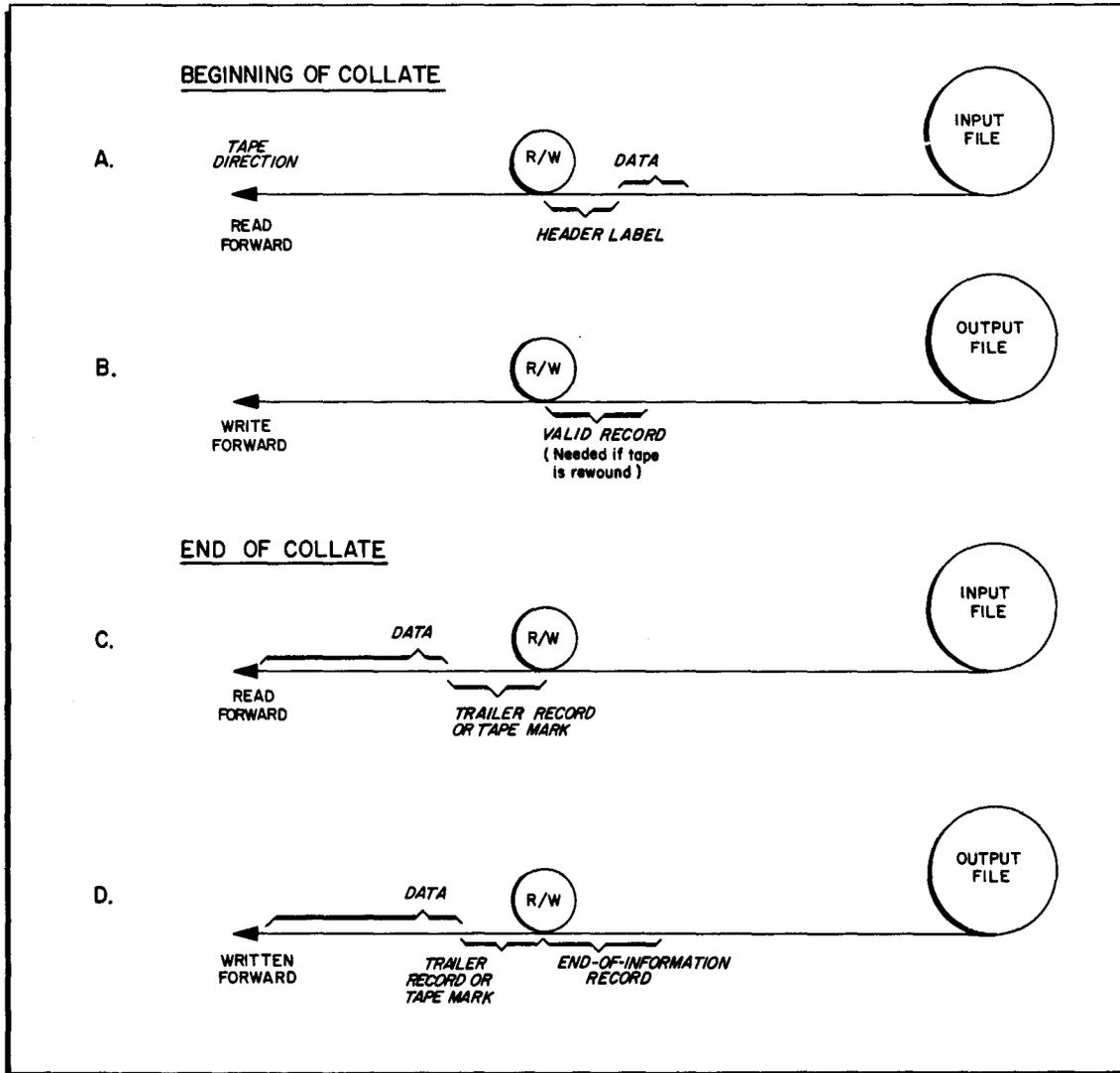


Figure 7-1. Collate C Tape Positioning



SECTION VIII  
COLLATE C OPERATING CONSIDERATIONS

INTRODUCTION

Collate C may be run as one program within a series, as the initial program in a series, or as a single program. Regardless of the method of running Collate C, the program will be loaded in the same manner as other programs functioning in the Mod 1 Operating System, i. e., by a call to a Loader-Monitor program.<sup>1</sup> Collate C may be stored on tape to be run under control of Tape Loader-Monitor C(3), Tape Loader-Monitor C(4), or Floating Tape Loader-Monitor C, or it could be stored on cards in binary run format to be run under control of Card Loader-Monitor B or the card loading option of Floating Tape Loader-Monitor C. Suggested in this section are various methods of running Collate C according to the type of Loader-Monitor used, including the considerations in preparing each method for operation.

COLLATE C PROGRAM FORMAT

Essential to discussing the mode of operation of Collate C is the sequence of the program's operational segments, which must be executed in the order listed below. Note that segment 00 is optional and may be used to call and load parameters from the card reader. Segments T1 and T2 are only required when a typewriter is used with the collate program.

Table 8-1. Collate C Program Format

<u>Program Name*</u>	<u>Segment Name</u>
AAECOL	00
AAECOL	T1
AAECOL	01
AAECOL	T2
AAECOL	S2
AAECOL	S3
AAECOL	S4
AAECOL	S5
AAECOL	S6
AAECOL	S7

\*When the Collate C program is assembled in four-character addressing mode, the program name is AAECOF.

CALLING AND LOADING COLLATE C WITH A TAPE LOADER-MONITOR

A Tape Loader-Monitor is used when Collate C is stored on a binary run tape (BRT).

<sup>1</sup>More extensive information on program loading under the SERIES 200/OPERATING SYSTEM - MOD 1 can be found in the Honeywell Information Bulletins listed in the foreword of this manual.

The calling and loading considerations for the collate program will depend upon the manner in which it is to be run, i. e. , the collate parameters may be placed in memory by the program executed before Collate C.

A general consideration is that a Tape or Card Loader-Monitor program assembled in three-character mode should not be used to load a version of Collate C assembled in four-character mode to include typewriter options. A three-character mode Loader-Monitor cannot load program segments above 32K; and, since the collate program loads its typewriter routines into upper memory according to the highest address available, as designated in parameter characters 49-54, it is possible that the typewriter routines for this configuration will not be loaded if the highest address specified is above 32K.

#### Programmed Call for Collate C on a BRT

When Collate C is run as a program within a series, it is useful to have the collate parameters left in memory by the program executed before Collate C and to include coding in that program which will initiate the call for the collate program. The Collate C program name AAECOL and first segment name 01 should be placed in the appropriate fields of the Loader-Monitor communication area to call in the program. Specification of segment 01 assumes the presence of parameters in memory.

NOTE: If Collate C has been assembled in four-character address mode, the program name becomes AAECOF, but the methods of running the program are the same as those explained for the three-character version, AAECOL.

#### Card Call for Collate C on a BRT with Parameters on Cards

If Collate C is to be run as the initial program of a series or as a single program, the collate parameters can be loaded from the card reader under control of segment 00. Once the Loader-Monitor program has been loaded into memory, a Console Call card can be used to call in collate segment 00 to enter the parameters from a card reader and initiate the loading of the collate program. The card reader should contain the necessary cards positioned in their proper order as described in Figure 8-1.

#### CALLING AND LOADING COLLATE C FROM CARDS

When Collate C program is punched on cards in BRT format, Card Loader-Monitor B or the card loading option of Floating Tape Loader-Monitor C is used to call and load the object deck.<sup>1</sup> The considerations for calling and loading the collate program vary according to the manner in which the program is run, i. e. , whether collate parameters can be loaded by the program executed before Collate C. As mentioned previously, a general consideration is that

---

<sup>1</sup>See footnote page 4-3.

a three-character mode Loader-Monitor should not be used to load a version of Collate C in four-character mode that includes typewriter options.

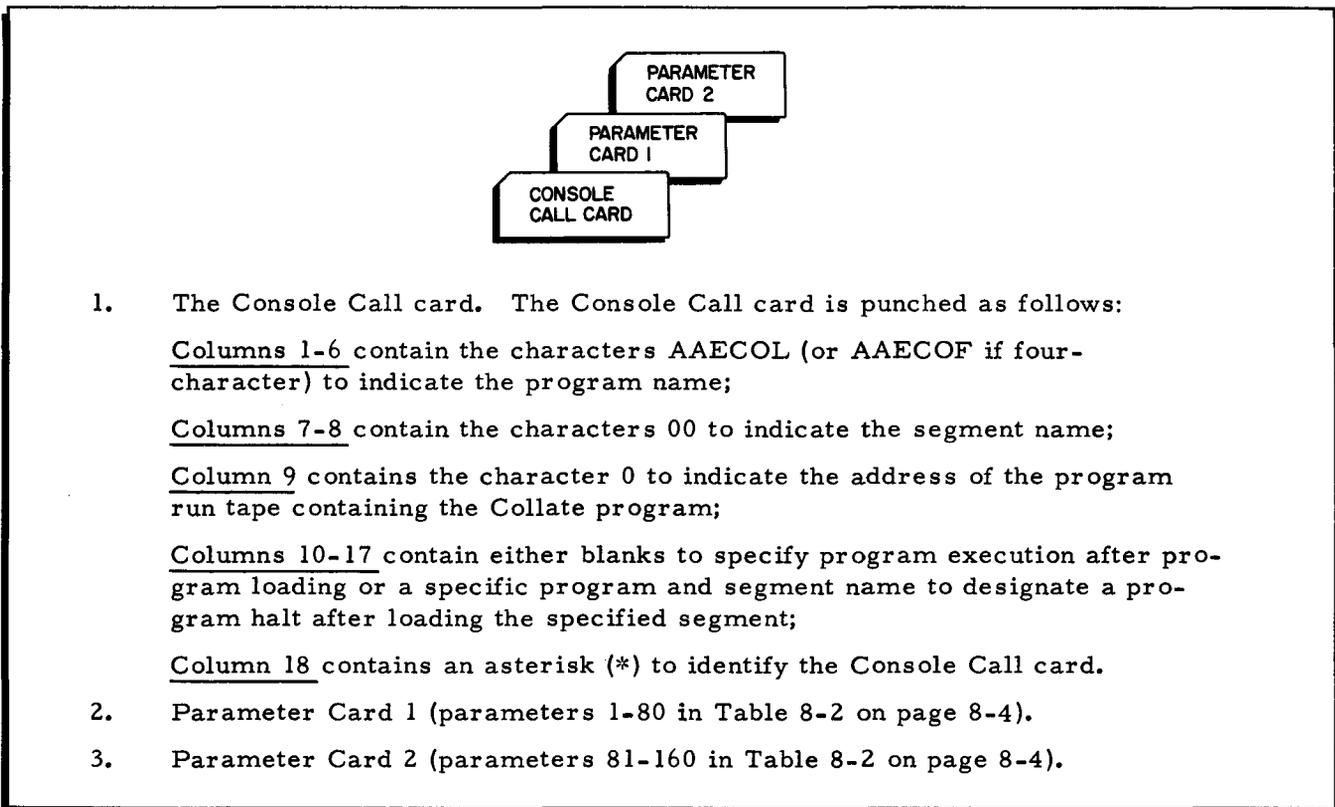


Figure 8-1. Parameter Card Input for Loading Collate C Parameters

#### Programmed Call for Collate C Program on Cards

If Collate C is loaded within a series of programs, it is recommended that the preceding program enter the collate parameters in their proper memory locations (the locations are specified in Table 8-2 on page 8-4). The program which precedes Collate C is used to initiate a call by entering the Collate C program name (AAECOL) and first segment name (01) in the appropriate fields of the Loader-Monitor communication area. The program run deck should be placed in the card reader in the order shown in Figure 9-2 on page 9-3.

The Collate C program deck is made up of segments, as indicated on page 8-1, each of which is identified by a segment header card, a card with an X8 (Q) or an X84 (\*) punch in column 1. After the preceding program has entered the parameters in memory and initiated a call for Collate C, the Loader-Monitor searches forward, reads the 01 segment header card (with the collate program name in columns 11 through 16 and the segment name in columns 17 and 18), and begins loading the Collate C program.

Card Call for Collate C Program and Parameters on Cards

When Collate C is on cards in BRT format and is to be run as the initial program in a series or as a single program, the collate parameters are punched on parameter cards and placed in the program deck between segments 00 and 01 (the parameter cards are placed after the end-of-segment card for 00, which contains an X4 "M" punch in column 1). When typewriter segment T1 is used, the parameter cards should be placed after the end-of-segment card for T1. The program run deck should be placed in the card reader in the order shown in Figure 9-1 on page 9-3. Once the Loader-Monitor has been loaded, it will search forward for a loading unit read the 00 segment header card, and load the collate program.

CALLING AND LOADING OWN-CODING

The user must provide a method of loading Collate C own-coding. Own-coding can be called and loaded in the same manner as any other segment of a loading unit, but it must be loaded before the first collate segment is loaded.

COLLATE C PARAMETER SUMMARY

Table 8-2 lists and defines the parameters for Collate C, gives the octal locations for the first character of each parameter field, and describes the specific parameter configurations (giving options where possible) to be entered into memory.

NOTE: In Table 8-2, "T" is used to designate a tape control unit; "t" specifies a tape drive attached to the control unit; and "x," unless otherwise specified, represents a decimal digit.

Table 8-2. Collate C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration
Tape Unit Specifications	1-2	2477	Address of primary input file	Tt
	3-4	2501	Number of reels in primary file	xx
	5-6	2503	Address of second input file	Tt
	7-8	2505	Number of reels in second input file	xx
	9-10	2507	Address of third input file (optional)	Tt or blank
	11-12	2511	Number of reels in the third input file (optional)	xx or blank
	13-14	2513	Address of fourth input file (optional)	Tt or blank
	15-16	2515	Number of reels in fourth input file (optional)	xx or blank

## SECTION VIII. COLLATE C OPERATING CONSIDERATIONS

Table 8-2 (cont). Collate C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration																																																																																																																
Tape Unit Specifications (cont)	17-18	2517	Address of fifth input file (optional)	Tt or blank																																																																																																																
	19-20	2521	Number of reels in fifth input file (optional)	xx or blank																																																																																																																
	21-22	2523	Address of output tape	Tt																																																																																																																
	23-24	2525	Address of alternate output tape (optional)	Tt or blank																																																																																																																
Item and Record Sizes	25-28	2527	Number of characters per item	xxxx																																																																																																																
	29-32	2533	Number of items per input record	xxxx																																																																																																																
	33-36	2537	Number of items per output record	xxxx																																																																																																																
Padding Character	37	2543	Padding character	Any alphanumeric character																																																																																																																
Label Indicators	38-39	2544	<p>Label indicators:</p> <table border="0"> <tr> <td></td> <td style="text-align: center;"><u>beginning</u></td> <td style="text-align: center;"><u>end of file</u></td> <td></td> </tr> <tr> <td></td> <td style="text-align: center;"><u>of file</u></td> <td style="text-align: center;"><u>end of file</u></td> <td></td> </tr> <tr> <td>ΔΔ -no label</td> <td>tape mark</td> <td></td> <td></td> </tr> <tr> <td>ΔT -no label</td> <td>tape mark,</td> <td></td> <td></td> </tr> <tr> <td></td> <td>trailer,* and</td> <td></td> <td></td> </tr> <tr> <td></td> <td>tape mark</td> <td></td> <td></td> </tr> <tr> <td>SS-std.</td> <td></td> <td>std. trailer</td> <td></td> </tr> <tr> <td>label</td> <td></td> <td></td> <td></td> </tr> <tr> <td>SS-std.</td> <td>tape mark,</td> <td></td> <td></td> </tr> <tr> <td>label</td> <td>std. trailer,</td> <td></td> <td></td> </tr> <tr> <td>and tape</td> <td>and tape</td> <td></td> <td></td> </tr> <tr> <td>mark</td> <td>mark</td> <td></td> <td></td> </tr> <tr> <td>ST-std.</td> <td>tape mark,</td> <td></td> <td></td> </tr> <tr> <td>label</td> <td>std. trailer,</td> <td></td> <td></td> </tr> <tr> <td></td> <td>and tape</td> <td></td> <td></td> </tr> <tr> <td></td> <td>mark</td> <td></td> <td></td> </tr> <tr> <td>NΔ-non-std.</td> <td>tape mark</td> <td></td> <td></td> </tr> <tr> <td>label</td> <td></td> <td></td> <td></td> </tr> <tr> <td>LL-std. (120-</td> <td>tape mark,</td> <td></td> <td></td> </tr> <tr> <td>char.)</td> <td>std. (120-</td> <td></td> <td></td> </tr> <tr> <td>IBM</td> <td>char.)IBM</td> <td></td> <td></td> </tr> <tr> <td>(1410/</td> <td>(1410/7010)</td> <td></td> <td></td> </tr> <tr> <td>7010)</td> <td>trailer, and</td> <td></td> <td></td> </tr> <tr> <td>label and</td> <td>tape mark</td> <td></td> <td></td> </tr> <tr> <td>tape mark</td> <td></td> <td></td> <td></td> </tr> <tr> <td>NT-non-std.</td> <td>tape mark,</td> <td></td> <td></td> </tr> <tr> <td>label</td> <td>trailer, and</td> <td></td> <td></td> </tr> <tr> <td></td> <td>tape mark</td> <td></td> <td></td> </tr> </table>		<u>beginning</u>	<u>end of file</u>			<u>of file</u>	<u>end of file</u>		ΔΔ -no label	tape mark			ΔT -no label	tape mark,				trailer,* and				tape mark			SS-std.		std. trailer		label				SS-std.	tape mark,			label	std. trailer,			and tape	and tape			mark	mark			ST-std.	tape mark,			label	std. trailer,				and tape				mark			NΔ-non-std.	tape mark			label				LL-std. (120-	tape mark,			char.)	std. (120-			IBM	char.)IBM			(1410/	(1410/7010)			7010)	trailer, and			label and	tape mark			tape mark				NT-non-std.	tape mark,			label	trailer, and				tape mark			<p>ΔΔ, ΔT, SS, ST,LL NΔ, or NT (When specifying SS, the user should have one of the proper file label combinations given under "Label indicators", the labels in the given combinations cannot be interchanged.)</p>
	<u>beginning</u>	<u>end of file</u>																																																																																																																		
	<u>of file</u>	<u>end of file</u>																																																																																																																		
ΔΔ -no label	tape mark																																																																																																																			
ΔT -no label	tape mark,																																																																																																																			
	trailer,* and																																																																																																																			
	tape mark																																																																																																																			
SS-std.		std. trailer																																																																																																																		
label																																																																																																																				
SS-std.	tape mark,																																																																																																																			
label	std. trailer,																																																																																																																			
and tape	and tape																																																																																																																			
mark	mark																																																																																																																			
ST-std.	tape mark,																																																																																																																			
label	std. trailer,																																																																																																																			
	and tape																																																																																																																			
	mark																																																																																																																			
NΔ-non-std.	tape mark																																																																																																																			
label																																																																																																																				
LL-std. (120-	tape mark,																																																																																																																			
char.)	std. (120-																																																																																																																			
IBM	char.)IBM																																																																																																																			
(1410/	(1410/7010)																																																																																																																			
7010)	trailer, and																																																																																																																			
label and	tape mark																																																																																																																			
tape mark																																																																																																																				
NT-non-std.	tape mark,																																																																																																																			
label	trailer, and																																																																																																																			
	tape mark																																																																																																																			

## SECTION VIII. COLLATE C OPERATING CONSIDERATIONS

Table 8-2 (cont). Collate C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration
Output Label Verification	40	2546	Output label verification: $\Delta$ -do not verify label 1-verify output file label for "IHDR $\Delta$ "	$\Delta$ or 1
Banner Characters	41-42	2547	Banner characters: <u>input</u> <u>output</u> $\Delta$ -banner character 1-no banner character	$\Delta$ or 1 $\Delta$ or 1
Parity and Gap Size Indicators	43	2551	Input parity indicator: E-even parity $\Delta$ -odd parity	E or $\Delta$
	44	2552	Output parity and gap size indicator: L-odd parity, 3/4" gaps E-even parity, 3/4" gaps $\Delta$ -odd parity, short gaps	L, E, or $\Delta$
Error Options	45-46	2553	Error Options (for an uncorrectable read error): $\Delta\Delta$ -check P $\Delta$ -print the record and check PD-print and delete the record	$\Delta\Delta$ , P $\Delta$ , or PD
Printer Control Unit	47	2555	Printer control unit x-address of the printer control unit $\Delta$ -no printouts	x or $\Delta$
	48	2556	Not used	
Highest Collate Address	49-54	2557	Highest address available to the collate program (If the address is specified in decimal, leading zeros must be included when necessary.)	xxxxxx or an x for character 54, where x represents a number of 4K modules (leading blanks must be included)
Collating Sequence	55	2565	Collating sequence: T-output in IBM collating sequence $\Delta$ -output in Honeywell collating sequence	T or $\Delta$

## SECTION VIII. COLLATE C OPERATING CONSIDERATIONS

Table 8-2 (cont). Collate C Parameter Specification

Parameter Group	Parameter Number	Octal Locations of First Character of Field	Description	Parameter Configuration
Collating Sequence (cont)	56	2566	Ascending or descending sequence output: Δ-input and output in ascending sequence D-input and output in descending sequence	Δ or D
Rewind Option	57	2567	Rewind Option: Δ-all tapes are rewound at the beginning of the collate 1-do not rewind tapes	Δ or 1
Own-Coding	58-63	2570	Address of Collate C header-trailer own-coding (leading zeros must be specified)	xxxxxx or blank if not used
	64-69	2576	Address of Collate C item-by-item own-coding (leading zeros must be specified)	xxxxxx or blank if not used
Number of Output Records	70-74	2604	Number of data records output reel	xxxxx (leading zeros must be specified)
	75-78	2611	Not Used (must be blank)	
Parameter Card Identification	79-80	2615	Parameter card identification: C2 - used only if parameters are entered from the card reader	C2 or not used
Key Field Specifications	81-84	2617	Position of key field 1 (position of the leftmost, high-order characters in the major key field)	xxxx
	85-86	2623	Number of characters in key field 1	xx
	87-134	2625	Similar configurations for key fields 2-9 (see page 7-7).	
	135-138	2705	Position of key field 10 (If less than ten key fields are used, the remaining characters in the key field area must be blank)	xxxx or blank if not used
	139-140	2711	Number of characters in key field 10	xx or blank if not used

## SECTION VIII. COLLATE C OPERATING CONSIDERATIONS

Table 8-2 (cont). Collate C Parameter Specification

Parameter Group	Parameter Number	Octal Location of First Character of Field	Description	Parameter Configuration
File Name Change	141	2713	Input file name and reel number verification: F - verify file name and reel number Δ - do not verify file name and reel number	F or Δ
	142-151	2714	Output file name change option	An alphanumeric name of up to ten characters, or a blank field if the option is not desired
Loader-Monitor	152-157	2726	Next program name	xxxxxx or blank if not used (parameters 152-160 are blank if Collate C is not followed by automatic loading of another program)
	158-159	2734	Next segment name	xx or blank if not used
	160	2736	Search direction for next program B - search forward C - search backward	B, C, or Δ if not used

SECTION IX  
COLLATE C OPERATING PROCEDURES

PRELIMINARY CONSIDERATIONS

The preliminary considerations for Collate C are similar to those for Tape Sort C, as given on page 5-1.

LOADING COLLATE C FROM A BRT

Suggested methods for calling and loading Collate C are given below. In each instance given, the Tape Loader-Monitor used is considered to have been loaded previously.

Loading Collate C from a BRT with Parameters on Cards

If the program executed before the collate did not include instructions for loading the parameters and calling Collate C, the call to load the parameters and program can be issued from the card reader via a Console Call card. The Console Call card and the parameter cards are placed in the card reader input hopper as shown in Figure 8-1 on page 8-3. The operator should then perform the steps listed below.

1. Verify the following:
  - a. The Loader-Monitor is ready to accept a console call.
  - b. All peripheral devices to be used are cycled up.
  - c. The collate input and output tapes are mounted on the appropriate drives according to parameters (see parameters on page 8-4).
2. Press RUN on the control panel or console. (The Loader-Monitor reads the Console Call card which calls the collate segment 00 to read in the parameter cards and begin collate program loading and execution. If a check should occur during the loading or running of the program, it can be detected and corrected as explained on page 9-4.)

Loading Collate C from a BRT by Programmed Call

If there are programs preceding Collate C on the BRT, the program executed before the collate program can enter the parameters and initiate a call for the Loader-Monitor to load Collate C. Using this method to call and load Collate C eliminates further operator intervention once the tapes required for collating are mounted on their appropriate drives according to parameters.

NOTE: If a check should occur during the loading or running of Collate C, see "Collate C Checkpoints" starting on page 9-5 to determine the cause and necessary corrective action.

LOADING COLLATE C FROM AN OBJECT PROGRAM DECK (BRD)

Either Card Loader-Monitor B or the card loading option of Floating Tape Loader-Monitor C may be used to call and load Collate C; the suggested methods given below are used for calling and loading any object program deck. Use of the card loading option of Floating Tape Loader-Monitor C requires the insertion of an appropriately designated Console Call card before segment 00 in the collate program deck; a Console Call card is not required when using Card Loader-Monitor B.

Loading Collate C Program and Parameters from Cards

If the program executed before Collate C did not enter the collate parameters, the parameters are punched on two cards and inserted between segment 00 and the next program segment in the object deck.<sup>1</sup> A suggested format for the object program deck with the above configuration is shown in Figure 9-1.

Once the applicable Loader-Monitor is loaded and awaiting a console call, the operator should perform the steps listed below.

1. Verify the following:
  - a. The Loader-Monitor is ready to accept a console call. (A Console Call card is the first card in the deck if Floating Tape Loader-Monitor C is used.)
  - b. All peripheral devices to be used are cycled-up.
  - c. The tapes required for collating are mounted on the appropriate drives according to parameters (see parameters on page 8-4).
2. Press RUN. (The Loader-Monitor reads the 00 segment header card when called and proceeds loading and starting the Collate C program.)

Loading Collate C from an Object Program Deck by Programmed Call

When there are programs preceding Collate C in the object program deck, it is suggested that the program executed before the collate program enter parameters and initiate a call for the Loader-Monitor to load Collate C. If a check should occur during the loading or running of the Collate program, see "Collate C Checkpoints" starting on page 9-5 to determine the cause and the necessary corrective action.

If the programmed call method is used to call Collate C, a suggested object program deck format is shown in Figure 9-2.

---

<sup>1</sup>The parameters are placed after the end-of-segment card for 00, which contains an X4 (M) punch in column 1, or after the end-of-segment card for T1.

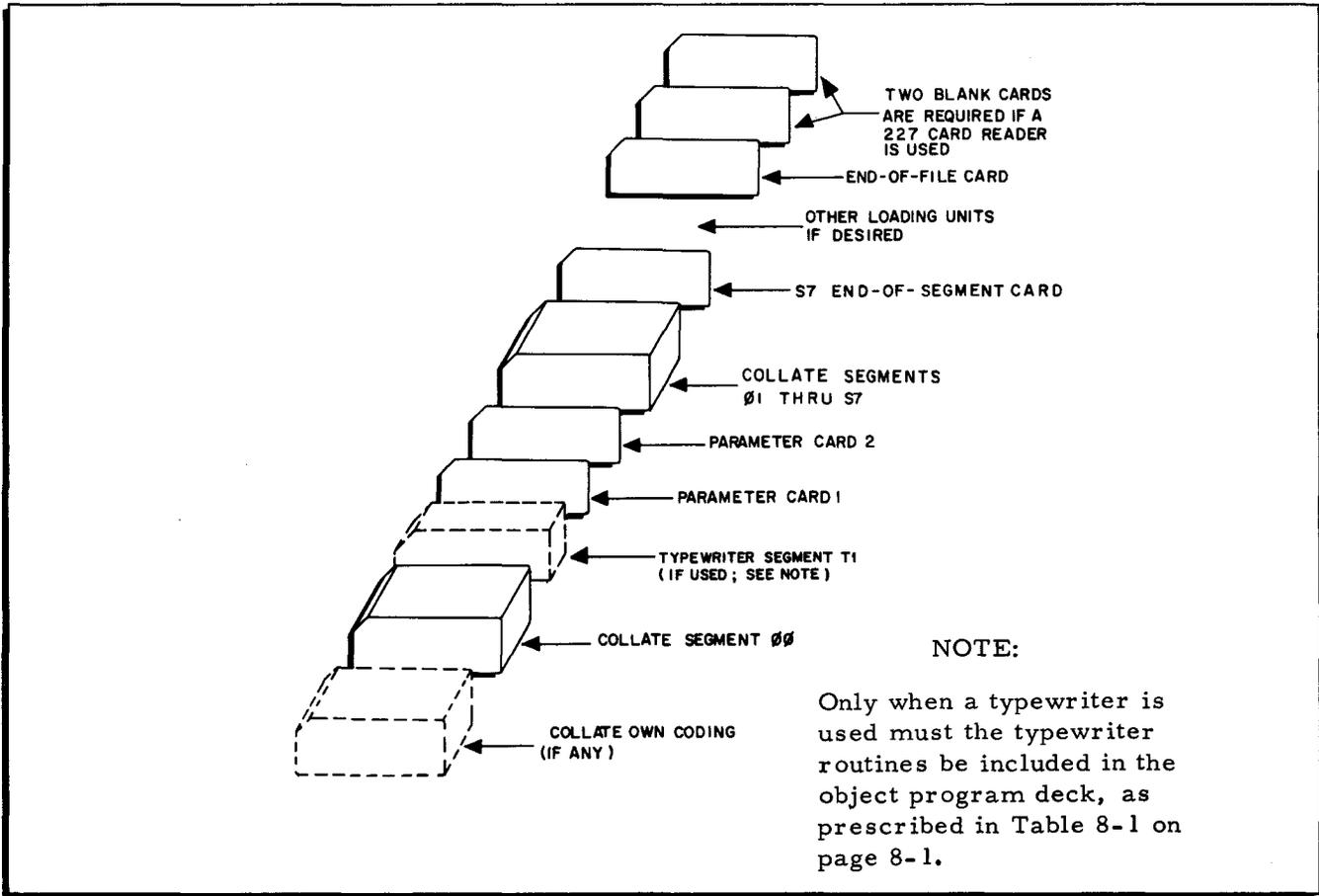


Figure 9-1. Suggested Object Program Deck Format for Loading Collate C with Parameter Cards

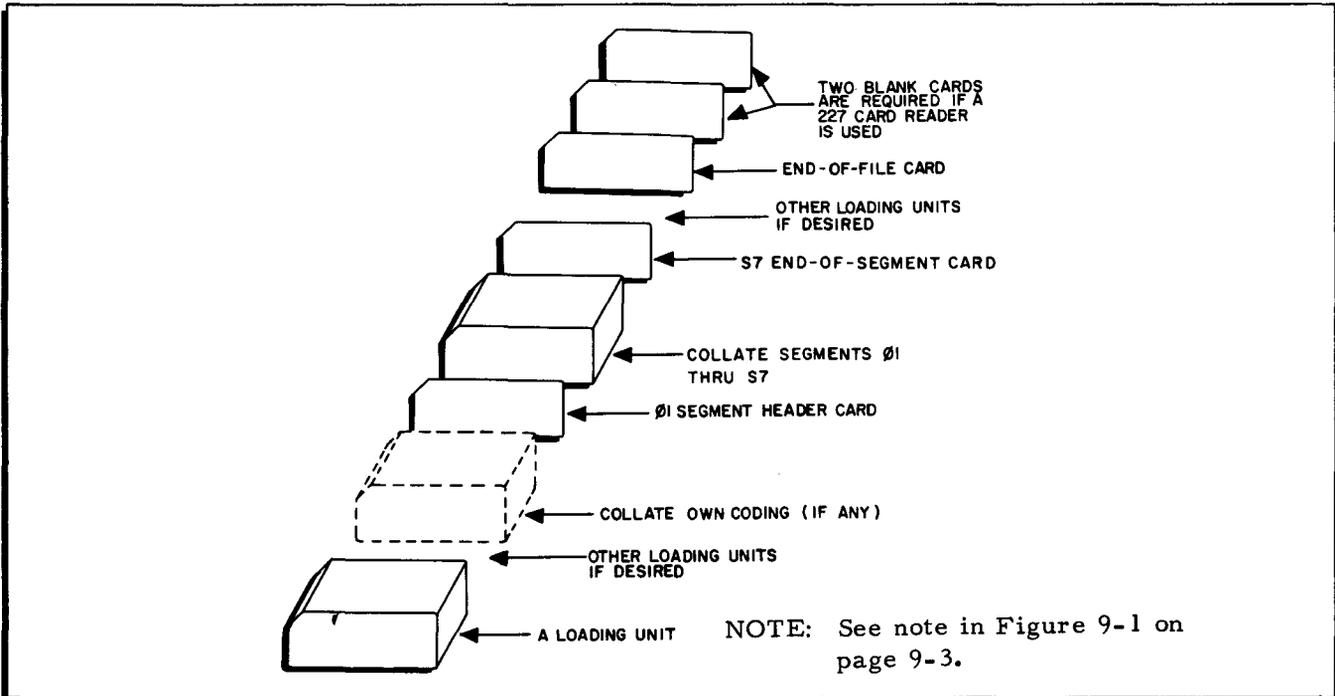


Figure 9-2. Suggested Object Program Deck Format for Loading Collate C without Parameter Cards

CHECK AND CHECKPOINT PROCEDURES

The operator should familiarize himself with the check and checkpoint procedures for Tape Sort C explained on page 5-4 as they are also relevant for Collate C, with one exception. The exception occurs when a checkpoint designates that the collate cannot be continued and must be rerun, for Collate C does not have a restart procedure for peripheral malfunctions as does Tape Sort C. Therefore, the operator must prepare the Loader-Monitor for a console call and rerun the collate.

If a typed message indicates that the Collate C program must be rerun, typing a response entry of "M" automatically returns control to the Loader-Monitor preparing it for a console call; if any other response is typed, the message "TERMINATE RUN" is listed, indicating the request for an "M" entry. If the rerun message occurred with a machine halt, the operator can enter octal 126 into the memory address register, set the three-character address mode indicator, clear the sector bits (bits 16 through 18) of the A-address register, and press RUN to prepare the Loader-Monitor for a console call.

Check Message Coding

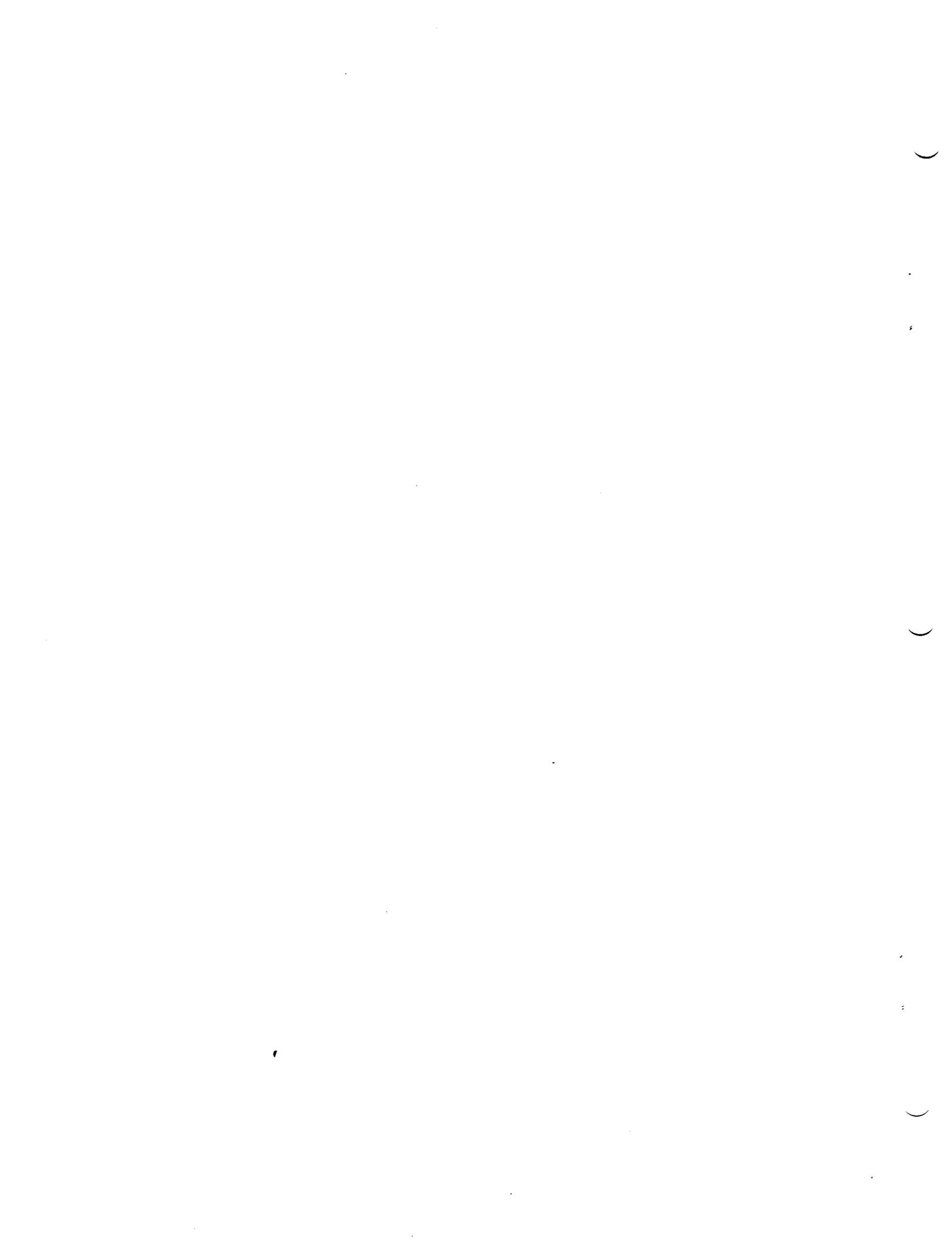
Table 9-1 lists the checkpoints by number and indicates the check message whether contained in the A- and B-address registers (e.g., A: 00000 B: 04T0D) or typed out (e.g., : TT D SEQ ER). The messages are listed beside the probable cause of the check and its possible corrective action, depending upon whether the check was coincidental with a typed message or a machine halt.

Check messages in most instances contain coded characters to make them more concise and meaningful. When messages are typed, they may include some of the following code characters: RD - read, WRT - write, ER - error, CD - card, TT - control unit address, D - tape drive number. The check messages as represented in the A- and B-address registers are divided into two categories for easy identification: (1) peripheral errors and (2) non-peripheral errors. When a particular check specifies a peripheral error, the B-address register contains a "0" in the fourth octal digit (from the right). With a non-peripheral error, the B-address register contains a "4" in the fourth octal digit. These configurations may include the following: "cu" for control unit address, "d" for tape drive number, and "bbbb" for the buffer address. The buffer address is the first (leftmost) location of the input buffer.

SECTION IX. COLLATE C OPERATING PROCEDURES

Table 9-1. Collate C Checkpoints

Checkpoint Number	Check Message	Cause	Corrective Action
1	A: 05555 B: 05555 : PARAM ER 79□	C2 not found in columns 79-80 of parameter card 1	Correct parameter card 1 and rerun the collate.
2	A: 00001 B: 04000 : KEY FIELDS□	Error in key field parameters	Correct the parameters and rerun the collate.
3	A: 00002 B: 04000 : TAPE ASSIGN□	Error in tape parameters	NOTE: The next thirteen checkpoints are parameter error conditions, and each error has the same corrective action as given above for checkpoint two.
4	A: 00003 B: 04000 : HIGH ADDRESS□	Error in highest address available; decimal parameter contains a non-decimal character	
5	A: 00004 B: 04000 : PARITY OPTION□	Parity/gap parameter incorrect	
6	A: 00005 B: 04000 : LABEL OPTION□	Label parameter incorrect	
7	A: 00006 B: 04000 : OWN CODE ADDR□	Own-coding address lies within the Collate C program area	
8	A: 00007 B: 04000 : MEMORY LIMIT ERROR□	Insufficient memory	
9	A: 00011 B: 04000 : OUTPUT FILE CK□	Parameter incorrect for output tape label verification	
10	A: 00012 B: 04000 : BANNER OPTION□	Banner character parameter incorrect	
11	A: 00013 B: 04000 : ER OPT OR PCU□	Read error parameter incorrect	
12	A: 00014 B: 04000 : COLLATING SEQ□	Translation parameter incorrect	
13	A: 00015 B: 04000 : ASC DESC SEQ□	Ascending/descending parameter incorrect	
14	A: 00016 B: 04000 : REWIND OPTION□	Rewind parameter incorrect	
15	A: 00017 B: 04000 : CK FILENM OPT□	Input file name and reel number verification parameter incorrect	
16	A: 00000 B: 00000 : CD RD ER□	Parameter card read error	Correct the parameter card, refeed it, and press RUN or type "G."
17	A: bbbbb B: 0cud : TT D RD ER□	Read error: data record (64 attempts have been made to reread the record)	To try to reread the record, set SENSE switches 1 and 2 OFF and press RUN or type "G." To accept the record, set SENSE switch 1 ON and press RUN or type "G." To delete the record, set SENSE switch 2 ON and press RUN or type "G."
18	A: bbbbb B: 0cu2d : TT D WRT ER□	Write error (a routine to backspace, erase, and rewrite has been executed nine times)	For nine more attempts to rewrite, press RUN or type "G."
19	A: bbbbb B: 0cu5d : TT D HEADER RD ER□	Read error on header record	For another read attempt, press RUN or type "G"; if unsuccessful, set SENSE switch 1 ON and press RUN or type "G" to accept the header as is.
20	A: 00001 B: 0cu3d : TT D MOUNT NEXT REEL□	Multi-reel output	Mount next output reel on the indicated drive and press RUN or type "G."
21	A: 00000 B: 04t0d : TT D SEQ ER□	Sequence error (input files not in sequence)	The input on the indicated drive is not in sequence; correct and rerun the collate.
22	A: 02222 B: 0cu0d : TT D MOUNT NEXT REEL□	Multi-reel input	Mount the next input reel on the indicated drive and press RUN or type "G." Checkpoint 23 occurs if correct reel was not mounted.
23	A: 02222 B: 0cu0d : TT D MOUNT NEXT REEL□	File name and reel number verification has failed	Mount the correct reel of input on the indicated drive and press RUN or type "G."
24	A: 00003 B: 0cu0d : TT D RETEN, CYCLE NOT EXPIRED□	Retention cycle verification	To try to verify the retention cycle again, set SENSE switch 1 OFF and press RUN or type "G." If another reel has been mounted or it is desired to accept the tape, set SENSE switch 1 ON and press RUN or type "G."
25	A: 07777 B: 0cu0d : END OF COLLATE xxxxxxxx ITEMS xxx PADDED xxxxx DELETED□ :TTD OUTPUT TAPE E.O.J.	End of job	The B-address register contains the address and number of the drive that contains the collated output. (If parameters 152-160 were blank, press RUN or type "M" to return control to the Loader-Monitor for a console call.)



APPENDIX A  
TIMING FORMULAS FOR TAPE SORT C AND COLLATE C

TAPE SORT C TIMING

The user can determine the total time required to run Tape Sort C for a particular application by substituting the appropriate values for the parameters listed in Table A-1. The designated equations can be solved accordingly, and the results are added to determine the total time in seconds. The timing formulas make the following assumptions:

1. Record size does not exceed the optimum size derived from the formula on page 2-3.
2. There is no own-coding.

Table A-1. Timing Formulas for Tape Sort C

Parameter	Definition
IF	Total number of items in the file.
CI	Number of characters per item.
IR <sub>p</sub>	Items per record (presort input).
CR	Number of characters per record = (CI) (IR <sub>p</sub> ).
LDR	Length of data record (in inches) = $\frac{CR}{\text{Density}}$ .
IRG	Interrecord gap (in inches).
RL	Record length (in inches) = LDR + IRG.
LF	Length of file = RL $\left( \frac{IF}{IR_p} \right)$ .
TF	File time (in seconds) = $\frac{LF}{\text{tape speed in inches/sec.}}$ .
RF	File rewind time (in seconds) = the lesser of TF or $\frac{TF}{3} + 12$ .
P	Power or way (number of merge tapes minus one).
HMA	Highest memory address available to the sort.
IR <sub>o</sub>	Items per record (final output).
IR <sub>m</sub>	Items per record (merge). If the result of a or b is less than IR <sub>p</sub> , IR <sub>m</sub> = IR <sub>p</sub> .
	<u>3-Character Version</u>
	(a) If IR <sub>o</sub> IR <sub>p</sub> , IR <sub>m</sub> = $\frac{HMA-5200-(TYP3)}{2(CI)(P+1)}$ rounded down
	(b) If IR <sub>o</sub> IR <sub>p</sub> , IR <sub>m</sub> = $\frac{HMA-5200-(TYP3)-2(CI)(IR_o)}{2P(CI)}$ rounded down
	where TYP3 = 0 with no typewriter; or 520 with a typewriter.

Table A-1 (cont). Timing Formulas for Tape Sort C

Parameter	Definition
$IR_m$ (cont)	<p><u>4-Character Version</u></p> <p>(a) If <math>IR_o \leq IR_p</math>, <math>IR_m = \frac{HMA-6200-(TYP4)}{2(CI)(P+1)}</math> rounded down</p> <p>(b) If <math>IR_o &gt; IR_p</math>, <math>IR_m = \frac{HMA-6200-(TYP4)-2(CI)(IR_o)}{2P(CI)}</math> rounded down</p> <p>where TYP4 = 0 with no typewriter; or 590 with a typewriter</p>
IS	<p>Items per string, where</p> <p>n = 1.6 for a two-way merge  n = 1.75 for a three way merge  n = 1.8 for a four-way merge  n = 1.85 for a five-way merge</p> <p><u>3-Character Version</u></p> $IS = n \left[ \frac{HMA - 5200 - (TYP3) - 2CR - 2(CI)(IR_m)}{CI + 12} \right]$ <p><u>4-Character Version</u></p> $IS = n \left[ \frac{HMA - 6200 - (TYP4) - 2(CI)(IR_m) - 2CR}{CI + 16} \right]$
NS	Number of strings = $\frac{IF}{IS}$ rounded up.
$RL_m$	Merge record length (in inches) = $\frac{(IR_m)(CI)}{\text{Density}} + IRG.$
$FL_m$	File length in merge (in inches) = $RL_m \left( \frac{IF}{IR_m} \right)$ .
$TF_m$	File time in merge (in seconds) = $\frac{FL_m}{\text{tape speed in inches/sec.}}$ .
NK	Number of key fields.
CK	Number of characters in sort key.
$RL_o$	Final output record length (in inches) = $\frac{(IR_o)(CI)}{\text{Density}} + IRG.$
$LF_o$	File length of final output (in inches) = $RL_o \left( \frac{IF}{IR_o} \right)$
CT	Cycle time of Series 200 Model used (in seconds) = (.000003) for Model 120 (.000002) for 200 (.0000015) for 1200 (.000001) for 2200
<u>Presort Time</u>	
CPT	<p>Central processor time (in seconds), where y is determined from <math>2^y \geq \frac{IS}{2}</math></p> <p><u>3-Character Version</u></p> $CPT = CT (IF) \left[ 378 + 4CI + 128NK + y(192+2CK) + \frac{560}{IR_p} + \frac{616}{IR_m} \right]$

Table A-1 (cont). Timing Formulas for Tape Sort C

Parameter	Definition				
CPT (cont)	<u>4-Character Version</u>				
	$CPT = CT (IF) \left[ 476 + 4CI + 160NK + y(240+2CK) + \frac{700}{IR_p} + \frac{770}{IR_m} \right]$				
TP	Total presort time (in seconds) = the greater of TF* or CPT + 2CT (CI) (IF).				
	<u>Merge Time</u>				
NP	Number of phases. In the table below, find the value just higher than and including NS. (Read NP to the left.) If NP = 0, skip merge and calculate last pass only.				
<u>NP</u>	<u>2-way</u>	<u>3-way</u>	<u>4-way</u>	<u>5-way</u>	
0	2	3	4	5	
1	3	5	7	9	
2	5	9	13	17	
3		17	25	33	
4	13		49	65	
5	21	57		129	
6		105	181		
7	55	193	349	497	
8	89		673	977	
9		653	1297	1921	
10	233	1201		3777	
11	377	2209	4819	7425	
12					
13	987	7473			
14	1597	13745			
15		25281			
16	4181				
%F	Percent factor = .75 (2-way), .62(3-way), .58(4-way), or .55(5-way).				
TPP	Time/phase in the merge = greater of a and b below.				
	<u>3-Character Version</u>				
	(a) $\%F \left[ TF_m^{**} + 1500CT \left( \frac{IF}{IR_m} \right) \right] + 5$				
	(b) $\%F \left[ IF \left( \frac{1366CT}{IR_m} + 2CT (25P+162+(P-1)CK+2CI) \right) \right]$				
	<u>4-Character Version</u>				
	(a) $\%F \left[ TF_m + 2000CT \left( \frac{IF}{IR_m} \right) \right] + 5$				

Table A-1 (cont). Timing Formulas for Tape Sort C

Parameter	Definition
TPP (cont)	(b) %F $\left[ IF \left( \frac{1707.5CT}{IR_m} + 2CT (32P+203+(P-1)CK+2CI) \right) \right]$
TM	Total merge time (in seconds) = TPP(NP)
<u>Last Pass Time</u>	
TL	Total last pass time in seconds = the greater of a and b below. <u>3-Character Version</u> (a) $TF_o^{***}$ (b) $IF \left[ 410CT + 2CT (4CK+123NK+2CI) \right] + \left[ \frac{IF}{IR_o} 1198CT + 4CT (CK+40P) \right]$ . <u>4-Character Version</u> (a) $TF_o$ (b) $IF \left[ 487.5CT + 2CT (4CK+154NK+2CI) \right] + \frac{IF}{IR_o} \left[ 1497.5CT + 4CT (CK+50P) \right]$
TT	Total sort time = TP + TM + TL.
<p>*When using a Model 120 with 13KC tape drives, TF is <math>TF + TF_m</math>.</p> <p>**With 13KC drives, <math>TF_m</math> is <math>2TF_m</math>.</p> <p>***With 13KC drives, <math>TF_o</math> is <math>TF_o + TF_m</math>.</p>	

COLLATE C TIMING

The user can determine the time required for a Collate C routine in a specific application by using the parameters and equations listed below. The timing formula makes the following assumptions:

1. Record size does not exceed the optimum size derived from the formula on page 6-3.
2. Input and output blocking (items per record) are the same.
3. There is no own-coding.

The time spent in mounting another reel in the case of multi-reel files is not included in the formulas.

Table A-2. Timing Formulas for Collate C

Parameter	Definition
IF	Total number of items to be combined.
CI	Number of characters per item.
$IR_p$	Number of items per record.
CR	Number of characters per record = (CI) ( $IR_p$ ).

Table A-2 (cont). Timing Formulas for Collate C

Parameter	Definition
LDR	Length of data record (in inches) = $\frac{CR}{\text{Density}}$ .
IRG	Interrecord gap (in inches).
RL	Record length (in inches) = LDR + IRG.
LF	Length of file = RL $\left( \frac{IF}{IR_p} \right)$ .
TF	File time (in seconds) = $\frac{LF}{\text{tape speed in inches/sec.}}$ .
RF	Records in file = $\frac{IF}{IR}$ .
NK	Number of key fields.
CK	Total number of characters in the collate key.
WF	Way factor = 0 if there are 2 input files = 2/3 for 3 input files = 5/4 for 4 input files = 7/5 for 5 input files
TIMI	Time per item (in seconds)'
	<u>3-Character Version</u> $TIMI = CT \left[ 309 + 4CK + CI + \frac{7NK}{2} + WF(CK + 36) \right]$ <u>4-Character Version</u> $TIMI = CT \left[ 389 + 4CK + CI + \frac{7NK}{2} + WF(CK + 36) \right]$
TIMR	Time per record (in seconds)
	<u>3-Character Version</u> $TIMR = CT \left[ 731 + 2CK + \frac{7NK}{2} + WF(CK + 36) \right]$ <u>4-Character Version</u> $TIMR = CT \left[ 916 + 2CK + \frac{7NK}{2} + WF(CK + 45) \right]$
TT	Total collate time (in seconds) = the greater of (a) and (b) below (a) TF* (b) IF (TIMI) + RF (TIMR)

\*When using a Model 120 with 13KC tape drives, TF becomes 2TF.

APPENDIX B  
PROGRAMMING AIDS FOR OWN CODING

SAMPLE OWN-CODING ROUTINES

The sample own-coding linkage routines presented in this appendix may be referenced by the user when coding his specific application. The routines contain the necessary coding to perform the basic own-coding functions of presort and last-pass header-trailer and item-by-item own-coding. The lines of coding below are numbered for reference by their corresponding verbal explanation given in Section III.

**EASYCODER**

CODING FORM

PROBLEM OWN-CODING PROGRAMMER W. DENNEEN DATE 12/16/65 PAGE 1 OF 1

CARD NUMBER	OPERATION CODE	LOCATION	OPERANDS	
			14 15	20 21
00010*				
00020*	PRESORT	ALABEL	DOWN-CODING	
00030*				
	ORG		11000	
00040	PSLAB	SCR	PSNEXX+MODE, 70	Save B-address for normal exit
00050		BCC	PSHDR, PSLAB, 20	header or trailer?
00060		LCA	PSTAD, X1	Trailer routine to replace
				existing Trailer
00070	PSNEXX	B	0	Branch to normal exit
00080	PSHDR	LCA	PSHAD, X1	Header routine to replace
				existing header
00090		B	PSNEX	Branch to normal exit
00100		NOP		
00110*	PRESORT	ITEMBY	ITEMDOWN-CODING	
00120*				
	ORG		11500	
00130	PSITM	SCR	PSNEX+MODE, 70	Save B-address for normal exit
00140	PSW1	B	PSISH	Branch to "one shot" routine
00150		BCC	PSQEN, PSITM, 20	Last input item?
00160*	PROCESS	ADETERMINATION		
00170		C	0-6+X1, PSNXT	Item to be added? (Modify the address in
00180		BCT	PSADD, EQU	X1 to reflect the field compared)
00190		C	0-13+X1, PSDDEL	Item to be deleted?
00200		BCT	PSDLX, EQU	
00210		C	0-25+X1, PSCCN	Item to be modified?
00220		BCT	PSMOD, EQU	
00230	PSNEX	B	*	Branch to normal exit
00240*	ITEMATO	ABEADDED		
00250	PSADD	LCA	PSITA, X1	Add own-coding item
00260		B	(ADRET-MODE+1)	Branch to add exit
00270*	ITEMATO	ABEADELETED		
00280	PSDLX	BA	ONE, DELCNT	Counter for deleted items
00290		B	(DELRT-MODE+1)	Branch to delete exit
00300*	ITEMATO	ABEMODIFIED		



APPENDIX B. PROGRAMMING AIDS FOR OWN CODING

CARD NUMBER	OPERATION	LOCATION	OPERATION CODE	OPERANDS	
				14 15	20 21
00860	SI		IBUF		Add item routine to move own coding item to output buffer beginning at address in X1.
00870	EXM		IBUF+49, 0+X1, 53		
00880	CI		(LPADX-MODE+1)		
00890	B		(LPADX-MODE+1)		Branch to add exit
00900*	MODIFY AN ITEM				
00910	LPMOD		MCW	CON, 0-17+X1	Modify the existing item with the contents of CON
00920	B		LPNEX		Branch to normal exit
00930*	TERMINATION		FAOWN-CODING		
00940	LPNDX		B	(LPTERX-MODE+1)	Branch to terminate exit
00950*	ONE SHOT CODING				
00960	LPISH		HA	OCT25, LPSW1	Stop branch to "one-shot"
00970	LCA		X1, LPADX		Add exit
00980	LCA		X2, LPDLX		Delete exit
00990	LCA		X3, LPTERX		Terminate exit
01000	B		LPNEX		Branch to normal exit
1010	NOP				

COMPUTER-GENERATED INDEX

ACTION  
 CORRECTIVE ACTION,  
 TAPE SORT C CHECKPOINT IDENTIFICATION AND  
 CORRECTIVE ACTION, 5-7

ADDING  
 \* AN ITEM, 3-4, 3-8

ADDRESS  
 MEMORY ADDRESS,  
 HIGHEST MEMORY ADDRESS, 2-8, 7-5

ASCENDING  
 \* OR DESCENDING SEQUENCED OUTPUT, 2-8, 7-5

BACKWARDS INPUT  
 READ BACKWARDS INPUT, 3-2

BANNER CHARACTERS, 2-5, 7-4

BASIC  
 \* CONCEPTS OF TAPE SORT C, 1-1  
 \* CONCEPTS,  
 COLLATE C BASIC CONCEPTS, 6-1

BEGINNING  
 \* OF COLLATE, 7-8  
 \* OF SORT, 2-11

BRANCH  
 PRESORT ONE-SHOT BRANCH,  
 SUMMARY FLOW CHART OF PRESORT "ONE-SHOT" BRANCH,  
 3-3

BRT  
 CARD CALL FOR COLLATE C ON A BRT WITH PARAMETERS ON  
 CARDS, 8-2  
 CARD CALL FOR TAPE SORT C ON A BRT WITH PARAMETERS  
 ON CARDS, 4-3  
 LOADING COLLATE C FROM A BRT, 9-1  
 LOADING COLLATE C FROM A BRT BY PROGRAMMED CALL, 9-1  
 LOADING COLLATE C FROM A BRT WITH PARAMETERS ON  
 CARDS, 9-1  
 LOADING TAPE SORT C FROM A BRT, 5-1  
 LOADING TAPE SORT C FROM A BRT BY PROGRAMMED CALL,  
 5-1  
 LOADING TAPE SORT C FROM A BRT WITH PARAMETERS ON  
 CARDS, 5-1  
 PROGRAMMED CALL FOR COLLATE C ON A BRT, 8-2  
 PROGRAMMED CALL FOR TAPE SORT C ON A BRT, 4-2

CALL  
 CARD CALL FOR COLLATE C ON A BRT WITH PARAMETERS ON  
 CARDS, 8-2  
 CARD CALL FOR COLLATE C PROGRAM AND PARAMETERS ON  
 CARDS, 8-4  
 CARD CALL FOR TAPE SORT C ON A BRT WITH PARAMETERS  
 ON CARDS, 4-3  
 CARD CALL FOR TAPE SORT C PROGRAM AND PARAMETERS ON  
 CARDS, 4-4  
 PROGRAMMED CALL,  
 LOADING COLLATE C FROM A BRT BY PROGRAMMED CALL,  
 9-1  
 LOADING COLLATE C FROM AN OBJECT PROGRAM DECK BY  
 PROGRAMMED CALL, 9-2  
 LOADING TAPE SORT C FROM A BRT BY PROGRAMMED  
 CALL, 5-1  
 LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK  
 BY PROGRAMMED CALL, 5-4  
 PROGRAMMED CALL FOR COLLATE C ON A BRT, 8-2  
 PROGRAMMED CALL FOR COLLATE C PROGRAM ON CARDS, 8-3  
 PROGRAMMED CALL FOR TAPE SORT C ON A BRT, 4-2  
 PROGRAMMED CALL FOR TAPE SORT C ON CARDS, 4-3

CALLING  
 \* AND LOADING COLLATE C FROM CARDS, 8-2  
 \* AND LOADING COLLATE C WITH A TAPE LOADER-MONITOR,  
 8-1  
 \* AND LOADING OWN-CODING, 8-4  
 \* AND LOADING TAPE SORT C FROM CARDS, 4-3  
 \* AND LOADING TAPE SORT C OWN-CODING, 4-5  
 \* AND LOADING TAPE SORT C WITH A TAPE LOADER-MONITOR,  
 4-2

CAPABILITIES  
 RESTART CAPABILITIES, 1-8

CARD  
 \* CALL,  
 CARD CALL FOR COLLATE C ON A BRT WITH PARAMETERS  
 ON CARDS, 8-2  
 CARD CALL FOR COLLATE C PROGRAM AND PARAMETERS  
 ON CARDS, 8-4  
 CARD CALL FOR TAPE SORT C ON A BRT WITH  
 PARAMETERS ON CARDS, 4-3  
 CARD CALL FOR TAPE SORT C PROGRAM AND PARAMETERS  
 ON CARDS, 4-4  
 \* IDENTIFICATION,  
 PARAMETER CARD IDENTIFICATION, 2-9, 7-6  
 \* INPUT,  
 (CONT.)

CARD (CONT.)  
 CARD INPUT TO TAPE SORT C, 5-2  
 PARAMETER CARD INPUT FOR LOADING COLLATE C  
 PARAMETERS, 8-3  
 PARAMETER CARD INPUT FOR LOADING TAPE SORT C  
 FROM TAPE, 4-4  
 TAPE SORT C LABEL OPTIONS FOR CARD INPUT, 4-6

CARDS  
 CALLING AND LOADING COLLATE C FROM CARDS, 8-2  
 CALLING AND LOADING TAPE SORT C FROM CARDS, 4-3  
 CARD CALL FOR COLLATE C ON A BRT WITH PARAMETERS ON  
 CARDS, 8-2  
 CARD CALL FOR COLLATE C PROGRAM AND PARAMETERS ON  
 CARDS, 8-4  
 CARD CALL FOR TAPE SORT C ON A BRT WITH PARAMETERS  
 ON CARDS, 4-3  
 CARD CALL FOR TAPE SORT C PROGRAM AND PARAMETERS ON  
 CARDS, 4-4  
 LOADING COLLATE C FROM A BRT WITH PARAMETERS ON  
 CARDS, 9-1  
 LOADING COLLATE C PROGRAM AND PARAMETERS FROM CARDS,  
 9-2  
 LOADING TAPE SORT C FROM A BRT WITH PARAMETERS ON  
 CARDS, 5-1  
 LOADING TAPE SORT C PARAMETERS AND PROGRAM FROM  
 CARDS, 5-2  
 PARAMETER CARDS,  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 COLLATE C WITH, 9-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 COLLATE C WITHOUT, 9-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITH, 5-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITHOUT, 5-5  
 PROGRAMMED CALL FOR COLLATE C PROGRAM ON CARDS, 8-3  
 PROGRAMMED CALL FOR TAPE SORT C ON CARDS, 4-3

CHANGE  
 OUTPUT FILE NAME CHANGE, 2-10, 7-7

CHARACTERISTICS  
 \* OF COLLATE C, 6-1  
 \* OF TAPE SORT C, 1-1

CHARACTERS  
 BANNER CHARACTERS, 2-5, 7-4  
 PARAMETER CHARACTERS,  
 EFFECTS OF PARAMETER CHARACTERS 15, 16, AND 19,  
 2-13  
 PARAMETER CHARACTERS FOR COLLATE KEY FIELDS 1-10,  
 7-7  
 PARAMETER CHARACTERS FOR KEY FIELDS 1-THROUGH 10,  
 2-10  
 TAPE POSITIONING FOR EFFECTS OF CHARACTERS 15, 16,  
 AND 19, 2-14

CHART  
 SUMMARY FLOW CHART OF PRESORT "ONE-SHOT" BRANCH, 3-3

CHECK  
 \* AND CHECKPOINT PROCEDURES, 5-4, 9-4  
 \* AND CHECKPOINT TERMINOLOGY FOR COLLATE C, 6-2  
 \* AND CHECKPOINT TERMINOLOGY FOR TAPE SORT C, 1-2  
 \* MESSAGE CODING, 5-6, 9-4  
 \* OPTION,  
 LAST-PASS CHECK OPTION, 2-7

CHECKPOINT  
 \* IDENTIFICATION,  
 TAPE SORT C CHECKPOINT IDENTIFICATION AND  
 CORRECTIVE ACTION, 5-7  
 \* PROCEDURES,  
 CHECK AND CHECKPOINT PROCEDURES, 5-4, 9-4  
 \* TERMINOLOGY,  
 CHECK AND CHECKPOINT TERMINOLOGY FOR COLLATE C,  
 6-2  
 CHECK AND CHECKPOINT TERMINOLOGY FOR TAPE SORT  
 C, 1-2

CHECKPOINTS  
 COLLATE C CHECKPOINTS, 9-5

CODING  
 CHECK MESSAGE CODING, 5-6, 9-4

COLLATE  
 BEGINNING OF COLLATE, 7-8  
 \* C BASIC CONCEPTS, 6-1  
 \* C CHECKPOINTS, 9-5  
 \* C LABEL OPTIONS, 7-3  
 \* C OPERATING CONSIDERATIONS, 8-1  
 \* C OPERATING PROCEDURES, 9-1  
 \* C OWN-CODING, 7-9  
 \* C PARAMETER SPECIFICATION, 8-4  
 \* C PARAMETER SUMMARY, 8-4  
 (CONT.)

COMPUTER-GENERATED INDEX

COLLATE (CONT.)

- " C PROGRAM FORMAT, 8-1
- " C PROGRAM SPECIALIZATION, 7-1
- " C TAPE POSITIONING, 7-9
- " C TIMING, A-4
- CARD CALL FOR COLLATE C ON A BRT WITH PARAMETERS ON CARDS, 8-2
- CARD CALL FOR COLLATE C PROGRAM AND PARAMETERS ON CARDS, 8-4
- CHARACTERISTICS OF COLLATE C, 6-1
- CHECK AND CHECKPOINT TERMINOLOGY FOR COLLATE C, 6-2
- END OF COLLATE, 7-8
- HOW TO USE COLLATE C OWN-CODING, 7-9
- " KEY FIELDS 1-10,
  - PARAMETER CHARACTERS FOR COLLATE KEY FIELDS 1-10, 7-7
- " KEYS AND KEYFIELDS, 6-3
- LOADING COLLATE,
  - CALLING AND LOADING COLLATE C FROM CARDS, 8-2
  - CALLING AND LOADING COLLATE C WITH A TAPE LOADER-MONITOR, 8-1
  - PARAMETER CARD INPUT FOR LOADING COLLATE C PARAMETERS, 8-3
  - SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING COLLATE C WITH, 9-3
  - SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING COLLATE C WITHOUT, 9-3
- LOADING COLLATE C FROM A BRT BY PROGRAMMED CALL, 9-1
- LOADING COLLATE C FROM A BRT WITH PARAMETERS ON CARDS, 9-1
- LOADING COLLATE C FROM A BRT, 9-1
- LOADING COLLATE C FROM AN OBJECT PROGRAM DECK BY PROGRAMMED CALL, 9-2
- LOADING COLLATE C FROM AN OBJECT PROGRAM DECK, 9-2
- LOADING COLLATE C PROGRAM AND PARAMETERS FROM CARDS, 9-2
- " PROCESS, 6-4
  - SAMPLE COLLATE PROCESS, 6-2
- " PROCESSING OF TWO INPUT FILES, 6-5
- PROGRAMMED CALL FOR COLLATE C ON A BRT, 8-2
- PROGRAMMED CALL FOR COLLATE C PROGRAM ON CARDS, 8-3
- TIMING FORMULAS FOR COLLATE C, A-4
- TIMING FORMULAS FOR TAPE SORT C AND COLLATE C, A-1

COLLATING SEQUENCE TRANSLATION, 7-5

CONCEPTS

- BASIC CONCEPTS OF TAPE SORT C, 1-1

CONCEPTS

- BASIC CONCEPTS,
  - COLLATE C BASIC CONCEPTS, 6-1

CONSIDERATIONS

- " FOR PROGRAMMING OWN-CODING, 3-9
- OPERATING CONSIDERATIONS,
  - COLLATE C OPERATING CONSIDERATIONS, 8-1
  - TAPE SORT C OPERATING CONSIDERATIONS, 4-1
- PRELIMINARY CONSIDERATIONS, 5-1, 9-1

CONTROL

- PRINTER CONTROL, 2-6, 7-4

CORRECTIVE ACTION

- TAPE SORT C CHECKPOINT IDENTIFICATION AND CORRECTIVE ACTION, 5-7

DATA

- " RECORDS,
  - NUMBER OF DATA RECORDS PER OUTPUT REEL, 7-6
- " REQUIREMENTS, 1-5, 6-3

DECK

- " FORMAT,
  - SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING COLLATE C WITH, 9-3
  - SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING COLLATE C WITHOUT, 9-3
  - SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING TAPE SORT C WITH, 5-3
  - SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING TAPE SORT C WITHOUT, 5-5
- OBJECT PROGRAM DECK,
  - LOADING COLLATE C FROM AN OBJECT PROGRAM DECK, 9-2
  - LOADING COLLATE C FROM AN OBJECT PROGRAM DECK BY PROGRAMMED CALL, 9-2
  - LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK, 5-2
  - LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK BY PROGRAMMED CALL, 5-4

DELETING

- " AN ITEM, 3-4, 3-7

DESCENDING SEQUENCED OUTPUT

- ASCENDING OR DESCENDING SEQUENCED OUTPUT, 2-8, 7-5

DESCRIPTION (CONT.)

DESCRIPTION

- SUMMARY DESCRIPTION OF TAPE SORT C, 1-3

DESIGNATION

- PARAMETER DESIGNATION, 2-1

DIRECTION

- SEARCH DIRECTION FOR LAST-PASS OWN-CODING, 2-10
- SEARCH DIRECTION FOR NEXT PROGRAM, 2-11, 7-8
- SEARCH DIRECTION FOR RESTART SEGMENT, 5-10

EFFECTS

- " OF PARAMETER CHARACTERS 15, 16, AND 19, 2-13
- TAPE POSITIONING FOR EFFECTS OF CHARACTERS 15, 16, AND 19, 2-14

END

- " OF COLLATE, 7-8
- " OF SORT, 2-11

ENTRIES

- RESTART ENTRIES, 5-9

ERROR OPTIONS, 2-6, 7-4

FAILURE

- RESTART FAILURE, 5-10

FIELDS

- KEY FIELDS, 2-9, 7-6
- SORT KEYS AND KEY FIELDS, 1-7
- " 1-THROUGH,
  - PARAMETER CHARACTERS FOR KEY FIELDS 1-THROUGH 10, 2-10
- " 1-10,
  - PARAMETER CHARACTERS FOR COLLATE KEY FIELDS 1-10, 7-7

FILE NAME

- " CHANGE,
  - OUTPUT FILE NAME CHANGE, 2-10, 7-7
- INPUT FILE NAME AND REEL NUMBER VERIFICATION, 7-7
- " VERIFICATION,
  - INPUT FILE NAME VERIFICATION, 2-10

FILES

- INPUT FILES,
  - COLLATE PROCESSING OF TWO INPUT FILES, 6-5

FLOW CHART

- SUMMARY FLOW CHART OF PRESORT "ONE-SHOT" BRANCH, 3-3

FORMAT

- PROGRAM FORMAT, 4-1
- COLLATE C PROGRAM FORMAT, 8-1
- TAPE SORT C PROGRAM FORMAT, 4-1
- SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING COLLATE C WITH, 9-3
- SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING COLLATE C WITHOUT, 9-3
- SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING TAPE SORT C WITH, 5-3
- SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING TAPE SORT C WITHOUT, 5-5

FORMULAS

- TIMING FORMULAS FOR COLLATE C, A-4
- TIMING FORMULAS FOR TAPE SORT C AND COLLATE C, A-1
- TIMING FORMULAS FOR TAPE SORT C, A-1

GAP

- RECORD GAP,
  - PARITY AND RECORD GAP, 2-6, 7-4

HASH TOTALS, 3-9

HEADER-TRAILER OWN-CODING

- LAST-PASS HEADER-TRAILER OWN-CODING, 3-6
- PRESORT HEADER-TRAILER OWN-CODING, 3-1

HISTORY

- PROGRAM HISTORY, 1-7
- SORT PROGRAM HISTORY, 2-7

IDENTIFICATION

- CHECKPOINT IDENTIFICATION,
  - TAPE SORT C CHECKPOINT IDENTIFICATION AND CORRECTIVE ACTION, 5-7
- PARAMETER CARD IDENTIFICATION, 2-9, 7-6

INPUT

- CARD INPUT,
  - TAPE SORT C LABEL OPTIONS FOR CARD INPUT, 4-6
  - CARD INPUT TO TAPE SORT C, 5-2
  - ENTERING THE INPUT TO BE SORTED, 4-5
  - " FILE NAME AND REEL NUMBER VERIFICATION, 7-7
  - " FILE NAME VERIFICATION, 2-10
- " FILES,
  - COLLATE PROCESSING OF TWO INPUT FILES, 6-5
- MULTI-REEL INPUT, 3-2
- OWN-CODING INPUT, 3-6
- PARAMETER CARD INPUT FOR LOADING COLLATE C PARAMETERS, 8-3
- PARAMETER CARD INPUT FOR LOADING TAPE SORT C FROM TAPE, 4-4
- READ BACKWARDS INPUT, 3-2

(CONT.)

COMPUTER-GENERATED INDEX

INPUT (CONT.)  
 " RECORDS,  
   UNREADABLE INPUT RECORDS, 1-7, 6-4  
 " TAPE,  
   POSITIONING THE INPUT TAPE, 5-10  
   TAPE INPUT,  
     TAPE SORT C LABEL OPTIONS FOR TAPE INPUT, 2-5  
 INSPECTING  
   " OR MODIFYING AN ITEM, 3-4, 3-7  
 INTRODUCTION, 1-1, 4-1, 6-1, 8-1  
 ITEM  
   ADDING AN ITEM, 3-4, 3-8  
   " AND RECORD SIZES, 2-3, 7-2  
   DELETING AN ITEM, 3-4, 3-7  
   INSPECTING OR MODIFYING AN ITEM, 3-4, 3-7  
   SORT ITEM,  
     WORD MARK LOCATIONS IN A SORT ITEM, 3-5  
 ITEM-BY-ITEM OWN-CODING  
   LAST-PASS ITEM-BY-ITEM OWN-CODING, 3-7  
   PRESORT ITEM-BY-ITEM OWN-CODING, 3-2  
 KEY FIELDS, 2-9, 7-6  
   SORT KEYS AND KEY FIELDS, 1-7  
 " 1-THROUGH,  
   PARAMETER CHARACTERS FOR KEY FIELDS 1-THROUGH  
     10, 2-10  
 " 1-10,  
   PARAMETER CHARACTERS FOR COLLATE KEY FIELDS  
     1-10, 7-7  
 KEYFIELDS  
   COLLATE KEYS AND KEYFIELDS, 6-3  
 KEYS  
   COLLATE KEYS AND KEYFIELDS, 6-3  
   SORT KEYS AND KEY FIELDS, 1-7  
 LABEL  
   " OPTIONS,  
     COLLATE C LABEL OPTIONS, 7-3  
     TAPE SORT C LABEL OPTIONS FOR CARD INPUT, 4-6  
     TAPE SORT C LABEL OPTIONS FOR TAPE INPUT, 2-5  
   " VERIFICATION,  
     OUTPUT LABEL VERIFICATION, 7-3  
 LABELS  
   TAPE LABELS, 2-4, 7-3  
 LAST-PASS  
   " CHECK OPTION, 2-7  
   " HEADER-TRAILER OWN-CODING, 3-6  
   " ITEM-BY-ITEM OWN-CODING, 3-7  
   " OWN-CODING, 2-9, 3-6  
     SEARCH DIRECTION FOR LAST-PASS OWN-CODING, 2-10  
     TERMINATING LAST-PASS OWN-CODING, 3-8  
   " OWN-CODING PROGRAM NAME, 2-10  
   " OWN-CODING SEGMENT NAME, 2-10  
   " PROCESS DETERMINATION, 3-7  
 LOADER-MONITOR  
   TAPE LOADER-MONITOR,  
     CALLING AND LOADING COLLATE C WITH A TAPE  
       LOADER-MONITOR, 8-1  
     CALLING AND LOADING TAPE SORT C WITH A TAPE  
       LOADER-MONITOR, 4-2  
 LOADING  
   " COLLATE,  
     CALLING AND LOADING COLLATE C FROM CARDS, 8-2  
     CALLING AND LOADING COLLATE C WITH A TAPE  
       LOADER-MONITOR, 8-1  
     LOADING COLLATE C FROM A BRT BY PROGRAMMED CALL,  
       9-1  
     LOADING COLLATE C FROM A BRT WITH PARAMETERS ON  
       CARDS, 9-1  
     LOADING COLLATE C FROM A BRT, 9-1  
     LOADING COLLATE C FROM AN OBJECT PROGRAM DECK BY  
       PROGRAMMED CALL, 9-2  
     LOADING COLLATE C FROM AN OBJECT PROGRAM DECK,  
       9-2  
     LOADING COLLATE C PROGRAM AND PARAMETERS FROM  
       CARDS, 9-2  
     PARAMETER CARD INPUT FOR LOADING COLLATE C  
       PARAMETERS, 8-3  
     SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
       COLLATE C WITH, 9-3  
     SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
       COLLATE C WITHOUT, 9-3  
   " OWN-CODING,  
     CALLING AND LOADING OWN-CODING, 8-4  
   " TAPE SORT,  
     CALLING AND LOADING TAPE SORT C FROM CARDS, 4-3  
     CALLING AND LOADING TAPE SORT C OWN-CODING, 4-5  
     CALLING AND LOADING TAPE SORT C WITH A TAPE  
       LOADER-MONITOR, 4-2  
 (CONT.)

LOADING (CONT.)  
 LOADING TAPE SORT C FROM A BRT BY PROGRAMMED  
 CALL, 5-1  
 LOADING TAPE SORT C FROM A BRT WITH PARAMETERS  
 ON CARDS, 5-1  
 LOADING TAPE SORT C FROM A BRT, 5-1  
 LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK  
 BY PROGRAMMED CALL, 5-4  
 LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK,  
 5-2  
 LOADING TAPE SORT C PARAMETERS AND PROGRAM FROM  
 CARDS, 5-2  
 PARAMETER CARD INPUT FOR LOADING TAPE SORT C  
 FROM TAPE, 4-4  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITH, 5-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITHOUT, 5-5  
 LOCATIONS  
   WORD MARK LOCATIONS IN A SORT ITEM, 3-5  
 MACHINE REQUIREMENTS, 1-2, 6-2  
 MAGNETIC TAPE UNITS, 1-7, 6-4  
 MARK LOCATIONS  
   WORD MARK LOCATIONS IN A SORT ITEM, 3-5  
 MEMORY ADDRESS  
   HIGHEST MEMORY ADDRESS, 2-8, 7-5  
 MERGE, 1-4  
   SUMMARY OF PRESORT, MERGE, AND LAST PASS, 1-6  
 MESSAGE  
   " CODING,  
     CHECK MESSAGE CODING, 5-6, 9-4  
     RESTART MESSAGE, 5-10  
 MODIFYING  
   INSPECTING OR MODIFYING AN ITEM, 3-4, 3-7  
 MULTI-REEL  
   " INPUT, 3-2  
   " OUTPUT, 3-9  
 NAME  
   " CHANGE,  
     OUTPUT FILE NAME CHANGE, 2-10, 7-7  
     INPUT FILE NAME AND REEL NUMBER VERIFICATION, 7-7  
     LAST-PASS OWN-CODING PROGRAM NAME, 2-10  
     LAST-PASS OWN-CODING SEGMENT NAME, 2-10  
     PROGRAM NAME,  
       NEXT PROGRAM NAME, 2-10, 7-7  
     PROGRAM SEGMENT NAME,  
       NEXT PROGRAM SEGMENT NAME, 2-11  
     SEGMENT NAME,  
       NEXT SEGMENT NAME, 7-7  
   " VERIFICATION,  
     INPUT FILE NAME VERIFICATION, 2-10  
 NUMBER  
   " OF DATA RECORDS PER OUTPUT REEL, 7-6  
   " VERIFICATION,  
     INPUT FILE NAME AND REEL NUMBER VERIFICATION,  
       7-7  
 OBJECT PROGRAM DECK  
   " FORMAT,  
     SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
       COLLATE C WITH, 9-3  
     SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
       COLLATE C WITHOUT, 9-3  
     SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
       TAPE SORT C WITH, 5-3  
     SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
       TAPE SORT C WITHOUT, 5-5  
     LOADING COLLATE C FROM AN OBJECT PROGRAM DECK, 9-2  
     LOADING COLLATE C FROM AN OBJECT PROGRAM DECK BY  
       PROGRAMMED CALL, 9-2  
     LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK, 5-2  
     LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK BY  
       PROGRAMMED CALL, 5-4  
 ONE-SHOT BRANCH  
   SUMMARY FLOW CHART OF PRESORT "ONE-SHOT" BRANCH, 3-3  
 OPERATING  
   " CONSIDERATIONS,  
     COLLATE C OPERATING CONSIDERATIONS, 8-1  
     TAPE SORT C OPERATING CONSIDERATIONS, 4-1  
   " PROCEDURES,  
     COLLATE C OPERATING PROCEDURES, 9-1  
     TAPE SORT C OPERATING PROCEDURES, 5-1  
 OPTION  
   ERROR OPTIONS, 2-6, 7-4  
 LABEL OPTIONS,  
   COLLATE C LABEL OPTIONS, 7-3  
   TAPE SORT C LABEL OPTIONS FOR CARD INPUT, 4-6  
   TAPE SORT C LABEL OPTIONS FOR TAPE INPUT, 2-5  
 (CONT.)

COMPUTER-GENERATED INDEX

OPTION (CONT.)  
 LAST-PASS CHECK OPTION, 2-7  
 REWIND OPTION, 7-6  
 OPTIONAL PRINTOUTS, 6-4  
 OUTPUT  
 DESCENDING SEQUENCED OUTPUT,  
 ASCENDING OR DESCENDING SEQUENCED OUTPUT, 2-8,  
 7-5  
 \* FILE NAME CHANGE, 2-10, 7-7  
 \* LABEL VERIFICATION, 7-3  
 \* MULTI-REEL OUTPUT, 3-9  
 \* REEL,  
 NUMBER OF DATA RECORDS PER OUTPUT REEL, 7-6  
 OWN-CODING, 2-8, 7-6  
 CALLING AND LOADING TAPE SORT C OWN-CODING, 4-5  
 COLLATE C OWN-CODING, 7-9  
 HOW TO USE COLLATE C OWN-CODING, 7-9  
 \* INPUT, 3-6  
 LAST-PASS HEADER-TRAILER OWN-CODING, 3-6  
 LAST-PASS ITEM-BY-ITEM OWN-CODING, 3-7  
 LAST-PASS OWN-CODING, 2-9, 3-6  
 SEARCH DIRECTION FOR LAST-PASS OWN-CODING, 2-10  
 LOADING OWN-CODING,  
 CALLING AND LOADING OWN-CODING, 8-4  
 PRESORT HEADER-TRAILER OWN-CODING, 3-1  
 PRESORT ITEM-BY-ITEM OWN-CODING, 3-2  
 PRESORT OWN-CODING, 3-1, 2-9  
 \* PROGRAM NAME,  
 LAST-PASS OWN-CODING PROGRAM NAME, 2-10  
 PROGRAMMING OWN-CODING,  
 CONSIDERATIONS FOR PROGRAMMING OWN-CODING, 3-9  
 \* ROUTINES,  
 SAMPLE OWN-CODING ROUTINES, 8-1  
 \* SEGMENT NAME,  
 LAST-PASS OWN-CODING SEGMENT NAME, 2-10  
 TAPE SORT C OWN-CODING, 3-1  
 TERMINATING LAST-PASS OWN-CODING, 3-8  
 TERMINATING PRESORT OWN-CODING, 3-5  
 PADDING, 2-4, 7-2  
 PARAMETER  
 \* CARDS,  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 COLLATE C WITH, 9-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 COLLATE C WITHOUT, 9-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITH, 5-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITHOUT, 5-5  
 \* CHARACTERS,  
 EFFECTS OF PARAMETER CHARACTERS 15, 16, AND 19,  
 2-13  
 PARAMETER CHARACTERS FOR COLLATE KEY FIELDS  
 1-10, 7-7  
 PARAMETER CHARACTERS FOR KEY FIELDS 1-THROUGH  
 10, 2-10  
 \* DESIGNATION, 2-1  
 \* PREPARATION, 7-1  
 \* SPECIFICATION,  
 COLLATE C PARAMETER SPECIFICATION, 8-4  
 TAPE SORT C PARAMETER SPECIFICATION, 4-6  
 \* SUMMARY,  
 COLLATE C PARAMETER SUMMARY, 8-4  
 TAPE SORT C PARAMETER SUMMARY, 4-6  
 PARAMETER CARD  
 \* IDENTIFICATION, 2-9, 7-6  
 \* INPUT,  
 PARAMETER CARD INPUT FOR LOADING COLLATE C  
 PARAMETERS, 8-3  
 PARAMETER CARD INPUT FOR LOADING TAPE SORT C  
 FROM TAPE, 4-4  
 PARAMETERS  
 CARD CALL FOR COLLATE C ON A BRT WITH PARAMETERS ON  
 CARDS, 8-2  
 CARD CALL FOR COLLATE C PROGRAM AND PARAMETERS ON  
 CARDS, 8-4  
 CARD CALL FOR TAPE SORT C ON A BRT WITH PARAMETERS  
 ON CARDS, 4-3  
 CARD CALL FOR TAPE SORT C PROGRAM AND PARAMETERS ON  
 CARDS, 4-4  
 LOADING COLLATE C FROM A BRT WITH PARAMETERS ON  
 CARDS, 9-1  
 LOADING COLLATE C PROGRAM AND PARAMETERS FROM CARDS,  
 9-2  
 LOADING TAPE SORT C FROM A BRT WITH PARAMETERS ON  
 CARDS, 5-1  
 LOADING TAPE SORT C PARAMETERS AND PROGRAM FROM  
 (CONT.)

PARAMETERS (CONT.)  
 CARDS, 5-2  
 PARAMETER CARD INPUT FOR LOADING COLLATE C  
 PARAMETERS, 8-3  
 PARITY  
 \* AND RECORD GAP, 2-6, 7-4  
 PASS  
 LAST PASS, 1-5  
 SUMMARY OF PRESORT, MERGE, AND LAST PASS, 1-6  
 POSITIONING  
 \* OF WORK TAPE 1, 5-9  
 TAPE POSITIONING, 2-11, 7-8  
 COLLATE C TAPE POSITIONING, 7-9  
 TAPE POSITIONING FOR EFFECTS OF CHARACTERS 15, 16,  
 AND 19, 2-14  
 TAPE POSITIONING FOR RESTARTING, 5-9  
 TAPE POSITIONING IN THE TAPE SORT C PROGRAM, 2-12  
 \* THE INPUT TAPE, 5-10  
 PRECAUTIONS  
 RESTART PRECAUTIONS, 5-9  
 PRELIMINARY CONSIDERATIONS, 5-1, 9-1  
 PREPARATION  
 PARAMETER PREPARATION, 7-1  
 PRESORT, 1-3  
 \* HEADER-TRAILER OWN-CODING, 3-1  
 \* ITEM-BY-ITEM OWN-CODING, 3-2  
 \* ONE-SHOT BRANCH,  
 SUMMARY FLOW CHART OF PRESORT "ONE-SHOT" BRANCH,  
 3-3  
 \* OWN-CODING, 3-1, 2-9  
 TERMINATING PRESORT OWN-CODING, 3-5  
 \* PROCESS DETERMINATION, 3-4  
 SUMMARY OF PRESORT, MERGE, AND LAST PASS, 1-6  
 PRINTER CONTROL, 2-6, 7-4  
 PRINTOUTS  
 \* OPTIONAL PRINTOUTS, 6-4  
 PROCEDURES  
 CHECKPOINT PROCEDURES,  
 CHECK AND CHECKPOINT PROCEDURES, 5-4, 9-4  
 OPERATING PROCEDURES,  
 COLLATE C OPERATING PROCEDURES, 9-1  
 TAPE SORT C OPERATING PROCEDURES, 5-1  
 RESTART PROCEDURES, 5-6  
 PROCESS  
 COLLATE PROCESS, 6-4  
 LAST-PASS PROCESS DETERMINATION, 3-7  
 PRESORT PROCESS DETERMINATION, 3-4  
 SAMPLE COLLATE PROCESS, 6-2  
 PROCESSING  
 COLLATE PROCESSING OF TWO INPUT FILES, 6-5  
 PROGRAM  
 CARD CALL FOR COLLATE C PROGRAM AND PARAMETERS ON  
 CARDS, 8-4  
 CARD CALL FOR TAPE SORT C PROGRAM AND PARAMETERS ON  
 CARDS, 4-4  
 \* DECK,  
 LOADING COLLATE C FROM AN OBJECT PROGRAM DECK,  
 9-2  
 LOADING COLLATE C FROM AN OBJECT PROGRAM DECK BY  
 PROGRAMMED CALL, 9-2  
 LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK,  
 5-2  
 LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK  
 BY PROGRAMMED CALL, 5-4  
 \* DECK FORMAT,  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 COLLATE C WITH, 9-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 COLLATE C WITHOUT, 9-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITH, 5-3  
 SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
 TAPE SORT C WITHOUT, 5-5  
 \* FORMAT, 4-1  
 COLLATE C PROGRAM FORMAT, 8-1  
 TAPE SORT C PROGRAM FORMAT, 4-1  
 \* HISTORY, 1-7  
 SORT PROGRAM HISTORY, 2-7  
 LOADING COLLATE C PROGRAM AND PARAMETERS FROM CARDS,  
 9-2  
 LOADING TAPE SORT C PARAMETERS AND PROGRAM FROM  
 CARDS, 5-2  
 \* NAME,  
 LAST-PASS OWN-CODING PROGRAM NAME, 2-10  
 NEXT PROGRAM NAME, 2-10, 7-7  
 PROGRAMMED CALL FOR COLLATE C PROGRAM ON CARDS, 8-3  
 SEARCH DIRECTION FOR NEXT PROGRAM, 2-11, 7-8  
 (CONT.)

COMPUTER-GENERATED INDEX

PROGRAM (CONT.)

- " SEGMENT NAME,
  - NEXT PROGRAM SEGMENT NAME, 2-11
- " SPECIALIZATION,
  - COLLATE C PROGRAM SPECIALIZATION, 7-1
  - TAPE SORT C PROGRAM SPECIALIZATION, 2-1
  - TAPE POSITIONING IN THE TAPE SORT C PROGRAM, 2-12
- PROGRAMMED CALL
  - " FOR COLLATE C ON A BRT, 8-2
  - " FOR COLLATE C PROGRAM ON CARDS, 8-3
  - " FOR TAPE SORT C ON A BRT, 4-2
  - " FOR TAPE SORT C ON CARDS, 4-3
  - LOADING COLLATE C FROM A BRT BY PROGRAMMED CALL, 9-1
  - LOADING COLLATE C FROM AN OBJECT PROGRAM DECK BY PROGRAMMED CALL, 9-2
  - LOADING TAPE SORT C FROM A BRT BY PROGRAMMED CALL, 5-1
  - LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK BY PROGRAMMED CALL, 5-4
- PROGRAMMING OWN-CODING
  - CONSIDERATIONS FOR PROGRAMMING OWN-CODING, 3-9
- READ BACKWARDS INPUT, 3-2
- RECORD
  - " GAP,
    - PARITY AND RECORD GAP, 2-6, 7-4
  - " SIZE,
    - ITEM AND RECORD SIZES, 2-3, 7-2
    - RECORD SIZE DETERMINATION, 2-3
- RECORDS
  - DATA RECORDS,
    - NUMBER OF DATA RECORDS PER OUTPUT REEL, 7-6
  - UNREADABLE INPUT RECORDS, 1-7, 6-4
- REEL
  - " NUMBER VERIFICATION,
    - INPUT FILE NAME AND REEL NUMBER VERIFICATION, 7-7
  - OUTPUT REEL,
    - NUMBER OF DATA RECORDS PER OUTPUT REEL, 7-6
- REQUIREMENTS
  - DATA REQUIREMENTS, 1-5, 6-3
  - MACHINE REQUIREMENTS, 1-2, 6-2
- RESTART
  - " CAPABILITIES, 1-8
  - " ENTRIES, 5-9
  - " FAILURE, 5-10
  - " MESSAGE, 5-10
  - " PRECAUTIONS, 5-9
  - " PROCEDURES, 5-6
  - " SEGMENT,
    - SEARCH DIRECTION FOR RESTART SEGMENT, 5-10
- RESTARTING
  - TAPE POSITIONING FOR RESTARTING, 5-9
- REWIND OPTION, 7-6
- ROUTINES
  - SAMPLE OWN-CODING ROUTINES, B-1
- SAMPLE
  - " COLLATE PROCESS, 6-2
  - " OWN-CODING ROUTINES, B-1
- SEARCH DIRECTION
  - " FOR LAST-PASS OWN-CODING, 2-10
  - " FOR NEXT PROGRAM, 2-11, 7-8
  - " FOR RESTART SEGMENT, 5-10
- SEGMENT
  - " NAME,
    - LAST-PASS OWN-CODING SEGMENT NAME, 2-10
    - NEXT PROGRAM SEGMENT NAME, 2-11
    - NEXT SEGMENT NAME, 7-7
  - RESTART SEGMENT,
    - SEARCH DIRECTION FOR RESTART SEGMENT, 5-10
- SEQUENCE TRANSLATION
  - COLLATING SEQUENCE TRANSLATION, 7-5
- SEQUENCED OUTPUT
  - ASCENDING OR DESCENDING SEQUENCED OUTPUT, 2-8, 7-5
- SIZE
  - RECORD SIZE DETERMINATION, 2-3
  - RECORD SIZES,
    - ITEM AND RECORD SIZES, 2-3, 7-2
- SORT
  - BEGINNING OF SORT, 2-11
  - END OF SORT, 2-11
  - " ITEM,
    - WORD MARK LOCATIONS IN A SORT ITEM, 3-5
  - " KEYS AND KEY FIELDS, 1-7
  - LOADING TAPE SORT,
    - CALLING AND LOADING TAPE SORT C FROM CARDS, 4-3
    - CALLING AND LOADING TAPE SORT C OWN-CODING, 4-5
    - CALLING AND LOADING TAPE SORT C WITH A TAPE (CONT.)

SORT (CONT.)

- LOADER-MONITOR, 4-2
- PARAMETER CARD INPUT FOR LOADING TAPE SORT C FROM TAPE, 4-4
- SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING TAPE SORT C WITH, 5-3
- SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING TAPE SORT C WITHOUT, 5-5
- LOADING TAPE SORT C FROM A BRT BY PROGRAMMED CALL, 5-1
- LOADING TAPE SORT C FROM A BRT WITH PARAMETERS ON CARDS, 5-1
- LOADING TAPE SORT C FROM A BRT, 5-1
- LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK BY PROGRAMMED CALL, 5-4
- LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK, 5-2
- LOADING TAPE SORT C PARAMETERS AND PROGRAM FROM CARDS, 5-2
- " PROGRAM HISTORY, 2-7
- TAPE SORT,
  - BASIC CONCEPTS OF TAPE SORT C, 1-1
  - CARD CALL FOR TAPE SORT C ON A BRT WITH PARAMETERS ON CARDS, 4-3
  - CARD CALL FOR TAPE SORT C PROGRAM AND PARAMETERS ON CARDS, 4-4
  - CARD INPUT TO TAPE SORT C, 5-2
  - CHARACTERISTICS OF TAPE SORT C, 1-1
  - CHECK AND CHECKPOINT TERMINOLOGY FOR TAPE SORT C, 1-2
  - PROGRAMMED CALL FOR TAPE SORT C ON A BRT, 4-2
  - PROGRAMMED CALL FOR TAPE SORT C ON CARDS, 4-3
  - SUMMARY DESCRIPTION OF TAPE SORT C, 1-3
  - TAPE POSITIONING IN THE TAPE SORT C PROGRAM, 2-12
  - TIMING FORMULAS FOR TAPE SORT C AND COLLATE C, A-1
  - TIMING FORMULAS FOR TAPE SORT C, A-1
  - TAPE SORT C CHECKPOINT IDENTIFICATION AND CORRECTIVE ACTION, 5-7
  - TAPE SORT C LABEL OPTIONS FOR CARD INPUT, 4-6
  - TAPE SORT C LABEL OPTIONS FOR TAPE INPUT, 2-5
  - TAPE SORT C OPERATING CONSIDERATIONS, 4-1
  - TAPE SORT C OPERATING PROCEDURES, 5-1
  - TAPE SORT C OWN-CODING, 3-1
  - TAPE SORT C PARAMETER SPECIFICATION, 4-6
  - TAPE SORT C PARAMETER SUMMARY, 4-6
  - TAPE SORT C PROGRAM FORMAT, 4-1
  - TAPE SORT C PROGRAM SPECIALIZATION, 2-1
  - TAPE SORT C TIMING, A-1
- SORTED
  - ENTERING THE INPUT TO BE SORTED, 4-5
- SPECIALIZATION
  - PROGRAM SPECIALIZATION,
    - COLLATE C PROGRAM SPECIALIZATION, 7-1
    - TAPE SORT C PROGRAM SPECIALIZATION, 2-1
- SPECIFICATION
  - PARAMETER SPECIFICATION,
    - COLLATE C PARAMETER SPECIFICATION, 8-4
    - TAPE SORT C PARAMETER SPECIFICATION, 4-6
  - TAPE UNIT SPECIFICATIONS, 2-1, 7-1
- SUGGESTED OBJECT PROGRAM DECK FORMAT
  - " FOR LOADING COLLATE C WITH, 9-3
  - " FOR LOADING COLLATE C WITHOUT, 9-3
  - " FOR LOADING TAPE SORT C WITH, 5-3
  - " FOR LOADING TAPE SORT C WITHOUT, 5-5
- SUMMARY
  - " DESCRIPTION OF TAPE SORT C, 1-3
  - " FLOW CHART OF PRESORT "ONE-SHOT" BRANCH, 3-3
  - " OF PRESORT, MERGE, AND LAST PASS, 1-6
  - PARAMETER SUMMARY,
    - COLLATE C PARAMETER SUMMARY, 8-4
    - TAPE SORT C PARAMETER SUMMARY, 4-6
- TAPE, 7-8
  - " INPUT,
    - TAPE SORT C LABEL OPTIONS FOR TAPE INPUT, 2-5
  - INPUT TAPE,
    - POSITIONING THE INPUT TAPE, 5-10
  - " LABELS, 2-4, 7-3
  - " LOADER-MONITOR,
    - CALLING AND LOADING COLLATE C WITH A TAPE LOADER-MONITOR, 8-1
    - CALLING AND LOADING TAPE SORT C WITH A TAPE LOADER-MONITOR, 4-2
  - PARAMETER CARD INPUT FOR LOADING TAPE SORT C FROM TAPE, 4-4
  - " POSITIONING, 2-11, 7-8
    - COLLATE C TAPE POSITIONING, 7-9
- (CONT.)

COMPUTER-GENERATED INDEX

TAPE (CONT.)

TAPE POSITIONING FOR EFFECTS OF CHARACTERS 15,  
16, AND 19, 2-14  
TAPE POSITIONING FOR RESTARTING, 5-9  
TAPE POSITIONING IN THE TAPE SORT C PROGRAM,  
2-12

" SORT,  
BASIC CONCEPTS OF TAPE SORT C, 1-1  
CALLING AND LOADING TAPE SORT C FROM CARDS, 4-3  
CALLING AND LOADING TAPE SORT C OWN-CODING, 4-5  
CALLING AND LOADING TAPE SORT C WITH A TAPE  
LOADER-MONITOR, 4-2  
CARD CALL FOR TAPE SORT C ON A BRT WITH  
PARAMETERS ON CARDS, 4-3  
CARD CALL FOR TAPE SORT C PROGRAM AND PARAMETERS  
ON CARDS, 4-4  
CARD INPUT TO TAPE SORT C, 5-2  
CHARACTERISTICS OF TAPE SORT C, 1-1  
CHECK AND CHECKPOINT TERMINOLOGY FOR TAPE SORT  
C, 1-2  
LOADING TAPE SORT C FROM A BRT BY PROGRAMMED  
CALL, 5-1  
LOADING TAPE SORT C FROM A BRT WITH PARAMETERS  
ON CARDS, 5-1  
LOADING TAPE SORT C FROM A BRT, 5-1  
LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK  
BY PROGRAMMED CALL, 5-4  
LOADING TAPE SORT C FROM AN OBJECT PROGRAM DECK,  
5-2  
LOADING TAPE SORT C PARAMETERS AND PROGRAM FROM  
CARDS, 5-2  
PARAMETER CARD INPUT FOR LOADING TAPE SORT C  
FROM TAPE, 4-4  
PROGRAMMED CALL FOR TAPE SORT C ON A BRT, 4-2  
PROGRAMMED CALL FOR TAPE SORT C ON CARDS, 4-3  
SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
TAPE SORT C WITH, 5-3  
SUGGESTED OBJECT PROGRAM DECK FORMAT FOR LOADING  
TAPE SORT C WITHOUT, 5-5  
SUMMARY DESCRIPTION OF TAPE SORT C, 1-3  
TAPE POSITIONING IN THE TAPE SORT C PROGRAM,  
2-12  
TAPE SORT C CHECKPOINT IDENTIFICATION AND  
CORRECTIVE ACTION, 5-7  
TAPE SORT C LABEL OPTIONS FOR CARD INPUT, 4-6  
TAPE SORT C LABEL OPTIONS FOR TAPE INPUT, 2-5  
TAPE SORT C OPERATING CONSIDERATIONS, 4-1  
TAPE SORT C OPERATING PROCEDURES, 5-1  
TAPE SORT C OWN-CODING, 3-1  
TAPE SORT C PARAMETER SPECIFICATION, 4-6  
TAPE SORT C PARAMETER SUMMARY, 4-6  
TAPE SORT C PROGRAM FORMAT, 4-1  
TAPE SORT C PROGRAM SPECIALIZATION, 2-1

TAPE SORT C TIMING, A-1  
TIMING FORMULAS FOR TAPE SORT C, A-1  
TIMING FORMULAS FOR TAPE SORT C AND COLLATE C,  
A-1  
" UNIT SPECIFICATIONS, 2-1, 7-1  
" UNITS,  
MAGNETIC TAPE UNITS, 1-7, 6-4  
WORK TAPE,  
POSITIONING OF WORK TAPE 1, 5-9

TERMINATING  
" LAST-PASS OWN-CODING, 3-8  
" PRESORT OWN-CODING, 3-5

TERMINOLOGY  
CHECKPOINT TERMINOLOGY,  
CHECK AND CHECKPOINT TERMINOLOGY FOR COLLATE C,  
6-2  
CHECK AND CHECKPOINT TERMINOLOGY FOR TAPE SORT  
C, 1-2

TIMING  
COLLATE C TIMING, A-4  
" FORMULAS,  
TIMING FORMULAS FOR COLLATE C, A-4  
TIMING FORMULAS FOR TAPE SORT C AND COLLATE C,  
A-1  
TIMING FORMULAS FOR TAPE SORT C, A-1  
TAPE SORT C TIMING, A-1

TOTALS  
HASH TOTALS, 3-9

TRANSLATION  
COLLATING SEQUENCE TRANSLATION, 7-5

UNIT SPECIFICATIONS  
TAPE UNIT SPECIFICATIONS, 2-1, 7-1

UNITS  
MAGNETIC TAPE UNITS, 1-7, 6-4  
UNREADABLE INPUT RECORDS, 1-7, 6-4

VERIFICATION  
INPUT FILE NAME VERIFICATION, 2-10  
OUTPUT LABEL VERIFICATION, 7-3  
REEL NUMBER VERIFICATION,  
INPUT FILE NAME AND REEL NUMBER VERIFICATION,  
7-7

WORD MARK LOCATIONS IN A SORT ITEM, 3-5

WORK TAPE  
POSITIONING OF WORK TAPE 1, 5-9

1-THROUGH  
KEY FIELDS 1-THROUGH,  
PARAMETER CHARACTERS FOR KEY FIELDS 1-THROUGH  
10, 2-10

1-10  
COLLATE KEY FIELDS 1-10,  
PARAMETER CHARACTERS FOR COLLATE KEY FIELDS  
1-10, 7-7

NOTES

NOTES

HONEYWELL EDP TECHNICAL PUBLICATIONS  
USERS' REMARKS FORM

TITLE: TAPE SORT C AND COLLATE C  
SOFTWARE MANUAL

DATED: MAY, 1966

FILE NO: 123.6005.021C.2-018

ERRORS NOTED:

Fold

SUGGESTIONS FOR IMPROVEMENT:

Fold

FROM: NAME \_\_\_\_\_

DATE \_\_\_\_\_

COMPANY \_\_\_\_\_

TITLE \_\_\_\_\_

ADDRESS \_\_\_\_\_

\_\_\_\_\_

Cut Along Line

**BUSINESS REPLY MAIL**

No postage stamp necessary if mailed in the United States  
POSTAGE WILL BE PAID BY

**HONEYWELL**

ELECTRONIC DATA PROCESSING DIVISION

**60 WALNUT STREET**

**WELLESLEY HILLS, MASS. 02181**

ATT'N: TECHNICAL COMMUNICATIONS DEPARTMENT

FIRST CLASS  
PERMIT NO. 39531  
WELLESLEY HILLS  
MASS.

Cut Along Line

**Honeywell**  
ELECTRONIC DATA PROCESSING