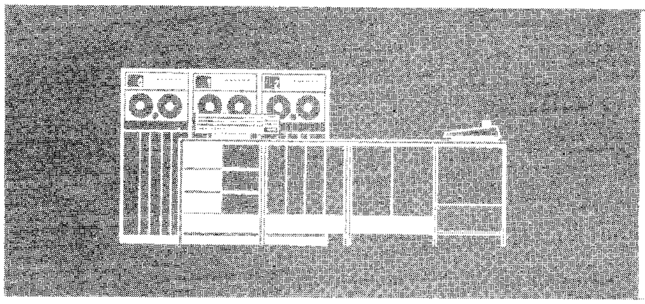
R. CRANE, U.B.C. TABULATING.



SORT 2 AND COLLATE 2

HONEYWELL INFORMATION BULLETIN

GENERAL SYSTEMS	Honeywell 200 Programming and Operating
SUBJECT............................	Sort 2 and Collate 2: Magnetic Tape Sorting and Collating Programs for the Honeywell 200
DATE..............................	October 9, 1964

This preliminary information bulletin will be superseded by more complete descriptions and operational characteristics in forthcoming publications.

HONEYWELL EDP, WELLESLEY HILLS, MASSACHUSETTS 02181

DSI-328
51064
Printed in U.S.A.

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

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I
BASIC CONCEPTS

INTRODUCTION

A large percentage of data processing involves sorting, which is the arrangement of randomly ordered data in a particular sequence and, thus, the creation of a sequentially ordered data file.

Sort 2 is a general-purpose program which performs basic sorting functions on a Honeywell 200 equipped with 1/2-inch magnetic tape. Because it is a general-purpose program, Sort 2 can adapt itself to different data formats and equipment configurations which enable it to handle a wide variety of sorting applications. The information used to specialize Sort 2 for a particular sorting application is inserted at the beginning of the sort program by means of parameters specified by the user.

A Sort 2 program is composed of three logical segments: presort, merge, and last pass. The presort segment accepts the input data and produces ordered groups of data on from two to five tape reels. The merge segment combines the groups until one contiguous group remains on each reel, and the last pass further combines these groups into a single output tape 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 prepared using the Extended Easycoder Assembly language with Advanced Programming Instructions, are referred to as "own-coding" (see Section III). Own-coding can be used to pre-process input files, eliminate duplicate data, and process output data as its final position in the output file is established.

CHARACTERISTICS OF SORT 2

The activities performed by Sort 2 are governed by the following characteristics:

1. Sorts fixed-length items blocked one or more 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. Sorts according to control information contained in up to ten sort key fields in each item;
5. Sorts up to one full reel of records (input or output data can be contained on any number of reels, but the total input data cannot exceed one full reel);
6. Labels output tapes as specified by the user;
7. Provides remedial activities for handling unreadable records;

8. Provides for the inclusion of own-coding elements; and

9. Provides the final output on a specified tape drive (work tape 1).

MACHINE REQUIREMENTS

Sort 2 is used with an H-200 having the following equipment configuration:

1. A minimum main memory capacity of 8, 192 characters, but up to 32K may be used to advantage.

2. A minimum of three tape drives using 1/2-inch tape. Up to five additional tape 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. A card reader or additional tape drive for program loading.

SUMMARY DESCRIPTION OF SORT 2

Sort 2 is a tri-segmented program. The presort segment begins the actual sorting by reading a number of items from an input file, arranging these items in ordered sequences called "strings," and writing these strings on from two to five output tapes. The merge segment sequentially combines the presorted strings into fewer and longer strings during a series of merge phases which ultimately result in only one long string on each work tape. The last-pass segment sequentially combines these strings into one contiguous sequence – the sorted file.

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, minimizing both the total number of strings and the time required to complete the merge segment of the program. These features are especially evident if any pre-ordering exists in the input file (i. e., if some groups of items happen to be in nearly the final output sequence).

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

1. A number of items are transferred from the input file to item storage areas in the main memory.

2. The addresses of the items are stored in pairs in another area in the main memory.

3. From each pair, the address which corresponds to the item having the lower key is selected and paired with another address which was similarly selected.

4. The pairing of selected addresses continues until the address of the item having the lowest key is determined. The item associated with this key becomes the initial item of a string.

5. The item is moved to an output area, and the item storage area in which it was initially stored receives a new item from the input tape.

6. The key of the new item is compared to the key of the item it replaced. If the new key is equal or larger, the new item is considered as an eventual entry in the string currently being formed. If the new key is smaller, the item remains in storage, but the key will not be considered until the creation of the next string.

7. Formation of the current string is terminated when all the keys in storage are of lower value than the key of the last item moved to the output area.

8. A new string is then created by repeating steps 2 – 7 above.

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 phases needed in 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 eliminates all rewind time from the merge segment; for example, once the presorted data has 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 strings in a prescribed ratio on all but one of the work tapes (see A and B in Figure 1).

2. The merge now combines strings taken from these input work tapes (A and B) to produce longer strings on the work tape left empty, 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.

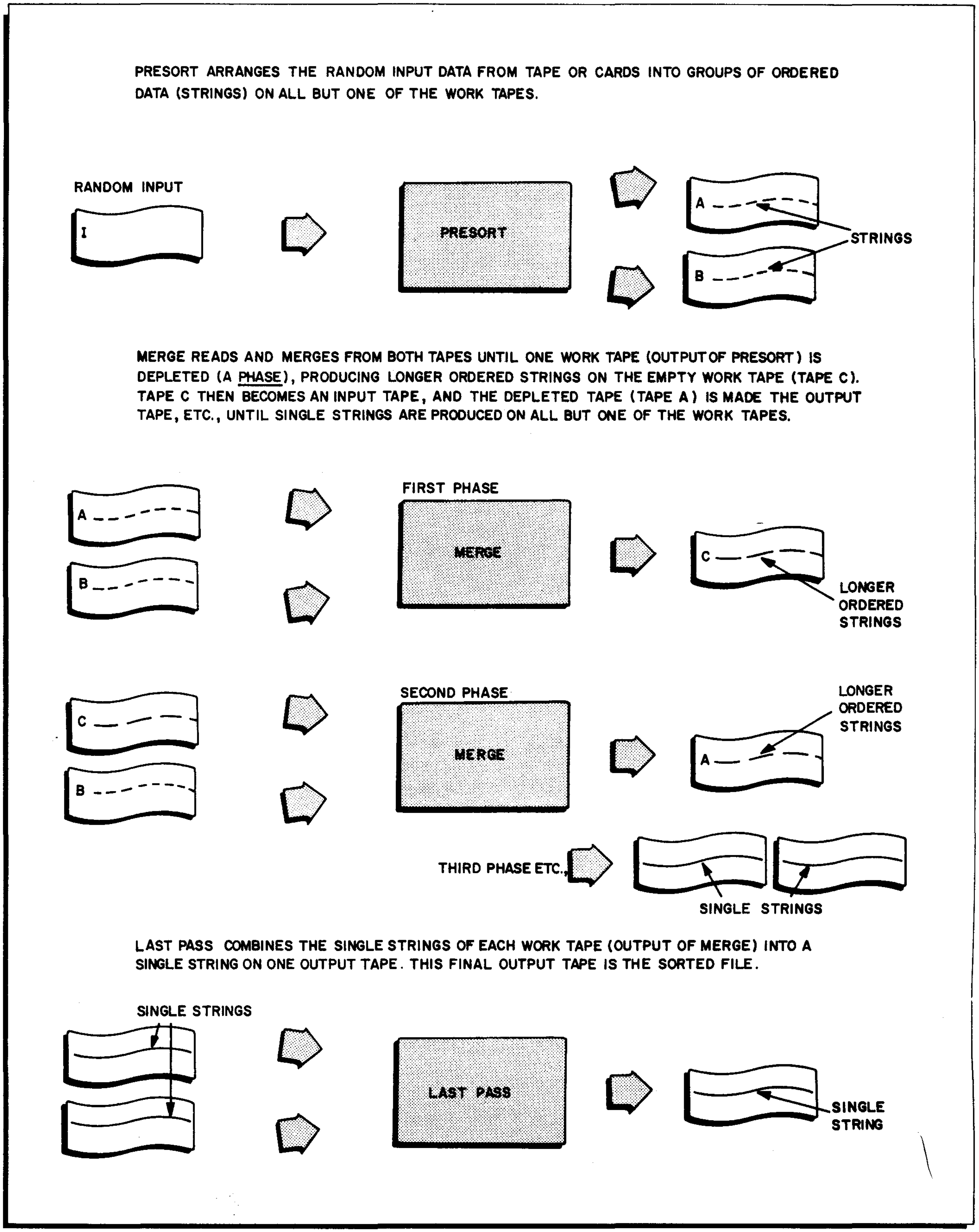
4. The previous phase's output tape (C) will now act as an input work tape to be read backward immediately for the next merge phase. The data on this new input work tape will be sequentially combined with the data remaining on the input work tape (B) which was not depleted in the previous merge phase.

5. The merge continues through as many phases as necessary to produce a single string on all but one of the work tapes. Note that a "phase" is completed each time an input work tape has been depleted, and a "pass" is completed each time all the input data are processed.

The Last Pass

The last pass performs the last merge pass 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. In this illustration, the Sort 2 program uses a minimum of three tape drives, and the primary input



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

MERGE READS AND MERGES FROM BOTH TAPES UNTIL ONE WORK TAPE (OUTPUT OF PRESORT) IS DEPLETED (A PHASE), PRODUCING LONGER ORDERED STRINGS ON THE EMPTY WORK TAPE (TAPE C). TAPE C THEN BECOMES AN INPUT TAPE, AND THE DEPLETED TAPE (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. THIS FINAL OUTPUT TAPE IS THE SORTED FILE.

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

tape (I) is to be read forward. 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 the merge output data being written after the end-of-file record of the primary input data.

DATA REQUIREMENTS

The Sort 2 program processes fixed-length items; and Sort 2 input and output files are usually blocked, that is, several items are combined to form one tape record. The following formulas may be used to determine the maximum and optimum record sizes for both input and output:

$$\text{Maximum record length} = \frac{\text{HMA - MON - PROG}}{\text{NTm} + 2} .$$

$$\text{Optimum record length} = \frac{\text{HMA - MON - PROG}}{\text{2NTm}} .$$

HMA – Highest memory address available to the sort (parameter characters 45-50).

MON – Monitor requirements (1, 340 characters).

PROG – Sort 2 program requirements (approximately 3500 characters).

NTm – Number of tapes used by merge.

For a given sorting application, the user establishes the record size according to the above formulas. It should be noted that the sort will operate most efficiently when the record does not exceed the optimum size.

SORT KEYS

The items to be sorted are identified and sequentially ordered by means of keys. A key is a field of one or more characters within an item, and it is designated as the basis for developing a sequential relationship among all the items to be sorted. Each input item may contain up to ten key fields. These key fields may 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 81-140. 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 (on which the file will be sorted on first) is specified first.

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 parameter characters 40-41.

One or more of the following error responses may be specified. The unreadable record can be:

1. Printed and the machine directed to halt;
2. Printed and deleted from the sort (presort and last pass only); or
3. Not printed or deleted and the machine directed to halt.

Note: If either halt option (1 or 3) is used, there are further options available after the halt: the record may be processed, deleted, or reread, according to the setting of SENSE switches 1 and 2.

MAGNETIC TAPE UNITS

Tape drives for 1/2-inch tape are used by the Sort 2 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.

SECTION II

SORT 2 PROGRAMMING PROCEDURES

PARAMETER DESIGNATION

The information which specializes Sort 2 for a particular sorting application is described to the program by means of parameters designated by the user.

If the Sort 2 program is to be run separately, the parameter characters are normally punched in three parameter cards, as described in Section IV, and read in from the card reader. If Sort 2 is to be entered automatically as one of a series of programs, the user will find it convenient to include coding in the preceding program which initializes the sort by setting up the parameter characters in the memory locations shown in Table II. This eliminates the need for manual intervention at the card reader when the preceding program calls the Loader/Monitor to load Sort 2. Even though the parameter characters are set up by the preceding program (and are therefore essentially fixed), minor modifications can be entered manually from the control panel when Sort 2 is loaded.

Parameter information is summarized in Table II, page 13, the parameter characters are loaded into memory starting at location 2477_8. The following commentary offers an explanation for each parameter field.

Note: When parameters are specified, leading zeros must be designated. For example, the field specifying the number of characters per item (parameter characters 20-23) is designated 0080 if there are 80 characters per item.

Tape Unit Specifications (Parameter Characters 1-19)

The Sort 2 program 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 in characters 1-16. Two characters are required to address a tape drive: the first character specifies the tape control unit, and the second character designates a particular tape drive attached to that unit.

Characters 1-2 specify the address (tape control designation followed by drive number) of the primary input tape. In the event that the input to Sort 2 is on punched cards, character 1 contains a "C" and character 2 is blank COL 3 = "J" COL 4 = 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 an alternate input reel is not used, characters 3 and 4 are blank. If card input is to be used, character 3 specifies the address of the card reader, and character 4 is blank.

Characters 5-6 designate the address of the first work tape. Characters 7-8 designate the address of the second work tape. Characters 9-10 designate the address of an optional third work tape, but these characters are blank if work tapes 3, 4, and 5 are not used. Characters 11-12 designate the address of an optional fourth work tape, but these characters are blank if work tapes 4 and 5 are not used. Characters 13-14 designate the address of an optional fifth work tape, but these characters are blank if no fifth work tape is used.

Characters 15-16 specify the merge work tape. For a merge work tape, the user may specify 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 character 5-14 (see "Effects of Characters 15, 16, and 19" on page 17). If these characters specify a "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. In the event of multi-reel input with alternate drive specified and "ΔR" indicated, the merge work tape will be on the drive from which the second last input reel was removed.

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

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 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.

Item and Record Sizes (Parameter Characters 20-31)

Characters 20-23 specify the number of characters per item. Note that item size must remain fixed throughout the sort. The maximum allowable item size depends on the maximum record size stated for input or output (see page 5). Characters 24-27 designate the number of items per input record. The maximum record size depends on the number of tapes used in the merge and the available memory capacity. Characters 28-31 designate the number of items per output record. The record size limits for output records are the same for input records.

Padding (Parameter Character 32)

If input and output blocking are different, a short record may result at the end of the sorted file. The Sort program automatically pads this record (adds the proper number of padding characters to fill the record) using the character specified in parameter character 32.

Tape Labels (Parameter Characters 33-34)

Characters 33-34 specify tape labeling options as indicated in Table I.

Table I. Label Options

If characters 33-34 are	the beginning of file has	and the end of file is indicated by
ΔΔ	no label	tape mark
ΔT	no label	tape mark and trailer
SS	standard label	standard trailer
SS	standard label and tape mark	tape mark and standard trailer
NΔ	non-standard label	tape mark
*NT	non-standard label	tape mark and trailer

*NT may designate standard leader with tape mark and standard trailer.

The output tape will have the same type of labeling as the input tape, unless the user formulates a change through own-coding. If standard labels and trailers are specified, the output tape header is identical to the header of the first input tape except that the serial number of the output tape will be retained. If non-standard labels are specified, the header label of the first input reel and the trailer record, if any, on the last input reel are copied unchanged onto the final output tape.

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 a banner character or specifies a "1" if banner characters are not present. Character 37 is blank if the first character of each output record is to contain a banner character (56_8) or specifies 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 specifies 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, the Sort 2 program proceeds according to the error option indicated in characters 40-41. These characters are blank if it is desired to halt the program. Once the program halts, the user may reread the record by setting SENSE switches 1 and 2 OFF and pressing the RUN button, correct the record in memory and process it by setting SENSE switch 1 ON and pressing the RUN button, or, during the presort or last pass, delete the record by setting SENSE switch 2 ON and pressing the RUN button. Characters 40-41 specify "PD" if it is desired to print the record (and delete it from the file during the presort or last-pass segment of the program). If it is desired to print the unreadable record and halt the program, characters 40-41 specify "PΔ." The settings of SENSE switches 1 and 2 determine the next action to be taken. During the merge, an unreadable record may be printed and the program halted, but the record can not be deleted.

Printer Control Unit (Parameter Character 42)

Character 42 designates the address of the printer if one is used to print the unreadable record. Otherwise, the character is ignored.

Last-Pass Halt Option (Parameter Character 44)

A "1" specified in character 44 causes the sort to stop 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 presort is specified in characters 45-50. The address is expressed as either a decimal number with leading zeros or a number of 4K modules with leading blanks and must be lower than the origin of presort own-coding (if any).

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

Sort 2 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. The collating sequence is exercised through translation table overlays in the presort and last-pass segments. The character code of the output depends on the presence or absence of the 051 code compatibility feature.

The translation tables provided with the Sort permit the user to sort in IBM sequence. By overlaying these tables with two 64-character translation tables of another desired code, one in the presort and one in the last pass, the user can obtain another desired collating sequence. The output character code will depend on the use of the 051 feature. Use of the tables for sorting in IBM sequence implies that the IBM input tapes are recorded in binary coded decimal, even parity, and that the H-200 is equipped with the 051 code compatibility feature.

Characters 51-52 are blank when no data 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 an IBM tape that is to be sorted in Honeywell sequence. Characters 51-52 specify "TT" when the translation tables are used in the presort and last-pass segments, and the input is on either a Honeywell or IBM tape that is to be sorted in IBM sequence.

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 specifies a "D".

Own-Coding (Parameter Characters 54-77)

Own-Coding may be used in conjunction with the presort and last-pass segments of the Sort 2 program to perform limited data processing, as described in Section III. For example, during the presort, before each item is processed, own-coding may be used for input editing or other item pre-processing. All own-coding addresses are expressed in decimal values with leading zeros; each own-coding address delimits the memory area to be used by the sort segment involved.

Presort Own-Coding (Parameter Characters 54-65)

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

Last-Pass Own-Coding (Parameter Characters 66-77)

Characters 66-71 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 72-77 designate the address to which the last pass will branch (1) after the last pass has been specialized and (2) after each item has been placed in the output buffer. Characters 72-77 are blank if this option is not used.

79,80 = S2 WHEN LOADING PARAMETERS FROM CARDS

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. Decimal values with leading zeros are used in specifying the six characters. The location within the item of the high-order character of each key field (counting the first character in the item as 0001) is designated in parameter characters 81-84, 87-90, 93-96, 99-102, 105-108, 111-114, 117-120, 123-126, 129-132, and 135-138 as needed for key fields 1-10, respectively. The number of characters in each key field are indicated in parameter characters 85-86, 91-92, 97-98, 103-104, 109-110, 115-116, 121-122, 127-128, 133-134, and 139-140 as needed for key fields 1-10, respectively. If fewer than ten sort key fields are specified, the unused parameter characters are blank.

Input File Name Check (Parameter Characters 141-150)

If standard labels are used (characters 33-34), the file name of each input reel can be checked without own-coding, provided that characters 141-150 specify the file name. These characters are blank if no file name check is desired.

Output File Name Change (Parameter Characters 151-160)

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

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

Characters 161-166 designate the name of the last-pass own-coding program. The last-pass branches to the Loader/Monitor program to initiate a search for the own-coding program and segment names. These characters are blank if no last-pass own-coding is used.

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

Characters 167-168 designate the name of the last-pass own-coding segment. These characters are blank if no last-pass own-coding is used.

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 program and segment names. Character 169 designates a "B" for a search forward, or a "C" for a search backward.

Next Program Name (Parameter Characters 170-175)

Characters 170-175 designate the name of the program which is to follow Sort 2. There is a branch to the Loader/Monitor program to initiate a search for the next program and segment names. Characters 170-178 are blank if Sort 2 is not followed by automatic loading of another program.

Next Segment Name (Parameter Characters 176-177)

Characters 176-177 designate the segment name of the next program to be loaded.

Search Direction for 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.

Table II. Parameter Specification

Parameter Characters	Octal Location of First Character of Field	Description
1-2	2477	Address of primary input tape
3-4	2501	Address of alternate input tape (optional)
5-6	2503	Address of first work tape
7-8	2505	Address of second work tape
9-10	2507	Address of third work tape (optional)
11-12	2511	Address of fourth work tape (optional)
13-14	2513	Address of fifth work tape (optional)
15-16	2515	Address of merge work tape
17-18	2517	Total number of input reels
19	2521	Option to read primary input tape forward or backward
20-23	2522	Number of characters per item
24-27	2526	Number of items per input record
28-31	2532	Number of items per output record
32	2536	Padding characters
33-34	2537	Label indicators
35	2541	Not used
36	2542	Input banner character option
37	2543	Output banner character option
38	2544	Input parity indicator
39	2545	Output parity and gap-size indicator
40-41	2546	Error options
42	2550	Printer control unit
43	2551	Not used
44	2552	Option to stop after the merge segment
45-50	2553	Highest address available to the presort
51-52	2561	Collating sequence translation
53	2563	Ascending or descending sequenced output
54-59	2564	Address of presort header-trailer own-coding
60-65	2572	Address of presort item-by-item own-coding

Table II (cont). Parameter Specification

Parameter Characters	Octal Location of First Character of Field	Description
66-71	2600	Address of last-pass header-trailer own-coding
72-77	2606	Address of last-pass item-by-item own-coding
78-80 79-80	2614	Not used 52
81-84	2617	1st key field
85-86	2623	1st key field
87-90	2625	2nd key field
91-92	2631	2nd key field
.		.
.		.
.		.
135-138	2705	10th key field
139-140	2711	10th key field
141-150	2713	Input file name check
151-160	2725	Output file name change
161-166	2737	Last-pass own-coding program name
167-168	2745	Last-pass own-coding segment name
169	2747	Search direction for last-pass own-coding
170-175	2750	Next program name
176-177	2756	Next segment name
178	2760	Search direction for next program

TAPE POSITIONING

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

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 is one.

Work Tapes

C. Work tapes may be positioned at any point. Beyond the read-write head, there must be one valid record (two valid records if the tape is rewound) 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.

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 2E.

Work Tapes

G. Work tape 1, the final output tape, is postioned between the trailer record and an end-of-recorded-information record.

H. If work tapes were rewound before the sort began, they 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 3.

Characters 15, 16, and 19	
A. Δ, R, and Δ	The primary tape is read forward, rewound, and locked at the end of the presort.
B. Δ, 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 Δ t, t = Address of primary input tape.	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.
D. (t, t), and B t, t = Address of primary input tape.	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.
E. (t, t), and Δ or B t, t = Address of work tape.	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.

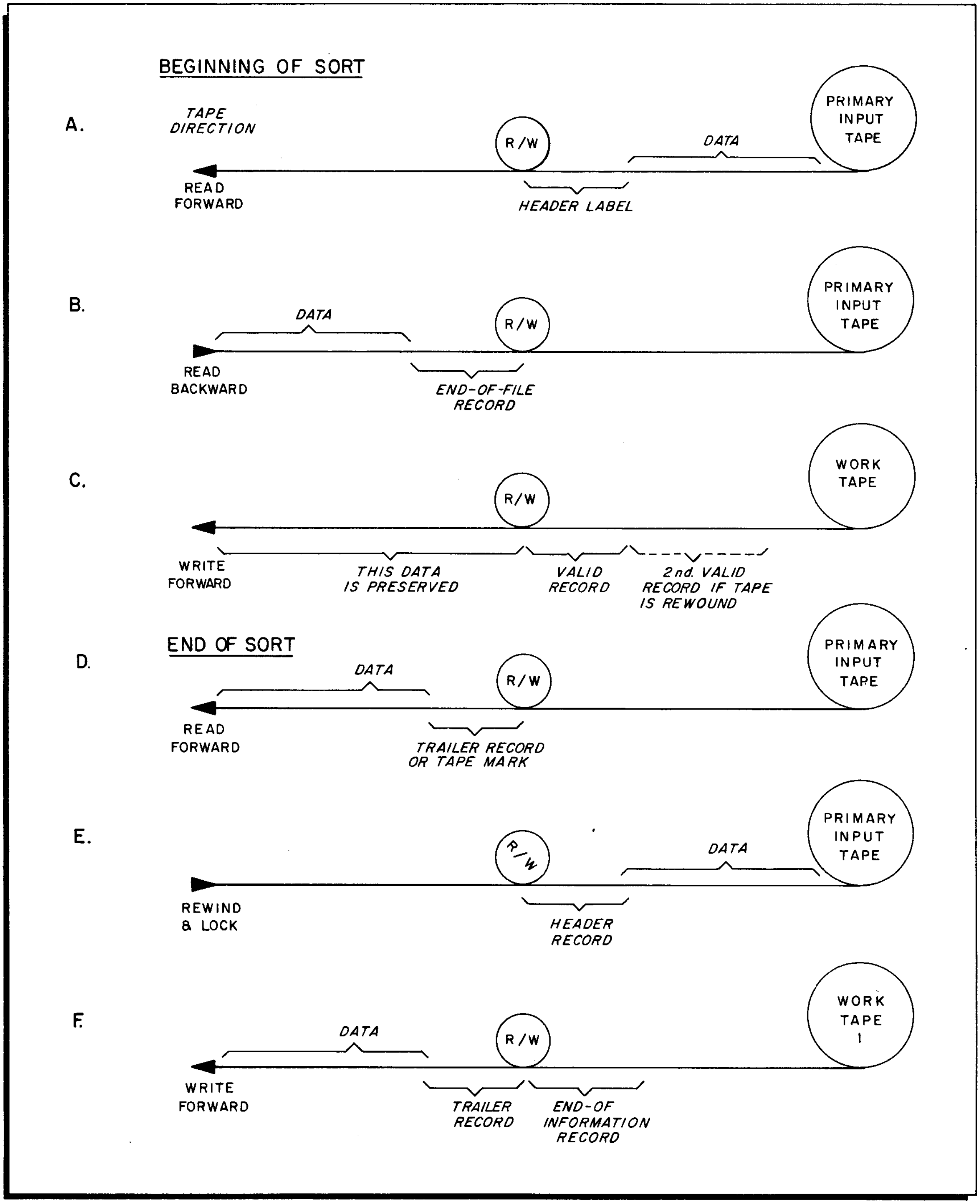


Figure 2. Tape Positioning in the Sort Program

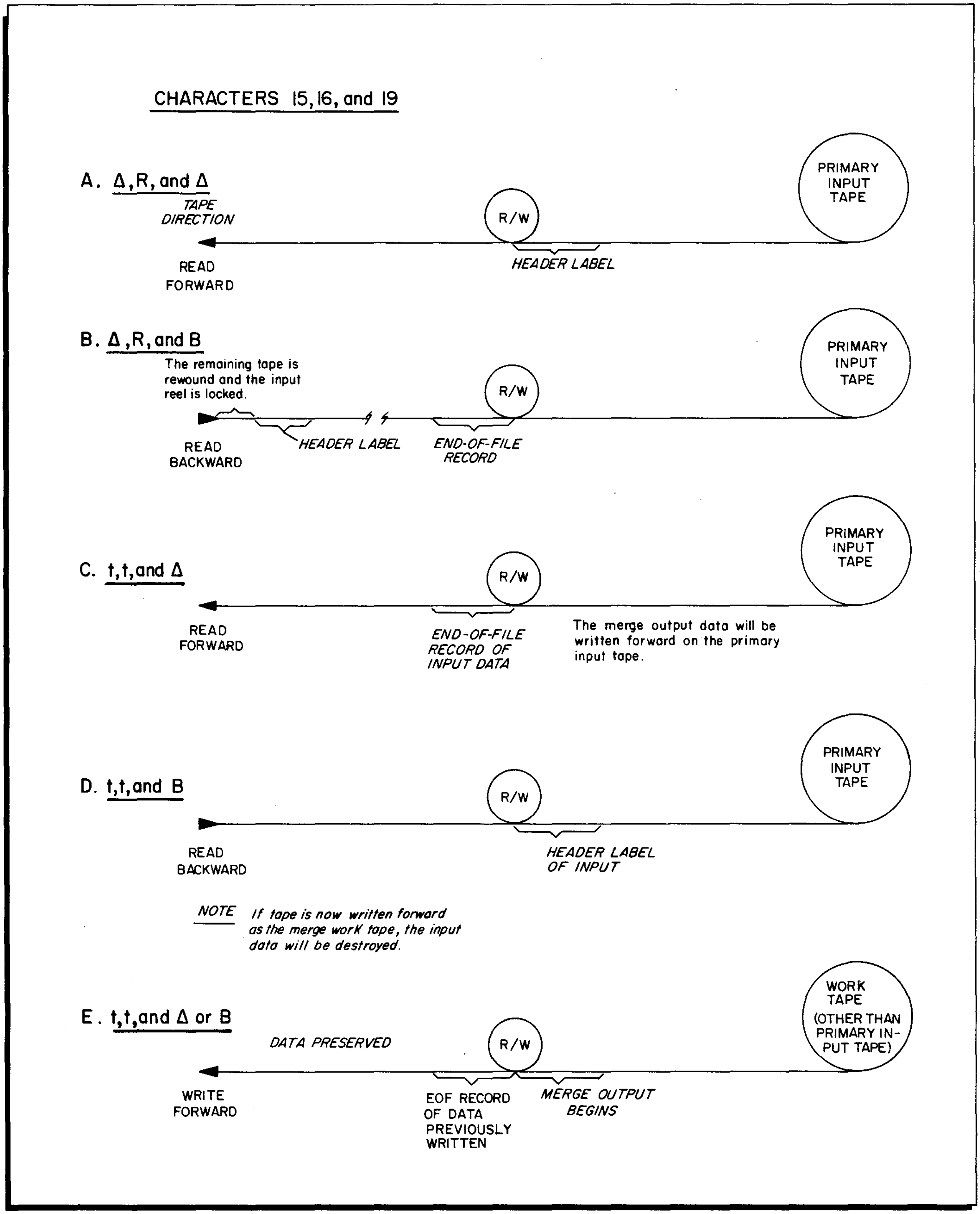

CHARACTERS 15, 16, and 19
A. Δ, R, and Δ
TAPE DIRECTION
READ FORWARD
R/W
HEADER LABEL
PRIMARY INPUT TAPE
B. Δ, R, and B
The remaining tape is rewound and the input reel is locked.
READ BACKWARD
HEADER LABEL
END-OF-FILE RECORD
R/W
PRIMARY INPUT TAPE
C. t, t, and Δ
READ FORWARD
R/W
END-OF-FILE RECORD OF INPUT DATA
The merge output data will be written forward on the primary input tape.
PRIMARY INPUT TAPE
D. t, t, and B
READ BACKWARD
R/W
HEADER LABEL OF INPUT
PRIMARY INPUT TAPE
NOTE If tape is now written forward as the merge work tape, the input data will be destroyed.
E. t, t, and Δ or B
DATA PRESERVED
WRITE FORWARD
R/W
EOF RECORD OF DATA PREVIOUSLY WRITTEN
MERGE OUTPUT BEGINS
WORK TAPE (OTHER THAN PRIMARY INPUT TAPE)

Figure 3. Tape Positioning for Effects of Characters 15, 16, and 19

SECTION III
OWN-CODING

GENERAL DESCRIPTION

Own-coding routines may be written by the user in Extended Easycoder assembly language and used in the presort and last-pass segments of the Sort 2 program. The presence of these routines is indicated to the sort program by parameter characters 54-77. 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 an item's size.

During the presort segment, own-coding is executed before each item is moved from the input buffer for processing by the sort program. Last-pass own-coding is executed after the item has been processed by the last-pass segment and moved to the output buffer. The starting location of an own-coding routine delimits the memory area to be used by the sort; thus, the routine should be written for high-address portions of memory.

If header-trailer own-coding is used, the address which references a label references the first character of that label. When item-by-item own-coding is being used, the address which references an item references the last character of that item. Index registers 1, 2, and 3 are used to load, store, and reference addresses. These rules apply to addresses supplied to own-coding, as well as to addresses supplied to the sort program by own-coding.

PRESORT OWN-CODING

Header-Trailer Own-Coding

When header-trailer own-coding is used during the presort segment, the initial address of the programmer's own-coding routine is designated in parameter characters 54-59. The address of the first character of the header or trailer label that is read is loaded into index register 1 before the branch to own-coding. The presort then branches to own-coding, and the first instruction of the own-coding routine should be an SCR of the B-address register. Own-coding may then be used to modify the header or trailer label as follows:

1. The header or trailer may be inspected and verified.
2. The header or trailer may be modified without changing its length.
3. The header or trailer may be modified and have changes in its length.

Index register 1 contains the address of the first character of the modified header or trailer. The user must reposition record marks in the event the header or trailer record length changes, so that a record mark is placed one character beyond the last (rightmost) character of the modified label.

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 character of the new label and placing a record mark one character beyond the last character of the new label. In the presort, own-coding may not be used to insert such a label where no label exists on the input; but last-pass header-trailer own-coding may be used to add a label when none exists on the input (see page 27).

> Note: Parameter characters 33-34 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 54-59, regardless of whether header-trailer own-coding was specified.

Upon completion of header-trailer own-coding, the program is re-entered by branching to the address stored from the B-address register.

Presort header and trailer own-coding are similar in operation, with one exception: When a header record is read, the presort sets an item mark at the own-coding address before branching to the own-coding routine; but if a trailer record is read, an item mark is not set at the own-coding address.

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

If item-by-item own-coding is used during the presort segment of the Sort 2 program, the initial address of the programmer's own-coding routine is designated in parameter characters 60-65. The presort performs a "one-shot" branch to the own-coding location before any item processing. The "one-shot" branch offers the user an opportunity to modify the presort program, store the address of the next program instruction from the B-address register, and store the contents of index registers 1, 2, and 3. The contents of these registers are stored for the following reasons:

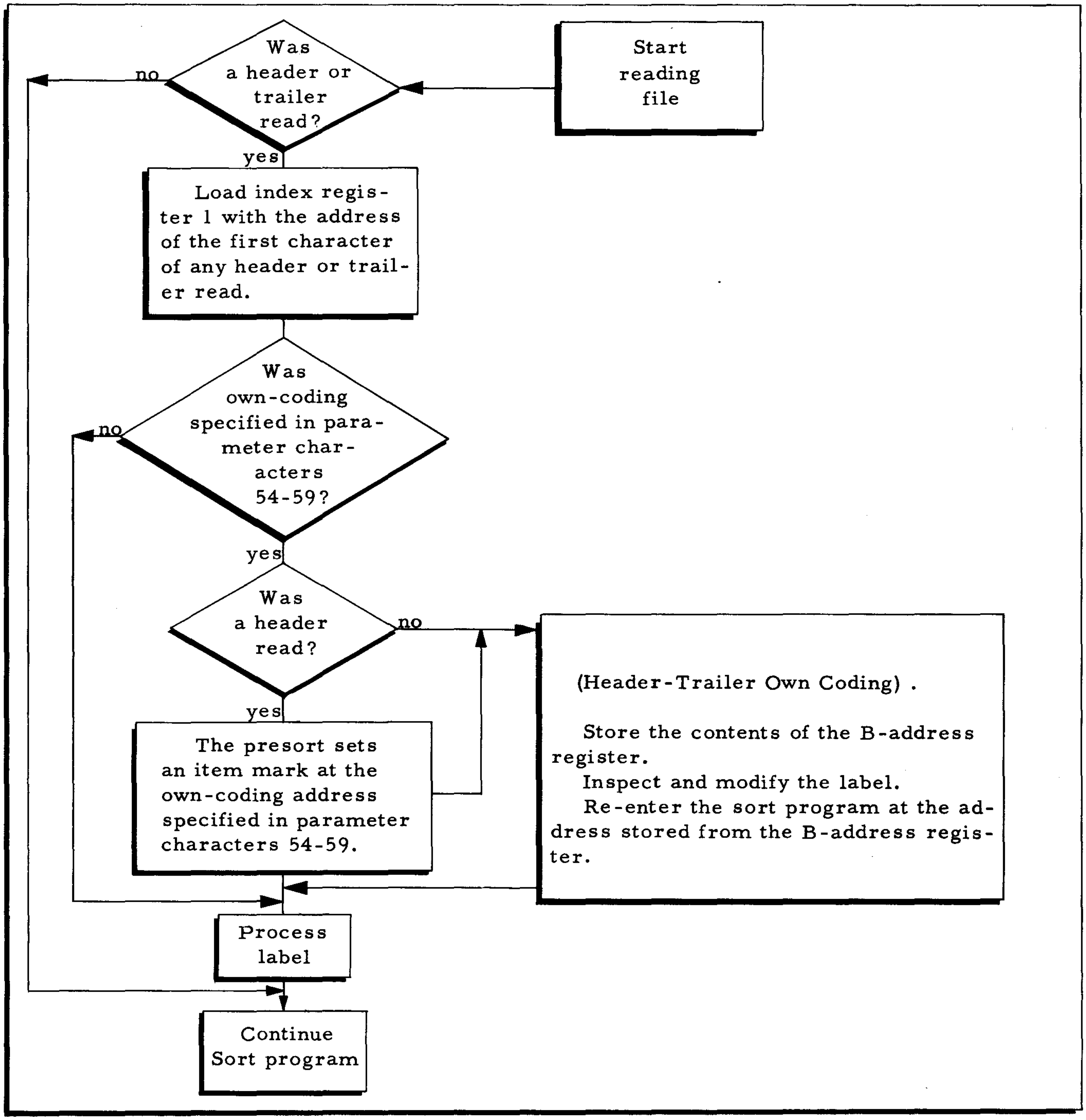

Start reading file
Was a header or trailer read?
no
yes
Load index register 1 with the address of the first character of any header or trailer read.
Was own-coding specified in parameter characters 54-59?
no
yes
Was a header read?
no
yes
The presort sets an item mark at the own-coding address specified in parameter characters 54-59.
(Header-Trailer Own Coding) .
Store the contents of the B-address register.
Inspect and modify the label.
Re-enter the sort program at the address stored from the B-address register.
Process label
Continue Sort program

Figure 4. Summary Flow Chart of Presort Header-Trailer Own-Coding

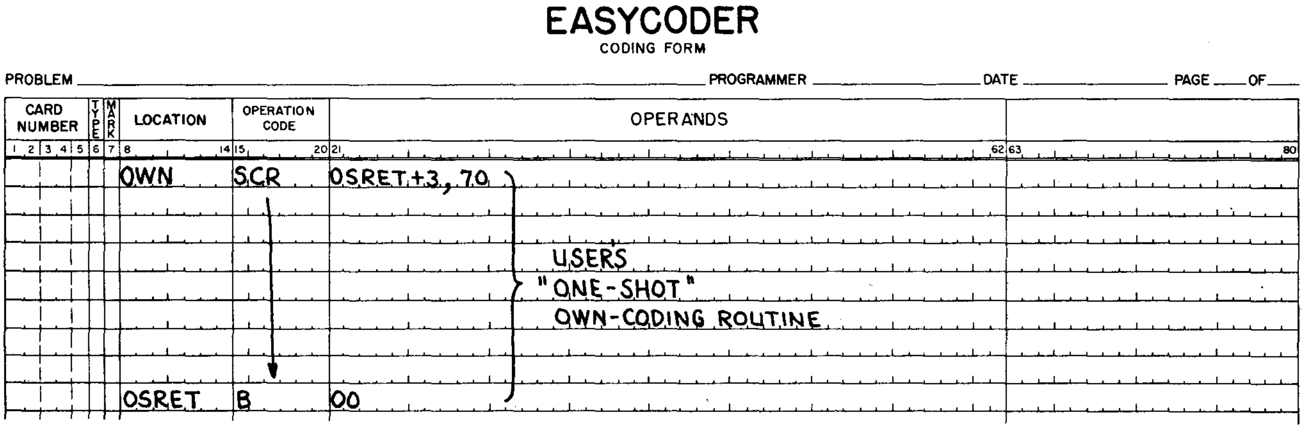
1. The address contained in the B-address register will designate the program re-entry location to which the own-coding routine branches to re-enter the presort from this "one-shot" routine.

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 word mark 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.

The first instruction of the user's own-coding routine should be an SCR (Store Control Register) instruction which stores the contents of the B-address register. The last instruction of the own-coding routine must be a Branch to the stored contents of the B-address register. An example of this method of opening and closing an own-coding routine, using the "one-shot" routine, is shown below.



EASYCODER
CODING FORM

PROBLEM ______ PROGRAMMER ______ DATE ______ PAGE ___ OF ___

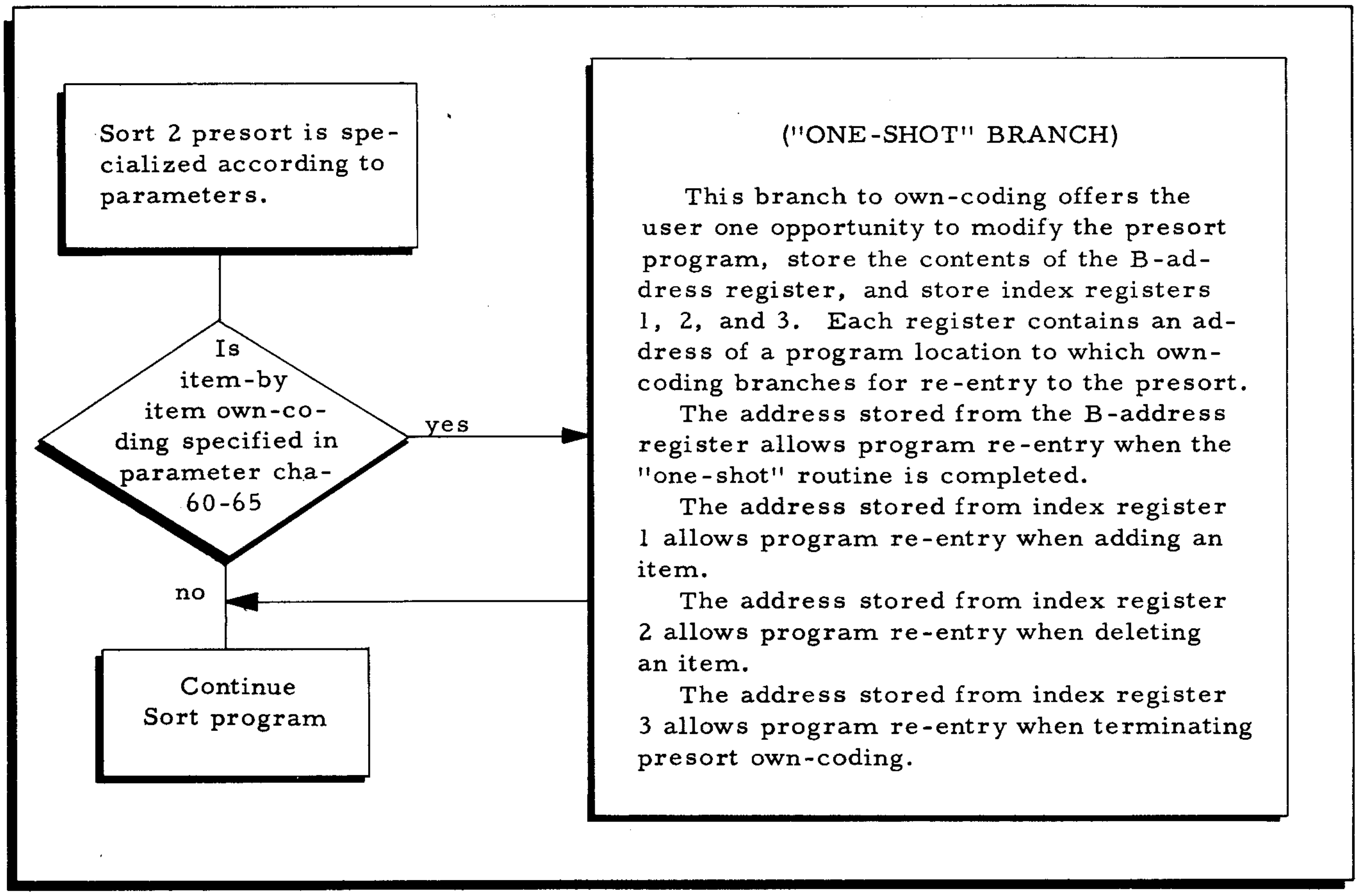
	CARD NUMBER 1-5	TYPE 6	MARK 7	LOCATION 8-14	OPERATION CODE 15-20	OPERANDS 21-62	63-80
1				OWN	SCR	OSRET+3, 70	
2							
3							
4						USERS	
5						"ONE-SHOT"	
6						OWN-CODING ROUTINE	
7							
8							
9				OSRET	B	00	

OWN is the location of the first instruction in the user's "one-shot" own-coding routine (parameter character 60-65). This instruction is an SCR instruction which stores the contents of the B-address register (70_8) in location OSRET + 3. Location OSRET + 3 is the A-address of the last instruction of the routine. The Branch instruction (OSRET) causes a branch to the address stored from the B-address register at the beginning of the routine, and this address provides re-entry to the Sort 2 program.

The registers described on pages 21-22 perform prescribed services for certain routines, as described in the following paragraphs.

Inspecting or Modifying an Item

Before processing each item, the presort branches to the own-coding location specified in parameter characters 60-65. The address of the last character of the item to be inspected or modified has been loaded into index register 1. The first own-coding instruction must be an SCR (Store Control Register) of the B-address register. The item may now be inspected and modified, after which the presort segment is re-entered at the location specified by the address stored from the B-address register.


Sort 2 presort is specialized according to parameters.
Is item-by item own-coding specified in parameter cha- 60-65
yes
no
Continue Sort program

("ONE-SHOT" BRANCH)

This branch to own-coding offers the user one opportunity to modify the presort program, store the contents of the B-address register, and store index registers 1, 2, and 3. Each register contains an address of a program location to which own-coding branches for re-entry to the presort.

The address stored from the B-address register allows program re-entry when the "one-shot" routine is completed.

The address stored from index register 1 allows program re-entry when adding an item.

The address stored from index register 2 allows program re-entry when deleting an item.

The address stored from index register 3 allows program re-entry when terminating presort own-coding.

Figure 5. Summary Flow Chart of Presort "One-Shot" Branch

Adding an Item

After item inspection, an item may be added to the data being sorted. Instructions in the own-coding routine are used to load the address of the last character of the item to be added into index register 1. Own-coding re-enters the presort at the address stored from index register 1 during the "one-shot" branch. The item to be added must have the input format specified by the sort parameters (i.e., item length, key location, and key lengths must be the same). Word marks must appear in the following locations:

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

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

Figure 6 shows an item to be added to the input data of a Sort 2 program. 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.

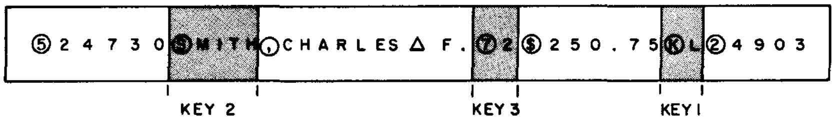


Figure 6. Word Mark Locations in a Sort Item

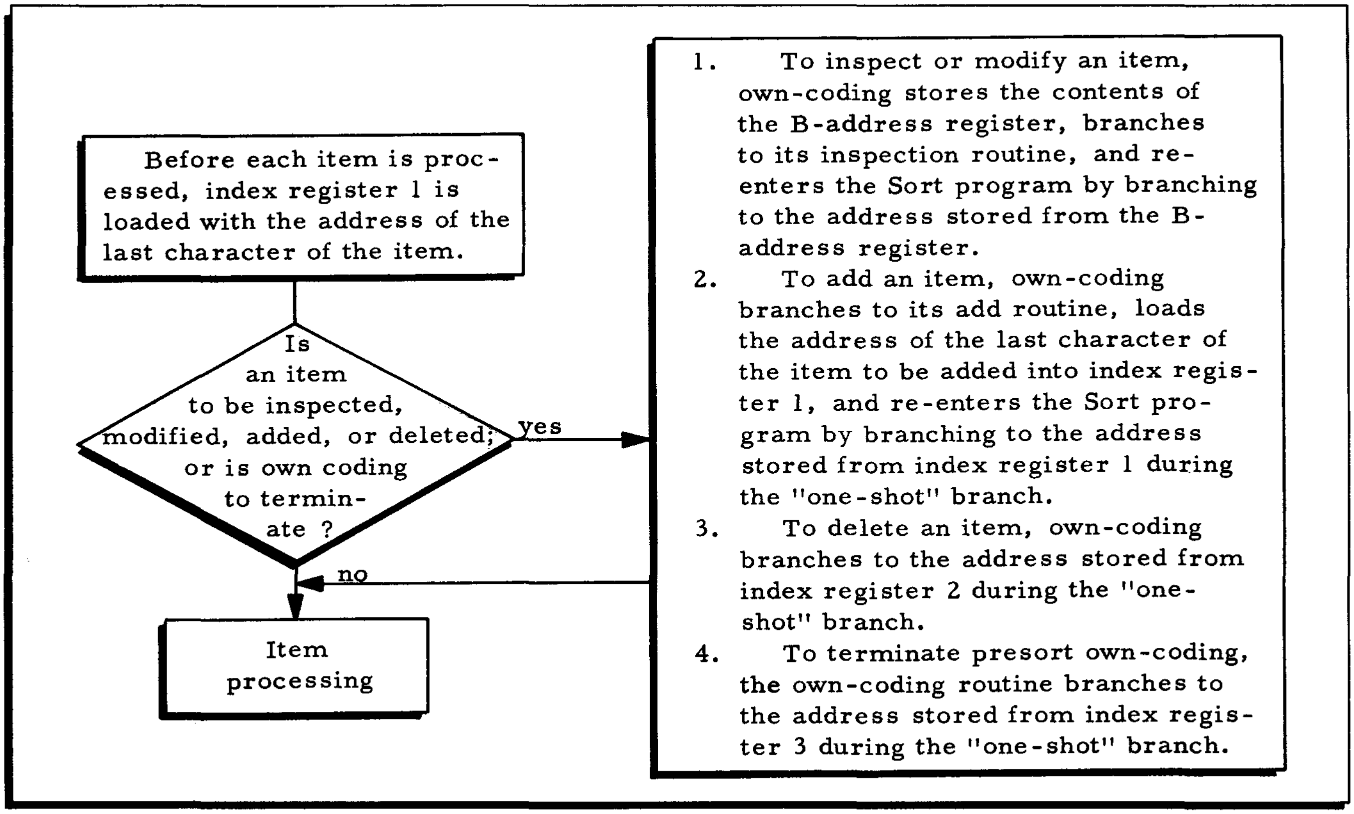
Deleting an Item

After inspecting an item, the user may wish to delete it. Own-coding must branch to the address stored from index register 2 during the "one-shot" branch. The Sort program then deletes the item.

Terminating Presort Own-Coding

If 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. If the branch is made before the entire input has been processed, it must follow the item inspection and modification routines (i.e., an item is neither being added nor deleted at the time of the branch).



Before each item is processed, index register 1 is loaded with the address of the last character of the item.
Is an item to be inspected, modified, added, or deleted; or is own coding to terminate ?
yes
no
Item processing
1. To inspect or modify an item, own-coding stores the contents of the B-address register, branches to its inspection routine, and re-enters the Sort program by branching to the address stored from the B-address register.
2. To add an item, own-coding branches to its add routine, loads the address of the last character of the item to be added into index register 1, and re-enters the Sort program by branching to the address stored from index register 1 during the "one-shot" branch.
3. To delete an item, own-coding branches to the address stored from index register 2 during the "one-shot" branch.
4. To terminate presort own-coding, the own-coding routine branches to the address stored from index register 3 during the "one-shot" branch.

Figure 7. Summary Flow Chart of Presort Item-by-Item Own-Coding

When the presort segment has processed all of the input data (an end-of-file record is read), an item mark is automatically set at the own-coding address specified in parameter characters 60-65 and own-coding may either terminate or add more items. 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 more items are to be added after all input data has been processed, own-coding loads index register 1 with the address of the last character of the item to be added, and re-enters the Sort program by branching to the address stored from index register 1 during the "one-shot" branch. The added item is processed by the Sort 2 program, and control is returned to the own-coding routine for another item. A cycle develops 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 by index register 1 during the "one-shot" branch. This cycle continues until control is returned to the own-coding routine 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.

The address of a program location stored from a register during the "one-shot" branch is also referred to as the re-entry address for the specified routine (inspect or modify, add, etc.).

Read Backwards Input

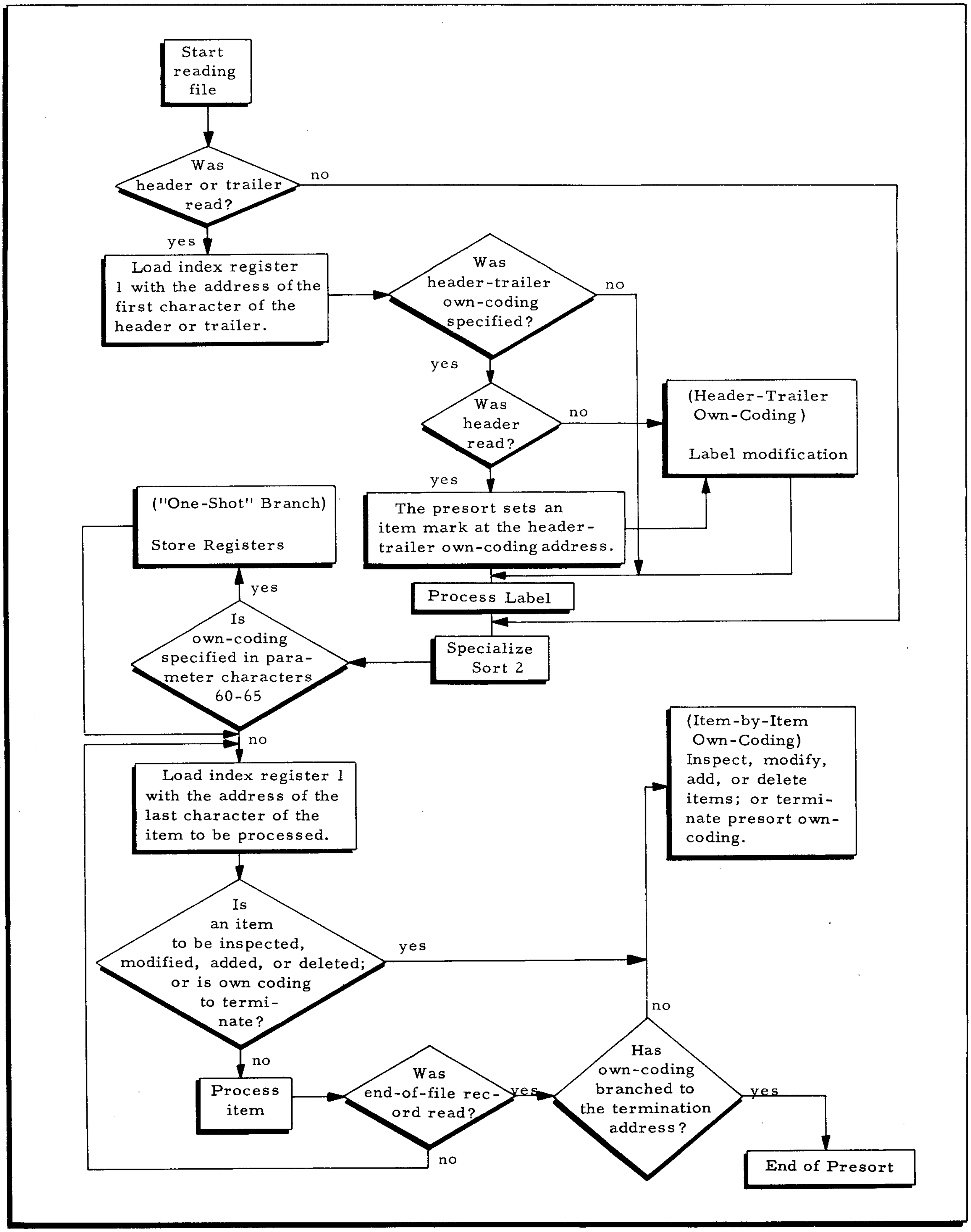
If the input is read backwards, item-by-item own-coding is the same as when input is read forward. For example, the last character of an item is the character occupying the numerically highest memory location whether the item was read into memory backwards or forwards.

LAST-PASS OWN-CODING

Last-Pass Header-Trailer Own-Coding

When header-trailer own-coding is used during the last-pass segment, the initial address of the programmer's own-coding routine is designated in parameter characters 66-71. The address of the first character of the header or trailer label that is read is stored in index register 1 before the branch to own-coding. The last pass then branches to own-coding, and the first own-coding instruction should be an SCR 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.

Inspection, modification, or replacement of labels in header-trailer own-coding is performed in the same manner as in the presort, with one exception. In the last pass, if a change



Start reading file
Was header or trailer read?
no
yes
Load index register 1 with the address of the first character of the header or trailer.
Was header-trailer own-coding specified?
no
yes
Was header read?
no
(Header-Trailer Own-Coding)
Label modification
yes
("One-Shot" Branch)
Store Registers
The presort sets an item mark at the header-trailer own-coding address.
Process Label
yes
Is own-coding specified in parameter characters 60-65
Specialize Sort 2
no
(Item-by-Item Own-Coding) Inspect, modify, add, or delete items; or terminate presort own-coding.
Load index register 1 with the address of the last character of the item to be processed.
Is an item to be inspected, modified, added, or deleted; or is own coding to terminate?
yes
no
no
Process item
Was end-of-file record read?
yes
Has own-coding branched to the termination address?
yes
no
End of Presort

Figure 8. Summary Flow Chart of Presort Own-Coding

is made to the length of a header or trailer, the user must set the label up in his own-coding, as if the label were to replace the existing header or trailer. 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 setup 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 label during the last pass if one was not present on the input. If own-coding was specified in parameter characters 66-71, an anticipatory branch to own-coding will occur whether or not a header or trailer was present on the input. Own-coding may then be used to add a header or trailer by writing the label on the output tape, and the program is re-entered by branching to the address stored from the B-address register.

Last-pass header and trailer own-coding are similar in operation, with one exception: When a header record is to be written, own-coding must set an item mark at the initial own-coding location; but, if a trailer record is to be written, no item mark is set at the own-coding location.

Last-Pass Item-by-Item Own-Coding

If item-by-item own-coding is used during the last-pass segment of the Sort 2 program, the initial address of the programmer's own-coding routine is designated in parameter characters 72-77. The last pass performs a "one-shot" branch to the own-coding location before processing any item. This branch is similar to the "one-shot" branch of presort own-coding. The registers perform the same functions in the last pass as they did in the presort (see page 21).

Inspecting or Modifying an Item

As each item is placed in the output buffer, the last pass branches to the own-coding location specifiedin parameter character 72-77. The address of the last character of the item to be inspected or modified has been loaded into index register 1. The first own-coding instruction must be an SCR of the B-address register. The item is then inspected or modified, and the Sort program is re-entered at the location specified by the address stored from the B-address register.

Adding an Item

The user may wish to add one or more items after the current item has been processed. An own-coding instruction is needed to place an item mark in the add re-entry address. Coding control is then returned to the inspection re-entry address, or it is given to the delete re-entry address for deletion of the current item. The next time a branch is made to own-coding, the last-pass segment expects an item to be added at the address specified in index register 1. A Branch instruction to the add re-entry address must be coded to re-enter the Sort program.

As long as the item mark remains in the location specified by the add re-entry 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 re-entry address. When adding an item, the word marks in the output buffer must not be changed (see Figure 6 on page 24).

Deleting an Item

After inspecting an item, the user may wish to delete it from the Sort. Own-coding branches to the delete re-entry address, and the Sort program deletes the item.

Terminating Last-Pass Own-Coding

The own-coding routine branches to the terminate re-entry address when all own-coding processing is completed. This branch may be made before or after the last pass has processed all of its tape input. If the branch is made before all tape input has been processed, it must follow item inspection and modification routines (i. e., an item is neither being added nor deleted at the time of the branch).

When the last-pass segment has processed all of its tape input, an item mark is set at the own-coding address specified in parameter characters 72-77. 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 regardless of the setting of the item mark at the add re-entry address. Index register 1 will contain not the address of the last tape input item, but the address at which an item may be added. If there are no items to be added, the Sort program is re-entered at the terminate re-entry address. The last pass will then execute its end-of-job routine.

If an item is to be added after all tape input has been processed, the item is added at the address specified by the current setting of index register 1. The Sort program is then re-entered at the add re-entry address. This cycle continues, adding an item at the address specified by the current setting of index register 1 and branching to the add re-entry 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 re-entry 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; and the user continues to have access to header-trailer and item-by-item own-coding.

SECTION IV

SORT 2 OPERATING PROCEDURES

LOADING THE SORT 2 PROGRAM

Sort 2 may be run as one program within a series of programs, as the initial program in a series, or as a single program. When the sort is to be run as a program within a series, it is assumed the program will be loaded from tape in the same manner as other programs operating under the PLUS system,[1] i.e., by a call to Loader/Monitor. If the sort is run as the initial program in a series or as a single program, it is loaded into memory by means of a Console Call card or by any other available method.

Loading Sort 2 Parameters

If Sort 2 is run within a series of programs, it is assumed that the parameters specializing the sort have been loaded into memory by the preceding program. The Sort 2 program name is AADS2Δ and the segment name of the first segment is 01. These characters are used to call in the Sort program.

When Sort 2 is run as the initial program of a series or as a single program, the Sort parameters may be bootstrapped into memory from a card reader after the Loader/Monitor program has been loaded. The card reader should contain the following cards for bootstrapping:

1. Parameter card 1 (parameter characters 1-80 in Table II, on page 13).
2. Parameter card 2 (parameter characters 81-160 in Table II).
3. Parameter card 3 (parameter character 161-178 in Table II).
4. The Sort 2 Console Call Card. The Console Call Card is punched as follows:

   Columns 1-6 : contain the characters AADS2Δ to indicate the program name.

   Columns 7-8 : contain the character 01 to indicate the segment name.

   Column 9: contains the character 0 to indicate the address of the run tape containing the Sort program.

   Columns 10-17: contain the characters AADS2Δ01 to indicate that the Loader/Monitor is to halt after loading the first segment of the Sort program.

   Column 18: contains an asterisk (*) to identify the Console Call Card

These four cards must be ordered as in Figure 9.

[1] Additional information on the Plus Loader/Monitor is presented in the Honeywell information bulletin entitled Plus – Tape Loader/Monitor (DSI-327).

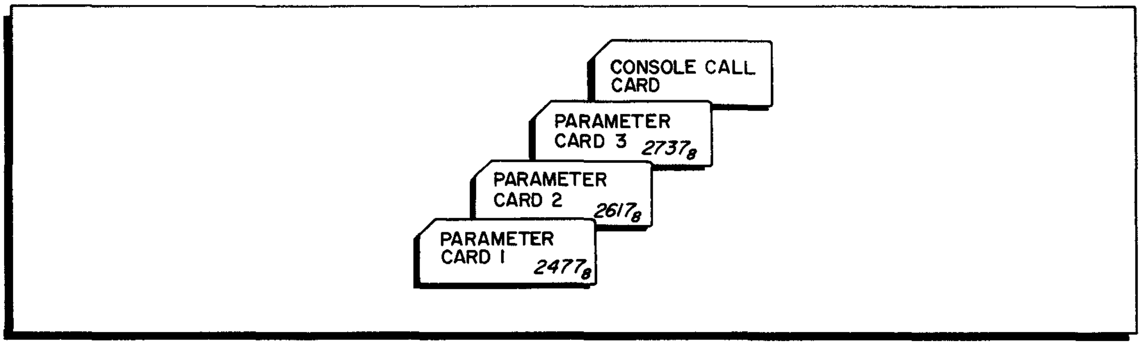
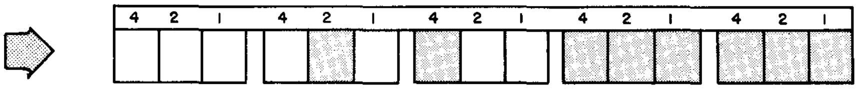

CONSOLE CALL CARD
PARAMETER CARD 3 2737_8
PARAMETER CARD 2 2617_8
PARAMETER CARD I 2477_8

Figure 9. Card Input for Bootstrapping Sort 2

Loading the Parameter Card Data

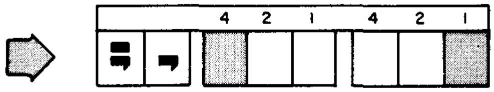
If the parameters of the Sort program are to be bootstrapped, the operator should perform the steps described below. (The machine is stopped and conditioned for card loading.)

1.



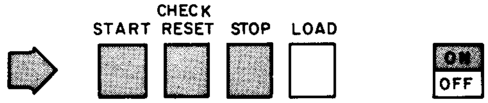
Enter location 2477_8 on the control panel by depressing the corresponding ADDRESS buttons. This address is automatically placed into the instruction address register, the A-address register, and both counters associated with RWC1. (Depressed buttons are indicated in gray.)

2.



Designate the card reader control by setting the CONTENTS buttons to the logical address of the control (octal 41).

3.



Initialize the card reader by insuring that the CARD READER ON/OFF switch is ON, then depressing the STOP, CHECK RESET, and START buttons, in that order. Ascertain that the card reader has clutched the first card.

4.

Depress the BOOTSTRAP button on the control panel. This causes parameter card 1 to be read.

5. Repeat steps 1, 2, and 4, but enter location 2617_8 in step 1.

6. Repeat steps 1, 2, and 4, but enter location 2737_8 in step 1. The Console Call Card is automatically read after parameter card 3.

After the parameter cards have been bootstrapped, the instruction address register is set to 00126 and the RUN button is pressed. The Loader/Monitor program should halt with the configuration 07002 in the B-address register. Press the RUN button again and the Loader/Monitor program will halt with the configuration 04000 in the B-address register. Press the RUN button once again to start the Sort program.

SORT 2 PROGRAM FORMAT

Sort 2 consists of the following segments which are assumed to be in this order:

Program Name	Segment Name
AADS2Δ	Presort 01
AADS2Δ	02
AADS2Δ	03
AADS2Δ	04
AADS2Δ	05
AADS2Δ	06
AADS2Δ	07
AADS2Δ	08
AADS2Δ	Merge M1
AADS2Δ	M2
AADS2Δ	M3
AADS2Δ	Last Pass G1
AADS2Δ	G2
AADS2Δ	G3

The sort will in all cases search forward for its next segment. If last-pass own-coding was specified, the coding program and segment names will be assumed to be preceding segment AADS2ΔG2 on the BRT (Binary Run Tape), and the Loader/Monitor will search for the names in the direction specified in parameter character 169. After loading the last-pass own-coding (if any), the last-pass will resume searching forward for the next segment.

PROGRAM HALTS

A halt numbering system is used by Sort 2 to identify the various halts which may occur during the running of the program. The contents of the A-address register and the B-address register display the particular halt configuration on the control panel. The causes of the halts and the necessary corrective actions to be taken by the operator when they occur are shown in Table III.

Program halts are divided into two categories for easy identification: (1) peripheral errors and (2) non-peripheral errors. When a particular halt specifies a peripheral error, the B-address register contains a 0 in the second octal digit. With a non-peripheral error, the B-address register contains a 4 in the second octal digit.

In Table III, t = tape control address, d = tape drive, and bbbbb = buffer address. The buffer address (bbbbb) in the presort is the address of the first location of the input buffer if the input was read foward; if the input was read backward, the buffer address is the address of the last location of the input buffer. The buffer address in the merge and last pass is the address of the last location of the input buffer.

Table III. Halt Codes

A-Address Register	B-Address Register	Segment(s) Involved	Cause	Corrective Action
00000	04000	Merge and last pass	Data or record out of sequence in merge, or record out of sequence in last pass	Rerun sort.
00001	04000	Presort	Invalid parameters 1-80	Correct parameter(s) and re-run sort.
00002	04000	Presort	Incorrect input file name: standard header	Rerun sort with correct tape or parameter.
00003	04000	Presort	Duplication of tape drive assignments	Correct parameters and rerun sort.
00004	04000	Presort	Overlap in key fields	Correct parameters and rerun sort.
00005	04000	Presort	Invalid parameters 81-160	Correct parameter(s) and rerun sort.
00007	04000	Presort	Insufficient memory to sort	Correct parameter and rerun sort.
00001	00t0d	Presort	1 HDR record missing	Rerun sort.
00002	00t0d	Merge and last pass	Parameter record missing	Rerun sort.

Table III (cont). Halt Codes

A-Address Register	B-Address Register	Segment(s) Involved	Cause	Corrective Action
00003	00t 0d	Last pass	Output retention incorrect during multi-reel output	Press RUN to override, or mount correct output reel.
00004	00t 0d	Last pass	To change tape density or to reassign the logical address of work tape 1 to an unused drive	Press RUN when output tape is ready.
07777	00t 0d	Presort	Halt for input rewind	Press RUN when new tape is mounted for merge work tape.
bbbbb	00t 1d	All Segments	Read error: data record	To reread nine more times, set SENSE switches 1 and 2 OFF and press RUN. To process record (assuming that record was corrected via control panel or the halt was erroneous), set SENSE switch 1 ON and press RUN. To delete the record, set SENSE switch 2 ON and press RUN (SENSE switch 2 has no effect on merge).
bbbbb	00t 2d	All Segments	Write Error	Press RUN to erase and rewrite.
0000x	00t 3d	All Segments	End of Tape	Rerun sort if x = 0. Mount a new output reel and press RUN if x = 1.
bbbbb	00t 4d	Presort	Data record of incorrect length	To drop the record, set SENSE switch 2 ON and press RUN; otherwise, sort must be rerun.
bbbbb	00t 5d	Presort and last-pass	Read error: header record	To try to reread nine more times, press RUN. To process the record, set SENSE switch 1 ON and press RUN.
bbbbb	00t 7d	Presort	Read error: trailer record	To try to reread nine more times, press RUN. To process the record, set SENSE switch 1 ON and press RUN.
bbbbb	00t 6d	Presort	Header or trailer record of incorrect length (standard header or trailer $\neq$ 80 characters, or non-standard header or trailer exceeds record length)	To process the record, set SENSE switch 1 ON and press RUN. If a standard header and/or trailer was specified, this action supplies 80 characters. If a non-standard header and/or trailer is specified, this action supplies a record equal to the size of an input buffer. If the record is not be to processed, the sort must be rerun.

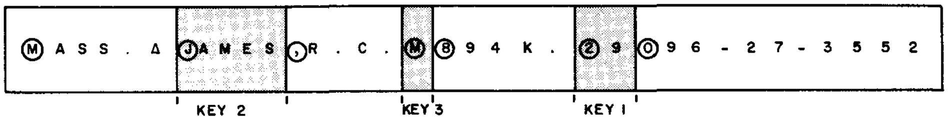
Table III (cont). Halt Codes

A-Address Register	B-Address Register	Segment(s) Involved	Cause	Corrective Action
ddddd	040x1	Last pass	Number of items in Presort differs from No. of items in L. P.	Presort is high if x = 0. Presort is low is x = 1. ddddd = amount of discrepancy (decimal).
07777	07777	Last pass	End of job	

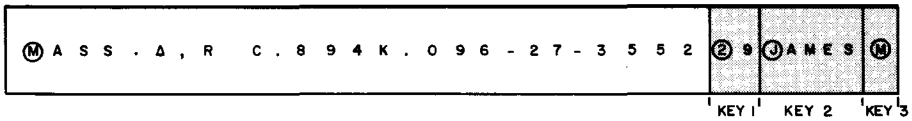
READ ERROR CORRECTION

When correcting a read error during the merge or last-pass segment of the program, the operator must consider that the presort rearranges the format of the items.

The random input items are rearranged by the presort into a format determined by the number and location of the keys in the items. The keys are placed in the last (rightmost) locations of each item according to their relative importance (the first key precedes the second etc.). The remaining (non-key) information is compressed to the left. For example, the input item:



consists of a person's resident state, the person's name (KEY 2), specific code data, sex (KEY 3), code data, age (KEY 1), and social security number. The items are to be sorted first by age (KEY 1), then by name within the age group, and finally by sex. The presort segment rearranges the item into the following format:



If this item is contained in an unreadable record and if unreadable records are designated to be printed (parameter character 40-42), the item will appear on the printout in the latter format during the merge or last-pass segment of the program.

Note: The single characters ",", "8", and "0" of the input item contained word marks in the original format. Because of the reformatting by the presort segment, these characters do not contain punctuation in the merge and last-pass segments. Therefore, when a character is to be changed by the operator due to an error condition during the merge or the last-pass, the character should first be displayed on the control panel. The punctuation of the character should be noted by the operator before he enters the correct character (see "Correcting Errors from the Control Panel," below).

Correcting Errors from the Control Panel

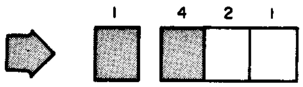
The programmer specifies in the parameter card the action to be taken by the program when an unreadable record is encountered. The designated action may be one of the following:

1. The record is to be printed, and the machine halts.
2. The record is to be printed and eliminated (and the program continues in sequence).
3. The machine halts without eliminating or printing the record.

If a program halt (1 or 3 above) occurs, the operator may reread the record, correct it from the control panel, and subsequently allow processing to continue, or he may delete it (see "Error Options," page 10).

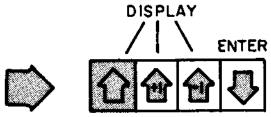
To correct an unreadable record which was printed (1 above), the operator must first check the printed record to determine which character(s) is in error. To correct and unreadable record which was not printed (3 above), he must first display the current record via the control panel. For example, to display an unreadable record occurring during a presort with the input read forward, the operator performs the following control panel activities:

1.



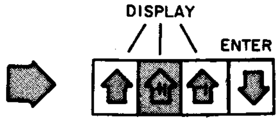
Enter the address (14) of the A-address register by depressing the corresponding CONTROL buttons. (Depressed buttons are indicated in gray.)

2.



Depress the CONTROL DISPLAY button. This causes the address contained in the A-address register to be displayed in the ADDRESS indicators. This address is the beginning location of the buffer that contains the record in error.

3.



Depress the CONTROL DISPLAY + 1 button repeatedly to display successive characters, from left to right, of the record. Each time the DISPLAY + 1 button is pressed, a character (including punctuation) is displayed in the CONTENTS indicators.

Once the record has been displayed and the character(s) in error have been noted (or the printed record is compared and the correction(s) noted), the operator enters the correct character(s) (including punctuation) into memory and directs the program to continue.

SECTION V
COLLATE 2

Collate 2 is a general-purpose program designed to combine from two to five files of identically ordered format into one sequentially ordered data file - the collated file.

Although Collate 2 may be used to combine the output files of several sorting operations, it is performed as a separate program, completely disassociated from the sorts which produced the files to be collated. In the collate routine, each input file may be contained on one or more magnetic tape reels; all reels containing a given file are processed sequentially from a single tape unit (i. e., no alternate drive may be specified). The information that specializes Collate 2 for a particular collating application is inserted at the beginning of the Collate program by means of parameters specified by the user.

CHARACTERISTICS OF COLLATE 2

The activities performed by Collate 2 are governed by the following characteristics:

1. Combines two to five ordered files into a single ordered file;
2. Processes fixed-length items blocked one or more per record;
3. Collates on up to ten key fields;
4. Allows manual correction or removal of unreadable records;
5. Allows label record changes;
6. Provides for the inclusion of own-coding elements;
7. Translates collating sequences; and
8. Writes output in ascending or descending sequence.

Figure 10 illustrates a collate process which uses three tape drives, the minimum number needed for collating.

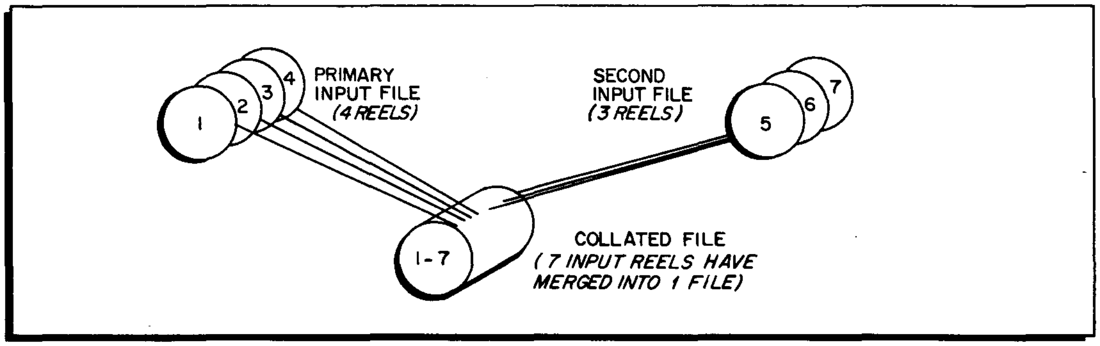

1
2
3
4
PRIMARY INPUT FILE (4 REELS)
SECOND INPUT FILE (3 REELS)
5
6
7
1-7
COLLATED FILE (7 INPUT REELS HAVE MERGED INTO 1 FILE)

Figure 10. Collate Process

MACHINE REQUIREMENTS

Collate 2 is used with an H-200 having the following equipment configuration:

1. A minimum main memory capacity of 8, 192 characters.
2. A minimum of three tape drives using 1/2-inch 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. A card reader or additional tape drive for program loading.

DATA REQUIREMENTS

The Collate 2 program processes fixed-length items blocked one or more per record. When blocking records, the number of items per record, specified by the user, is limited only by the maximum record length. The maximum record length depends on the number of files to be collated and the size of main memory; however, the collate is most efficient when the record is of an optimum size. The following formulas may be used to determine the maximum and optimum record sizes:

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

$$\text{Optimum record length} = \frac{\text{HMA} - \text{MON} - \text{PROG} - (\text{NTC} - 1)(2\text{NKC}) - \text{NKC}}{2\text{NTC}}$$

HMA – Highest memory address available to collate (parameter characters 49-54).

MON – Monitor requirements (1340 characters).

PROG – Collate program (3000 characters).

NTC – Number of input files plus one.

NKC – Number of key field characters in an item.

COLLATE KEYS

The files to be collated are made up of items that are identified by key fields. Key fields must be fixed in length and location for all items throughout all files to be collated. Within the key fields, any bit configuration is permissible. Each file item may contain up to ten key fields, and these key fields are specified in decreasing order of importance by parameters specified by the user.

MAGNETIC TAPE UNITS

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

PARAMETER PREPARATION

The information which specializes Collate 2 for a particular collating application is described to the program by means of parameters specified by the user. Parameter information is summarized in Table VI, page 43 . Leading zeros must be included when applicable. Like the sort parameters, collate parameters may be read into memory from the card reader for single program operation, or they may be set by the preceding program for automatic linking of a series of programs.

Tape Unit Specifications (Parameter Characters 1-24)

The Collate 2 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 1-24. Two 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 1-2 specify the address (tape control designation followed by drive number) of the primary input file. Header and trailer labels for the output file will be taken from this file. Characters 3-4 designate the number of reels in the primary input file. Characters 5-6 designate the address of the second input file. Characters 7-8 designate the number of reels in the second input file.

Characters 9-10 designate the address of an optional third input file, but these characters are blank if only two files are combined. Characters 11-12 designate the number of reels in the third file, and these characters will be blank if there is no third file. Characters 13-14 designate the address of an optional fourth input file, but these characters are blank if only two or three files are combined. Characters 15-16 designate the number of reels in the fourth file, and these characters will be blank if there is no fourth file. Characters 17-18 designate the address of an optional fifth input file, but these characters are blank if only two, three, or four files are combined. Characters 19-20 designate the number of input reels in the fifth file, and these characters are blank if there is no fifth file.

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

Item and Record Sizes (Parameter Characters 25-36)

Characters 25-28 specify the number of characters per item in decimal with leading zeros. The size of the items must remain fixed throughout the collate; however, the number of items per record may be separately specified for the input and the output. All input files must have the same number of items per record. Note that when specifying parameters, leading zeros must be designated.

Characters 29-32 designate the number of items per input record. The maximum number of items per record is limited only by the maximum record size (see page 38). Characters 33-36 designate the number of items per output record. The size limits for output records are the same as for input records.

Padding Character (Character 37)

If the last data record of the output reel is lacking one or more items because of re-blocking or own-coding, the Collate 2 program will automatically "pad" this record with the character represented in parameter character 37.

Tape Labels (Parameter Characters 38-39)

Characters 38-39 specify tape labeling options as indicated in Table IV.

Table IV. Label Options

If columns 38-39 are punched with	the beginning of file has	and the end of file is indicated by
Δ Δ	no label	tape mark
Δ T	no label	tape mark and trailer
S S	standard label	standard trailer
S S	standard label and tape mark	tape mark and standard trailer
NΔ	non-standard label	tape mark
NT	non-standard label	tape mark and 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 will be identical to the header of the primary input tape, except that the serial number of the output tape will be retained. The output trailer will be identical with 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. 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 unchanged onto the final output reel.

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 a 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 a banner character (56_8), or specifies "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, the Collate 2 program will proceed according to the error option indicated in characters 45-46. These characters are blank if it is desired to halt the program. Once the program halts, the user may reread the record by setting SENSE switches 1 and 2 OFF and pressing the RUN button, correct the record in memory and process it by setting SENSE switch 1 ON and pressing the RUN button, or delete the record by setting SENSE switch 2 ON and pressing the RUN button. Characters 45-46 specify "PD" if it is desired to print the record and delete it from the file. If it is desired to print the unreadable record and halt the program, characters 45-46 specify "PΔ". The settings of SENSE switches 1 and 2 will determine the next action to be taken.

Printer Control Unit (Parameter Character 47)

Character 47 designates the address of the printer if a printer is used to print out the unreadable record. The character will be blank if no printer is used.

Highest Memory Address (Parameter Characters 49-54)

The highest memory address available to the collate is specified in character 49-54. The address is expressed as either a decimal number with leading zeros or a number of 4K modules with leading blanks.

Collating Sequence (Parameter Character 55)

Collate 2 may be used with any collating sequence. The translation table provided with the Collate permits collating in IBM sequence, but by overlaying this 64-character translation table with a table of another desired code, the user can obtain any other 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 H-200 is equipped with the 051 code compatibility feature.

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

Ascending or Descending Sequence (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 specifies a "D" if all input files are in descending sequence and the final output file 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 specifies a "1" if the tapes are not to be rewound.

Own-Coding (Parameter Characters 58-69)

Own-coding may be executed during the collate to perform additional data processing. Own-coding routines can be used to inspect, modify, or replace file labels; and they may be used to modify, add, or delete items - they cannot change an item's size. All own-coding addresses are expressed as decimal values with leading zeros. Collate own-coding may be loaded into any available memory locations of either higher or lower order than those containing the Collate program.

Characters 58-63 designate the own-coding address to which the collate will branch after reading a header or trailer label. Characters 64-69 designate the address to which the collate will branch (1) after the collate has been specialized and (2) after each item has been placed in the output buffer. Characters 58-69 are blank if these options are not used.

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

Characters 70-74 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 70-74 specify this decimal number with leading zeros.

79,80 = C2

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. Decimal values with leading zeros are used in specifying the six characters. The location within the item of the high-order character of each key field (counting the first character in the item as 0001) is designated in parameter characters 81-84, 87-90, 93-96, 99-102, 105-108, 111-114, 117-120, 123-126, 129-132, and 135-138 as needed for key fields 1-10, respectively. The number of characters in each key field is indicated in parameter characters 85-86, 91-92, 97-98, 103-104, 109-110, 115-116, 121-122, 127-128, 133-134, and 139-140 as needed for key fields 1-10, respectively. If fewer than ten collate key fields are specified, the unused parameter characters are blank.

Input File Name and Reel Number Check (Parameter Character 141)

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

Character 141 is blank if it is not desired to check the file name and reel number, or specifies an "F" if the check 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 owncoding. The new output file name is specified in characters 142-151, and the collate will make the change. These characters are blank if no change is desired.

Next Program Name (Parameter Characters 152-157)

Characters 152-157 designate the name of the program which is to follow Collate 2. There is a branch to the Loader/Monitor program to initiate a search for the next program and segment names. Characters 152-160 are blank if Collate 2 is not followed by automatic loading of another program.

Next Segment Name (Parameter Characters 158-159)

Characters 158-159 designate the segment name of 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.

Table V. Collate 2 Parameter Specification

Parameter Characters	Octal Location of First Character of Field	Description
1-2	2477	Address of primary input file
3-4	2501	Number of reels in primary file
5-6	2503	Address of second input file

Table V (cont). Collate 2 Parameter Specification

Parameter Characters	Octal Location of First Character in Field	Description
7-8	2505	Number of reels in second input file
9-10	2507	Address of third input file (optional)
11-12	2511	Number of reels in third input file (optional)
13-14	2513	Address of fourth input file (optional)
15-16	2515	Number of reels in fourth input file (optional)
17-18	2517	Address of fifth input file (optional)
19-20	2521	Number of reels in fifth input file (optional)
21-22	2523	Address of output tape
23-24	2525	Address of alternate output tape (optional)
25-28	2527	Number of characters per item
29-32	2533	Number of items per input record
33-36	2537	Number of items per output record
37	2543	Padding character
38-39	2544	Labels
40	2546	Not used (must be blank)
41	2547	Input banner character option
42	2550	Output banner character option
43	2551	Input parity indicator
44	2552	Output parity and gap-size indicator
45-46	2553	Error options
47	2555	Printer control unit
48	2556	Not Used (Must be blank)
49-54	2557	Highest memory address available
55	2565	Collating sequence translation
56	2566	Ascending or descending sequenced output
57	2567	Rewind option
58-63	2570	Address of header-trailer own-coding
64-69	2576	Address of item-by-item own-coding
70-74	2604	Number of data records per output reel
75-80	2611	Not used
81-84	2617	1st key location
85-86	2623	1st key length
87-90	2625	2nd key location
91-92	2631	2nd key length
.		.
.		.
135-138	2705	10th key location
139-140	2711	10th key length
141	2713	Input file name and reel number check
142-151	2714	Output file name change
152-157	2726	Next program name
158-159	2734	Next segment name
160	2736	Search direction for next program

TAPE POSITIONING

Note: The letters used to label the following paragraphs are also used to designate the corresponding diagrams in Figure 11.

Beginning of Collate

Input Files

A. If all tapes are to be rewound at the beginning of the collate (parameter character 57), the input tapes may be positioned at any points. 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 (1ERI).

OWN-CODING

Own-coding routines are written by the programmer and included in the Collate 2 program. The presence of these routines is indicated to the Collate program by means of parameter characters. Characters 58-63 designate a branch to a header-trailer own-coding address, where the user may inspect, modify, or replace the tape labels. Characters 64-69 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 Own-Coding

The preparation of Collate 2 own-coding routines and the registers referenced by such routines are identical to those described for Sort 2 last-pass own-coding on page 25.

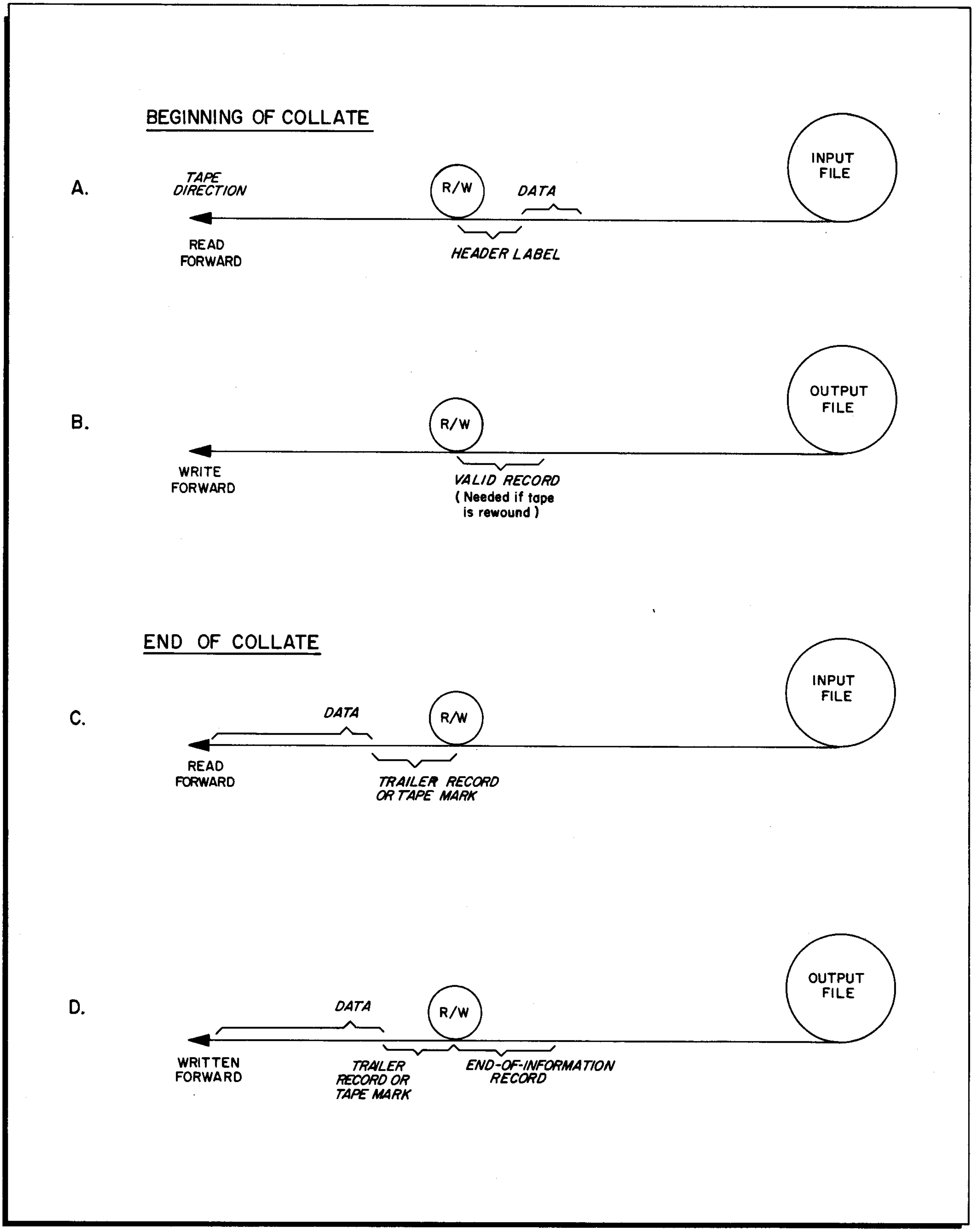


Figure 11. Collate Tape Positioning

HONEYWELL
ELECTRONIC
DATA
PROCESSING
WELLESLEY HILLS,
MASSACHUSETTS 02181