File No. 1401/1460-33
Form C24-1420-5

IBM Systems Reference Library

Sort Programs for IBM 1401-1311 and 1460-1311
Specifications and Operating Procedures
Generative Program: Sort 6
Object Programs: Sorts 61, 62, 63, and 64

Sort 6 Program	*1401-SM-062*
Sort 61 Program	*1401-SM-067*
Sort 62 Program	*1401-SM-064*
Sort 63 Program	*1401-SM-065*
Sort 64 Program	*1401-SM-066*

This reference publication discusses these topics:

Sort 6 Specifications describes the requirements for generating a sort object program.

Sort 6 Operating Procedures describes the Sort 6 program deck, preparation of the Autocoder system pack and library for generation, and the operating procedures to be followed when generating a sort object program.

Sort Object-Program Specifications describes the characteristics of object programs generated by Sort 6 and the requirements for executing the object programs. Sorts 61, 62, 63, and 64 are discussed in this section.

Sort Object-Program Operating Procedures describes the IBM-supplied generalized object decks (Sorts 61, 62, 63, and 64), the insertion of user-prepared control cards in the object deck, the operating procedures to be followed when executing the object program, and the halts and messages that are associated with the sort object programs.

For a list of associated publications and abstracts, see the *IBM 1401 and 1460 Bibliography*, Form A24-1495.

Preface

This reference publication contains the specifications and operating procedures for the Sort 6 generative program and for sort object programs generated by Sort 6.

Four sort object programs, Sorts 61, 62, 63, and 64 are available from IBM. The specifications for these programs are presented under the heading, *Sort Object Program Specifications.*

The selection of the Sort 6 generative program or any of the four sort object programs, will depend on the needs of the individual users. The sort object program decks generated by Sort 6 are available to users who may not require the versatility of Sort 6 and whose needs are satisfied by the supplied sort object programs.

Sixth Edition, July 1966

This is a reprint of C24-1420-4 incorporating changes released in the following Technical Newsletter:

Form Number	Pages Affected	Date
N21-0035-0	25, 27, 41, 63, 64, 68.1	March 15, 1965

Significant changes or additions to the specifications contained in this publication will be reported in subsequent revisions or Technical Newsletters.

Copies of this and other IBM publications can be obtained through IBM Branch Offices. A form is included at the back of this manual for readers' comments. If this form has been removed, address comments to: IBM Corporation, Programming Publications, Dept. 425, Rochester, Minn. 55901.

Contents

Sort 6 is a generative program that produces a generalized sort object program to be used on an IBM 1401-1311 or 1460-1311 Data Processing System. The object program consists of an assignment phase that checks control-card information, and six phases that sort the records according to the control-data fields within the records. Figure 1 shows Sort 6 and sort object program operations.

The following components make up the Sort 6 program:

1. The sort prephase: part 1 (SORT) and part 2 (MOSHK)
2. Nine sets of library routines: LODER macro, SORT macro, phase 0 (assignment phase), phase 1, phase 2, phase 3, phase 4, phase 5, and phase 6.

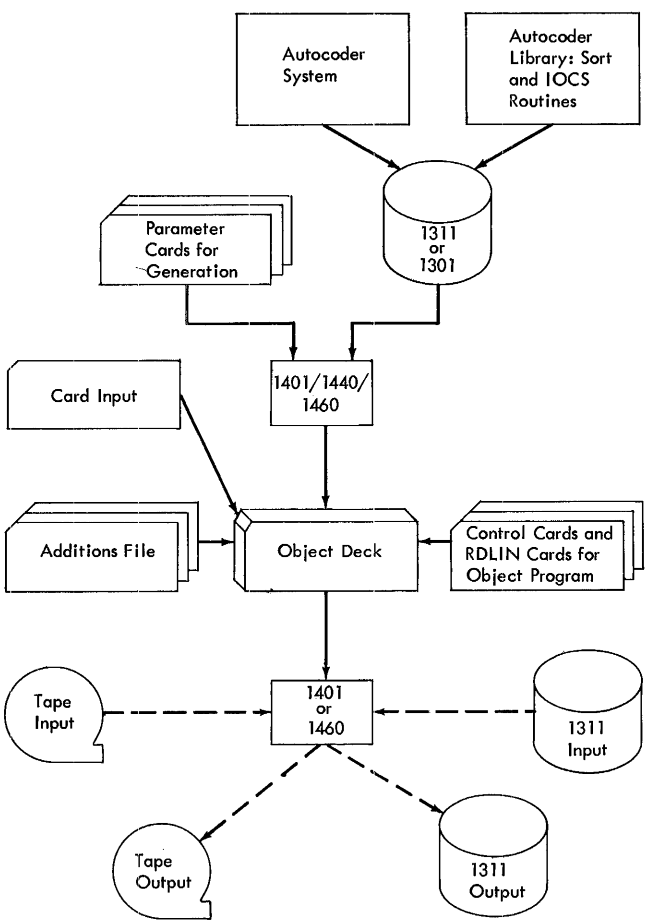

Figure 1. Sort 6 and Sort Object Program Operations

Parameter cards, prepared by the user, supply the Sort 6 program with the general characteristics of the user's sort applications. The result of the complete Sort 6 generation is a sort object deck that contains an assignment phase and phases 1-6.

Phase 0 is an assignment phase that initiates the execution of the object program. Phases 1-6 contain the routines that accomplish the sort. The kind of output desired determines the number of phases to be executed. In control card 1, the user can request termination after phase 4, 5, or 6. For further information on the kinds of output, see *Output Available.*

All sort object programs:

- Sort blocked or unblocked fixed-length records in the move mode.
- Sort numeric or alphameric records in ascending or descending collating sequence according to control data. Each record can contain a maximum of ten control-data fields.
- Can delete characters from sorted records
- Utilize the IBM 1401/1460 disk IOCS program
- Provide checkpoint and restart procedures

Other capabilities of the object program depend on the parameters specified. An object program can include routines for:

- Card, tape and/or disk input
- Additions from cards with disk input
- Disk and/or tape output
- Deletion or selection of records according to class
- Deletion according to control data
- Hash totals and a sequence check
- Linkage to user routines
- Utilization of the direct-seek feature and/or console printer (without a buffer feature).

At object time, the user's control cards supply the object program with the specific characteristics of the sort application. This procedure permits the utilization of the same object program for several sort applications that differ in their specific characteristics.

Required Programs

In addition to the Sort 6 programs, the following programs are required for the generation of a sort object program:

- 1401, 1440, 1460 Autocoder (on Disk), program number 1401-AU-008
- 1401, 1460 Input/Output Control System (on Disk), Version 2, program number 1401-IO-068.

Machine Requirements

The Sort 6 program can be used to generate a sort object program on any system that satisfies the minimum machine requirements for the 1401, 1440, 1460 Autocoder (on disk) program. The requirements are:

An IBM 1401 system with:

4,000 positions of core storage
High-Low-Equal Compare Feature
One IBM 1311 Disk Storage Drive
One IBM 1402 Card Read-Punch
One IBM 1403 Printer.

An IBM 1440 system with:

4,000 positions of core storage
One IBM 1301 Disk Storage or
one IBM 1311 Disk Storage Drive
One IBM 1442 Card Reader
One IBM 1443 Printer.

An IBM 1460 system with:

8,000 positions of core storage
One IBM 1301 Disk Storage or
one IBM 1311 Disk Storage Drive
One IBM 1402 Card Read-Punch
One IBM 1403 Printer.

The following can be used if available:

IBM 1404 Printer
IBM 1444 Card Punch
IBM 1447 Console without a buffer feature.

Related Information

The user should be familiar with the following SRL publications:

Autocoder (on Disk) Language Specifications for IBM 1401, 1440, and 1460, Form C24-3258

Autocoder (on Disk) Program Specifications and Operating Procedures for IBM 1401, 1440, and 1460, Form C24-3259

Input/Output Control System (on Disk) Specifications for IBM 1401 and 1460, Form C24-1489.

Sort 6 Generation

A Sort 6 generation is performed under control of the 1401, 1440, 1460 Autocoder system. Only one phase of the sort object program can be generated per job. The input for the generation job is a deck that contains Autocoder control cards, parameter cards for the phase to be generated, and any user routines that are to be incorporated in the phase. All user routines must be written in the Autocoder language.

The sort prephase, which is part of the Autocoder system, checks the parameter cards and translates the information into a format that is recognized by the Autocoder macro generator. The macro generator selects the appropriate library routines for the phase being generated and modifies the routines according to the parameters supplied by the user. Although the Sort 6 library routines are selected by the macro generator, they cannot be used as subroutines in other Autocoder programs because the sort prephase sets permanent switches that are used by the sort library routines.

When the phase is being generated, user routines, if included in the job deck, are assembled. Thus, the result of each generation job is an object program that contains one sort phase and any user routines that are to be executed during that phase of the sort job.

If the general characteristics of the users sort application(s) change, only those phases affected by the change need to be regenerated. For example, if the user wants tape output but has an object program that handles only disk output, he can regenerate phase 6 to revise his object program.

Requirements for Generation

The requirements for Sort 6 generation are:

1. Sort prephase incorporated in the Autocoder system.
2. SORT macro, LODER macro, appropriate phase, and 1401, 1460 disk IOCS library routines in the same Autocoder library.
3. A job deck for each phase that is to be generated.

Sort Prephase

The sort prephase must be incorporated in the preprocessor portion of the 1401-1440-1460 Autocoder system. The prephase is transferred from cards to the disk unit (SYSTEM file) by the Autocoder update job. Because a fixed area on the SYSTEM file is reserved for the sort prephase, the update job need be performed only once.

All the control and data cards required for the update job are included in the Sort 6 program deck. The first card in the prephase section of the deck is an AUTOCODER RUN card; the last card is an END card.

Library Routines

The SORT macro, the LODER macro, and the 1401, 1460 disk IOCS routines required for every phase generation must be in the same Autocoder library.

The sets of phase routine(s) associated with the phase(s) to be generated must also be in the library. For example, if the assignment phase is the only phase to be generated, only the phase-0 routines are required in addition to the SORT macro, LODER macro, and IOCS routines.

The user should perform an Autocoder library change job to transfer the necessary routines to the library. All necessary INSER cards are included in the Sort 6 program deck. A LIBRARY ASGN card is required if the Autocoder library has been relocated. (See *Sort 6 Operating Procedures* for a discussion of the librarian-control cards that are required for the job.)

Job Deck

A job deck is required for each phase generation. The deck must contain Autocoder control cards and parameter cards, and may contain user routines that are written in the Autocoder language.

Autocoder Control Cards

The Autocoder cards that are required for the generation of each phase are: RUN, CTL, SORT, and END. A JOB card may be inserted ahead of the CTL card.

RUN Card

The RUN card is punched in the following manner:

Columns	*Contents*
6-14	AUTOCODER
16-18	RUN
21-24	THRU
26-31	OUTPUT

Note: AUTOCODER RUN THRU OUTPUT is the conventional assembly that yields an object program deck in the condensed-loader format.

JOB Card

A JOB card, punched in the following manner, can be included:

Columns	*Contents*
16-18	JOB
21-72	*Any identification*
76-80	*Any identification*

CTL Card

Figure 2 shows the format of a CTL card that is to be used with Sort 6. The object machine size, specified in column 21, must be the same for the generation of all phases that make up the object program. For example, if phase 1 is generated for a 4K machine, it cannot be inserted in a 12K object deck.

SORT Card

The SORT card is punched in the following manner:

Columns	*Contents*
16-19	SORT

This card causes the sort prephase to be read into core storage from the SYSTEM file.

END Card

The END card is punched in the following manner:

Columns	*Contents*
16-18	END

Parameter Cards

Parameter cards specify the general characteristics of the sort application. All parameter cards are punched in the following manner:

Columns	*Contents*
1-5	Blank
6-15	Name of the parameter card
16-20	Blank
21-80	Parameters. A comma must follow each parameter but the last on a coding-sheet line. Leave no columns blank between column 21 and the last character of the last parameter. The parameters can be in any order.

Figure 3 is a summary of the parameter cards. The cards that are designated as mandatory are required for each phase generation.

If disk is the only kind of input specified, and additions, deletions by class, and/or selections are not specified, the generator includes only the routines that

Columns	Indicates	Punch (Meaning)	Assumptions if the Columns Are Left Blank
16-18	Mnemonic operation code	CTL	
21	Object machine size	1 (4K) 2 (8K) 3 (12K) 4 (16K)	4K
22	Modify address feature available	1 (Yes)	No, if the object machine has 4K; Yes, if the object machine has 8, 12, or 16K.
23	This column must be left blank		
24	Multiply-Divide feature available	1 (yes)	No
25	Object machine	0 (1401) 6 (1460)	Processor machine
26	Punch device	P (1402)	P if the object machine is a 1401 or 1460
27	Read device	P (1402)	P if the object machine is a 1401 or 1460
28	Print device	P (1403)	P if the object machine is a 1401 or 1460
29	Disk device	1 (1311)	1311
30	Source Statement Diagnostics	1 (yes) N (no)	Yes
31	Label Table or Cross Reference Listing	L (label table) N (neither)	Cross reference listing
32-36	Read-in area	00001	00001
37-41	Loader location	Assignment phase-03100 Phase 1-00334 Phase 2-02101 for 4K; 02250 for 8K, 12K or 16K Phase 3-00100 Phase 4-00100 Phase 5-02001 for 4K; 02251 for 8K, 12K, or 16K Phase 6-00100	

Figure 2. CTL Card Format

are necessary for the user's object program. In this case, the object program will contain the disk-input routines with full label checking.

If card input, tape input, the additions option, the deletions by class option, or the selections option is specified in the parameter cards, the object program will contain all (card, tape, and disk) input routines, full label-checking routines, the additions routines, and the deletions by class and selections routine.

Therefore, even though the requirements of the sort application change, none of the phases may have to be regenerated. For example, if the current object program has been generated to write the output on tape, and the user desires to have the output written in disk storage, only phase 6 may have to be regenerated. This would be the case if the current phase 6 contained only tape-output routines.

When the job deck for generation is prepared, only those cards that specify parameters that affect the phase to be generated are required. (*Exception:* The INPUTMEDIA and OUTPTMEDIA cards are always required; the TAPE card is required if the INPUTMEDIA card specifies TAPE.) If any non-required parameter cards are included, they are bypassed.

INPUTMEDIA Card

This card specifies the kind(s) of input that the object program is to accommodate. CARD, DISK, and TAPE are the valid parameters. Although three kinds of input can be specified, the object program can accommodate only one of the following kind(s) of input per sort job: cards or disk or tape or disk with card additions. The proper input routines are selected at object time on the

Name of Parameter Card	Valid Operands	Specifies	Mandatory for Generation of All Phases	Phase Affected
INPUTMEDIA	CARD, DISK, TAPE	Kinds of input	Yes	Phase 1
TAPE	UNLOADIN	Tape rewind and unload option	No	Phase 1
	UNLOADOUT	Tape rewind and unload option	No	Phase 6
	DUMP or SCAN	Procedure for unreadable blocks or wrong length records (tape input).	Yes, if TAPE is specified in the INPUTMEDIA card.	Phase 1
OUTPTMEDIA	DISK, TAPE	Kind(s) of output	Yes	Phase 6
PHASE	0, 1, 2, 3, 4, 5, or 6	Phase to be generated	Yes	Phase specified
SYSTEMSPEC	DIRECT	Direct seek feature	No	Phases 1-6
	CONPRINT	Console printer without the buffer feature.	No	Phases 0-6
SELECTDLET	SELDELCLASS	Deletion or selection by class	No	Phase 1
	CONTROLDATA	Deletions by control data	No	Phase 4
ADDITIONS	CARD	Card additions (disk input only)	No	Phase 1
CHECKS	SEQHASH	Hash totals and sequence check	No	Phase 2, 3, and 4
EXITS	EXIT1, EXIT2, EXIT3, EXIT4	Linkage to phase 1 user routines	No	Phase 1
	EXIT5, EXIT6	Linkage to phase 4 user routines	No	Phase 4
	EXIT7, EXIT8	Linkage to phase 6 user routines	No	Phase 6
USERAREA	1	Assembly of user routines in phase 1	No	Phase 1
	4	Assembly of user routines in phase 4	No	Phase 4
	6	Assembly of user routines in phase 6	No	Phase 6
DSKLABLOUT	ALL	Checking of existing disk header labels; generation and writing of disk header labels for the output area(s).	No	Phase 6
TAPLABLIN	STANDARD or NONSTANDARD	Kind of labels on the input tape (Specify ALL or IDENT and A, B, or C if STANDARD is specified.)	No	Phase 1
	TM	Presence of tape mark following the header label.	No	Phase 1
	ALL or INDENT	Kind of label checking desired.	No	Phase 1
	A, B, C	Type of standard labels	No	Phase 1
TAPLABLOUT	STANDARD or NONSTANDARD	Kind of labels to be generated and written on the output tape (Specify ALL and A, B, or C if STANDARD is specified.)	No	Phase 6
	ALL	Checking of file retention period in existing output tape header labels	No	Phase 6
	A, B, C	Type of standard labels to be generated.	No	Phase 6

Figure 3. Parameter-Card Summary

basis of information punched in the object-program control cards.

If the object program is to process card additions, include the ADDITIONS parameter card in the source deck. (CARD need not be specified in the INPUTMEDIA card unless *all* the input is from cards.)

If TAPE is specified in the INPUTMEDIA card, the TAPE parameter card is also required.

The INPUTMEDIA card must be included in the job deck for each phase (0-6) generation.

TAPE Card

This card specifies that certain tape rewind options are desired and also the procedure to be followed for unreadable blocks or wrong-length records when input is from tape. The valid parameters are:

UNLOADIN: Rewind and unload the input tape when end-of-file is reached. This parameter should be specified if input and output tapes are to be mounted on the same drive.

UNLOADOUT: Rewind and unload the output tape after closing the tape.

DUMP OR SCAN: Specify DUMP if unreadable tape blocks or wrong-length records are to be written on tape unit 4. The dump tape will be rewound and unloaded at the end of phase 1.

Specify SCAN if unreadable tape blocks or wrong-length records are to be corrected through the use of the scan option. When this option is specified, the program halts to give the operator an opportunity to investigate the error.

If TAPE is specified in the INPUTMEDIA card, one of these parameters must be specified.

OUTPTMEDIA Card

This card specifies the kind of output that the object program is to accommodate. DISK and TAPE are the valid parameters. The object program can accommodate only one kind per sort job and will contain the output routine(s) associated with the parameter(s) specified. The proper output routine is selected at object time on the basis of information punched in the object-program control cards.

The OUTPTMEDIA card must be included in the job deck for each phase (0-6) generation.

PHASE Card

This card specifies the object program phase that is to be generated. Only one parameter can be specified per generation job. The phases of the program are numbered as follows:

0–Phase 0 (assignment phase)
1–Phase 1
2–Phase 2
3–Phase 3
4–Phase 4
5–Phase 5
6–Phase 6

SYSTEMSPEC Card

This card specifies the availability of the direct-seek feature and/or console printer on the object system. If the direct-seek feature is to be used, the parameter is DIRECT. If the console printer is to be used, the parameter is CONPRINT. When the object deck contains the CONPRINT routines, all messages will be printed by the console printer.

Note. A console printer with a buffer feature cannot be used.

ADDITIONS Card

This card specifies the inclusion of a routine that can handle card additions to disk input. The required parameter is CARD. At object time, the routine is selected if control card 1 specifies additions.

CHECKS Card

This card specifies the inclusion of routines that accumulate hash totals during phases 2, 3, and 4 and perform a sequence check during phase 4.

The required parameter is SEQHASH. If this parameter card is used for the generation of phase 2, 3, or 4, it must be used for the generation of all three.

The routines are selected at object time if the user specifies hash totals and sequence check in control card 1.

SELECTDLET Card

The valid parameters and associated routines are:

1. SELDELCLASS for deleting and selecting records from the input file before the sorting process begins (phase 1). At object time, records may be either deleted or selected. The choice is specified in control card 1 for the object program.

   The position in the records and the character controlling the deletion or selection also are specified in the control card.

If the input is from cards or tape, all the records will be written on the disk input area, but only those specified will be sorted.

2. CONTROLDATA for deleting records on the basis of control data after the records have been sorted (phase 4). At object time, if deletions are specified in control card 1, the user must insert, in the object deck, cards containing the control data for the records to be deleted.

EXITS Card

This card specifies the particular exits that are to be used during a sort job. The exits available are:

Phase	*Exit*	*Available*
Phase 1	EXIT 1	During the disk or tape input routine.
Phase 1	EXIT 2	Before card additions have been processed
Phase 1	EXIT 3	During the deletions or selections routine.
Phase 1	EXIT 4	During the card input routine.
Phase 4	EXIT 5	After the records have been sorted and before each record's 10-character address is written in disk storage. The 10-character address consists of a 6-character disk address, a 2-character block location factor, and a 2-character adjustment factor.
Phase 4	EXIT 6	During the deletions routine.
Phase 6	EXIT 7	During the disk or tape output routine.
Phase 6	EXIT 8	At the end of the sort job.

Any combination of EXIT parameters can be specified. Sort 6 generates a branch instruction for each exit associated with the phase being generated. It also generates the label (ENTRY*x*) to be used for the return to object program main line. No label is generated for exit 8 because it occurs after the sort job has been completed.

If the EXITS parameter card is included in the job deck, a USERAREA card and the appropriate user routine(s) are also required. See *User Programming.*

USERAREA Card

This card specifies that a user routine is to be assembled during the generation job. A USERAREA card is required whenever an EXITS card is included in a phase -1, phase -4, or phase -6 job deck. The relationship between the EXITS parameters and the USERAREA parameters is:

EXITS	*USERAREA*
EXIT1, EXIT2, EXIT3 and/or EXIT4	1
EXIT5 and/or EXIT6	4
EXIT7 and/or EXIT8	6

Punch the number of the phase (1, 4, 6) in which the user routine is to be generated.

DSKLABLOUT Card

This card specifies that existing labels on the disk output areas are to be checked, and that new standard header labels are to be generated according to output RDLIN card information and written on the disk label track for each output area.

If, during the checking of labels, an unexpired file is found to be within the limits specified in an output RDLIN card, the program halts and the user is given the option of placing a new pack on the drive or of using the pack currently on the drive.

The required parameter for the DSKLABLOUT card is ALL.

TAPLABLIN Card

If labels are present on the input tape, use the TAPLABLIN card to supply the following information about the labels:

1. The kind of labels: Specify STANDARD or NONSTANDARD. If NONSTANDARD is specified, a branch to EXIT0 is generated and a user-routine designed to process or bypass the labels must be included in the phase-1 job deck.
2. The presence of a tape mark after the input header labels: Specify TM.
3. The kind of standard-label checking desired: Specify ALL or IDENT. ALL or IDENT must be specified if the labels are standard. If ALL is specified, the file serial number, reel sequence number, file name and creation date of each input header label will be checked against input area RDLIN card information. If IDENT is specified, only the file name is checked. See *Object Program Specifications: RDLIN Cards.*
4. The type of standard labels: Specify A for 120-character labels, B for 80-character labels, or C for 84-character labels. A, B, or C must be specified if the labels are standard.

TAPLABLOUT Card

If a label is to be written on the output tape, use the TAPLABOUT card to supply the following information about the labels:

1. The kind of label: Specify STANDARD or NONSTANDARD. If STANDARD is specified, labels will be generated according to output RDLIN card information and written on the output tape. See *Object Program Specifications: RDLIN Cards.*

If NONSTANDARD is specified, a branch to EXIT9 is generated and a user routine designed to write the output label must be included in the phase-6 job deck.

2. Checking of existing labels: Specify ALL if a standard label is to be generated. The retention period in the existing labels will be checked. If the retention period has not expired, a halt occurs.

   If a nonstandard label is to be written, the object program does not check the retention period. The user-routine (EXIT9) should check the retention period in any existing labels.

3. The type of standard labels: Specify A for 120-character labels, B for 80-character labels, or C for 84-character labels. A, B, or C must be specified if a standard label is to be generated.

   If A is specified, a tape mark is written after the label. If B or C is specified, no tape mark is written.

User Programming

Exits are provided at various points in phases 1, 4, and 6 of the object program to permit the inclusion of user routines.

To utilize exits 1-8:

1. Prepare an EXITS parameter card that specifies the exit(s) that are to be used and a USERAREA parameter card that specifies the phase(s) during which the routine(s) are to be assembled.
2. Code the routine(s) in the disk Autocoder language and punch the routine(s) into cards.
3. Include the EXITS and USERAREA parameter cards and the routine(s) in the job deck for the generation of the phase in which the routine(s) will be used.

To utilize exits 0 and 9:

1. Code the nonstandard tape-label processing routine in the Autocoder format and punch the routine into cards.
2. Include the routine for processing input labels in the job deck for phase-1 generation. Include the routine for processing output labels in the source deck for phase-6 generation.

During the generation job, Sort 6 develops linkage to the user's routine(s) and assembles the routine(s).

Writing the Routines

The following items must be taken into consideration when writing user routines:

1. All labels must be unique. It is suggested that all labels, except the label of the first instruction to be executed, end in zz. The first instruction to be executed must be labeled EXIT*x* when *x* is the number of the exit. Sort 6 develops linkage to the user routine by generating a branch to EXIT*x*.
2. The last instruction must be a branch to ENTRY*x* where *x* is the number of the exit. (*Exception:* see *Exit 8.*)
3. The number of core locations available for user routines depends on the phase in which the exit is used, the combination of parameters specified for generation, the object-machine size, and the routines that are to be selected at object time. Figure 4 shows the number of positions available for user programming in phases 1, 4, and 6.

4a. Routines for exits 0 and 9. The first statement must be a LTORG statement with an asterisk in column 21. No ORG statement is required because Sort 6 determines the origin. The EXIT0 routine must immediately follow the parameter cards in the phase-1 job deck; the EXIT9 routine must immediately follow the parameter cards in the phase-6 job deck.

4b. Routines for exits 1-4. These routines can be assembled in any order. For example, the exit-4 routine could follow the exit-1 routine in the generation job deck. The first two statements of the first routine to be assembled in the phase-1 generation must be a LTORG statement with an asterisk in column 21 and an ORG statement.

4c. Routines for exits 5 and 6. The procedure is the same as that for routines for exits 1-4.

4d. Routines for exits 7 and 8. If routines for both exits are to be assembled, the exit-7 routine must precede the exit-8 routine. The first statement in each of the routines must be a LTORG statement with an asterisk in column 21. The second statement in each routine must be an ORG statement.

5. To determine the operand of the ORG statement, approximate the number of core positions required for the routine(s) and subtract it from the highest core position available (3999 for 4K, 7999 for 8K, 11999 for 12K, 15999 for 16K). Use the result as the operand of the ORG statement. See Figure 4 for the number of positions available for user routines.

6. Do not use the INCLD, CALL, or any IOCS macro instructions in any user routines.

   Do not use LTORG and END statements within user routines.

7. If the routine uses the index registers, the contents of the affected positions must be saved. After the user routine has been executed and before the branch back to the object program main line, the contents, including word marks, of the positions affected must be restored. Any position in any of these areas may contain a word mark.

Phase and Exits	Factors that Affect the Number of Positions Available: Generation Time	Factors that Affect the Number of Positions Available: Object Time	Object Machine Size	Number of Positions Available	Remarks
Phase 1: EXIT0	TAPLABLIN card specifies NONSTANDARD		4K 8K 12K 16K	200	Sort 6 generates the ORG statement for the routine. These 200 positions are not counted in the number of positions available for EXIT1, EXIT2, EXIT3, and EXIT4 routines.
Phase 1: EXIT1 EXIT2, EXIT3, EXIT4	IMPUTMEDIA card specifies DISK only		4K 8K 12K 16K	900 3540 5200 8540	If more than one of the limiting factors applies to the sort application, the smallest number is the maximum number of positions available. Examples: Assume that the object machine size is 8K. If the SELECTDLET SELDELCLASS parameter card is included in the job deck for the generation of phase 1 and the user intends to delete records at object time, 2607 positions are available for user programming. If the SELECTDLET SELDELCLASS parameter card is included in the job deck and the user does not intend to select records at object time, 3010 positions are available. The inclusion of user routines reduces the number of input blocks that can be read into core storage at one time. To determine the number of core positions required for one block: N = 100 x Number of sectors required for the block. The maximum number of blocks that can be read is reduced by one for every N or part of N positions occupied by user routines.
	INPUTMEDIA card specifies TAPE or CARD		4K 8K 12K 16K	375 3010 5610 8010	
	SELECTDLECT card specifies SELDELCLASS				
	ADDITIONS card specifies CARD				
		Control Card 1 will specify tape input	4K 8K 12K 16K	225 2840 5440 7840	
		Control card 1 will specify card input	4K 8K 12K 16K	100 2730 5330 7730	
		Control card 1 will specify deletions by class	4K 8K 12K 16K	7 2607 5207 7607	
		Control card 1 will specify card additions	4K 8K 12K 16K	80 2680 5280 7680	
Phase 4: EXIT5, EXIT6	CHECKS card specifies SEQHASH		4K 3K 12K 16K	100 100 300 600	The inclusion of user routines do not affect the efficiency of the sort.
		Control card 1 will specify hash totals and a sequence check	Subtract the total length of the control data fields from the appropriate number above.		
			4K 8K 12K 16K	50 200 400 700	
Phase 6: EXIT7, EXIT8			4K 8K 12K 16K	100 1550 3500 5200	The inclusion of user routines reduces the number of sectors of data that can be written at one time. The number of sectors is reduced by 1 for each 100 or part of 100 core positions occupied by user routines.
Phase 6: EXIT9	TAPLABLOUT card specifies NONSTANDARD		4K 8K 12K 16K	900	Sort 6 generates the ORG statement for the routine. The positions available for EXIT9 are not counted in number of positions available for EXIT7 and EXIT8 routines.
	TAPLABLOUT card specifies NONSTANDARD and DSKLABLOUT card specifies ALL		4K 8K 12K 16K	450	

Figure 4. Exits

Use of Exits

For each exit, the following information is given:

Sort Job Characteristic: The object-time characteristic that determines whether the user routine is to be executed.

Availability: The point in the object program when the exit is available.

Status: The sort operation performed immediately before or after the exit.

Symbolic Address: Represents either the address of data that the user may want to modify or the address of the core-storage positions that contain the address of data. All of the symbolic addresses can be found on the program listing.

Use: A suggestion for a user routine.

Remarks: Any restrictions or other information that pertains to the use of the exit.

See Figure 4 for the number of core-storage positions available for the user routines. The user must adhere to the rules given under *Writing the Routines.*

Phase-1 Exits

The exits that are available during phase 1 are exit 0, exit 1, exit 2, exit 3, and exit 4. If the TAPLABLIN parameter card with NONSTANDARD specified is used for generation, exit 0 is generated. If any of the other phase-1 exits are to be generated, the EXITS and USERAREA parameter cards must be included in the phase-1 generation job deck.

Exit 0

Sort Job Characteristic: Tape input with nonstandard labels.

Availability: Beginning of and during phase 1.

Status: Tape input files have not been opened.

Symbolic Address: P1RD2+4 is a 1-position field that contains a group-mark with a word-mark.

Use: This routine must open the input file and check or bypass nonstandard tape labels. When end-of-file or end-of-reel is reached, the routine must close the tape, and, if a new tape is mounted, the routine must open the new tape before branching back to the phase-1 main line.

Remarks:

1. Any group marks with word marks that are set by the user's routine must be cleared before returning to the object-program main line.
2. At object time, the routine is placed in the checkpoint area of the disk-storage work area. Thus, it does not occupy any core storage positions during phase-1 main line.

Exit 1

Sort Job Characteristic: Tape or disk input.

Availability: After each group of records has been read into core storage. The group that is read into core storage is not necessarily the same size as the user's input block because the sort program takes advantage of all available core storage by reading in a multiple of the user's blocks. The number of positions available for the sort input area depends on the specific characteristics of the sort application, the core-storage capacity of the object machine, and the size of the user routines. See Figure 4.

Status: No processing has been performed.

Symbolic Addresses: LFTEDG is a 3-position field that contains the address of the position before the input area,

TOPAR is a 3-position field that contains the address of the position before the group mark which indicates the end of the input area.

Use: Modify the records or perform calculations from data in the records.

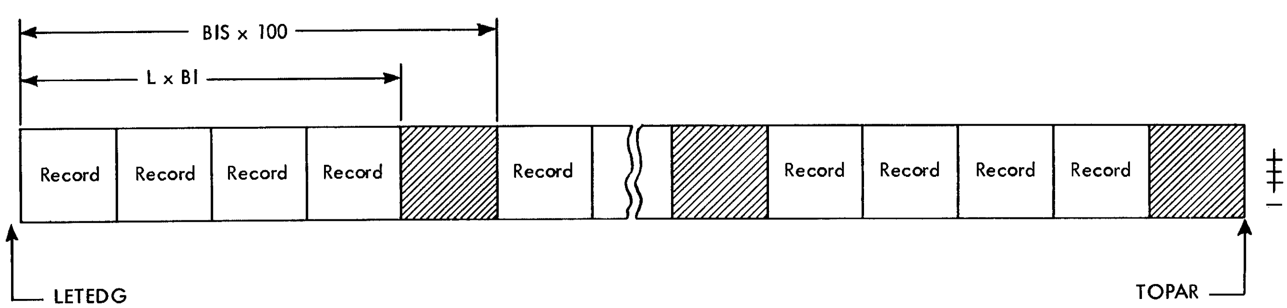

Figure 5. Input Area for Disk Input

Remarks: The number of positions in the core-storage input area is always a multiple of 100. The sort object program reads as many of the user's blocks as possible into the input area.

If the input is from disk and the user's input block does not fill the last sector of the disk area required by the input blocking factor, the number of unused positions must be taken into consideration when the starting address of the next block in the core-storage input area is determined.

Figure 5 shows the record arrangement in core storage when the input blocking factor is four and the block length is not equal to a multiple of one-hundred.

If the input is from tape, the records may be reblocked by the sort program before they are written in the disk-storage input area. The group of user blocks in the core-storage input area will not fill the area if the length of the group of blocks is not a multiple of 100. This fact should be taken into consideration when the end of the group of blocks in the input area is determined.

The user's routine must contain the necessary programming to check the first five positions of each block for a 1EOFb to determine when the last record has been processed.

Figure 6 shows the record arrangement in core storage when the input blocking factor is four and the length of the group of blocks is not equal to a multiple of one-hundred.

Exit 2

Sort Job Characteristic: Card additions to disk input.

Availability: Immediately after the additions routine has been initiated.

Status: No records have been processed.

Symbolic Addresses: Not applicable.

Use: Modify the additions routine.

Remarks: Note that this exit is available only once.

Exit 3

Sort Job Characteristics: Selections or deletions on the basis of a controlling character.

Availability: Immediately before each record is checked for the controlling character.

Status: The tag (control-data field, 6-character disk address, 2-character block location factor) has been formed and stored in a core-storage output area.

Symbolic Addresses: LFTEDG is a 3-position field that contains the address of the low-order position of the record about to be processed.

DEINST+3 is a 3-position field that contains the address of the controlling character about to be processed.

XCDMOV+6 is a 3-position field that contains the address of the low-order position of the control-data field portion of the tag in the output area.

CWCH4+6 is a 3-position field that contains the address of the low-order position of the whole tag.

Use: Insert the controlling character.

Remarks: All records will be written in the disk input area. Only the tags of the selected or not deleted records will be sorted.

Exit 4

Sort Job Characteristic: Card input (not additions).

Availability: After each group of card records has been read. The number of records read depends on the specific characteristics of the sort application, the core-storage capacity of the object machine, and the size of the user routines. See Figure 4.

Status: No processing has been performed.

Symbolic Addresses: BEGN is a 3-position field that contains the address of the position before the group of records in the input area.

AUXBI is a 2-position field that contains the number of records in the input area. A- and B-bits are

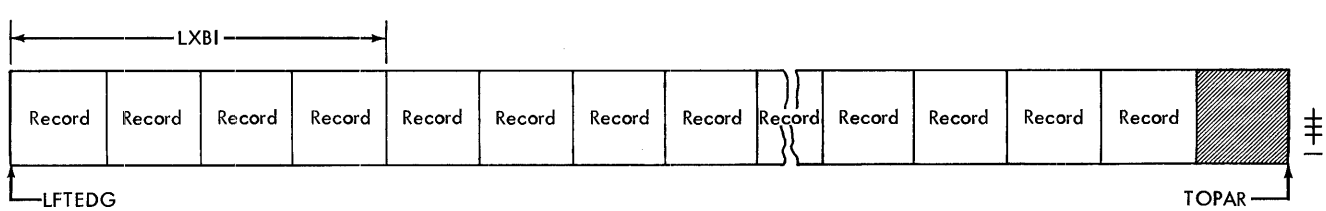

Figure 6. Input Area for Tape Input

present in the units position of the field. If, for example, 25 records are present in the input area, AUXBI contains 2E. If 100 records are present, AUXBI contains +? (? represents a plus zero).

Use: Modify the records or perform calculations from data in the records.

Remarks: The user should establish a count of the records he processes in each block and compare the count with the contents of AUXBI to determine when the end of the block has been reached.

Phase-4 Exits

Exits 5 and 6 are available during phase 4.

Exit 5

Sort Job Characteristics: Phase 4 (8-character address), phase 5, or phase 6 output.

Availability: After the 8-character address (phase 4 output) or the 10-character address (phase 5 or 6 output) of each record is formed in the output area.

Status: Last merge pass that results in a sequenced string of tags (8- or 10-character address) is being performed.

Symbolic Addresses: OUTCOM+6 is a 3-position field that contains the low-order address of the 8- or 10-character address in the output area.

CNVBKT is a 3-position field that contains the low-order address of the tag.

Use: Modify or delete the address.

Remarks: Both the 8-character address and the control data are available. However, the exit is *not* available if phase 4 control-data output is desired.

Exit 6

Sort Job Characteristic: Deletion by control data.

Availability: Before the control-data portion of each tag is checked, and before the control data and address are moved to the output area.

Status: Last merge pass that results in a sequenced string of tags (control data, 8-character or 10-character address) is being performed.

Symbolic Addresses: CNVBKT is a 3-position field that contains the address of the low-order position of the control-data portion of the tag.

Use: Modify the deletions routine or control data.

Phase-6 Exits

The exits that are available during phase 6 are exit 7, exit 8, and exit 9. If the TAPLABLOUT parameter card with NONSTANDARD specified is used for generation, exit 9 is generated. If any of the other phase-6 exits are to be generated, the EXITS and USERAREA parameter cards must be included in the phase-6 generation job deck.

Exit 7

Sort Job Characteristic: Phase-6 output.

Availability: Immediately before one or more of the user's output blocks are written in the disk-storage output area.

Status: Records are being read into core storage from the disk-storage input area.

Symbolic Addresses: MVE+6 is a 3-position field that contains the low-order address of the last record moved to the output area.

FIGIT8 is a 3-position field that contains the address of the position before the first record in the output area.

TOUT2, TOP2, and TOP1 are the low-order addresses of blocks to be written in the disk-storage output area.

CKPT—3, CYLN2, CYLN, SWCH2, MDFY+4, and NOTST are test fields.

Use: Modify sorted records.

Remarks: Phase 6 writes one or more of the user's blocks during each disk-write operation. The number of blocks written depends on such factors as the number of records in each of the user's blocks and the number of sectors required for each block.

The formats of the blocks written during the disk-write operations are shown in Figure 7.

BOS is the number of output sectors required for an output block. L is the length of an output record. BO is the output blocking factor. UNUSD represents the difference between BOS x 100 and L x BO. The actual value of the constant UNUSD for a particular sort application can be found in the field with the symbolic label UNUSD.

Disk Output: The user routine for exit 7 should first test the constant CKPT—3 for AB bits. This is the first part of a 2-part test that is used to determine whether the block to be written is the last block. If CKPT—3 has AB bits, the program is processing the last group of records which, however, may not be the last block. Determine whether CYLN2 (4K object machine) or CYLN (8K, 12K, 16K object machine) is greater than, equal to, or less than the specified output blocking factor.

If CYLN2 or CYLN is equal to or less than this factor, the last output block will be written after the branch to ENTRY7. The last output block is always in the single block format and, if necessary, will be padded with blanks. CYLN2 and CYLN are both three characters in length and contain AB bits over the units position.

If it has been determined that the program is not processing the last block, the user routine should then determine the format of the output block. The type of disk output format and whether or not the program has finished processing the current block can be determined by the following tests:

1. If either SWCH2 or MDFY+4 contains a word mark, phase 6 has processed a single block, and the write-disk operation will be performed after the branch back to ENTRY7. Neither SWCH2 nor MDFY+4 will contain a word mark if the first output block is to be written. However, the first block written is always a single block.

2. If the 1-position constant NOTST contains a word mark, phase 6 is processing a large multiple block. If MVE+6 is equal to TOP1, the block will be written after the branch back to ENTRY7.

3. If the tests for a single block (SWCH2 or MDFY+4 contain a word mark) and a large multiple block (NOTST contains a word mark) fail, a short multiple block is being processed by phase 6. In this case compare MVE+6 to TOP2. If they are equal, a write-disk operation will be performed after the branch back to ENTRY7.

The type of disk output format should be determined each time the object program branches to

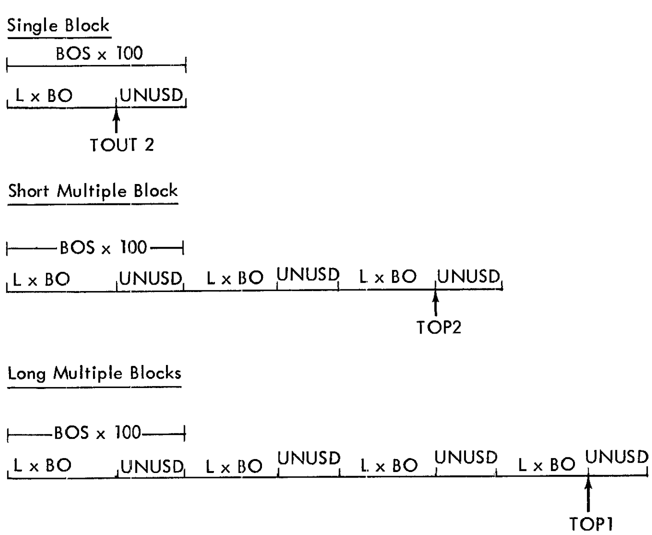

Figure 7. Output Formats

EXIT7 because the output format may change for each write-disk operation.

Tape Output: The single block format is always used. Thus MVE+6 always contains the address of TOUT2.

If CKPT—3 contains AB bits, the last block is written immediately after the branch to ENTRY7.

Exit 8

Sort Job Characteristic: Phase-6 output.

Availability: End of sort job.

Status: All records have been written on tape or disk.

Symbolic Addresses: Not applicable.

Use: Begin execution of another program.

Remarks: The routine cannot re-enter the sort object program because the Sort 6 program does not generate the label ENTRY8.

Exit 9

Sort Job Characteristic: Tape output with nonstandard labels.

Availability: Beginning of and during phase 6.

Status: Output files have not been opened.

Symbolic Address: CYLN is the address of a 1-position test field; 99 is the actual address of a 1-position field that contains a group-mark with word-mark.

Use: The routine must open the output file and check or bypass any labels on the tape and may write labels on the output tape. When end-of-file occurs or end-of-reel is reached, the routine must close the tape, and, if a new output tape is mounted, the routine must open this tape before branching back to phase-6 main line.

If the test field named CYLN contains a word mark, an end-of-file condition exists; otherwise, an end-of-reel condition exists.

Remarks:

1. Group marks with word marks that are set by the user's routine must be cleared before branching to the object-program main line.

2. The routine that closes the tape should test the field named CYLN to determine whether an end-of-reel or end-of-file condition exists. CYLN contains a word mark if the end of file has been reached.

3. At object time, the routine is written in the checkpoint area of the disk-storage work area. Thus, it does not occupy any core-storage positions during phase-6 main line.

Sort 6 Operating Procedures

This section contains a description of the Sort 6 program deck, a brief discussion of the Autocoder jobs that pertain to Sort 6, and the operating procedures to be followed when performing the job.

The user must be familiar with the following items in the *Autocoder Program Specifications and Operating Procedures:* Processor Jobs, Library Jobs, and Preparing a Stack of Jobs.

Program Deck Description

The Sort 6 deck contains a sort prephase, which must be transferred to the System file, and nine sets of Autocoder library routines.

Part 1 of the sort-prephase section is identified by SORT in columns 76-79; part 2 is identified by MOSHK in columns 76-80. The entire prephase section consists of approximately 70 cards. Each card contains a sequence number in columns 72-75; the cards are numbered sequentially beginning with 0001.

The library-routine section consists of approximately 14,000 cards, which contain a sequence number in columns 1-5, and identification in columns 76-80. The names and identification of the routine-sets are:

Name	*Identification*
SORT macro	62002
LODER macro	62L02
Phase 0 (assignment phase)	62A02
Phase 1	62102
Phase 2	62202
Phase 3	62302
Phase 4	62402
Phase 5	62502
Phase 6	62602

Preparing the Update Job

An update job transfers the sort prephase to its fixed location on the System file. In addition to the control and data cards that are supplied in the program deck, the card boot, SYSTEM ASGN, and HALT cards are required for the job. These cards are described in Figure 8.

Formats

Label Field (Columns 6-15)	Operation Field (Columns 16-20)	Operand Field (Columns 21-72)	Remarks
SYSTEM	ASGN	1301 UNIT 0 1311 UNIT n	n represents unit number and can be 0, 1, 2, 3, 4. The SYSTEM ASGN card immediately follows the card boot. The card boot, supplied with the Autocoder system deck, is required for all system operations.
LIBRARY	ASGN	1301 UNIT n, START nnnnnn, END nnnnnn 1311 UNIT n, START nnnnnn, END nnnnnn	n represents the number of the unit and can be 0, 1, 2, 3, or 4. nnnnnn is a disk address; it must be a multiple of 20.
AUTOCODER	RUN		
INITIALIZE	OPTN		
LIBRARY	OPTN		
	END		
	HALT		This must be the last card in a stack of jobs.

Note: Leave blanks in the operand field as indicated above.

Figure 8. Control-Card Formats

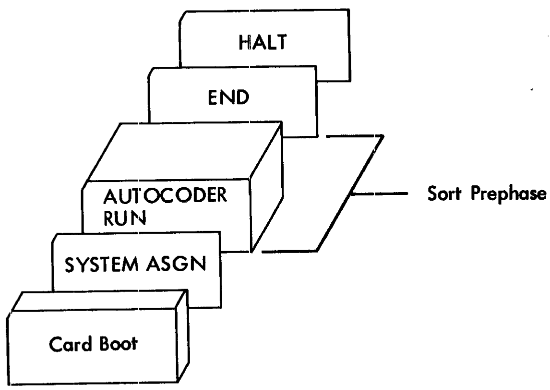

Figure 9. Update Job

Figure 9 shows the arrangement of cards for an update job. An AUTOCODER RUN card is the first card in the sort-prephase section of the program deck; an END card is the last.

Preparing Library Jobs

The SORT macro, LODER macro, and 1401, 1460 disk IOCS library routines must be in the same Autocoder library for each generation job. The set(s) of library routines for the phase(s) to be generated must also be in the library.

Perform a library change run to transfer the appropriate routines to an Autocoder library. The model statements that make up a library routine are stored in the Autocoder library as follows: The model statement is compressed by eliminating high-order blanks. The statements are stored as variable-length records in 2-sector blocks. The library table requires twelve sectors.

Figure 10 shows the names of the routines that make up each set of sort routines and the approximate number of statements in each set of routines.

Note: An exact number of statements is not given because the number may change when a new version or modification level of Sort 6 is issued.

Methods of Preparing the Library

If disk storage is available, the most efficient method of preparing the library is to define a new library and insert the required IOCS and sort routines.

If disk storage is at a minimum, delete routines from an existing library and then insert the required IOCS and sort routines. If the required IOCS routines are currently in the library, they need not be deleted and re-inserted.

NAME OF SET	ROUTINE NAMES	APPROXIMATE NUMBER OF STATEMENTS
Sort Macro	SORT	560
Loder Macro	LODER	90
Phase 0 (Assignment Phase)	PHSAA PHSAB PHSAD PHSAH	3260
Phase 1	PHSA8 PHSA9 PHSB0 PHSB1 PHS1A PHS1B PHS1C PHS1D PHS1E PHS1F PHS1G PHS1I PHS1J PHS1K PHS1M PHS1Z	2025
Phase 2	PHS2A PHS2B PHS2C PHS2D	1240
Phase 3	PHS3A PHS3B PHS3C PHS3D PHS3E PHS3F PHS3G PHS3H	1500
Phase 4	PHS4A PHS4B PHS4C PHS4D PHS4E PHS4F PHS4G PHS4H PHS4I PHS4J PHS4K PHS4L PHS4M PHS4N PHS4O	1690
Phase 5	PHS5A PHS5B PHS5C PHS5D	2000
Phase 6	PHS6A PHS6B PHS6C PHS6D PHS6E PHS6F PHS6G PHS6H	2450

Figure 10. Sort Library Routines

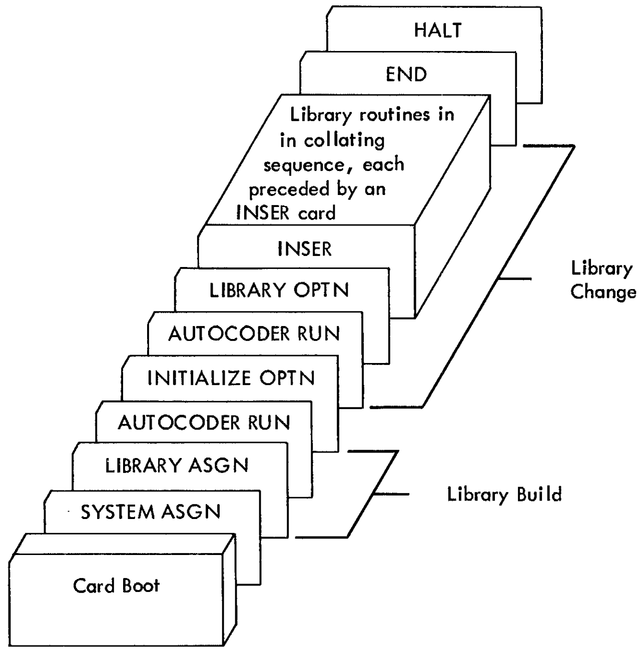

Figure 11. Preparing a New Library

New Library

The required IOCS and all sort routines will fit on a single system pack. In addition to the CARD BOOT, SYSTEM ASGN, and HALT cards, which are required for every stack of Autocoder jobs, the following control cards are required to build the new library and insert the routines:

Library Build:	AUTOCODER RUN INITIALIZE OPTN
Library Change:	AUTOCODER RUN LIBRARY OPTN INSER (an INSER card precedes each library routine in the program deck) END

The formats of the cards to build and change a library are shown in Figure 8. The arrangement of the input cards is shown in Figure 11.

Existing Library

The procedure for modifying an existing library depends on the size and current contents of the library and on the number of routines to be inserted. The user

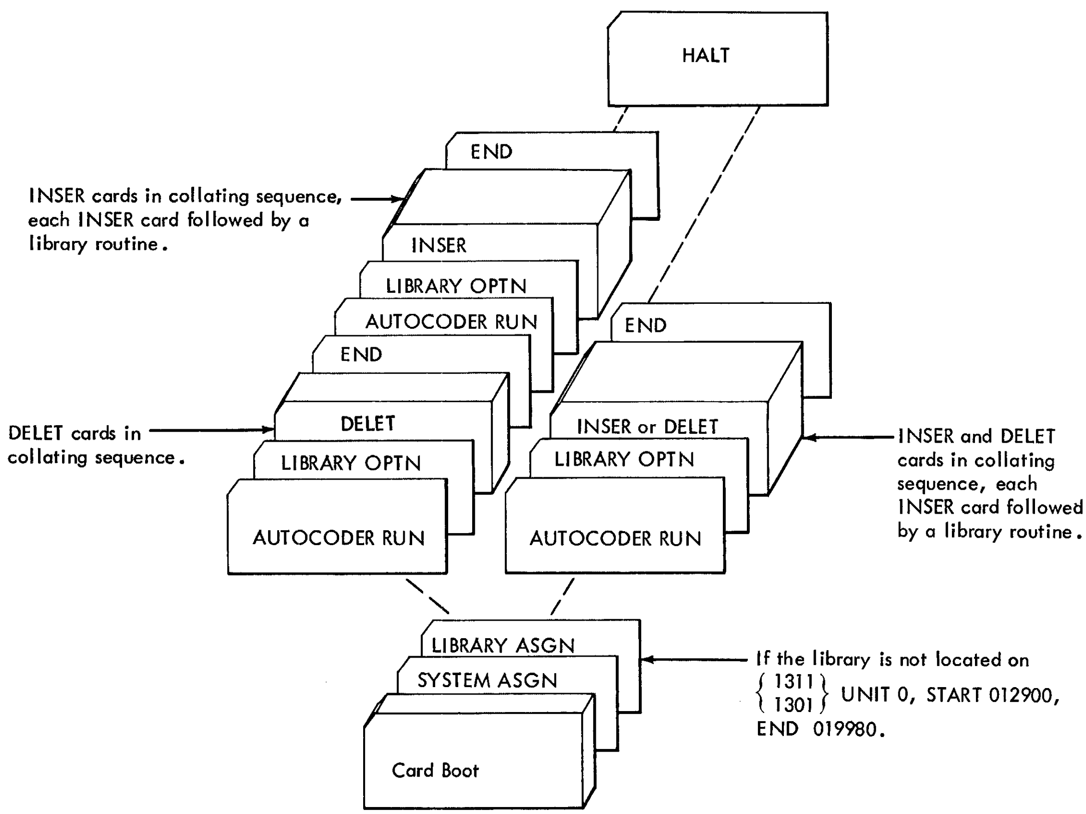

Figure 12. Methods of Modifying an Existing Library

can delete and insert routines during the same library-change job or he can perform two library-change jobs: one to delete routines and one to insert the required IOCS and sort routines. The second method is more practical. Because routines must be deleted and inserted in collating sequence, there is a possibility that a sufficient number of routines will not be deleted before routines are inserted. This can cause the library capacity to be exceeded. The arrangement of the cards for both methods of modifying the library is shown in Figure 12.

Preparing Generation Jobs

Use the following as a guide when preparing the job deck for a phase generation:

1. AUTOCODER RUN THRU OUTPUT card
2. JOB card (optional)
3. CTL card
4. SORT card
5. Parameter cards
6. Exit-0 or exit-9 routine, if required. The first card must contain LTORG* in columns 16-21.
7. Other user routines, if required.
   a. Phase-1 routines (exits 1-4) and phase-4 routines (exits 5 and 6) can be in any order. The first card must contain LTORG* in columns 16-21. The second card must contain ORG in columns 16-18 and an actual address starting in column 21.
   b. Phase-6 routines (exits 7 and 8) must be in ascending numerical order. The first card in each routine must contain LTORG* in columns 16-21. The second card in each routine must contain ORG in columns 16-18 and an actual address starting in column 21.
8. END card.

The card boot and SYSTEM ASGN cards must be the first cards in the stack; a HALT card must be the last. The LIBRARY-ASGN card must follow the SYSTEM ASGN card if the library is not on

$\left\{\begin{matrix}1311\\1301\end{matrix}\right\}$ UNIT 0, START 012900, END 019980.

See Figure 8. The formats of the cards for the generation-job decks are given under *Requirements for Generation.* Figures 13 and 14 show sample job decks for the generation of phases 1 and 6, respectively. Figure 15 shows a stack of generation jobs.

Performing Jobs

To prepare the stack, arrange the input cards in this order:

1. Card boot
2. SYSTEM ASGN card
3. Any required ASGN cards and job deck(s). The job decks can be in any order.
4. HALT card.

To perform a stack run when the Autocoder system resides on 1311:

1. Place the system pack on the disk drive referred to in the SYSTEM ASGN control card, and ready the drive.
2. Ready all required machines (disk units, reader, punch, printer).
3. Ready the console:
   a. Set the I/O check-stop switch off.
   b. Set the check-stop switch and disk-write switch on.
   c. Set the mode switch to RUN.
   d. Press CHECK RESET and START RESET.
4. Load the program.
   a. IBM 1402 Card Read-Punch: Press LOAD.
   b. IBM 1442 Card Reader: Press START on the reader, and PROGRAM LOAD on the console.
5. When the system attempts to read the last card:
   a. IBM 1402 Card Read-Punch: Press START.

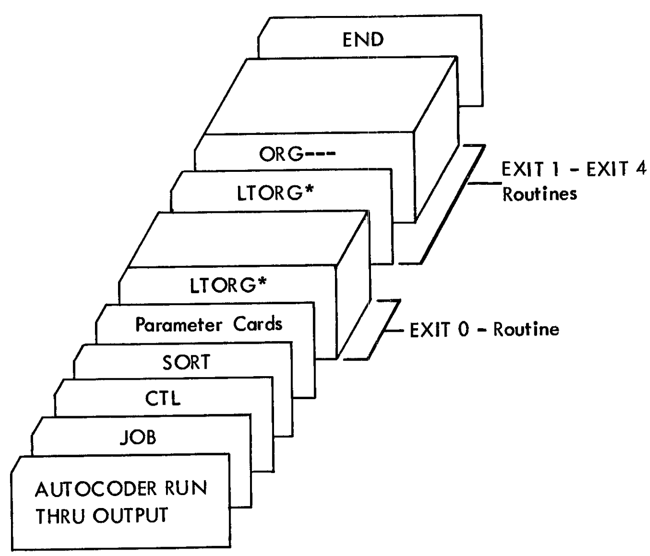

Figure 13. Sample Job Deck for Phase-1 Generation

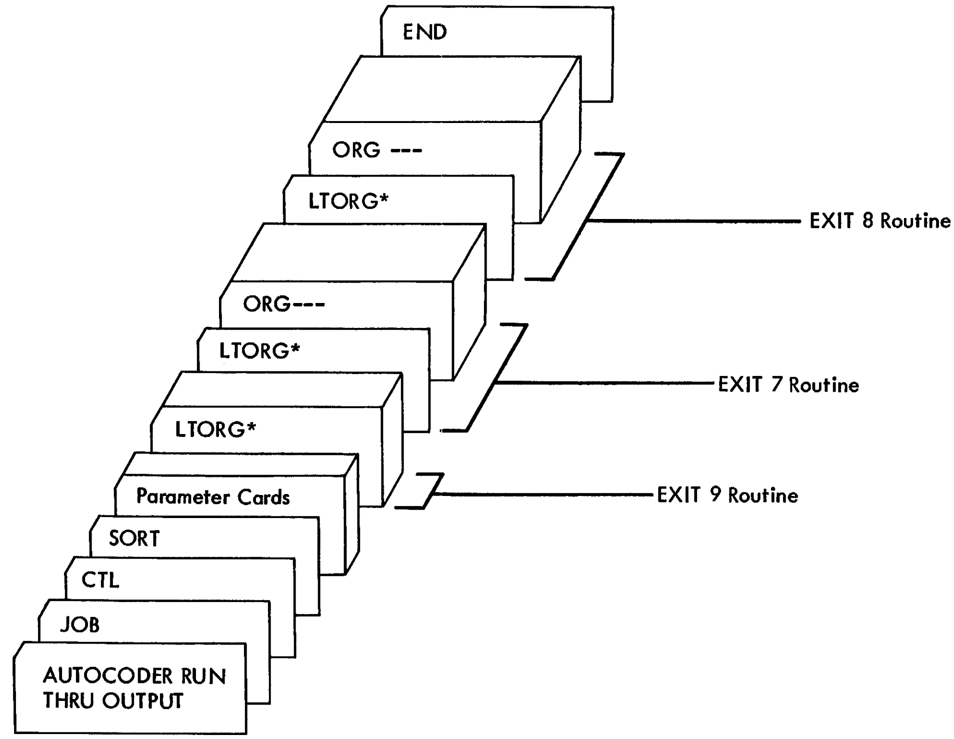

Figure 14. Sample Job Deck for Phase-6 Generation

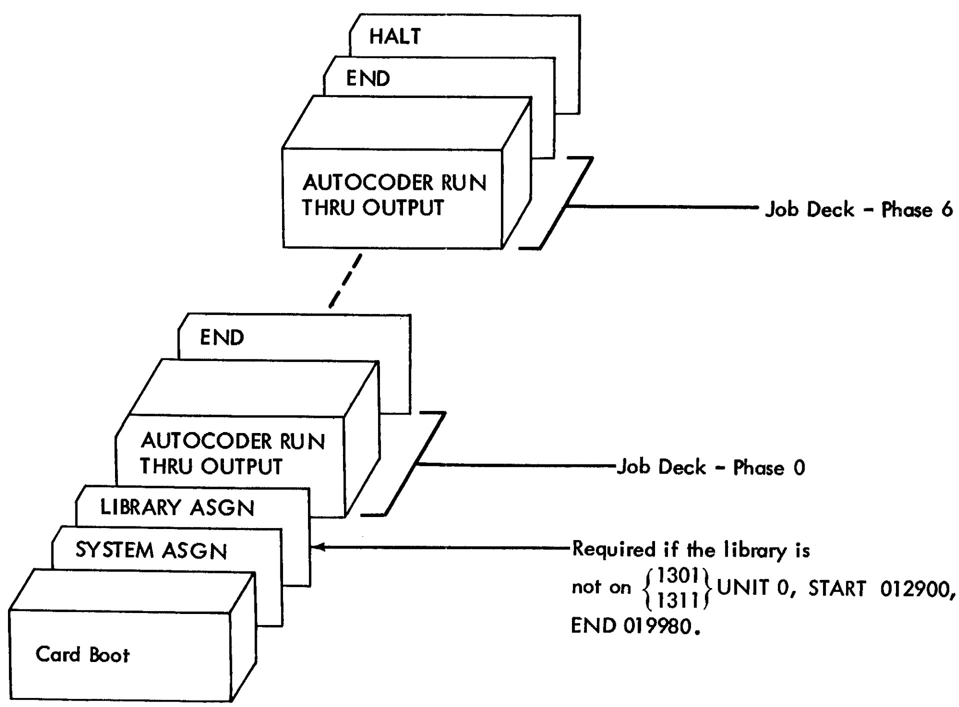

Figure 15. Stack of Generation Jobs

b. IBM 1442 Card Reader: Press START on the card reader.

To perform a stack run when the Autocoder system resides on 1301:

Follow steps 2-5 of the instruction for a system that resides on 1311.

Results of a Generation Job

The results of each generation job are:

1. CTL card diagnostics, if the card is invalid.
2. Source statement diagnostics, if specified in the CTL card.
3. Label table or a cross-reference listing if specified in the CTL card.
4. Program listing.
5. An object deck in the condensed loader format (object program statements preceded by a 7-card loader: two clear cards, a bootstrap card and four load cards).

These are the results of an AUTOCODER RUN THRU OUTPUT.

Prephase Listing Comments

During the sort-prephase routine, the parameter cards are analyzed. If no errors are detected, the parameter information is converted into an acceptable macro-generator input format. The generation of the phase then proceeds automatically.

If an error is detected, the contents of the invalid parameter card are printed. A comment that defines the error is printed at the right of the card image.

After all invalid parameters have been printed, the program halts. The user must correct the card(s) and restart the generation job.

The error comments that can appear on the listing are as follows:

LABEL ERROR

This comment is printed if the card contains a misspelled label.

LABEL ALREADY ENCOUNTERED

This comment is printed if two cards contain the same label.

OPERAND ERROR

This comment is printed if the card contains a misspelled operand. The sort prephase checks only the number of positions required for each specified parameter. Assume for example, that the TAPE parameter card specifies UNLOADIN, SCAN. The prephase checks eight positions for UNLOADIN and four for SCAN.

THE PARAMETER CARD LABELLED XXXXXXXXXX HAS BEEN OMITTED, RESTART*

This comment is printed at the end of the listing if a required parameter card has been omitted. The label of the missing card appears in the message.

THE PARAMETER CARD LABELLED XXXXXXXXXX IS IN ERROR, RESTART*

This comment is printed if the tape parameter cards contain incorrect specifications.

Halts and Messages

Refer to the *Autocoder Program Specifications and Operating Procedures* for a list of the halts and messages that can occur during a stack run.

Sort Object Program Specifications

Sort object programs are generalized programs that have been generated by the Sort 6 program. Four sort object programs: Sorts 61, 62, 63, and 64 are available from IBM.

All sort object programs can sort blocked or unblocked fixed-length records in ascending or descending standard 1401, 1460 collating sequence. Records are sorted on the basis of control-data fields within the records. As output, the user can obtain sorted addresses, sorted control data, or sorted records.

Other capabilities of sort object programs depend on the parameters used for the Sort 6 generation. An object program becomes a specialized sort program by means of control and RDLIN cards, which supply the program with the specific characteristics of the sort application.

IBM-Supplied Sort Object Programs

Sort 61, 62, 63, and 64 are object programs that have been generated by the Sort 6 program. Sorts 61 and 62 use the normal-seek operation to read and write disk records; Sorts 63 and 64 use the direct-seek operation.

The parameters used to generate Sorts 61, 62, 63, and 64 and the size of the systems on which they are to be used are:

Name of Program	*Machine Size*		*Parameters*
Sort 61	4K	INPUTMEDIA	DISK,CARD,TAPE
Sort 62	8K	OUTPTMEDIA	DISK,TAPE
		SELECTDLET	SELDELCLASS,CONTROLDATA
		ADDITIONS	CARD
		CHECKS	SEQHASH
		DSKLABLOUT	ALL
		TAPE	UNLOADIN,UNLOADOUT,SCAN
		TAPLABLIN	STANDARD,ALL,B
		TAPLABLOUT	STANDARD,ALL,B
Sort 63	4K	Same as Sorts 61 and 62	
Sort 64	8K	SYSTEMSPEC	DIRECT

Users of IBM-supplied sort object programs should refer to the *Sort 6 Specifications: Parameter Cards* section of this publication for a discussion of the routines specified by the parameters.

Note: The IBM-supplied programs do not provide linkage to user-routine or routines that use the console printer.

User-Generated Sort Object Programs

The routines incorporated in user-generated programs depend on the parameters supplied at generation time. The advantages of a user-generated program are: user-routines can be included, and a console printer without a buffer feature can be used.

Also, the sort time can be reduced if:

1. The INPUTMEDIA and OUTPTMEDIA cards specified DISK only.
2. The SYSTEMSPEC DIRECT, if applicable, was the only optional parameter card used for generation of the sort program.

When the INPUTMEDIA card specifies TAPE, all optional phase-1 routines are included in the object program. User routines also increase sort time.

Machine Requirements

The 1401 system on which the object program is to be run must have at least:

- 4,000 positions of core storage.
- One IBM 1311 Disk Storage Drive, Model 4.
- An IBM 1403 Printer, Model 1, or an IBM 1404 Printer, or a Console Printer.
- An IBM 1402 Card Read-Punch.
- High-Low-Equal Compare special feature.

The 1460 system on which the object program is to be run must have at least:

- 8,000 positions of core storage.
- One IBM 1311 Disk Storage Drive, Model 1.
- An IBM 1403 Printer, Model 2, or a Console Printer.
- An IBM 1402 Card Read-Punch.

If the object machine is equipped with the Direct Seek special feature, the user can specify that this feature be used. If the system on which the object program is to be run has a buffered console printer, this printer cannot be used.

Related Information

The user should be familiar with the SRL publication, *Input/Output Control System (on Disk) Specifications for IBM 1401 and 1460,* Form C24-1489, because the 1401, 1460 disk IOCS routines control the input to and output from sort object programs.

The collating sequence used by the sort object program is given in the *System Operation Reference Manual for IBM 1401 and 1460,* Form A24-3067.

Record Requirements

Allowable Record Configurations

All sort object programs accommodate fixed-length records, blocked or unblocked, in the single sector move mode. Input records can be from cards (unblocked only), disk, or tape. The minimum number of records must be at least the number of records in a block, plus one. If the records are unblocked, there must be at least two records.

Record Length

The minimum record length for records to be sorted is:

1. Five characters for disk input and output.
2. Ten characters for blocked tape input and output.
3. Thirteen characters for unblocked tape input and output.
4. Five characters for card input.

The maximum record length depends on the core-storage capacity of the object machine:

Core Capacity	*Record Length*
4K	400 characters
8K	1,200 characters
12K	2,000 characters
16K	3,000 characters

The record length must be the same for all records to be sorted because the object program can accommodate only fixed-length records.

Blocking

Each block of tape or disk records must contain the same number of records. Blocking does not apply to card records.

Block lengths have the same limitations as record lengths. To determine the maximum blocking factor, divide the maximum block length by the record length and round the result low. The maximum allowable blocking factor is 100.

If the storage media is disk, the optimum blocking factor is the number of records that fills or most nearly fills the least number of sectors.

Example: Assume that the record length is 80 characters. If the blocking factor is three, three sectors can be processed as a unit. The three sectors contain 240 data characters and 60 unused positions. If the blocking factor is five, four sectors can be processed as a unit. The four sectors contain 400 data characters. Therefore, the optimum blocking factor is five.

If the storage media is tape, the optimum blocking factor is the maximum blocking factor. The start-stop time in the tape units is reduced when large blocks are processed.

Using the optimum blocking factor reduces the time required for execution of phase 1 because the object program reads all records during this phase and forms the tags (control data, 6-character disk address, 2-character block location factor), which are to be sorted during phases 2, 3, and 4.

The number of blocks that are read into core storage during phase 1 depends on the amount of core storage available for the input area. The capacity of the object machine and the positions required by user routines are determining factors.

Card Input and Additions

The following discussion applies to input and additions cards.

A single card can contain only one record. A record can consist of as many cards as necessary.

No card can contain ENDCD in columns 6-10.

If the user wants the object program to check the records, he must not punch any data characters in columns 1-10 (70-character format). If no checking is to be performed, control data and/or record data can be punched in columns 1-80 (80-character format).

The format of records to be checked is:

Columns	*Contents*
1-8	Identification number. This number must be unique for each record.
9-10	Sequence number (01-43). The cards that make up a record must be numbered consecutively.
11-80	Data

The object program examines the identification and sequence numbers to ensure that the correct record is read in the proper sequence. If a record contains an incorrect number of cards, an incorrect identification number, or is out of sequence, an error message is printed and a halt occurs. The error-message indicates the identification and sequence number of the erroneous card.

Restrictions

Records to be sorted must not contain:

1. A group mark in any position.
2. A record mark in any position except the last. If the last position is part of a control data field, it cannot contain a record mark.
3. A tape mark in the first position if tape input or output is specified in control card 1.

Labeling

Disk input/output files can be either labeled or unlabeled. Tape files can contain either standard or nonstandard labels.

The object program uses RDLIN card information and 1401, 1460 disk IOCS routines to process disk labels and standard tape labels.

If disk files are unlabeled the user must supply RDLIN cards that specify the drive number and limits of the files.

Tape input records with type A (120-character) labels must contain a tape mark after each header label. Input records with type B (80-character) labels or type C (84-character) labels must not contain tape marks after the labels unless the user has specified TM in the TAPLABLIN parameter card. If tape files contain nonstandard labels, the user must supply routines, at generation time, to process or bypass the labels.

For information regarding label formats, refer to the *1401, 1460 Disk IOCS Specifications* publication. Also, see *Figure 26: Schematics of Standard Disk and Tape Header Labels.* For a discussion of label processing, see *Object Program Routines: Labeling.*

Note: If the input file is contained in more than one disk area, only the last area can contain a 1EOFb trailer label. When the program encounters a 1EOFb trailer label, it stops reading the defined input area(s).

Control-Data Fields

The object program sorts on as many as ten control data fields. These fields control the sequencing of records and can be located anywhere in the record as long as they do not overlap. The maximum total length of this control data is 189 characters.

The location of each control-data field must be specified in control card 2. If more than one control data field is used, the user must specify which field is to be compared first, which second, and so forth. The control-data fields should be referred to in the order of their significance with the most significant one designated as control field 1, etc.

Processing time is dependent upon the size and number of the control fields. Therefore, whenever possible and to the user's advantage, these two factors should be limited to save time.

Object Program Features

This section describes the capabilities of optional routines associated with sort object programs. Some of the optional routines are incorporated in every object program, others are incorporated only if the parameters used for generation requested their inclusion. With the exception of user routines, the optional routines are selected at object time only if requested in control card 1. If a user routine is associated with an optional routine, it is selected only if the optional routine is requested; otherwise, it is always selected. See *Sort 6 Specifications: User Programming.*

Input/Output Routines

The object program can contain card, tape, and/or disk input routines and tape and/or disk output routines. The kinds of routines to be included must have been specified in the INPUTMEDIA and OUTPTMEDIA parameter cards. See *Object Program Description* for a synopsis of the input/output routines.

Card Additions

If the input is from disk, card records can be added to the file. The ADDITIONS card with CARD specified must have been used for generation. The routine is selected at object time if the user requests it in control card 1.

Labeling

Disk labels and standard tape labels are processed by 1401, 1460 disk IOCS routines. For the contents of these labels, refer to the 1401, 1460 disk IOCS publication. Also, see *Figure 26 and 27: Schematics of Standard Disk and Tape Header and Trailer Labels.* At object time the user must supply RDLIN cards that contain label information and file limits. Several tracks in the sort work area are reserved for label processing.

Disk Labels

If the labels on disk files are to be checked, the user must specify the kind of checking in control card 1. Labels on the work area and work cylinder will be checked initially as output areas. However, on a restart run, these areas will be checked as input areas. If output disk files are to be labeled, the DSKLABLOUT parameter card must have been used at generation time.

The IOCS routines process disk labels in the following manner:

1. For disk input:
   a. Compares all fields in the header labels with the contents of the input RDLIN card(s) if full checking is specified (control card 1), or
   b. Compares the file name in the header label with the file name in the input RDLIN card if partial checking is specified (control card 1).

2. For disk output:
   a. Checks existing labels on the output file to ensure that the limits specified in the RDLIN card(s) for the new output file(s) are such that no unexpired file(s) will be destroyed.
   b. Generates a header label according to RDLIN and date-card information and writes the label on the first available sector of the label track in the first output area; repeats this procedure for each additional output area.
   c. Writes a 1EORb trailer label as the first and only record in the last output block in each output area, except the last; writes a 1EOFb trailer label after the last record in the last output area. The length of the trailer label is the same as the length of the records in the file. The 1EORb or 1EOFb is located in the first five positions of the trailer label. The user should not expect the balance of the record containing 1EORb or 1EOFb to be cleared.

Tape Labels

If input tape labels are to be processed, the TAPLABLIN parameter card must have been used at generation time. If labels are to be written on the output tape, the TAPLABLOUT parameter card must have been used.

If nonstandard labels (input and/or output) are to be used, routines to process or bypass the labels must have been assembled at generation time. See *Sort 6 Specifications: EXIT0 and EXIT9.*

The 1401, 1460 disk IOCS routines can process three types of standard tape labels:

1. Type A—120-character label
2. Type B— 80-character label
3. Type C— 84-character label

The standard labels are processed in the following manner:

1. For input tape labels:
   a. Compares the file serial number, reel sequence number, file name, and creation date in the header labels with the corresponding fields in the input area RDLIN cards if ALL was specified in the TAPLABLIN card, or
   b. Compares the file name in the header labels with the file names specified in the input area RDLIN cards if IDENT was specified.
2. For output tape labels:
   a. Checks existing labels on the output file to determine whether the retention period has expired.
   b. Generates an output header label according to RDLIN and date card information and writes the label on the output tape.
   c. Writes a tape mark, 1EORb trailer label and a tape mark after the last record on all output tapes except on the last reel when a tape mark, a 1EOFb trailer label, and a tape mark are written.

If the number of tape output records is not a multiple of the user's specified output block size, the program pads the last block of records. For example, the user might specify an output blocking factor (BO) of five records. If the last block of output contains only two actual records, the program pads the last block with three dummy records composed of all nines (if ascending sequence has been specified) or all blanks (if descending sequence has been specified).

The format of the trailer label will be the same as that of the header label.

Note: Unlabeled input and output disk files can be processed. RDLIN cards that specify the drive number and file limits must be inserted in the object deck.

Selection or Deletion Before Sorting

Records can be either selected or deleted before the sorting process begins. A record is selected or deleted if a specified character is present in a specified position. Control card 1 supplies the object program with the choice (either selection or deletion), the character, and the position. The SELECTDLET parameter card with SELDELCLASS specified must have been used at generation time.

Deletions by Control Data

Records can be deleted, in phase 4, on the basis of their control data. This deletion routine is in the object program if the SELECTDLET card with CONTROLDATA specified was used at generation time. The routine is selected at object time if deletions are specified in control card 1. Records to be deleted can contain a maximum of 80 control-data characters. The user must punch the control data of the records to be deleted into cards. The control data must begin in column 1 and be punched in the order that is specified in control card 2. Columns 6-10 of these cards cannot contain ENDCD.

At object time, the deletion cards must be inserted in the phase-4 section of the object deck. See Sort *Object Program Operating Procedures*.

Checking

The object program always accumulates a record count during phases 1, 2, 4, 5, and 6. At the end of phases 2, 4, and 5, the count is compared with the count accumulated at the end of the previous phase. If the compare is unequal, the program halts.

If the CHECKS card with SEQHASH specified was used for generation, a hash total can be accumulated during phases 2, 3, and 4 and a sequence check can be performed in phase 4.

The program adds the contents of certain positions in the control fields in each record to accumulate a hash total. The totals are compared to ensure that no characters have been lost and that the same number of records has been processed in each phase. The sequence check is performed to ensure that the records are in collating sequence.

The SEQHASH routine is selected at object time if the user specifies it in control card 1.

User Programming

Routines to process non-standard labels are incorporated in the object program if TAPLABLIN and/or TAPLABLOUT parameter cards that specified NONSTANDARD were used at generation time. Other user routines are incorporated if the EXITS and USERAREA parameter cards were used.

Record Marks

All sort object programs include a routine that can write a record mark after each output record. If record marks are requested in control card 1, the record length must be one position less than the maximum record length. If the output is to be written on disk, the user must be sure that the number of sectors allotted for each output block (BOS) can contain the original records plus the record marks. Assume, for example, that a user specified 100-character records, a blocking factor of five, and five sectors for each block. If he requested record marks, the allotted area would be exceeded.

Deletion of Characters from Records

All sort object programs have the ability to delete a specified number of characters from the low-order end of each record before it is written out as an output record. This deletion is performed during phase 6. The maximum number of characters that can be deleted from disk output records is 2,996 if record marks are requested; otherwise, the maximum number is 2,295. The record length, after the characters have been deleted, must not be less than the minimum record length required by the program.

The routine is selected if control card 1 specifies the number of characters to be deleted.

Checkpoint and Restart

The checkpoint routine, incorporated in every sort object program, writes checkpoint records at the end of phases 1, 2, and 4; after all phase-3 merge passes; and at various points during phases 5 and 6.

If it is necessary to stop processing because of an error or a lack of machine time, processing can be resumed at a point close to the point where the processing stopped.

Object-Program Description

This section describes the functions of the phases that make up the sort object program.

Assignment Phase

The assignment phase reads and analyzes the two control cards and checks them for logical errors. If a logical error is detected, a message indicating the nature of the error is printed. For example, messages occur if:

1. The sum of the lengths of the individual control-data fields does not equal the total specified in control card 2.

2. The control-data fields specified in control card 2 overlap.

The assignment phase reads disk RDLIN cards and performs disk label checking on the areas specified by the RDLIN cards. It also computes and stores constants used in subsequent phases of the program.

Phase 1

The process that takes place during phase 1 depends on the input medium.

Input from Cards

During phase 1, data records are read from cards into core storage and blocked, assigned 6-character disk addresses, and written in disk storage. An 8-character address (a 6-character disk address and a 2-character block location factor) is formed for each record. All the control-data fields are then extracted from each record in the order of their significance. (The order is specified in control card 2.) From the control-data fields and the 8-character address, a tag is formed for each record.

Input from Disk Storage

During phase 1, data records previously stored in disk storage (in at least one, but not more than four input areas) are read into core storage. The functions of extracting the control data, adding to it the 8-character address, and forming the tags operate in the same manner as explained under *Input from Cards*.

Input from Tape

For tape input to the program, the data records are read into core storage, reblocked, and assigned disk addresses. The functions of extracting the control data, adding to it the 8-character address, and forming the tags operate in the same manner as explained under *Input from Cards.*

Deletions or Selections and Additions

During phase 1, deletion or selection by class and additions can also be performed. Deletions by class permits the user to bypass records that have one particular character in a specific location in the record. With this feature, for example, all blank records can be bypassed.

Sorting by selection allows the user to sort only one class of records. This permits the user to select all those records that do have the particular character in a specific location in the record.

All the input records are written on the disk input area. The tags (control data and 8-character address) are selected or deleted. Phase 6 seeks and writes only those records whose tags were sorted.

By using the additions feature, it is possible to incorporate (into the sort operation) records that are not part of the main source of input. Only when the main source of input is disk storage can additional input from cards be processed.

Phase 2

During phase 2, an internal sort is performed on the number of tags that fit into a work area in core storage.

Phase 3

During phase 3, a series of merge passes is performed on the sequenced tags developed in phase 2.

Phase 4

During phase 4, the last merge pass of the sort is performed. At this time, the sequenced strings of tags created during the previous phases are merged into one string, and the control data is dropped from the tags.

The input to phase 5 is one string of 10-character addresses. These 10-character addresses are made up of 8-character addresses (a 6-character disk address and a 2-character block location factor) and a 2-character adjustment factor that indicates the location of the 8-character address in a block of input that will be read into core storage in phase 5.

If specified, deletions by control data can also be performed during phase 4. This is accomplished by reading pre-sorted cards containing control data into core storage. These cards designate which records of the file being sorted are to be deleted. The control data in these cards can be a maximum of 80 characters in length.

The user has the option of ending the program at the end of phase 4. If this option is specified, one of the following types of output can be specified:

1. Sorted 8-character addresses (a 6-character disk address and a 2-character block location factor).
2. Sorted control-data field(s).

Phase-4 output will be located in the work area and a message giving the starting address of the output will be printed. The characteristics of the sort job determine the starting address. However, the user can request that the output be relocated so that it will start at the beginning of the work area.

Phase 5

The input to phase 5 is one string of 10-character addresses. The addresses are in sequence according to their control data. Assume, for example, that the first 10-character address is 0090390501. The 01 indicates that the first record to be written in phase 6 is the sixth record in a block of records that has the disk address 009039. The second 10-character address would end in 02, the third in 03, and so on.

Phase 5 sorts C addresses under control of the 6-character disk addresses. The value of C, which is calculated by the sort object program, is printed at the end of the assignment phase.

By sorting according to the 6-character addresses, the sort object program is able to group the addresses of record blocks that are located on the same cylinder. This minimizes seek time in phase 6.

The blocks of C addresses are written in the disk-storage work area. If phase-5 output is specified, a message stating the starting address of the output is printed.

Phase 6

The input to phase 6 consists of blocks of C 10-character addresses that have been sorted on the basis of the 6-character portion of the address. (C is the phase-5 blocking factor; the value of C is printed in an assignment phase message.)

The method used for seeking and writing sorted records depends on the specific characteristics of the sort job. If the work cylinder is used, phase 6 sets up a table area in core storage and proceeds as follows:

1. Reads 10-character addresses from the disk work area into a core-storage work area.
2. Seeks and reads records from the disk input area into a core-storage input area.
3. Checks the block location factor in the address and moves the records to a hold portion of the core-storage work area.
4. Repeats steps 1 through 3 until the hold area is filled, and then:
   a. Each record is written on the work cylinder, and
   b. An entry is made in the table to indicate the location of each record on the work cylinder. The entries are arranged according to the 2-character adjustment factors, which indicate the relative positions of the records in phase-5 input block.
5. Repeats steps 2-4 until the blocks within the phase-5 block have been read into core storage and processed.
6. Uses the table to seek and read records (in collating sequence) from the work cylinder into core storage.
7. Writes the records in the disk output area or on tape.
8. Repeats steps 1-7 until all sort records have been written on disk or tape.

Under certain conditions, such as no tape output and no phase-6 exits have been generated, the object program may not use the work cylinder. The object program procedure is the same as steps 1-3 when the work cylinder is used, except that the selected records are not moved to a hold area. Instead, they are moved to their appropriate positions in the core-storage output area. For example, if the first record selected had a 2-character adjustment factor of 06, the record would be moved to the sixth record position in the core-storage output area. When the output area, which can hold C records, is filled, the records are written in the disk output area. The procedure is repeated until all records are written on disk.

Output Available

The user must specify, in control card 1, the kind of output desired. The kinds of output available are:

1. Sorted 8-character addresses (phase-4 halt)
2. Sorted control data (phase-4 halt)
3. Sorted 10-character addresses (phase-5 halt)
4. Sorted records (phase-6 halt)

Phase-4 Output

Either 8-character addresses or control data can be obtained when the program halts at the end of phase 4. The output will be written in the work area, and a message giving the starting address of the output will be printed.

Note: The output may not start at the beginning of the work area unless the user punches a 1 in column 54 of control card 1.

The 8-character addresses consist of 6-character disk addresses, each followed by a 2-character block indicator. The block indicator is the location of the record within the block addressed by the 6-character address. The 8-character addresses are written out in one continuous sequence. Each block of 200 characters contains 25 addresses.

The size of control data output blocks depends on the size of the object machine:

Object Machine Size	*Block Size*
4K (SEQHASH specified)	200-character
4K (SEQHASH not specified)	600-character
8K	1,000-character
12K	1,800-character
16K	3,000-character

The control data from as many records as possible will be blocked and written in the disk-storage work area. The remaining positions of each block will be blank. A 1EOFb trailer label will be written at the end of the output file.

The maximum number of records in a phase-4 output block is 399. If the object machine size is 16K and the total length of the control data fields is five, six, or seven characters, the number of records in a block will be limited by the maximum blocking factor (399) rather than by the maximum block size (3,000). Because this results in some unused disk area, the user may desire to reduce the machine size specified in control card 1 to make more efficient use of disk storage.

Phase-5 Output

A halt at the end of phase 5 results in blocks of C records, which consist of 10-character addresses that have been sorted according to the 6-character disk address. C is an internal sort blocking factor; the value of C is printed in an assignment phase message. The last two characters in the 10-character address indicate the relative position of the disk address and the block location factor within a phase-5 input block. If, for example, a 10-character address was 0190390508, the record indicated would be the sixth record in the block of records at address 019039. The 08 in the address 0190390508 indicates that the 8-character address is the eighth address in a phase-5 input block. The 2-position field that indicates the relative position of the disk address and the block location factor can range from 01-499.

Numbers greater than 99 are coded by placing bits over the tens and units positions of the indicator. Numbers from 100 to 199 contain an A-bit over the tens position, numbers from 200 to 299 contain a B-bit over the tens position, numbers from 300 to 399 contain an A-bit and a B-bit over the tens position, and numbers from 400 to 499 contain an A-bit over the units position. Thus, if the position indicator of a particular record is D9, it is record number 349 in the specified block of input to phase 5.

If the total number of 10-character addresses is not a multiple of the phase-5 input blocking factor, a 1EOFb trailer label follows the last address in the last block.

Phase-6 Output

Phase-6 output is written on tape or in the disk output area defined by the user's RDLIN cards. The records are blocked as specified in control card 1. A record mark may be added to each record if the user selected the option in control card 1. Characters may have been deleted from the low-order position of each record if the user requested the deletion.

Requirements for Execution

The user supplies the sort object program with the following information:

1. Limits of the disk areas to be used (RDLIN cards)
2. Label information, if desired (RDLIN cards and a date card)
3. Specific characteristics of the sort application (two control cards).

File Area Definition

To eliminate the possibility of disk-storage capacity problems or disk-storage location problems arising during execution of the sort object program, the user must know the size and location of certain required storage areas. These areas must then be reserved on disk for specific functions of the sort program.

The program always requires a defined work area, and a defined input area. If card records are to be added to disk records, at least two input areas must be defined (one for the card records to be added and one for the disk records). If the input medium is 1311 disk storage, the records may be stored in as many as four non-adjacent areas. The user defines these four input areas by specifying their high and low limits in RDLIN cards (see *Input RDLIN Cards: INP1, INP2, INP3, and INP4*).

If phase-6 output is desired, a work cylinder must be defined.

The location of each area required by the program is specified in RDLIN cards. Use the formulas discussed in this section to calculate the length of each area. After the length of each area has been calculated, decide the location of the areas and prepare RDLIN cards that specify the sector addresses of the areas. The RDLIN cards must be inserted in the object deck. See *RDLIN Cards* and *Preparing the Object Deck.*

There are no limitations governing the relative positions of the defined areas. However, the input area(s) must not overlap each other or any other area(s); the work area and the output area may overlap (see *Overlapping Work and Output Areas*). Restrictions that relate to the addresses specified in RDLIN cards are listed under *RDLIN Cards.*

The user should read the section *Optimum Placement of Disk-Storage Areas* before selecting the disk area that will contain the input area. If a given sort program is to be used repeatedly, the same RDLIN cards can be used for each execution of the program. However, recalculations must be performed each time the characteristics (file size, record length, disk availability, etc.) of the program change, provided the change is such that it will cause areas specified in the RDLIN cards to be incorrect.

A number of conditions have been taken into consideration in the compilation of the formulas given in this section. For example, the input blocking factor and block size may differ from the output blocking factor and block size. The formulas are designed to permit variables of this kind. The results obtained will provide the user with maximum utilization of storage as well as maximum efficiency of record movement.

In the following discussions, the variable information to be substituted into the formulas is equated with the symbol that represents it. The brackets $\lfloor \; \rfloor$ mean rounded to the next lowest integer; the brackets $\lceil \; \rceil$ mean rounded to the next highest integer.

The variable values inserted by the user must be exact, otherwise the results will be inaccurate. Charts of constants that the user must insert in the formulas for the calculation of certain areas are given in the following section.

Input Area

If the input is to be from cards or tape (these cannot be mixed) a disk input area must be defined. The limits of the area must be specified in an input area RDLIN card.

Card and tape input records (or card additions to disk input) must be converted to disk records before they can be processed by the program. The records are read from tape or cards into core storage. Phase 1 forms the required 8-character disk address portion of the tags and adds the control-data information from the records while they are in core storage. The records are then written in the disk input area, and the tags are written in the work area.

If the input is to be from disk, the user must know the location of his file, and must use RDLIN cards to specify the drive number and limits.

The size of the area required for card or tape input can be calculated with the following formula:

Basic-Input-Area Equation

$$NS = \left\lceil \frac{N+1}{\left\lfloor \frac{100V}{L} \right\rfloor} \right\rceil \times V$$

where

NS = Size of the input area in sectors.
N = Number of input records.
L = Input record length.
V = Number of sectors that can be written out by phase 1 at one time.

The value of NS must be calculated for each card or tape file size used.

Note: See *Card Additions.*

Before this basic-input-area equation can be solved, the user must calculate the value to be substituted for V. Card input causes V to have a specific value; tape input causes it to have a different value.

Because the core-storage capacity of the machine being used to run the program has a limitation (4K, 8K, 12K, or 16K), there is naturally a limitation on the amount of space available for processing records. This limitation is determined by such factors as machine size and program size. As a result, the maximum value of V (Vmax) must be taken into consideration when the actual value of V is calculated. Use the following formula to determine Vmax:

$$Vmax = \left\lfloor \frac{MS - Bmax - UA1 - PS1 - 23}{100} \right\rfloor$$

where

MS = Core-storage capacity of the object machine.
Bmax = Maximum sort block size:
400 for a 4K machine.
800 for an 8K machine.
1,400 for a 12K machine.
1,900 for a 16K machine.
UA1 = Number of core-storage positions required for the phase-1 user area (see *User Programming*).
PS1 = Phase-1 program size. This depends on certain factors specified in control card 1. See Figure 16 for approximate values of PS1. The values given should be used for planning purposes only. The actual PS1 values can be found in the next-to-last card generated by phase-1 as follows

Columns

11-13 If tape input and deletions or selection by class are not specified.
15-17 If card input and deletions or selection by class are not specified.
19-21 If the additions option is specified.
23-25 If card or tape input and deletion or selection by class are specified.

The values found in these columns are in actual machine language.

By subtracting Bmax, UA1, PS1, and 23 (number of positions required for programming incidentals) from MS, the number of core positions available for the phase-1 output is determined. When this number is divided by 100 (number of characters in a sector), and rounded low, the result (Vmax) is the number of sectors that can be written out by phase 1 at one time.

Figure 16. Values of PS1

Card Input

If all the input records are to be read in from punched cards, a calculable value (Vcard) must be obtained so it can be substituted for V in the basic-input-area equation. Vmax, the result of the formula just discussed, must be used as a basis for determining the value of Vcard.

If the value of Vmax is greater than the length of a single input record (L), substitute the record length for V in the basic-input-area equation and calculate the input area. Calculation of the value of Vcard is unnecessary because the input-record length prevents phase 1 from exceeding its maximum blocking factor.

However, if the input-record length is equal to or greater than the value of Vmax ($L \geqq Vmax$), the value of Vcard must be determined with the following formula:

$$Vcard = \left\lceil \frac{(n)\ (L)}{100} \right\rceil$$

To solve this equation, substitute successive values for n (starting with 1) to find the highest value of n that still results in the condition, Vcard is equal to or less than Vmax ($Vcard \leqq Vmax$). For example, assume that the value of Vmax is 3 sectors, and the input-record length (L) is 100 characters. Substituting 1 for n, multiplying it by 100, and dividing by 100 results in Vcard = 1. Vcard is less than Vmax (3); so 2 should be tried as a value for n. Substituting 2 for n, multiplying it by 100, and dividing by 100 results in Vcard = 2. Vcard is still less than Vmax (3); so 3 should be tried as a value for n. Substituting 3 for n, multiplying it by 100, and dividing by 100 and rounding high results in Vcard = 3. Vcard (3) is now equal to Vmax (3); therefore, any value greater than 3 will not satisfy all the conditions of the formula.

After the value of Vcard (the number of sectors of output produced by phase 1 at one time) has been determined, substitute it for V in the basic-input-area equation, and calculate the size of the input area.

Tape Input

If the input records are to be read from magnetic tape, a calculable value (Vtape) must be obtained so that it can be substituted for V in the basic-input-area equation. The solution of two equations, one of which uses the previously determined value of Vmax, must be performed before the value of Vtape is ultimately determined. The first equation is formulated as follows:

$$NCNT = \left\lfloor \frac{(Vmax)\ (100)}{(BI)\quad (L)} \right\rfloor$$

where

NCNT = Maximum number of input tape blocks that can be contained in core storage at one time.

BI = Input blocking factor.

Multiplying the number of sectors required for phase-1 output (Vmax) by 100 (the number of core positions in a sector) results in the number of core positions required for phase-1 output. Multiplying the input blocking factor (BI) by the input-record length (L) results in the number of core positions per block of input. Dividing the number of core positions required for phase-1 output by the number of core positions per block of input and rounding low results in the number of input tape blocks that can be contained in core storage at one time (NCNT).

The value of NCNT must be substituted in one of the following formulas to determine the value of Vtape:

1. The first formula can be used only if the maximum number of input tape blocks in core storage at one time (NCNT) multiplied by the input blocking factor (BI) results in a value that is equal to or less than 100 [$(NCNT)(BI) \leqq 100$]. This means that the maximum number of input records in core storage at one time will be 100 or less. Under these conditions, the formula used for finding the value of Vtape is:

   $$Vtape = \left\lceil \frac{(NCNT)\ (BI)\ (L)}{100} \right\rceil$$

   Multiplying the number of input tape blocks in core storage at one time (NCNT) by the input blocking factor (BI) results in the maximum number of input records in core storage at one time. This maximum multiplied by the record length (L) results in the number of core-storage positions required by phase-1 input at one time. Dividing the number of core-storage positions by 100 (number of positions in a sector) and rounding high results in the number of sectors required by phase-1 input at one time if the input medium is magnetic tape (Vtape) and $(NCNT)\ (BI) \leqq 100$.

   Substitute the value of Vtape for V in the basic-input-area equation, and calculate the length of the input area.

2. The second formula for finding the value of Vtape can be used only if multiplying the maximum number of input tape blocks in core storage at one time (NCNT) by the input blocking factor (BI) results in a value that is greater than 100 [$(NCNT)\ (BI) > 100$]. This means that the number of input records in core storage at one time will be greater than 100. When this condition exists, the formula used for

finding the value of Vtape is:

$$\text{Vtape} = \left\lceil \frac{(\text{TBI})\,(\text{L})}{100} \right\rceil$$

where

$$\text{TBI} = (\text{n})\,(\text{BI})$$

To solve this equation, substitute successive values for n (starting with 1) to find the highest value of n (the number of the user's input blocks that will be read into core storage at one time) that still results in the condition TBI is less than 100 ($\text{TBI} \leqq 100$). For example, assume that the input blocking factor (BI) is 12. Multiplying the values 1 through 7 by 12 will result in the condition $\text{TBI} \leqq 100$, but the *highest* value that can be substituted for n, while satisfying the condition $\text{TBI} \leqq 100$, is 8. TBI, in this case, has a substitute value of 96.

Multiplying the input blocking factor (BI) by the number of records per block (n), and controlling the result with 100 (which means that the *maximum* input blocking factor for this program is 100 records), results in the actual number of records per input block (TBI). Multiplying TBI by the record length (L) results in the number of storage positions (or characters) in each actual input block. Dividing positions per input block by 100 (positions per sector) and rounding high results in the number of sectors required at one time for output from phase 1 if the input medium is magnetic tape (Vtape) and $(\text{NCNT})(\text{BI}) > 100$.

Substitute the value of Vtape for V in the basic-input-area equation, and calculate the size of the input area.

Solving the Basic-Input-Area Equation

After finding the number of sectors of records that can be produced by phase 1 at one time (V), calculate the size of the input area required for the tape or card input records. The basic-input-area equation is:

$$\text{NS} = \left\lceil \frac{\text{N} + 1}{\left\lfloor \frac{100\text{V}}{\text{L}} \right\rfloor} \right\rceil \times \text{V}$$

Multiplying V (number of sectors) by 100 (positions per sector) results in the number of positions in the entire area. Dividing the number of positions in the entire area by the record length (L) and rounding low results in the maximum number of records in the phase-1 output area at one time. Dividing the number of tape or card input records increased by one (N + 1 allows room for a trailer label) by the number of records in the phase-1 output area at one time and rounding high results in the total number of blocks of output that will be produced by phase 1. Multiplying the total number of blocks of phase-1 output by the number of sectors in each output block (V) results in the total number of sectors (NS) required to store all the input records that are to be read into core storage from tape or cards.

Card Additions

If the sort object program is to sort disk input records, the user does not have to calculate an input area for the records. He specifies the disk area containing the records as his input area (see *RDLIN Cards: INP1, INP2, INP3, INP4*). However, if the user is going to make card additions to his disk records, he must calculate an input area for the card additions. Use the following formula for calculating the input area for card-addition records:

Additions Input Equation.

$$\text{NS} = \left\lceil \frac{\text{N} + 1}{\text{BI}} \right\rceil \times \text{BIS}$$

where

NS = Size of the card-additions input area in sectors.
N = Number of records to be added to the file.
BI = Number of records per input block.
BIS = Number of sectors per input block.

The values of BI and BIS substituted into this equation must be the same as BI and BIS for the disk file to which the card records will be added. If, for example, the disk input file is blocked with ten records per block (BI) and a block occupies two sectors (BIS), the card-addition records will also be blocked ten records per block and two sectors per block. (The card-addition records must, of course, be the same length and in the same format as the disk records.) The BI and BIS for the input file will have been specified in control card 1 (see *Control Cards*).

Dividing the number of records to be added (increased by one record) by the number of records per input block (BI) and rounding high results in the number of blocks required for the card additions. Multiplying the number of required blocks by the number of sectors per block results in the number of sectors required for the card-additions input area (NS). This area must be defined in an input area RDLIN card.

Work Area

The work area is the disk area in which the actual sorting takes place. Although it can be calculated with a single formula, it entails more than just one storage area. It actually consists of three adjacent, contiguous

Work Section	Work Section	Phase Routine Section

Figure 17. Schematic of the Work Area

sections (areas of disk): two work sections. and a section for storing the routines of various phases of the program (see Figure 17). Each of the two work sections must be large enough to contain all of the tags (8-character addresses and the attached control data) that are used to represent the input records during the sorting operation (see *Object Program Description*). Because control data will determine the sequence of the output records, each tag must contain the control-data information. Provision for control-data fields is part of the formula.

After all the tags (one for each record) have been written in the first section of the work area, the sequencing process, actually a sort and several merges, takes place (see *Object Program Description* for a complete discussion of the sequencing process). During the process of putting the tags in sequence, the program moves blocks of tags back and forth between the two work sections. At various points in the program, one or the other of the work sections will be filled with tags.

The third section in the work area is used to expedite execution of the program. As each phase of the program is needed, certain routines containing instructions for that phase are transferred from core storage to the third section in the work area. This reduces the amount of core storage required at any one time by the program.

The formula for calculating the length of the work area is:

$$W = 2H \left\lceil \frac{N}{\left\lfloor \frac{A}{CW} \right\rfloor} \right\rceil + P$$

where

W = Work area in sectors.

N = Number of input records.

CW = Total length of all the control-data fields in a single record, plus 8. (For example, if each input record contains five control-data fields of four characters each, the total length of the control-data fields is 20. A constant of 8 must be added to 20 to include the 8-character tag, giving CW a value of 28). The specified value of CW must always be 10 or greater. If the calculated CW < 10, set CW = 10.

A, H, and P = See Figure 18.

Dividing CW (control word) into A (the maximum number of characters written on disk by a phase) and rounding the result low results in the maximum number of control words that can be written in each block of a work section in the work area. Dividing N (the number of input records) by the result of the first division and rounding high results in the minimum number of blocks required to contain the control words for the entire input file. Multiplying 2 times H (the minimum number of sectors required by a phase to write a full block of data) times the number of blocks required to contain all the control words results in two times the number of sectors (the two work sections in the work area) required to store all the control words for the input file. The total sector requirement increased by P (the maximum number of sectors required to store certain routines for the individual phases of the program) results in the number of sectors required for the work area (W). The user must set aside a disk area by defining it in a RDLIN card.

The work area must be one continuous area, and it must be contained within one disk pack. If a label track is required, the work area cannot exceed 19,900 sectors; if a label track is not required, the work area cannot exceed 20,000 sectors.

Note: On rare occasions and under limited conditions, the result obtained from the work-area formula will be insufficient. If, for example, the output blocking factor for phase 4 (value of C) causes phase 4 to require more room in the work area than phases 1, 2, or 3 required, the original work-area calculation formula will produce an insufficient work area. When, for any reason, the work-area formula produces an insufficient work-area size, the following message will be printed out in the assignment phase messages. **NOTE** MFS HAS BEEN DECREASED BY XXXXXX RECORDS. If this message appears, a new work-area size must be calculated with the following formula:

$$W = 2 \left\lceil \frac{N-1}{C} \right\rceil \times \left\lceil \frac{C}{10} \right\rceil + P$$

C is the number of 10-character addresses in a phase-5 input block; the value of C is always given in an assignment-phase message.

where

N = Number of input records.

P = See Figure 18.

After the calculation has been completed, define the new work area in a RDLIN card and insert it in place of the invalid one that was originally used for work-area definition. The assignment phase must be rerun.

Overlapping Work and Output Areas

The work area can overlap the *end* of the output area or the *end* of the last output area if the sorted output records are to be written in non-contiguous sections. (For example, the output area may extend from one pack to another.) To calculate the number of work-

POSITIONS OF CORE STORAGE	A	H	P
4,000	399	4	118
8,000	799	8	125
12,000	1399	14	141
16,000	1899	19	151

Figure 18. Conditional Constant Values of A, H, and P

area sectors that can overlap the output area, subtract the constant P (see Figure 18) from the size of the work area. What occurs, in effect, is that the portion of the output area that is not overlapped is available for storage of the output records while the sorted tags are still in the last part of the work area. The size difference between the work area, which contains a comparatively limited amount of data, and the output area, which must be large enough to contain the entire output file, is such that the output records being written will never extend into the area containing the sorted tags.

Note: The low-limit address of the overlapping output area must be no greater than the low-limit address of the work area. The high-limit address of the output area must not be greater than the high-limit of the work area.

The program will subtract P from the available sectors contained within the limits defined by the last output RDLIN card if the work area overlaps the end of the output area. This allowance (P) should be made when the limits of the output area are being determined.

Output Area

The user of the sort object program should know the number of disk sectors required to store his sorted output file, particularly if he intends to make the most efficient use of his storage area. After the output blocking factor, the number of sectors per output block, and the total number of records to be sorted have been determined, the output sector requirement can be calculated with the following formulas. The output area need not be a single block of storage; the records can be written in intermittent areas of storage. However, the intermittent areas of storage must never require less than the minimum number of sectors that can be specified in a single output RDLIN card (see the following formula). If the program is to halt after phase 4 or 5, or if tape output is desired, no output area need be calculated.

Assume, for example, that a sorted alphabetical file is to be written on disk. The user may want to leave 50 blank sectors between each area of output records. He can specify a low-limit sector address, for each area, that is 50 sectors higher than the high-limit sector address of the preceding area of data. As many output RDLIN cards as are required may be used. If the user inserts more output RDLIN cards than the output file requires, the unnecessary cards are ignored by the program. If the user fails to calculate the output area, or if he makes an incorrect calculation, and the output area defined in the RDLIN cards is too small, the program halts and requests additional RDLIN cards.

The formula for calculating the number of sectors required for the output area is:

Equation 1

$$\left\lceil \frac{N+1}{BO} \right\rceil \times BOS$$

where

N = Total number of records to be sorted.

BO = Output blocking factor, which is the number of records per output block.

BOS = Number of sectors per output block.

Note: If multiple-output RDLIN cards will be used, add an additional BOS for each RDLIN except the last. The additional block is required for the record that will contain the 1EORb trailer label.

Dividing N (the number of records) by BO (the number of records per output block) and rounding high, results in the number of required output blocks. Multiplying the output block requirement by BOS (the number of sectors per output block), results in the number of sectors required for output.

If the user has specified deletions, the number of input records (N) will undoubtedly be greater than the number of output records. The program will automatically stop attempting to write output records when it reaches the last actual record, and it will then write an end-of-file trailer label in the next record area.

Assume, for example, that the input file requires 20,000 sectors of disk storage, that half of the input records will be deleted before the sort begins, and that the records are to be written in a disk-storage output area. Under these conditions, the last 10,000 sectors of output area will not be used when the output records are written in the output area (if the input blocking factor and input-record length are the same as the output blocking factor and output-record length). However, the user should make certain the stipulated output area is sufficient to contain all the input records, if no records are deleted.

If the output file is to be written out on tape, only those records that have been sorted will be transferred to the tape.

Note: The user is responsible for saving the deleted records (see *User Programming*), regardless of whether the output records are to be written out on tape or disk. The deleted records still exist in the initial input area.

The formula for calculating the minimum number of sectors defined by a single output RDLIN card is:

Equation 2

$$\left(\left\lfloor \frac{20{,}000}{LO} \right\rfloor \times \left\lceil \frac{BOS}{BO} \right\rceil \right) + BOS$$

where

LO = Length of records in the final output

Dividing 20,000 by LO (the output-record length—this may differ from the input-record length if the user has utilized the program features that permit deletion of characters from the low-order end of input records or the addition of record marks) and rounding low results in an approximation of the maximum number of output records that can be processed at one time during phase 6. Dividing BOS (the number of sectors per output block) by BO (the number of records per output block) and rounding high results in the number of sectors per output record. Multiplying the maximum number of output records by the number of sectors per output record results in the minimum number of sectors required for the output file (or the minimum number of sectors that can be specified in a single disk-output RDLIN card). The output area must be large enough to handle the maximum number of records phase 6 can write out at one time.

Equation 1 and equation 2 both result in a figure that can be called "the minimum number of sectors required for the output file." It is not likely that these two figures will be equal, or that the result of equation 2 will be greater than the result of equation 1. However, if the result of equation 2 is the larger it should be used and the output records should be stored in one output area that is defined by one disk output RDLIN card. In this case, the entire output area will not be used, although it has been specified in a RDLIN card, because the end-of-file trailer label will stop the writing process after the last output record has been written.

Multipack Output

The object program will accommodate multipack output. RDLIN card information must refer to the output pack on-line.

If the user has more than one drive on-line, output can be placed on several packs using one drive. The areas required on-line during phases 5 and 6 are the input area(s), work area, and work cylinder. These areas cannot be defined on a drive from which packs are to be removed during the processing of phases 5 and 6.

The user must mount a new disk pack on the drive when the capacity of the specified area has been reached.

Maximum File Size

Because the sort object program can process records on a system with a minimum of one and a maximum of five 1311 disk drives, the maximum file size (maximum number of records) varies according to the number of disk storage drives on-line. The maximum file size for any system configuration can be calculated with the following formula:

Basic Maximum File Size Equation

$$MFS = \left\lfloor \frac{(2 \times 10^6 \times D) - 100\,P - 2 \times 10^4}{Z + T} \right\rfloor$$

MFS = Maximum file size.
2×10^6 = Number of characters that can be stored on an IBM 1311.
D = Number of drives available to the sort program.
P = See Figure 18.
2×10^4 = Number of characters on the work cylinder. Do *not* use this factor unless phase-6 output is desired.
Z = See *Finding the Value of* Z.
T = See *Finding the Value of T.*

By determining the number of disk-storage positions available for the sort ($2 \times 10^6 \times D$) and subtracting the number of positions reserved for the sort program, and the work cylinder if phase-6 output is desired, the number of positions available for the sorting process is calculated. The number of positions available for sorting is then divided by the sum of *either* the number of characters required for each individual output record *or* the number of characters required for processing each tag (one of these two values will be substituted for Z) and added to the number of characters required for each input record in the disk input area (this value will be T). The result of this division is the maximum number of records (MFS) that can be sorted under the given system specifications and record specifications.

The assumption made in the formula for calculating the maximum file size is that a maximum overlap will exist between the work area and the output area (see *Overlapping Work and Output Areas*).

The approximated value of MFS can be used for calculating the work area, the output area, and the input area. The user simply substitutes MFS for N in the formulas used to calculate these three areas. The result of substituting MFS for N is a *maximum* work area, a *maximum* output area, and a *maximum* input area. Use of MFS instead of N means, of course, that the user must calculate Vcard, Vtape, NCNT, and TBI before he can calculate MFS. Therefore, the size of the input, output, and work areas will be calculated after the MFS equation has been solved.

Finding the Value of Z

$$Z = \text{The greater of} \quad \left\lceil \frac{BOL}{BO} \right\rceil \quad \text{or} \quad \left\lceil \frac{200H}{\left\lfloor \frac{A}{CW} \right\rfloor} \right\rceil$$

If the program is to be terminated after phase 4 or phase 5, or if the phase-6 output is to be written on tape, $\frac{BOL}{BO}$ will always equal 0, because the output area will not be required.

BO = Output blocking factor. (The number of records per output block.)

BOL = Output block length. (The number of characters in the output block. This is equivalent to the number of sectors required for an output block multiplied by 100.)

H = See Figure 18.

A = See Figure 18.

CW = Length of the control word. (The sum of the length of the control-data fields added to the 8-character disk address.)

Before the basic-maximum-file-size equation can be solved, the value of Z must be determined. To determine the value of Z, divide the output block length (BOL) by the output blocking factor (BO) and round the result high. The result of this division is the number of characters required for each individual output record.

Next, divide the maximum number of characters written on disk by a phase (A—Figure 18) by the value of CW (see the discussion of CW in the section *Work Area*) and round low. The result of this division is the maximum number of tags that can be written on disk or tape by a phase at one time. Multiply the minimum number of sectors required to write a block of tags during a phase (H—Figure 18) by 200 (the number of characters required to write a sector in the work area —each sector requires 100 characters, but the work area has two sections, each of which must be capable of containing a tag for each input record). The result of this multiplication is the total number of characters required in the combined sections of the work area. Dividing the number of characters required by a single phase for the two work-area sections by the number of tags written on disk by a phase at one time and rounding high results in the number of characters required to process each tag.

This value (characters per tag) is then compared to the number of positions required for each individual output record, and the greater of the two values can be substituted for Z in the basic-maximum-file-size equation. If the output area will not be required (program terminates at the end of phase 4 or phase 5, or tape output has been specified), the number of characters required to process each tag will always be substituted for Z, because the output area will be equal to zero.

Finding the Value of T

Equation a.

$$T = \left\lceil \frac{BIL}{BI} \right\rceil \quad \text{for disk input and/or additions}$$

Equation b.

$$T = \left\lceil \frac{100V\,card}{n} \right\rceil \quad \text{for card input}$$

Equation c.

$$T = \left\lceil \frac{100V\,tape}{TBI} \right\rceil \quad \text{for tape input if NCNT x BI} > 100$$

Equation d.

$$T = \left\lceil \frac{100V\,tape}{NCNT \times BI} \right\rceil \quad \text{for tape input if NCNT x BI} \leq 100$$

where

BIL = Input-block length. (The number of characters in an input block. This equals the number of sectors per input block multiplied by 100. For example, if three 80-character records are contained in three sectors, BIL = 300.)

BI = Input blocking factor. (The number of records per input block.)

TBI = The value substituted for TBI in the formula used to find the value of Vtape when (NCNT) (BI)>100. (See *Input Area: Tape Input*).

n = The value substituted for n in the formula used basic-input area equation (see *Input Area: Card Input*).

Vcard = The value that was determined with the card input formula and was substituted for V in the basic-input-area equation (see *Input Area: Card Input*).

Vtape = The value that was determined with the tape input formula was substituted for V in the basic-input-area equation (see *Input Area: Tape Input*).

NCNT = Maximum number of input tape blocks that can be contained in core storage at one time. The value of NCNT was determined with a formula and was used to find the value of Vtape (see *Input Area: Tape Input*).

In equation a, determine the value of T by dividing the input block length by the number of records per input block and rounding high. The value of T, in this case, is the number of characters per input record.

In equation b, multiply Vcard (number of sectors of output produced by phase 1 at one time) by 100 (number of characters per sector) to find the number of characters of output produced by phase 1 at one time. Divide the result of this multiplication by the value of n (number of records of a given length in a sector) that was substituted in the formula used to calculate Vcard, and round the result high. The value of T, if the user has card input, is the number of characters per record produced by phase 1 at one time.

In equation c, multiply Vtape (number of sectors of output produced by phase 1 at one time) by 100 (number of characters per sector) to find the number of characters of output produced by phase 1 at one time. Divide the result of this multiplication by the value that was substituted for TBI (TBI is the number of blocks of a given size that can be contained in core storage) and round high. The value of T, under the condition NCNT x BI $>$ 100, is the number of characters per record produced by phase 1 at one time.

In equation d, multiply Vtape (number of sectors of output produced by phase 1 at one time) by 100 (number of characters per sector) to find the number of characters produced by phase 1 at one time. Next, multiply NCNT (the number of user blocks of a given size that can be contained in core storage) by BI, the input blocking factor) to find the number of records that can be contained in core storage at one time. Divide the number of characters produced by phase 1 at one time by the number of records in a sector and round high. The value of T, under the condition NCNT x BI $\leq$ 100, is the number of characters required for each record.

After the MFS has been computed, the input, output, and work areas should be computed individually to make certain the program specifications governing placement of required areas are not violated. For example, the work area must be confined to a single disk pack.

Optimum Placement Disk-Storage Areas

The minimum sorting time for a given machine configuration is attained when the input/output areas, work area, and work cylinder are placed in a particular manner in disk storage.

The optimum placement of the areas depends on:

1. Whether the object program is to be run through phase 4, or through phase 6,
2. The number of disk storage drives on-line, and
3. Whether the direct-seek feature or normal-seek operation is to be used.

If the system has one drive on-line, and if the normal-seek operation or the direct-seek feature is to be used, and if the object program is to be run to the end of phase 4, then the work area should precede and be adjacent to the input area in disk storage (Figure 19). The work area should be placed as close as possible to the home position.

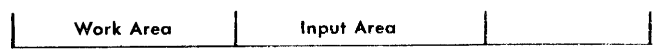

Figure 19. One Drive On-Line, Terminating in Phase 4

If the system has two or more disk drives on-line, and if the normal-seek operation or the direct-seek feature is to be used, and if the object program is to be run to the end of phase 4, then the input area and the work area should be placed on different drives (Figure 20).

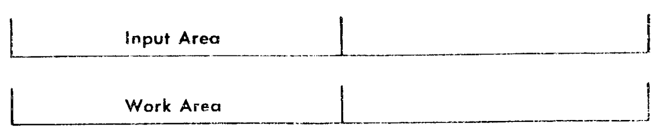

Figure 20. Two or More Drives On-Line, Terminating in Phase 4

If the system has one drive on-line, and if the normal-seek operation is to be used, and if the object program is to be run to the end of phase 6, then the work cylinder should be placed first, the input area second, and the output and work areas last (Figure 21). If the direct-seek feature is to be used, the work cylinder and input area should be interchanged.

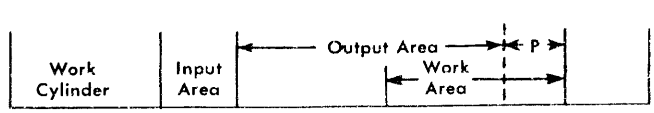

Figure 21. One Drive On-Line, Terminating in Phase 6

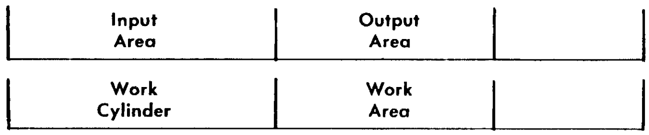

Figure 22. Two Drives On-Line, Terminating in Phase 6

If the system has two disk drives on-line, and if the normal-seek operation or the direct-seek feature is to be used, and if the object program is to be run to the end of phase 6, then the input area should be placed first and the output area second on one of the drives. On the other drive, the work cylinder should be placed first and the work area second (Figure 22). If, however, the input and output areas cannot be placed on one drive, the remainder of the output area should be placed on the second drive. In this case, the work cylinder should be placed first and the output and work areas last (Figure 23).

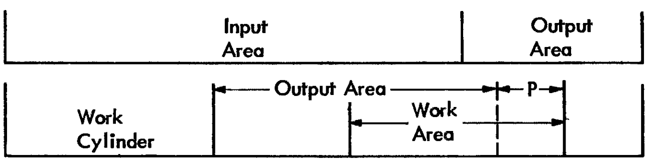

Figure 23. Two Drives On-Line, Terminating in Phase 6

If the system has three disk drives on-line, and if the normal-seek operation is to be used, and if the object program is to be run to the end of phase 6, then the input area should be on one drive and the output area, on another. The work cylinder should be placed first and the work area, second on the remaining drive (Figure 24). If the direct-seek feature is to be used, interchange the locations of the work area and work cylinder.

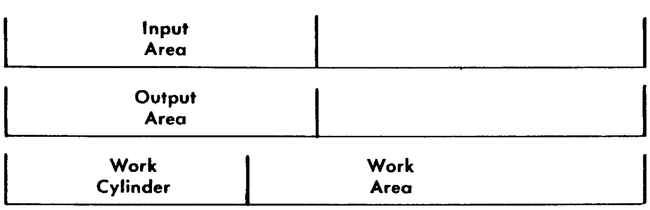

Figure 24. Three Drives On-Line, Terminating in Phase 6.

If the system has four or five drives on-line, and if the normal-seek operation or the direct-seek feature is to be used, and if the object program is to be run to the end of phase 6, then the four areas should be placed on different drives (Figure 25). If the input area extends beyond the limit of one or two drives, the user, when planning the placement of the additional areas, should assume that he has one or two fewer drives respectively.

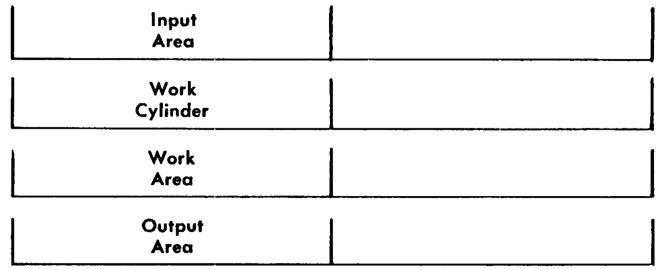

Figure 25. Four or Five Drives On-Line, Terminating in Phase 6

RDLIN Cards

RDLIN (read label information) cards supply the sort object program with label and disk-area information. Input RDLIN cards are required for every object-program execution. Output RDLIN cards are required for phase-6 disk output (sorted records) and for phase-6 tape output with labels.

Input RDLIN Cards

Input RDLIN cards have the following functions:

1. To define the work area
2. To define the input area(s)
3. To define the work cylinder
4. To supply label information (tape and/or disk).

Disk RDLIN Cards

Input RDLIN cards that define disk areas are inserted in the assignment-phase portion of the object deck. They can be in any order, but:

1. The last card must contain an L in column 5.
2. If card additions are to be sorted, the last input area RDLIN card must contain an A in column 5. Therefore, when additions are specified, the last input area RDLIN card cannot be the last card.

If unlabeled disk areas are to be used, the RDLIN card must contain identification, drive number, RDLIN and limits.

Note: There is a difference between input RDLIN cards and input-area RDLIN cards. The former refers to all the disk RDLIN cards that are inserted in the assignment-phase portion of the object deck; the latter refers to those RDLIN cards that define input areas (INP1, INP2, INP3, INP4).

Restrictions on Limits

The restrictions are:

1. Any area defined by a RDLIN card must be one continuous area and be contained within one disk pack.
2. The address ranges on the disk packs that contain the input areas, work area, and work cylinder must not be the same. For example, if the input area is on a pack with an address range of 000000-019999, the work area and work cylinder cannot be a different pack with the same address range.

Format

The format of an input disk RDLIN card is:

Columns	*Identification*
1-4	INP*x* = input area. *x* is the number (1-4) of the area. WKAR = work area. WCYL = work cylinder.
5	Last card: Punch an L in the last input RDLIN card. This card can be an INP*x*, WKAR, or WCYL RDLIN card. Additions: An *A* in the last input area RDLIN card specifies the disk area that is reserved for card additions to disk files. An INP*x* RDLIN card that contains an *A* in column 5 cannot be the last input RDLIN card.
6-10	Blank.
11	Drive number: If the limits specified in columns 55-66 are within the range 000000-019999, punch the number of the drive on which the pack is to be mounted.
12	Drive number: If the limits specified in columns 55-66 are within the range 020000-039999, punch the number of the drive on which the pack is to be mounted.
13	Drive number: If the limits specified in columns 55-66 are within the range of 040000-059999, punch the number of the drive on which the pack is to be mounted.
14	Drive number: If the limits specified in columns 55-66 are within the range 060000-079999, punch the number of the drive on which the pack is to be mounted.
15	Drive number: If the limits specified in columns 55-66 are within the range 080000-099999, punch the number of the drive on which the pack is to be mounted. *Note:* The disk-drive control numbers are 0, 2, 4, 6, and 8.
16-20	RDLIN
21-24	File retention period.
25-29	Creation date.
30-39	File identification.
40-44	File serial number.
45-49	Pack serial number.
50	Blank.
51-54	File sequence number.
55-60	Low limit.
61-66	High limit. (Note that Sort 6 requires that the high limit of the RDLIN card be the address of the *last* sector of the last block.)
67-80	Blank.

If the user desires to use the work area as an output area, he must specify the normal retention period. However, if the user wishes to utilize the label-check feature and does not want to retain the information on either the work cylinder or work area, he should specify the retention period of 0. This will protect these areas for a minimum of only one day.

Figure 26 shows the format of a standard disk header label.

Work Area RDLIN Card. This card is always required. It must contain WKAR in columns 1-4. The upper limit must be 0*xxx*99. If the sort object program is to be run on a 4K machine, the lower limit must be a multiple of 4.

Input Area RDLIN Cards. At least one input area RDLIN card is required. A maximum of four input areas can be defined. Only the last area can contain a 1EOFb trailer label.

The identification (columns 1-4) for input area RDLIN cards are:

ID	*Meaning*
INP1	Specifies the first or only input area. If the input medium is tape or cards, the area defined must be large enough to contain the entire area (only one input area can be used with either medium).
INP2	Specifies the second input area.
INP3	Specifies the third input area.
INP4	Specifies the fourth input area.

Work Cylinder RDLIN Card. This card is required if phase-6 output (sorted records) is desired. It must contain WCYL in columns 1-4 and define the limits of one actual cylinder (200 sectors).

Tape RDLIN Cards

If a tape input file contains standard labels that are to be checked, tape RDLIN cards are required. Nonstandard labels must be processed by user routines.

The tape RDLIN cards are inserted in the phase-1 portion of the object deck.

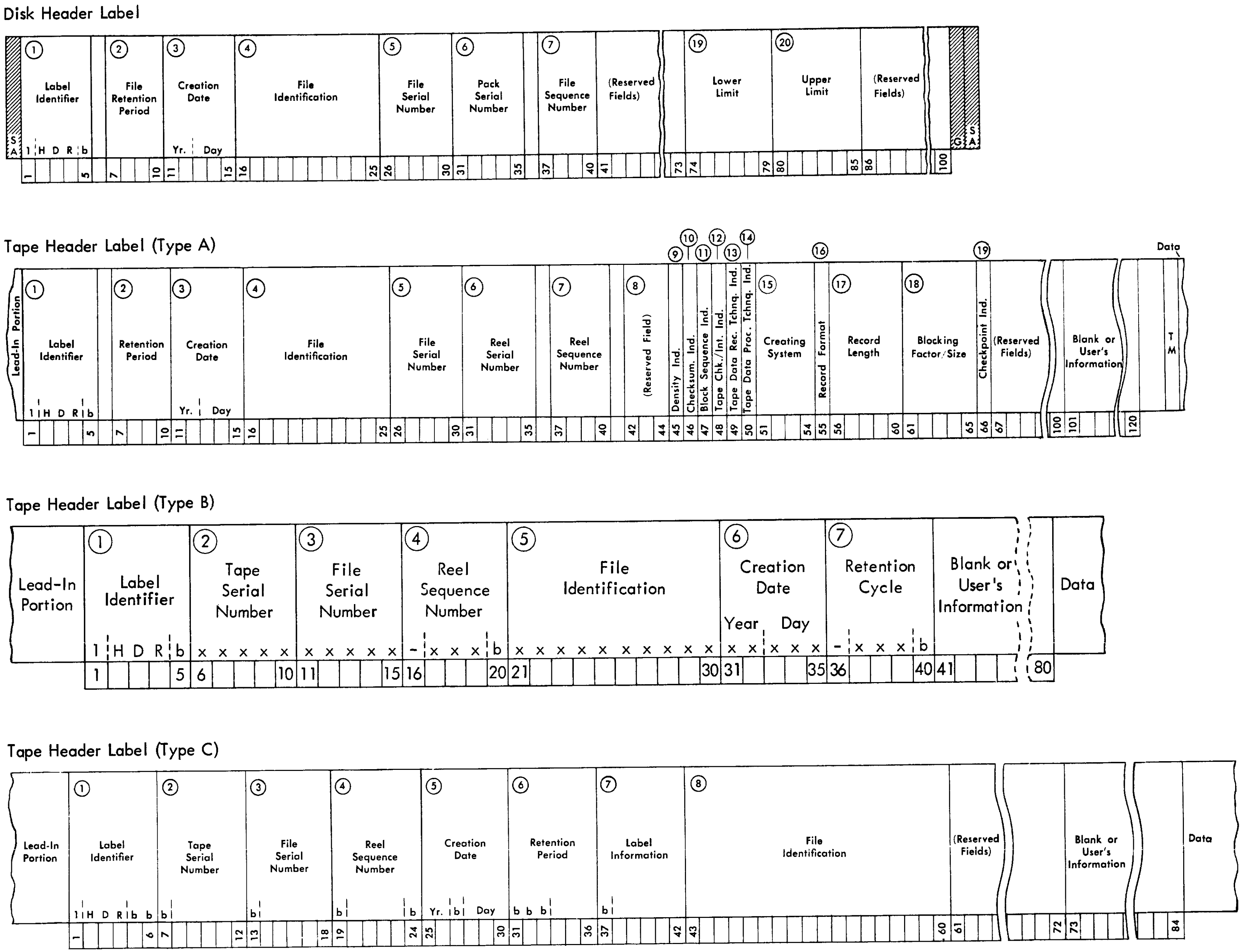

Figure 26. Schematics of Standard Disk and Tape Header Labels

Formats

1. Use the following format if type A (120-character) labels were specified at generation time:

Columns	*Indicate*
16-20	RDLIN
21-24	Retention period
25-29	Creation date
30-39	File identification
40-44	File serial number
45-48	Reel sequence number

2. Use the following format if type B (80-character) labels were specified at generation time:

Columns	*Indicate*
16-20	RDLIN
21-25	File serial number
26	Minus sign (11-punch)
27-29	Reel sequence number
30	Blank
31-40	File identification
41-45	Creation date
46	Minus sign (11-punch)
47-49	Retention period
50	Blank

3. Use the following format if type C (84-character) labels were specified at generation time:

Columns	*Indicate*
16-20	RDLIN
21	Blank
22-26	File serial number
27	Blank
28-31	Reel sequence number
32	Blank
33-34	Creation date: year
35	Blank
36-38	Creation date: day
39-41	Blank
42-44	Retention period
45	Blank
46-50	Label information
51-68	File identification

Figure 26 shows the format of standard tape header labels. (Figure 27 shows format of standard tape trailer labels.)

Output RDLIN Cards

Output RDLIN cards have the following functions:

1. To define the output area if phase-6 output (sorted records) is to be written on disk.
2. To supply label information (tape or disk)

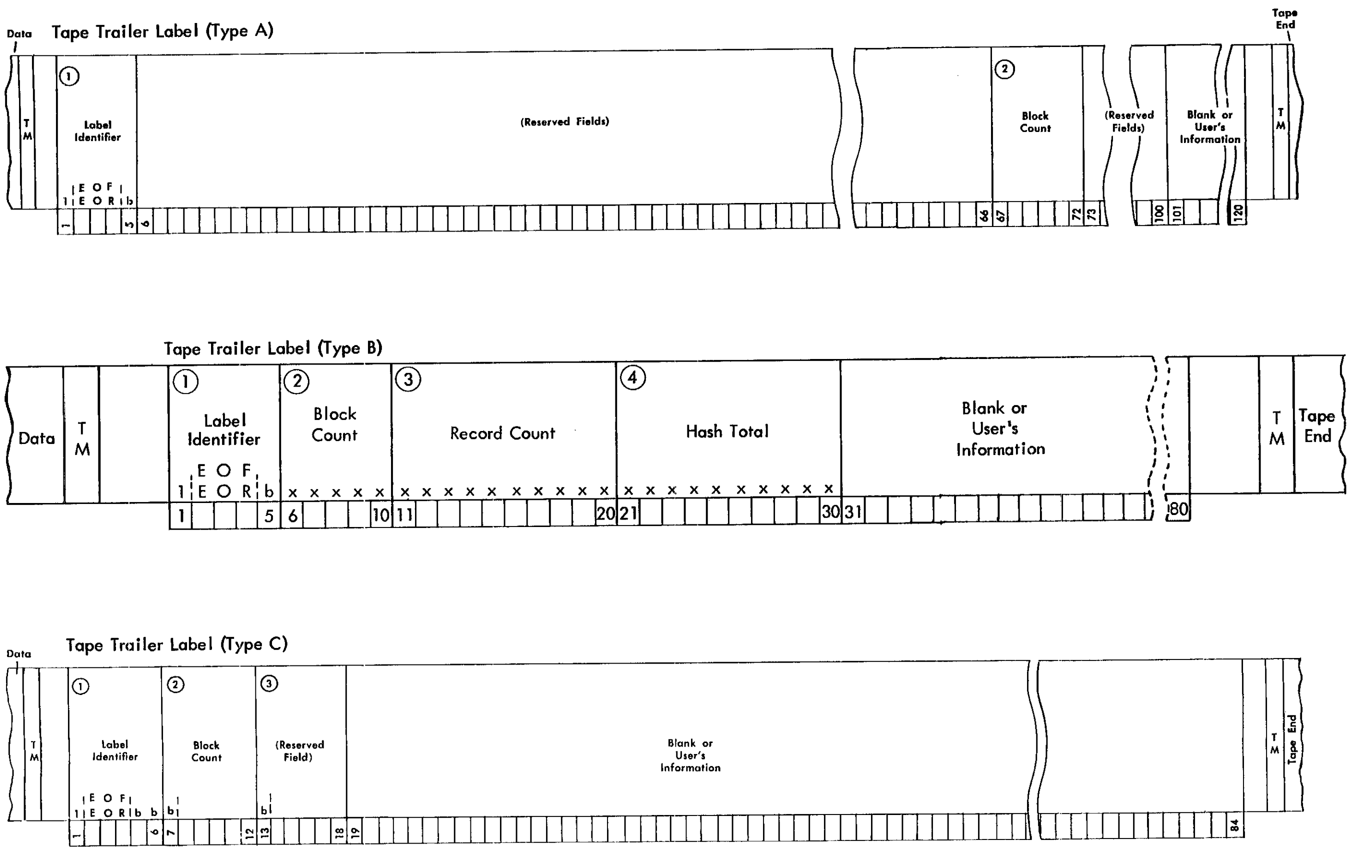

Figure 27. Schematics of Standard Tape Trailer Labels

Disk RDLIN Cards

Any number of disk RDLIN cards can be inserted in the phase-6 portion of the object deck. The cards can be in any order. The format of a disk RDLIN card is:

Columns	*Indicate*
1	Last card. Punch an *L* in the last output RDLIN card.
2-5	Overlap. Punch STOT in the last output RDLIN card if the area it defines is to overlap the first part of the work area.
6-80	Same as input RDLIN cards.

If the output and work areas are to overlap, the starting address of the output area must be less than or equal to the starting address of the work area. The high limit of the output area must not be greater than the high limit of the work area. Only the end of the output area can overlap the work area. See *File Area Definition: Overlapping Work and Output Areas.*

Each disk area, defined by a RDLIN card, must be contained within one disk pack.

If the object deck does not contain tape output routines and/or phase-6 user routines, two identical sets of RDLIN cards are required. One of these sets will be utilized by the sort program; the other will be ignored. The last card in each set must contain L or LSTOT depending on what was punched in the last card of the first set.

Sorts 61, 62, 63, and 64 do not require the two identical sets.

Tape RDLIN Cards

If standard tape labels are to be generated and written on the output tape, tape RDLIN cards are required. The formats are the same as those described under *Input RDLIN Cards.* Use the format for the type of label specified in the TAPLABLOUT card at generation time.

Date Card

If the user has specified checking of standard input or output tape or disk labels, a date card must be inserted in the assignment-phase portion of the object deck. The program uses the date-card information to check the retention period of existing labels and to write the creation date on output labels. The format of the date card is:

Columns	*Indicate*
1-3	Core-storage address at which the high-order character of the date will be loaded. Punch *082.*
4-5	Number of characters to be loaded from this card. Punch *05.*
6	Word separator (0-5-8 punch).
7-11	Today's date: year (xx) and day (xxx).

Control Cards

The control cards contain the information for a specific application of the generated object program. Two control cards must be prepared by the user for each file that is to be sorted. Punch the control cards as follows:

Control Card 1

Column	*Indicates*
1	*Input media.* 1. Punch a *C* if input is from cards. 2. Punch a *D* if input is from disk storage. 3. Punch a *2* if input is from tape. The input tape must be mounted on tape unit 2.
2	*Output media.* 1. Punch a *D* if the output is to be written in disk storage. 2. Punch a *3* if the output is to be written on tape. The output tape must be mounted on tape unit 3.
3-5	*Input blocking factor (B_i).* Punch the number of records (001-100) in each input block. The input blocking factor for unblocked records is 001.
6-8	*Output blocking factor (B_o).* Punch the number of records (001-100) that are to be written in each output block. The output blocking factor for unblocked records is 001. If the object program run is to halt at the end of phase 4 or phase 5, these columns must be left blank.
9-10	*Number of sectors (01-30) in each disk storage input block (BIS).* This entry must be made when input is from disk storage. Otherwise is must be blank.
11-12	*Number of sectors (01-30) in each disk storage output block (BOS).* This entry must be made when the output is to be written in disk storage. If the object program run is to halt at the end of phase 4 or phase 5, these columns must be left blank.
13-16	*Input record length (0005-3000).*
17	*Core storage capacity of the 1401 or 1460.* 1. Punch a *1* if the 1401 has 4,000 positions of core storage. 2. Punch a *2* if the 1401 has 8,000 positions of core storage. 3. Punch a *3* if the 1401 has 12,000 positions of core storage. 4. Punch a *4* if the 1401 has 16,000 positions of core storage.
18	*Desired sequence of the output file.* 1. Punch a *0* for ascending sequence. 2. Punch a *1* for descending sequence.
19	*Deletions by class in phase 1.* 1. Punch a *0* if there are to be no deletions. 2. Punch a *1* if records of a certain class are to be deleted during phase 1. Columns 21-25 of this control card must be punched in this case.

Column	Indicates
20	*Selections by class in phase 1.* 1. Punch a *0* if there is to be no selection. 2. Punch a *1* if only records of a certain class are to be selected from the file and sorted during phase 1. Columns 21-25 of this control card must be punched in this case.
21	*Character controlling the deletion or selection of records in phase 1.* This column must be punched if column 19 or column 20 contains a *1* unless the controlling character is a blank.
22-25	*Position in each record that controls the deletion or selection of records during phase 1.* Punch the position of the record in which the character punched in column 21 can be found. These columns must be punched if column 19 or column 20 contains a *1*.
26	*Card additions to the file.* 1. Punch a *0* if additions to the file are not to be made. 2. Punch a *1* if additions are to be made to the file.
27	*Deletion, in phase 4, of records on the basis of control data.* 1. Punch a *0* if there are no deletions to be made. 2. Punch a *1* if records are to be deleted during phase 4 on the basis of control data. In this case, cards containing the deletion control data must be properly inserted in the object program deck. (Refer to *Deletions by Control Data.*)
28-31	*Number of characters to be deleted from the low-order position of each output record (0000-2995).* 1. Punch *0000* if no characters are to be deleted. 2. Punch the number of characters (0001-2995) that are to be deleted from each record. *Note:* Number of characters deleted can be 2996 if a *1* is punched in column 36.
32	*Type of card input or additions.* 1. Punch a *0* if no card input or additions. 2. Punch a *1* if the input or additions consist of 70-character card records. 3. Punch a *2* if the input or additions consist of 80-character card records.
33-34	*Number of tape reels in the input file (00-99).* 1. Punch *00* if tape input is not specified. 2. Punch the number of reels in the input file (01-99) when input is from tape.
35	*Sequence check and hash total.* 1. Punch a *0* if hash totals are not to be accumulated in phases 2, 3, and 4, and the file is not to be sequence-checked during phase 4. 2. Punch a *1* if the hash totals are to be accumulated and the sequence-check is to be performed.
36	*Record marks in the output file.* 1. Punch a *0* if a record mark is not to follow each output record. 2. Punch a *1* if a record mark is to follow each output record. The output record length is increased by 1 in this case.
37	*Restart run.* 1. Punch a *0* if this is not a restart run. 2. Punch a *1* if this is a restart run.

Column	Indicates
38-42	*Starting address of the user written program, if present, in phase 1.* These columns must be left blank if a user routine is not required in phase 1.
43-47	*Starting address of the user written program, if present, in phase 4.* These columns must be left blank if a user routine is not required in phase 4.
48-52	*Starting address of the user written program, if present, in phase 6.* These columns must be left blank if a user routine is not required in phase 6.
53	*Output desired.* 1. Punch a *0* for sorted records (phase-6 halt). 2. Punch a *3* for sorted 8-character addresses (phase-4 halt). 3. Punch a *4* for the control data from each record (phase-*4* halt). 4. Punch a *5* for sorted 10-character addresses (phase-5 halt).
54	*Location of phase-4 output.* 1. Punch a *1* if the output is to be relocated so that the starting address of the output is to be the starting address of the work area. 2. Punch a *0* if the output is not to be relocated. *Note:* In either case, a halt occurs and a message giving the starting address of the output is printed. 3. Leave blank if phase-4 output is not desired.
55-66	*Leave blank.*
67	*Input disk labels.* 1. Punch a *0* if there are no input disk labels. 2. Punch a *1* if only the file name is to be checked on standard disk labels on the input file, and labels are written for work area and work cylinder. 3. Punch a *2* if full label checking is to be performed on each standard disk label on the input file, and labels are written for work area and work cylinder. *Note:* A disk RDLIN card containing the label information must be properly inserted in the object deck if this column contains a 1 or a 2.
68	*Output disk labels.* 1. Punch a *0* if no labels are to be written on the output disk pack. 2. Punch a *1* if the labels already in the label track are to be checked and then a new header label written for each section of the output file. A disk RDLIN card containing the label information must be properly inserted in the object deck in this case.
69	*Input tape labels* 1. Punch a *0* if the input tapes contain no labels. 2. Punch a *1* if the input tapes contain labels. The desired checking has been specified in the TAPLABLIN parameter card. A tape RDLIN card must be properly inserted in the object deck if standard label checking is to be performed.
70	*Output tape labels.* 1. Punch a *0* if labels are not to be generated and written on the output tape. 2. Punch a *1* if labels are to be written on the output tapes. The desired type of label has been specified in the TAPLABLOUT card. When a *1* is

Column	Indicates
	punched in this column and standard labels are specified, the retention period in the header label that is already written on the output tape is automatically checked to make sure that the data on the tape can be destroyed. A tape RDLIN card must be properly inserted in the object deck in this case.
71-78	*Leave blank.*
79	*Control card number.* Punch a *1* in this column.
80	*Leave blank.*

Control Card 2

Control card 2 specifies the location and size of each control data field. This card is required in all cases.

Column	Indicates
1-2	*The total number of control data fields in each input record (01-10).*
3-5	*The total number of characters in all control data fields (001-189).*
6-9	*The location, within each record, of the high-order position of control data field 1.* If control data field 1 is in positions 1-10 of each record, these columns are punched *0001.*
10-12	*The number of characters in control data field 1.*
13-16	*The location, within each record, of the high-order position of control data field 2.*
17-19	*The number of characters in control data field 2.*
20-23	*The location, within each record, of the high-order position of control data field 3.*
24-26	*The number of characters in control data field 3.*
27-30	*The location, within each record, of the high-order position of control data field 4.*
31-33	*The number of characters in control data field 4.*
34-37	*The location, within each record, of the high-order position of control data field 5.*
38-40	*The number of characters in control data field 5.*
41-44	*The location, within each record, of the high-order position of control data field 6.*
45-47	*The number of characters in control data field 6.*
48-51	*The location, within each record, of the high-order position of control data field 7.*
52-54	*The number of characters in control data field 7.*
55-58	*The location, within each record, of the high-order position of control data field 8.*
59-61	*The number of characters in control data field 8.*
62-65	*The location, within each record, of the high-order position of control data field 9.*
66-68	*The number of characters in control data field 9.*
69-72	*The location, within each record, of the high-order position of control data field 10.*
73-75	*The number of characters in control data field 10.*
76-78	*Leave blank.*
79	*Control card number.* Punch a 2 in this column.
80	*Leave blank.*

This section describes the program decks for Sort 61, 62, 63, and 64; the preparation of the object deck; and the machine operator procedures for running the sort object program.

Description of IBM-Supplied Object Decks

The core-storage capacity of the object system and the presence or absence of the direct-seek feature determines which IBM-supplied sort should be used:

Program Name	*ID*	*Machine Size*	*Direct Seek*
Sort 61	614*x*1	4K	No
Sort 62	628*x*1	8K, 12K, 16K	No
Sort 63	63D*x*1	4K	Yes
Sort 64	64H*x*1	8K, 12K, 16K	Yes

Each object deck contains the seven sort program phases (0-6). Each card contains a sequence number in columns 73-75 and identification in columns 76-80. The cards in each phase deck are numbered sequentially. The value of *x* identifies the phase of the object program and can be 0, 1, 2, 3, 4, 5, or 6. The parameters used in the generation of the sort object decks are given in *Sort Object Program Specifications.*

Preparing the Object Deck

The user must consult his program listing when preparing the object deck. Comments on the listing indicate insertion points for control cards, RDLIN cards, card input, card additions, and deletion cards.

Before the sort operation can be performed, the object program decks for the various phases must be combined into one deck (Figure 28). The procedure for combining the decks is:

1. Remove the first two cards from each phase object deck except the assignment-phase deck. These are the clear storage cards.
2. Remove the next to last card from both the phase-1 and phase-6 object program decks. Insert these cards immediately following the disk RDLIN cards in the assignment-phase object deck. If phase 6 is not to become part of the object program deck, the next to last card of the phase-6 object deck does not have to be inserted into the assignment-phase deck. If a Sort 61, 62, 63, or 64 object deck is being used, skip step 2.
3. Place the complete assignment-phase object deck first. Follow this deck with the phase object decks in ascending sequence according to phase number. If phase-5 output is desired, the phase-6 deck is not required. If phase-4 output is desired, neither the phase-5 nor phase-6 decks are required. The resulting deck is the complete object program deck.

Assignment Phase

If a date card is required, place it immediately after the 6-card loader. The date card is required if standard tape or disk output labels are to be processed.

Place the control cards immediately ahead of the card that is preceded by the following comments on the listing:

```
***CONTROL CARDS AND RDLIN CARDS GO***
***BEFORE THIS OVERLAY FIRST CARD***
```

Control card 1 must precede control card 2.

Place input RDLIN cards that define disk areas immediately after control card 2. The last input area may contain an EOF, and the input area associated with it must be the last defined input area in the sequence indicated in column 4 of the RDLIN card. The last card must contain an L in column 5.

Phase 1

If any of the following comments are applicable to the sort job, place the specified card(s) immediately ahead of the card that is preceded by the comment on the listing:

```
*TAPE RDLIN CARD PRECEDES THIS CARD*
*CARD INPUT BEFORE THIS CARD*
*ADDITIONS BEFORE THIS CARD*
```

Phase 4

If records are to be deleted on the basis of their control data, place the cards containing the control data immediately ahead of the card that is preceded by the following comment on the listing:

```
*DELETIONS BEFORE THIS CARD*
```

There must be one deletion card for each unique control-data field. All the records that contain the same control data will be deleted by the single deletion card. The cards must be in the collating sequence specified in control card 1.

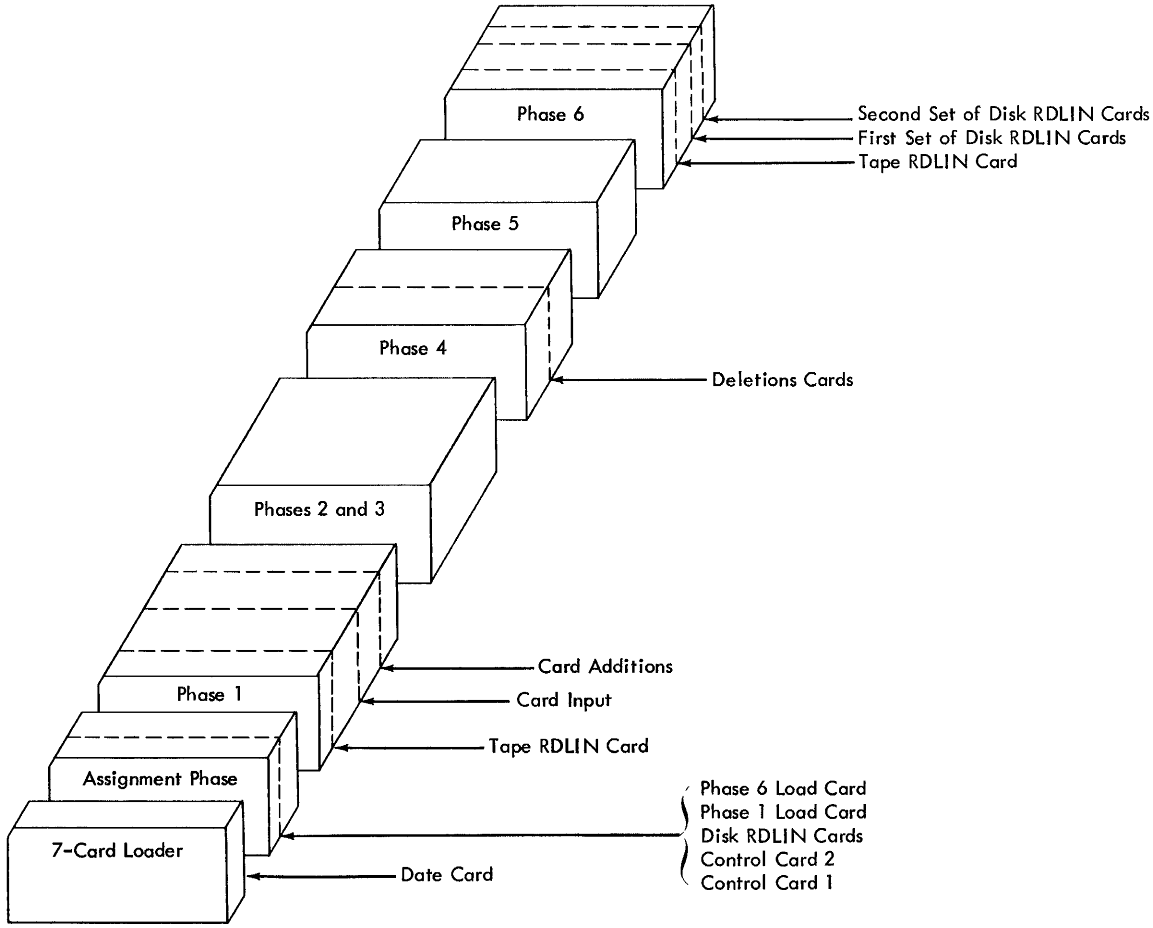

Figure 28. Approximate Insertion Points for User-Prepared Cards

Phase 6

The RDLIN cards that define the disk output area(s) must be placed immediately ahead of the card that is preceded by the first appearance of the following comment on the program listing.

DISK RDLIN CARDS PRECEDE THIS CARD

An additional set of disk RDLIN cards is required if the sort object deck does *not* have at least one of the following characteristics:

1. Phase-6 user routines.
2. Tape output routines (TAPE was *not* specified in the OUTPTMEDIA parameter card when the sort object program was generated).

The additional set of disk RDLIN cards must be identical to the first set. Insert the identical set immediately ahead of the card that is preceded by the second appearance of the following comment on the program listing:

DISK RDLIN CARDS PRECEDE THIS CARD

The identical set is not required when Sort 61, 62, 63, or 64 is used because tape output routines are included in the object deck.

The sequence of the RDLIN cards determines the sequence of the use of the output areas. The last card must contain an L in column 1. If the second set is required, the last card of the set also must contain an L in column 1.

The RDLIN card that is to be used to generate standard output tape labels must be placed immediately ahead of the card that is preceded by the following comment on the program listing:

TAPE RDLIN CARD PRECEDES THIS CARD

Executing the Object Program

To execute the sort object program:

1. Insert forms on the printer or console printer and install an appropriately punched carriage tape.

2. Mount any tape(s) required on the unit(s) specified in control card 1.
3. Mount the required disk pack(s) on the proper drive(s). The work area and all input areas must be on-line. The work cylinder must be on-line if phase 6 is to be executed.
4. Turn sense switch A on.
5. Turn the I/O check stop switch on.
6. Place the prepared object deck in the read hopper of the 1402.
7. Press CHECK RESET, START RESET, and LOAD.
8. When the system attempts to read the last card, press START.

If a card-read error occurs while loading the program with the I/O check stop switch on, the procedure is:

1. Remove the cards from the hopper.
2. Press NPRO to run out the non-processed cards.
3. Place the last three cards (two non-processed cards and the card in error) and the remainder of the deck in the hopper.
4. Press START.

Checkpoint, Interrupt, and Restart

Conditions sometimes necessitate interrupting the sort object program run before the sorting job being performed is complete. To allow for this, a feature that permits the user to stop processing at any point during the sort run and later to resume processing is incorporated into the program.

Processing can be interrupted and restarted because the program periodically writes a checkpoint-information area during the running of the program. The checkpoint information area is written at the completion of phases 1, 2, 3 and 4; after every phase-3 merge pass; and periodically during phases 5 and 6. The checkpoint information area contains the necessary data to allow the program to perform a restart after an interruption has occurred.

If phase-4 output is specified, the program cannot be interrupted and restarted after the last phase-4 object card has been read.

If the output is to be written on disk, the program cannot be restarted if interrupted from the time that the first card of the last phase-5 program overlay is read until the first time halt number 11 occurs (B-address register 655). This condition results when the program is to halt upon the completion of phase 6.

If the output is to be written on tape, the program cannot be restarted if interrupted from the time that the first card of the last phase-5 program overlay is read until some records have been written on the output tape. With multireel output, do not interrupt between reels. Wait until some records have been written on the next tape.

The checkpoint information is written on the last five tracks of the disk-storage work area. The user must make sure that the checkpoint information, work area, and input areas are not destroyed between the time of interrupt and the subsequent restart. If the information is destroyed, the sort must be rerun from the beginning. The disk utility program (1401-UT-053) cannot be used to copy or print the last five tracks of the work area, because the area is written in both the move and load modes.

Press the stop key to interrupt the program at any point. When the program is restarted, processing will begin from the point at which the last checkpoint information area was written prior to the interrupt. There is no restart in phase 1. When restarting in phase 2, 3, 4, or 5, simply remount the disk packs being used at the time of the interrupt, punch a *1* in column 37 of control card 1; then reload the complete object program deck including control cards, and RDLIN cards.

Each disk pack must be remounted on a 1311 disk-storage drive with the same number as the one from which it was removed. If the program was interrupted in phase 6, the restart procedure varies depending upon whether all RDLIN cards had been read or not before the interrupt occurred.

If the interrupt occurs after one or more RDLIN cards have been read, the RDLIN cards that were read must be removed from the deck before processing is restarted. One exception to the rule is that the last RDLIN card containing LSTOT in columns 1-5 or only L in column 1 must always be present when restarting in phase 6. To restart the sort after the required RDLIN cards have been removed, follow the restart procedure described for the other phases.

If phase 6 is interrupted and the preceding case does not apply, restart the sort as described for the other phases.

If the user specifies disk-input label checking, the work cylinder and work area are checked initially as output areas. However, on a restart run, these areas are checked as input areas.

If the user specifies disk-output label checking and restarts in phase 6, halt number 13 (B-address register 777) may occur. This halt occurs when the retention period on the output area that was being processed at the time of interrupt is checked.

Input/Output Error Routines

The tape and disk input/output operations performed during the running of the object program are performed by the 1401, 1460 disk IOCS routines. Refer to the *IOCS Specifications* publication for a description of these routines.

When the input is from tape and a block of records is found to be unreadable, the procedure followed is that which was specified in the TAPE parameter card at generation time. If DUMP was specified, the unreadable blocks are written on the tape mounted on tape unit 4. If SCAN was specified, unreadable record blocks cause a program halt so that the operator can investigate the error.

Halts and/or Messages

The following messages are printed during an object program run primarily to inform the machine operator of the current status of the program.

Assignment-Phase Messages

1. The information punched into each control card that is inserted into the object deck is printed exactly as it is punched into the card.

2. BCW-XXX BCWS-XXX LCF-XXX MFS-XXXXXX
   BI-XXX BO-XXX BIS-XXX BOS-XXX L-XXXX

   This message shows various calculated and specified parameters of the sort run.

   BI = input blocking factor.

   BO = output blocking factor.

   BIS = number of sectors required for each input block.

   BOS = number of sectors required for each output block.

   L = record length.

   BCW = number of tags that will be blocked together during phase 1. This value equals:

$$\left\lfloor \frac{A}{CW} \right\rfloor$$

where

$\lfloor\ \rfloor$ = rounded low

CW = control word length

A = 399 for a 1401 with 4,000 positions of core storage

A = 799 for a 1401 or 1460 with 8,000 positions of core storage

A = 1399 for a 1401 or 1460 with 12,000 positions of core storage

A = 1899 for a 1401 or 1460 with 16,000 positions of core storage

BCWS = number of sectors required to contain one block of tags formed during phase 1. This value equals: $\frac{A+1}{100}$

LCF = total number of characters in all control data fields.

MFS = maximum file size.

C = the number of addresses in a phase-4 output block.

3. **NOTE** MFS HAS BEEN DECREASED BY XXXXXX RECORDS

   This message appears only if the factor C has entered into the calculation of the work-area size. If the decreased MFS is not large enough to accommodate the file, the alternate work-area formula must be used to recalculate the work-area size.

4. END OF ASSIGNMENT PHASE

Phase-1 Messages

1. SORT 6 PHASE ONE

2. RECORDS PROCESSED — XXXXXX
   END OF PHASE

This message prints upon the completion of phase 1. XXXXXX is the number of records that were processed during the phase.

Phase-2 Message

XXXXXX RECORDS PROCESSED — PHASE 2
END OF INTERNAL SORT

This message prints upon the completion of phase 2. XXXXXX is the number of records that were processed during the phase.

Phase-3 Message

END OF PHASE 3

This message prints upon the completion of phase 3.

Phase-4 Messages

1. SORT 6 PHASE FOUR

2. XXXXXX — RECORDS PROCESSED
   END OF PHASE FOUR

This message prints upon the completion of phase 4. XXXXXX is the number of records that were processed during the phase.

Phase-5 Message

XXXXXX ADDRESSES PROCESSED — PHASE 5
END OF ADDRESS ADJUSTMENT

This message prints upon the completion of phase 5. XXXXXX is the number of addresses that were processed during the phase.

Phase-6 Message

The only messages printed during phase 6 are accompanied by a halt.

Halts and Associated Messages

Figures 29, 30, 31, 32, 33, 34, and 35 are listings, by phase, of the halts and associated messages that can occur during the running of the object program. The information given for each halt consists of:

1. The contents of the B-address register, the I-address register, or the A-register when the halt occurs.
   The B-address register should be referred to first. If the tens and units positions are blank (except 7bb), the A-register identifies the halt. If a tape operation causes a halt, the tape SELECT light that is on identifies the tape unit concerned.
2. The message associated with the halt and/or the reason for the halt.
3. The procedure to be followed when the halt occurs.

NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1.	1	xxbb	A message is not associated with this halt. Reason: A disk error other than those defined elsewhere in this list has occurred.	Press the start key to retry the disk operation.
2.	2	xxbb	A message is not associated with this halt. Reason: The lower and upper limits in a standard header label for a disk input file, do not check.	Visually check the disk pack and label specifications. Press the start key, the start reset key, and then the start key to process the file with this condition. Mount a new pack, if required, and press the start key twice to recheck all header labels. The new pack must have the address range already specified by the RDLIN card.
3.	3 C21	xxbb	A message is not associated with this halt. Reason: The disk light indicates the reason for this halt. Disk light ON. Access inoperable condition on a disk header label read or write operation. Disk light OFF. Disk header label does not contain 1HDRb in the first five positions.	Press the start key, the start reset key, and then the start key to recheck all header labels on this pack for this logical file.
4.	4	xxbb	A message is not associated with this halt. Reason: Parity, wrong-length-record, or unequal-address-compare error detected while reading or writing a disk header label.	Press the start key to recheck all header labels on this pack for this logical file.
5.	5 C41	xxbb	A message is not associated with this halt. Reason: No header label found for the specified disk input file.	Visually check the disk pack and label specifications. Mount a new pack, if required, and press the start key to recheck all header labels. The new pack must have the address range already specified by the RDLIN card.
6.	6 C42	xxbb	A message is not associated with this halt. Reason: A RDLIN information is missing for a disk file.	Remove the cards from the hopper, run the cards out, place the proper cards (including correct RDLIN card) in the hopper, and press the start key, the start reset key, and the start key to read the RDLIN card.
7.		7bb	A message is not associated with this halt. Reason: A card error has occurred while reading the control cards or the assignment phase RDLIN cards.	Use the non-process run-out key on the 1402 to remove the cards from the reader. Remove the last three cards from the normal read stacker and place these cards in front of the card deck that was removed from the read hopper before the non-process run-out key was pressed. Place the deck in the read hopper; and press the start key to continue processing.
8.		001	A message is not associated with this halt. Reason: An error occurred while the program was attempting to read a RDLIN card.	Use the non-process run-out key on the 1402 to remove the cards from the reader. Remove the last three cards from the normal read stacker and place these cards in front of the card deck that was removed from the read hopper before the non-process run-out key was pressed. Place the deck in the read hopper; and press the start key to continue processing.
9.		002	A message is not associated with this halt. Reason: The program has read a card that was not a RDLIN card when a RDLIN card should have been read. This condition results if the RDLIN cards were not properly inserted into the object deck, the identification in columns 1-5 of the RDLIN card was incorrectly punched, or an L has not been punched in column 5 of the last RDLIN card.	Correct the condition and reload the program deck.
10.		003	A message is not associated with this halt. Reason: The disk storage input areas have been incorrectly specified in the RDLIN cards.	Correct the RDLIN cards and reload the program deck.
11.		004	A message is not associated with this halt. Reason: The first disk storage input area has been specified as an additions area.	Correct the RDLIN cards so that the additions area is the last specified disk storage input area and then reload the program deck.
12.		005	A message is not associated with this halt. Reason: A disk storage input area, other than the first or last, has been specified as an additions area.	Correct the RDLIN cards so that the additions area is the last specified disk storage input area and then reload the program deck.

Figure 29. Assignment Phase, Halts and Messages (Part 1 of 5)

NUMBER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
13.	006	A message is not associated with this halt. Reason: Control card 1 either has not been inserted in the program deck or it has been incorrectly inserted.	Correctly insert the control card and reload the program deck.
14.	007	A message is not associated with this halt. Reason: Control card 2 either has not been inserted in the program deck or it has been incorrectly inserted.	Correctly insert the control card and reload the program deck.
15.	008	A message is not associated with this halt. Reason: A required RDLIN card (INP1, or WKAR, or WCYL) either has not been inserted in the program deck or it has been incorrectly inserted.	Correctly insert the RDLIN card(s) and reload the program deck.
16.	010	WORK AREA IS INCORRECTLY DEFINED Reason: The disk storage work area either has not been specified or it has not been properly specified in the WKAR RDLIN card.	Correct the RDLIN card, properly insert it in the program deck, and reload the deck.
17.	011	CONTROL CARD #1 HAS INVALID CHARACTERS PUNCHED ON IT - CORRECT AND RELOAD PROGRAM Reason: A character, other than an allowable one has been punched in control card 1. BI, BO, BIS and/or BOS have been incorrectly specified.	Correct the control card and reload the program deck.
18.	012	There are two possible messages that can be associated with this halt. 1. WRONG INPUT MEDIA SPECIFIED Reason: A blank character has been punched column 1 of control card 1. 2. WRONG OUTPUT MEDIA SPECIFIED Reason: A blank character has been punched in column 2 of control card 1.	Correct the control card and reload the program deck.
19.	013	There are two possible messages that can be associated with this halt. 1. INPUT TAPE UNIT INVALIDLY SPECIFIED Reason: A character other than C, D, or 2 has been punched in column 1 of control card 1 when tape input is specified. 2. OUTPUT TAPE UNIT INVALIDLY SPECIFIED Reason: A character other than D or 3 has been punched in column 2 of control card 1 when tape output is specified.	Correct the control card and reload the program deck.
20.	014	CONTROL CARD # (1) (2) HAS INVALID CHARACTERS PUNCHED ON IT - CORRECT AND RELOAD THE PROGRAM Reason: An alphabetic character has been specified where a numeric character should have specified in the indicated control card.	Correct the control card and reload the program
21.	015	THE NO. OF CONTROL FIELDS IN INVALIDLY PUNCHED ON CC #2, CORRECT IT AND RELOAD THE PROGRAM Reason: The characters specified in columns 1-2 of control card 2 are incorrect.	Correct the control card and reload the program deck.
22.	016	CONTROL CARD #1 IS INVALIDLY PUNCHED Reason: A character, other than an allowable one, has been punched in control card 1.	Correct the control card and reload the program deck.
23.	017	INPUT AREA IS NOT A MULTIPLE OF BIS Reason: The limits of the disk storage input area when input is from disk storage must specify an area containing a number of sectors that is a multiple of the number of sectors required for an input block, or a 1EOF or a 1EOR trailer record must follow the last disk input record.	Correct the control card and reload the program deck if no 1EOF or 1EOR trailer record exists. Otherwise press the start key to continue.

Figure 29. Assignment Phase, Halts and Messages (Part 2 of 5)

NUMBER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
24.	018	NO. OF INPUT TAPES ARE INCORRECTLY SPECIFIED Reason: Either columns 33-34 of control card 1 are blank or they contain an invalid character and input is from tape.	Correct the control card and reload the program deck.
25.	019	THE BLOCK SIZE OR USER AREA IS TOO BIG Reason: An input block cannot be processed in the available core storage positions. This condition results if the combined phase 1 user area and the input block size exceeds the area available.	Correct the control card and/or condition and reload the program deck.
26.	020	BOTH DELETIONS AND SELECTIONS ARE SPECIFIED Reason: Both deletions and selections by class during phase 1 are specified	Correct the control card and reload the program deck.
27.	021	CARD TYPE 70 OR 80 IS NOT SPECIFIED Reason: Card input or additions has been specified and column 32 of control card 1 does not contain a 1 or 2.	Correct the control card and reload the program deck.
28.	023	ADDITIONS CANNOT BE HAD WITH CARD, TAPE INPUT Reason: Card additions cannot be specified when input is from tape or cards.	Correct the control card and reload the program deck.
29.	024	SEQUENCE INCORRECTLY SPECIFIED PUNCH A 1 OR 0 IN COLUMN 18 AND RELOAD PROGRAM Reason: Column 18 of control card 1 is either blank or contains other than a 0 or 1.	Correct the control card and reload the program deck.
30.	025	DELETIONS OR SELECTIONS ARE SPECIFIED WITHOUT A RELATIVE POS. OF CTL CHARACTER SPECIFIED Reason: Columns 22-25 of control card 1 are blank and deletions or selections by class during phase 1 has been specified.	Correct the control card and reload the program deck.
31.	026	ADDRESS OF A CONTROL FIELD IS SPECIFIED LARGER THAN L Reason: The location of a control data field specified in control card 2 extends beyond the specified length of an input record.	Correct the control card and reload the program deck.
32.	027	NO INPUT AREA SPECIFIED FOR ADDITIONS Reason: Additions to the file during phase 1 is specified, but only one input-area RDLIN card is in the deck.	Insert the required RDLIN card in the object deck and reload the program deck.
33.	028	SPECIFIED L LESS THAN MIN RECORD LENGTH Reason: The input record length specified in columns 13-16 of control card 1 is less than 5 for disk or card input, 10 for blocked tape input, or 13 for unblocked tape input.	Correct the control card and reload the program deck.
34.	029	USER AREA PH. 4 IS TOO LARGE Reason: The specified phase 4 user area exceeds allowable area.	Correct the control card and reload the program deck.
35.	030	CONTROL FIELD GREATER THAN 80 FOR DELETIONS IN PHASE 4 Reason: Deletions by control data during phase 4 is specified and the specified total number of characters in all control data fields exceeds 80.	Correct the control card and reload the program deck.

Figure 29. Assignment Phase, Halts and Messages (Part 3 of 5)

NUMBER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
36.	031	AREA X OVERLAPS WITH AREA X Reason: The disk storage areas specified in the input RDLIN cards overlap. The message refers to the areas as follows: AREA 1. Disk storage area in INP1 card. AREA 2. Disk storage area in INP2 card. AREA 3. Disk storage area in INP3 card. AREA 4. Disk storage area in INP4 card. AREA 5. Disk storage area in WKAR card. AREA 6. Disk storage area in WCYL card.	Correct the RDLIN cards and reload the program deck.
37.	032	CONTROL FIELDS ARE OVERLAPPING Reason: Two or more of the control data fields specified in control card 2 overlap.	Correct the control card and reload the program deck.
38.	033	L EXCEEDS LMAX - XXXX Reason: The input record length specified in columns 13-16 of control card 1 exceeds the maximum allowable record length.	Correct the control card and reload the program deck.
39.	034	There are two possible messages that can be associated with this halt. 1. BIL EXCEEDS LMAX - XXXX Reason: The specified 100 BIS for disk input results in a value that exceeds the maximum allowable block length. 2. BOL EXCEEDS LMAX - XXXX Reason: The specified 100 BOS for disk output results in a value that exceeds the maximum allowable block length.	Correct the control card and reload the program deck.
40.	035	CONTROL FIELD EXCEEDS MAX - 189 Reason: The specified total number of characters in all control data fields (columns 3-5, control card 2) exceeds 189 which is the maximum allowable.	Correct the control card and reload the program deck.
41.	037	There are two possible messages that can be associated with this halt. 1. L EXCEEDS INPUT BLOCK Reason: The specified record length exceeds the value BIS (100). 2. L EXCEEDS OUTPUT BLOCK Reason: The specified record length exceeds value BOS (100).	Correct the control card and reload the program deck.
42.	038	(BI) (BO) IS TOO LARGE Reason: Either the specified input blocking factor, or the specified output blocking factor exceeds 100.	Correct the control card and reload the program deck.
43.	039	There are two possible messages that can be associated with this halt. 1. INPUT BLOCK IS GREATER THAN BIS X 100 Reason: The specified input blocking factor multiplied by the specified input record length results in a value that exceeds the value obtained when the specified number of sectors per input block is multiplied by 100. 2. OUTPUT BLOCK IS GREATER THAN BOX X 100 Reason: The specified output blocking factor multiplied by the specified output record length results in a value that exceeds the value obtained when the specified number of sectors per output block is multiplied by 100.	Correct the control card and reload the program deck.

Figure 29. Assignment Phase, Halts and Messages (Part 4 of 5)

NUMBER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
44.	040	Two possible messages can be associated with this halt. 1. USER AREA OF PHASE 6 IS TOO LARGE, OR MIN. STORAGE REQ. IS EXCEEDED Reason: An output block cannot be processed in the available core storage positions. The combination of the user area and the output block length exceeds the area available. The user must either reduce the size of the phase 6 user area or reduce the size of the output block. 2. USER AREA OF PHASE 6 IS TOO LARGE, OR BO IS TOO LARGE Reason: An output block cannot be processed in the available core storage positions. The combination of the user area and the output block length exceeds the area available. The user must either reduce the size of the phase 6 user area or reduce the size of the output block.	Correct the control card and reload the program deck.
45.	041	RESTART IN PROGRESS, PRESS START IF DESIRED Reason: Either a 1 has been punched in column 37 of control card 1 or an incorrect character has been punched.	If the program is being restarted after an interrupt, press the start key to continue processing. Otherwise, correct the control card and reload the program deck.
46.	042	MACHINE SIZE NOT SPECIFIED, PRESS START FOR 4K Reason: Column 17 of control card 1 has a character other than 1, 2, 3, or 4.	If the program is being run on a 1401 with 4,000 positions of core storage, press the start key to continue processing. Otherwise correct control card 2 and reload the program deck.
47.	043	SPECIFIED AND COMPUTED TOTAL LENGTH OF CONTROL FIELDS ARE NOT EQUAL, PRESS START TO CONTINUE Reason: The specified total number of characters in all control data fields (columns 3-5 of control card 2) is not equal to the sum of the individually specified control data field lengths.	If the individually specified control data field lengths are acceptable, press the start key to continue processing. Otherwise, correct control card 2 and reload the program deck.
48.	044	There are two possible messages that can be associated with this halt. 1. BI X L IS GREATER THAN L MAX Reason: The specified input record length multiplied by the input blocking factor results in a value that exceeds the maximum allowable block length. 2. BO X L IS GREATER THAN L MAX Reason: The specified output record length multiplied by the output blocking factor results in a value that exceeds the maximum allowable block length.	Correct the condition and reload the program deck.
49.	222	A message is not associated with this halt. Reason: Seven unequal-address-compare detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
50.	333	A message is not associated with this halt. Reason: Seven parity-error-detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.
51.	444	A message is not associated with this halt. Reason: Access-inoperable error detected during a disk operation.	Determine the disk drive number, by examining the storage position specified by the A-address register. Alter-in a different number, if required. Press the start key to retry with the same or altered drive number.
52.	555	A message is not associated with this halt. Reason: Seven wrong-length-record detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
53.	777	A message is not associated with this halt. Reason: The portion of a pack specified for a random disk output file contains active records. This applies to labeled files only and is detected by the dates in the header label.	Mount a different disk pack and press the start key twice to recheck all header labels. Press the start key, the start reset key, and then the start key to use the original pack without altering. This automatically deletes the data in the file identification field of the header label.
54.	888	A message is not associated with this halt. Reason: The pack serial number of a labeled disk output file does not check.	Press the start key, the start reset key, and then the start key to write the disk header label with this condition. Press the start key twice to recheck all header labels.

Figure 29. Assignment Phase, Halts and Messages (Part 5 of 5)

NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1	+0 CBA82	xxbb	A message is not associated with this halt. Reason: First halt (parity error) when SCAN is specified. (The error stop switch on the tape adapter unit must be off.)	Set the tape-select switch to D, turn off the check stop switch on the auxiliary console, and press the start key to reread the error block for scanning. (Second halt at symbolic label BOCSCN+1 in the program listing.)
2	1	xxbb	A message is not associated with this halt. Reason: A disk error other than those defined elsewhere in this list has occurred.	Press the start key twice to retry the disk operation.
3	⌑ CBA84	xxbb	A message is not associated with this halt. Reason: A parity error was detected while reading the tape header label.	Press the start key to read the header label again.
4	$ CB821	xxbb	A message is not associated with this halt. Reason: A tape input header label does not check with the user's specifications.	Press the start key twice to recheck the header label. Press the start key, the start reset key and then the start key to process the reel of data without rechecking the header label.
5	/ CA1	xxbb	A message is not associated with this halt. Reason: A parity error was detected while reading a tape trailer label.	Press the start key to reread the trailer label.
6	@ C84	xxbb	A message is not associated with this halt. Reason: A parity error was detected while writing on the dump tape.	Press the start key to backspace, erase, and write the record again.
7	. BA821	xxbb	A message is not associated with this halt. Reason: A RDLIN information card is missing for a tape file.	Remove the cards from the hopper, run the cards out, place the proper cards (including correct RDLIN card) in the card reader, and press the start key, the start reset key, and the start key to re-read the RDLIN card.
8	* B84	xxbb	A message is not associated with this halt. Reason: The tape input trailer label does not check with the totals accumulated by IOCS.	Press the start key to ignore the condition and continue processing.
9	X CA421	xxbb	A message is not associated with this halt. Reason: End-of-reel has been reached on an input reel other than the last in a multi-reel file and label processing has been specified.	Mount the next reel of data for this file and press the start key.
10		103	A message is not associated with this halt. Reason: A card read error has occurred during the reading of a RDLIN card in phase 1.	Follow the procedure given for halt number 15.
11		104	A message is not associated with this halt. Reason: End-of-reel has been reached on an input reel other than the last in a multi-reel file and label processing has not been specified.	Mount the next reel of the input file and press the start key to continue processing.
12		141	A message is not associated with this halt. Reason: The maximum file size has been exceeded.	Disk input: Recalculate the size of the work area, change the limits in the RDLIN card, and reload the program deck. Card or tape input: Recalculate the size of the input and work areas, change the limits in the RDLIN cards, and reload the program deck.

Figure 30. Phase 1 Halts and Messages (Part 1 of 2)

NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
13		151	A message is not associated with this halt. Reason: Disk input: The specified disk storage input area does not contain a number of sectors that is a multiple of the number of sectors required for an input block (BIS) or a 1EOF or a 1EOR trailer record does not follow the last disk input record. Card or tape input: The specified disk storage input area has been exceeded.	Change the input area limits in the RDLIN card and reload the program deck.
14		161	CARD IN ERROR XXXXXXXXX Reason: If 70 column format card input is specified, this halt indicates that a sequence error has been detected in columns 9 and 10 of an input card, or the identification in columns 1-8 does not check with the previous card read of the record. The card containing the sequence error is the last card in the stacker when the halt occurs. The sequence number must range from 01-43. If 80 column card input is being read, this halt indicates that the total number of cards read was not a multiple of the cards per record. In this case the card identification is not printed.	If the input is in the 70 column card format, use the non-process run-out key to remove the cards from the reader. Remove the last three cards from the normal read stacker, correct the card in error, and place the cards, in the correct order, in front of the deck that was removed from the read hopper before the non-process run-out key was pressed. Place the deck in the read hopper and press the start key to continue processing. If the input is in the 80 column card format, use the non-process run-out key to remove the cards from the reader. Remove the last card from the normal read stacker, place the additional number of input cards required in front of this card and place these cards in front of the deck. Place the deck in the read hopper and press the start key to continue processing.
15		171	A message is not associated with this halt. Reason: A card read error has occurred when input is from cards.	Use the non-process run-out key on the 1402 to remove the cards from the reader. Remove the last three cards from the normal read stacker and place these cards in front of the card deck that was removed from the read hopper before the non-process run-out key was used. Place the deck in the read hopper and then press the start key to continue processing.
16		181	CARD IN ERROR XXXXXXXXXX Reason: If 70 column card additions are being made to the file, this halt indicates that a sequence error has been detected in columns 9 and 10 of an addition card, or the identification in columns 1-8 does not check with the previous card of the record read. The card containing the sequence error is the last card in the stacker when the halt occurs. The sequence number must range from 01-43. If 80 column card additions are being made to the file, this halt indicates that the total number of cards read was not a multiple of the cards per record. In this case the card identification is not printed.	Follow the procedure given for halt number 14.
17		191	A message is not associated with this halt. Reason: Additions have been specified and a card read error has occurred.	Follow the procedure given for halt number 15.
18		222	A message is not associated with this halt. Reason: Seven unequal-address-compare detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
19		333	A message is not associated with this halt. Reason: Seven parity-error detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)

Figure 30. Phase 1 Halts and Messages (Part 2 of 2)

NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1	1	xxbb	A message is not associated with this halt. Reason: A disk error other than those defined elsewhere in this list has occurred.	Press the start key twice to retry the disk operation.
2		201	PHASE 1 RECORD COUNT XXXXXX PRESS START TO ACCEPT PH2 RECORD COUNT OR RESTART PROGRAM Reason: The number of records processed in phase 2 does not equal the number processed in phase 1.	Either press the start key to accept the phase 2 record count and continue processing, or reload the program deck.
3		222	A message is not associated with this halt. Reason: Seven unequal-address-compare detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
4		333	A message is not associated with this halt. Reason: Seven parity-error detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
5		444	A message is not associated with this halt. Reason: Access-inoperable error detected during a disk operation.	Determine the disk drive number, by examining the storage position specified by the A-address register. Alter-in a different number, if required. Press the start key to retry with the same or altered drive number.
6		555	A message is not associated with this halt. Reason: Seven wrong-length-record detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)

Figure 31. Phase 2 Halts and Messages

HALT IDENTIFIED BY I-ADDRESS REGISTER			
NUMBER	I-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1.	881-4K 3134-8, 12, 16K	A message is not associated with this halt. Reason: The hash total accumulated in phase 3 does not equal the hash total accumulated in phase 2.	Either press the start key to accept the phase 3 hash total and continue processing, or reload the program deck.

HALTS INDENTIFIED BY B-ADDRESS REGISTER OR A-REGISTER				
NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1	1	xxbb	A message is not associated with this halt. Reason: A disk error other than those defined elsewhere in this list has occurred.	Press the start key twice to retry the disk operation.
2		222	A message is not associated with this halt. Reason: Seven unequal-address-compare detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
3		333	A message is not associated with this halt. Reason: Seven parity-error detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
4		444	A message is not associated with this halt. Reason: Access-inoperable error detected during a disk operation.	Determine the disk drive number, by examining the storage position specified by the A-address register. Alter-in a different number, if required. Press the start key to retry with the same or altered drive number.
5		555	A message is not associated with this halt. Reason: Seven wrong-length-record detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)

Figure 32. Phase 3 Halts and Messages

NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1.	1	xxbb	A message is not associated with this halt. Reason: A disk error other than those defined elsewhere in this list has occured.	Press the start key twice to retry the disk operation.
2.		7bb	A message is not associated with this halt. Reason: Deletions by control data have been specified and a card read error has occured.	Use the non-process run-out key on the 1402 to remove the cards from the reader. Remove the last three cards from the normal read stacker and place these cards in front of the card deck that was removed from the read hopper before the non-process run-out key was pressed. Place the deck in the read hopper; and press the start key to continue processing.
3.		222	A message is not associated with this halt. Reason: Seven unequal-address-compare detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
4.		333	A message is not associated with this halt. Reason: Seven parity-error detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
5.		444	A message is not associated with this halt. Reason: Access-inoperable error detected during a disk operation.	Determine the disk drive number, by examining the storage position specified by the A-address register, Alter-in a different number, if required. Press the start key to retry with the same or altered drive number.
6.		454	RECORDS PROCESSED XXXXXX STARTING ADDRESS OF OUTPUT XXXXXX END OF SORT *6* Reason: If either phase 4 address output or phase 4 control data output is specified, this halt and associated message occur at the end of the sort.	
7.		464	RECORD COUNT ERROR - PRESS START TO ACCEPT Reason: The number of records processed during phase 4 does not equal the number processed in phase 2.	Either press the start key to accept the phase 4 record count and continue processing, or reload the program deck.
8.		474	HASH TOTAL ERROR - PRESS START TO ACCEPT Reason: The hash total accumulated in phase 4 does not equal the hash total accumulated in phase 3.	Either press the start key to accept the phase 4 hash total and continue processing, or reload the program deck.
9.		484	A message is not associated with this halt. Reason: A record has been found to be out of sequence.	Either press the start key to accept the record out of sequence and continue processing, or reload the program deck and resort the file.
10.		555	A message is not associated with this halt. Reason: Seven wrong-length-record detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)

Figure 33. Phase 4 Halts and Messages

NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1	1	xxbb	A message is not associated with this halt. Reason: A disk error other than those defined elsewhere in this list has occurred.	Press the start key twice to retry the disk operation.
2		222	A message is not associated with this halt. Reason: Seven unequal-address-compare detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
3		333	A message is not associated with this halt. Reason: Seven parity-error detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)
4		444	A message is not associated with this halt. Reason: Access-inoperable error detected during a disk operation.	Determine the disk drive number, by examing the storage position specified by the A-address register. Alter-in a different number, if required. Press the start key to retry with the same or altered drive number.
5		501	XXXXXX ADDRESS PROCESSED - PHASE 4 - PRESS START FOR PH5 COUNT OR RESTART Reason: The number of records processed in phase 5 does not equal the number processed in phase 4.	Either press the start key to accept the phase 5 record count and continue processing, or reload the program deck.
6		502	PHASE 5 OUTPUT STARTS AT SECTOR XXXXXX Reason: If the user has specified in column 53 of control card 1 that a halt is to occur at the end of phase 5, this halt and associated message occur at the end of the phase. XXXXXX is the address of the first sector of the phase 5 output.	
7		555	A message is not associated with this halt. Reason: Seven wrong-length-record detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)

Figure 34. Phase 5 Halts and Messages

NUMBER	A-REGISTER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
1	1	xxbb	A message is not associated with this halt. Reason: A disk error other than those defined elsewhere in this list has occurred.	Press the start key twice to retry the disk operation.
2	3 C21	xxbb	A message is not associated with this halt. Reason: The disk light indicates the reason for this halt. Disk light ON. Access inoperable conditon on a disk header label read or write operation. Disk light OFF. Disk header label does not contain 1HDRb in the first five positions.	Press the start key, the start reset key, and then the start key to recheck all header labels on this pack for this logical file.
3	4	xxbb	A message is not associated with this halt. Reason: Parity, wrong-lengh-record, or unequal address-compare error detected while reading or writing a disk header label.	Press the start key to recheck all header labels on this pack for this logical file.
4	5 C41.	xxbb	A message is not associated with this halt. Reason: No header label found for the specified disk input file.	Visually check the disk pack and label specification. Mount a new pack, if required, and press the start key to recheck all header labels. The new pack must have the address range already specified by the RDLIN card.
5	¤ CBA84	xxbb	A message is not associated with this halt. Reason: Parity error detected while reading tape header label.	Press the start key to re-read the header label.
6	-0 B82	xxbb	A message is not associated with this halt. Reason: Expiration date for this tape has not been reached.	Press the start key twice to check the retention period again. Press the start key, the start reset key and then the start key to use this reel anyway.
7	L B21	xxbb	A message is not associated with this halt. Reason: Parity error detected while writing tape header label.	Press the start key to write the header label again.
8	X CA421	xxbb	A message is not associated with this halt. Reason: End-of-reel conditon on the output tape and label processing has been specified.	Mount the next reel of data for this file and press the start key.
9	, CA821	xxbb	A message is not associated with this halt. Reason: Parity error detected while writing tape trailer label.	Press the start key to write the trailer label again.
10	% A84	xxbb	A message is not associated with this halt. Reason: Thirty parity-error detections while attempting to write a block of data.	Press the start key to attempt 30 more times to write this block correctly.
11		222	A message is not associated with this halt. Reason: Seven unequal-address-compare detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control fie ld.)
12		333	A message is not associated with this halt. Reason: Seven parity-error detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk control field.)
13		444	A message is not associated with this halt. Reason: Access-inoperable error detected during a disk operation.	Determine the disk drive number, by examing the storage position specified by the A-address register. Alter-in a different number, if required. Press the start key to retry with the same or altered drive number.
14		555	A message is not associated with this halt. Reason: Seven wrong-length-record detections during a disk operation.	Press the start key twice to try ten more times. (The A-address register contains the high-order address of the disk-control field.)

Figure 35. Phase 6 Halts and Messages (Part 1 of 2)

NUMBER	B-ADDRESS REGISTER	MESSAGE AND/OR REASON	PROCEDURE
15.	611	A message is not associated with this halt. Reason: The tape output RDLIN card does not have RDLIN punched in columns 16-20.	Use the non-process run-out key on the 1402 to remove the cards from the reader. Remove the last three cards from the normal read stacker, correct the RDLIN card, and place these cards in front of the card deck that was removed from the read hopper before the non-process run-out key was pressed. Place the deck in the read hopper, and then press the start key to continue processing.
16.	622	A message is not associated with this halt. Reason: A card read error has occurred while the program was reading a disk output RDLIN card.	Follow the procedure given for halt number 15.
17.	633	A message is not associated with this halt. Reason: The remaining sectors in the last disk storage area (defined in the RDLIN card containing LSTOT in columns 1-5) cannot contain the remainder of the output file.	Use the non-process run-out key on the 1402 to remove the cards from the reader. Remove the last three cards from the normal read stacker and place these cards along with another RDLIN card containing the limits of a new disk storage output area in front of the card deck that was removed from the read hopper before the non-process run-out key was pressed. Place the deck in the read hopper, and then press the start key to continue processing.
18.	644	A message is not associated with this halt. Reason: End-of-reel has been reached on the output tape and label processing is not specified.	Mount another reel of tape on tape unit 3 and press the start key to continue processing.
19.	655	A message is not associated with this halt. Reason: This halt normally occurs before a disk output RDLIN card is read. This permits the user to make certain that the correct disk pack is mounted on the output disk storage drive. However, if core storage location 20 contains an N when the halt occurs, the output RDLIN card has already been read. The RDLIN card information is in the card read area in core storage. If the user desires to alter the RDLIN constants, he can do so by manual console operation.	Press the start key to continue processing if the correct output disk pack is mounted on the output disk storage drive. Otherwise, mount the correct disk pack on the output drive and press the start key to continue processing. If the last RDLIN card has been processed, insert an additional RDLIN card with an L punched in column 1 and press the start key to continue processing.
20.	677	A message is not associated with this halt Reason: A tape-read error was detected 100 times when repositioning the tape during a phase 6 restart.	Press the start key to retry the read operation 100 times.
21.	699	Two possible messages can be associated with this halt. 1. RECORDS PROCESSED XXXXXX LAST OUTPUT SECTOR XXXXXX END OF SORT *6* Reason: This message is printed at the completion of the sort run when disk storage output is specified. The message gives the number of records processed during phase 6 and the address of the last output record. 2. RECORDS PROCESSED XXXXXX END OF SORT *6* Reason: This message is printed at the completion of the sort run when tape output is specified. The message gives the number of records processed during phase 6.	
22.	777	A message is not associated with this halt. Reason: The portion of a pack specified for a random disk output file contains active records. This applies to labeled files only and is detected by the dates in the header label.	Mount a different disk pack and press the start key twice to recheck all header labels. Press the start key, the start reset key, and then the start key to use the original pack without altering. This automatically deletes the data in the file identification field of the header label.
23.	888	A message is not associated with this halt. Reason: The pack serial number in a labeled disk output file does not check.	Press the start key, the start reset key, and then the start key to write the disk header label with this condition. Press the start key twice to recheck all header labels.

Figure 35. Phase 6 Halts and Messages (Part 2 of 2)

Appendix

This section illustrates the preparation for execution of the Sort 63 program which has been generated for use on a 4K 1401 system that has the Direct Seek feature.

Assume that:

1. Two disk drives (0 and 2) are available.
2. The input is from tape (high-density).
3. The input tape contains standard Type B (80-character) labels.
4. The format of each record is as shown in Figure 36. CDF indicates the control-data field.
5. The number of records to be processed equals the maximum file size (MFS) rounded low to the nearest thousand.
6. Records are to be selected on the basis of a controlling character (transaction code).
7. Phase-6 output is desired.
8. The work area is to overlap the output area (Figure 37).
9. The records are in an ascending sequence.
10. The records are 80-character records.
11. The blocking input factor is five.

Disk Storage Requirements

The sector requirements for the input, work, and output areas must be calculated using the formulas discussed under *File Area Definition*. The calculations for the sample Sort 63 application are shown in the following subsections. A summary of the sector requirements and the disk address used to define the area is also shown (see Figure 38).

Maximum File Size

Calculate the maximum file size (MFS) and round low to the nearest thousand to determine the number of records to be sorted. Use the result to calculate the sector requirements for the input, work, and output areas. Before solving the MFS equation, the values of Vmax, NCNT, Vtape, Z, and T must be calculated.

$$\text{Vmax} = \left\lfloor \frac{\text{MS} - \text{Bmax} - \text{UA1} - \text{PS1} - 23}{100} \right\rfloor$$

MS = 4000 (object machine size)

Bmax = 400 (maximum sort block size for 4K)

UA1 = 0 (phase 1 user area)

PS1 = 3171 (phase-1 program size determined from columns 23-25 of the next to last card in the phase-1 program deck. See the Sort 63 program listing. Columns 23-25 contain the phase-1 program size when card or tape input and deletion or selection by class are specified in control card 1).

$$\text{Vmax} = \left\lfloor \frac{4000 - 400 - 3171 - 23}{100} \right\rfloor$$

$$= \left\lfloor \frac{4000 - 3594}{100} \right\rfloor$$

$$= \left\lfloor \frac{406}{100} \right\rfloor$$

= 4 sectors (maximum number of sectors that can be written by phase 1 during one disk-write operation)

$$\text{NCNT} = \left\lfloor \frac{(\text{Vmax})\ (100)}{(\text{BI})\ (\text{L})} \right\rfloor$$

BI = 5 (input blocking factor)

L = 80 (input record length)

$$\text{NCNT} = \left\lfloor \frac{(4)\ (100)}{(5)\ (80)} \right\rfloor$$

= 1 tape block (maximum number of input record blocks that can be contained in core storage at one time)

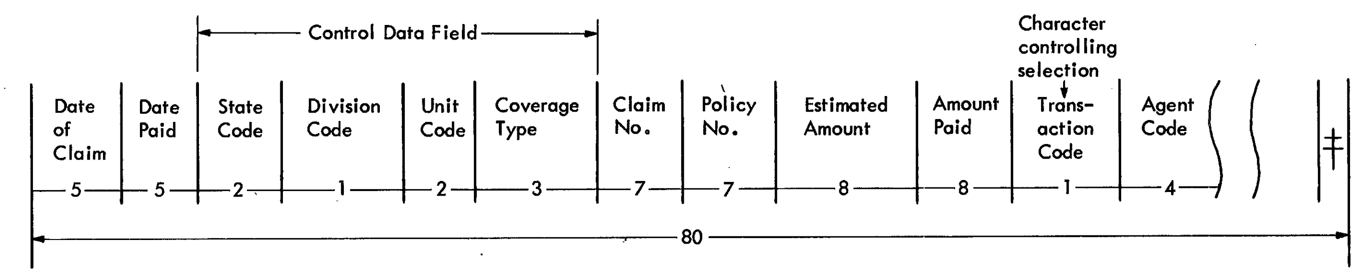

Figure 36. Record Format

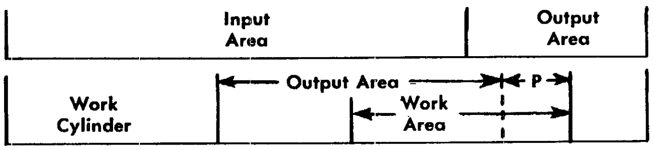

Figure 37. Two Drives On-Line, Terminating in Phase 6

(NCNT) (BI) $\leq$ 100; therefore, the formula for Vtape is:

$$\text{Vtape} = \left\lceil \frac{(\text{NCNT})\ (\text{BI})\ (\text{L})}{100} \right\rceil$$

$$= \left\lceil \frac{(1)\ (5)\ (80)}{100} \right\rceil$$

= 4 sectors (number of sectors that will be written by phase and during one disk-write operation)

(NCNT) (BI) $\leq$ 100; therefore, the formula for T is:

$$\text{T} = \left\lceil \frac{100\ \text{Vtape}}{\text{NCNT} \times \text{BI}} \right\rceil$$

$$= \left\lceil \frac{(100)\ (4)}{(1)\ (5)} \right\rceil$$

$$= \left\lceil \frac{400}{5} \right\rceil$$

= 80 characters (number of characters required for each input record)

$$\text{Z} = \text{the greater of } \left\lceil \frac{\text{BOL}}{\text{BO}} \right\rceil \text{ or } \left\lceil \frac{200\text{H}}{\left\lfloor \frac{\text{A}}{\text{CW}} \right\rfloor} \right\rceil$$

$$\text{Z} = \left\lceil \frac{\text{BOL}}{\text{BO}} \right\rceil$$

BOL = 400 (number of character per output block)

BO = 5 (output blocking factor)

$$\text{Z} = \frac{400}{5}$$

= 80 characters (number of characters required for each output record)

$$\text{Z} = \left\lceil \frac{200\text{H}}{\left\lfloor \frac{\text{A}}{\text{CW}} \right\rfloor} \right\rceil$$

Area	Sectors Required	Disk Addresses
Input	19,204	000000-019203
Work Area	8,118	030620-038799
Work Cylinder	200	020000-020199
Output	19,208	019204-019999 020200-038799

Figure 38. Sector Requirements and Disk Addresses

A = 399 (see Figure 18)

H = 4 (see Figure 18)

CW = 16 characters (number of characters in each tag: 8 control-data characters plus the 8-character disk address)

$$\text{Z} = \left\lceil \frac{(200)\ (4)}{\left\lfloor \frac{399}{16} \right\rfloor} \right\rceil$$

$$= \left\lceil \frac{800}{24} \right\rceil$$

= 34 characters (number of characters required to process each tag)

80 > 34 therefore, Z = 80

$$\text{MFS} = \left\lfloor \frac{(2 \times 10^6 \times \text{D}) - \text{Q} - 100\text{P}}{\text{Z} + \text{T}} \right\rfloor$$

2×10^6 = Number of characters per disk pack

D = 2 (number of disk drives available)

$\text{Q} = 2 \times 10^4$ = Number of characters required for the work cylinder

P = 118 (see Figure 18)

Z = 80 (number of characters per output record)

T = 80 (number of characters per input record)

$$\text{MFS} = \left\lfloor \frac{(2 \times 10^6 \times 2) - 2 \times 10^4 - 11800}{80 + 80} \right\rfloor$$

$$= \left\lfloor \frac{4 \times 10^6 - 318 \times 10^2}{160} \right\rfloor$$

$$= \left\lfloor \frac{3968200}{160} \right\rfloor$$

= 24,801 records

Input Area

Using 24,000 as the number of records to be sorted, calculate the number of sectors required for the disk input area:

$$\text{NS} = \left\lceil \frac{\text{N} + 1}{\left\lfloor \frac{100\text{V}}{\text{L}} \right\rfloor} \right\rceil \times \text{V}$$

N = 24,000 (number of records)

V = 4 (value of Vtape)

L = 80 (input record length)

$$\text{NS} = \left\lceil \frac{24{,}001}{\left\lfloor \frac{(100)\ (4)}{80} \right\rfloor} \right\rceil \times 4$$

$$= \left\lceil \frac{24{,}001}{5} \right\rceil \times 4$$

$= 4{,}801 \times 4$

$= 19{,}204$ sectors

Work Area

To calculate the number of sectors required for the work area:

$$W = 2H \left\lceil \frac{N}{\left\lfloor \frac{A}{CW} \right\rfloor} \right\rceil + P$$

H = 4 (see Figure 18)

N = 24,000 (number of records to be sorted)

A = 399 (see Figure 18)

CW = 16 (number of characters in each tag)

P = 118 (see Figure 18)

$$W = (2)\ (4) \left\lceil \frac{24{,}000}{\left\lfloor \frac{399}{16} \right\rfloor} \right\rceil + 118$$

$$= 8 \left\lceil \frac{24{,}000}{24} \right\rceil + 118$$

= 8,118 sectors

Output Area

To calculate the number of sectors required for the disk output area:

NS = the greater of Equation 1 or Equation 2

Note: If more than one RDLIN card is to be used to define the output area, add BOS for each RDLIN card except the last to allow for a 1EOR trailer label.

Equation 1

$$NS = \left\lceil \frac{N + 1}{BO} \right\rceil \times BOS$$

N = 24,000 (number of records to be sorted)

BO = 5 (output blocking factor)

BOS = 4 (number of sectors per output block)

$$NS = \left\lceil \frac{24{,}001}{5} \right\rceil \times 4$$

$= 4{,}801 \times 4$

= 19,204 sectors (number of sectors required for the output area)

Equation 2

$$NS = \left(\left\lfloor \frac{20{,}000}{LO} \right\rfloor \times \left\lceil \frac{BOS}{BO} \right\rceil \right) + BOS$$

LO = 80 (output record length)

BOS = 4 (number of sectors per output block)

BO = 5 (output blocking factor)

$$NS = \left(\left\lfloor \frac{20{,}000}{80} \right\rfloor \times \left\lceil \frac{4}{5} \right\rceil \right) + 4$$

= 254 sectors (minimum number of sectors that can be defined by an output disk RDLIN card)

$19{,}204 > 254$, therefore, NS = 19,204 sectors

One BOS must be added to NS because two RDLIN cards define the output area (see Figure 40). The additional BOS is required for the 1EORb trailer label which will be written in the first record location in the additional block. Therefore, the total number of sectors required for the output area is 19,208.

Disk Addresses

Figure 37 shows the optimum placement of the required disk areas when two drives are on-line and the input and output areas cannot be placed on one pack. Figure 38 shows the addresses that define the limits of the disk areas. Note that:

1. The high limit of the work area is of the form 0*xxx*99.
2. The low limit of the work area is a multiple of 4. If the object machine has only 4,000 positions of core storage, the low limit must be a multiple of 4.
3. The work cylinder is exactly 200 sectors. (It must be a single cylinder, not extended from one cylinder to the next.)
4. BOS (4) is added to the result of output-area equation 1 to allow for the 1EORb trailer label.

Input RDLIN Cards

Assume that the two disk packs to be used contain an expired multi-pack file. The coding for the input area, work area and work cylinder RDLIN cards is shown in Figure 39.

Assume that full label checking is to be performed on the standard 80-character input-tape header label. The specifications for the input tape RDLIN card are:

Punch	*Columns*	*Indicate*
	1-15	Blank
RDLIN	16-20	
13762	21-25	File serial number
–	26	Minus sign (11-punch)
001	27-29	Reel sequence number
	30	Blank

Punch	*Columns*	*Indicate*
QUARTERLY	31-40	File identification
64274	41-45	Creation date (September 30, 1964)
–	46	Minus sign (11-punch)
030	47-49	Retention period
	50-80	Blank

Figure 40 shows the coding for the input-tape RDLIN card.

Output RDLIN Cards

The output area is to be placed on two drives. Therefore, two output RDLIN cards are required. The specifications for the cards are shown in Figure 41.

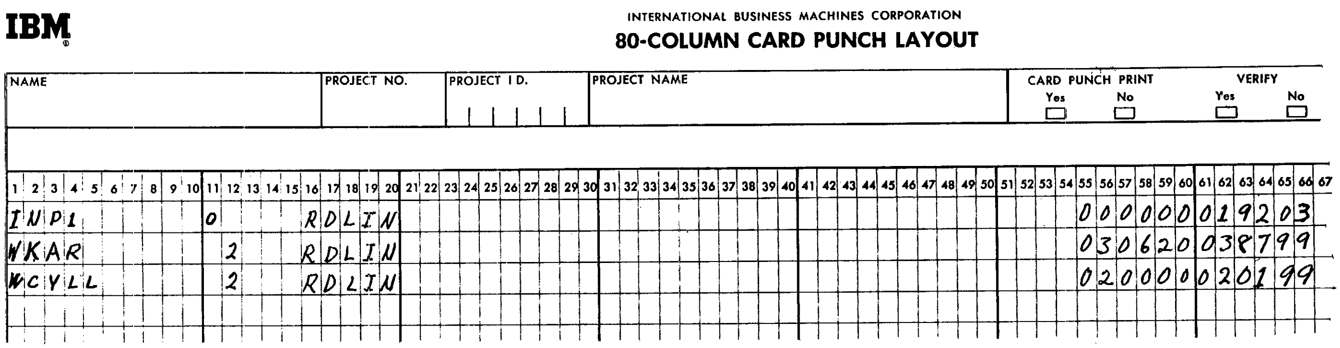

Figure 39. Input Disk RDLIN Cards

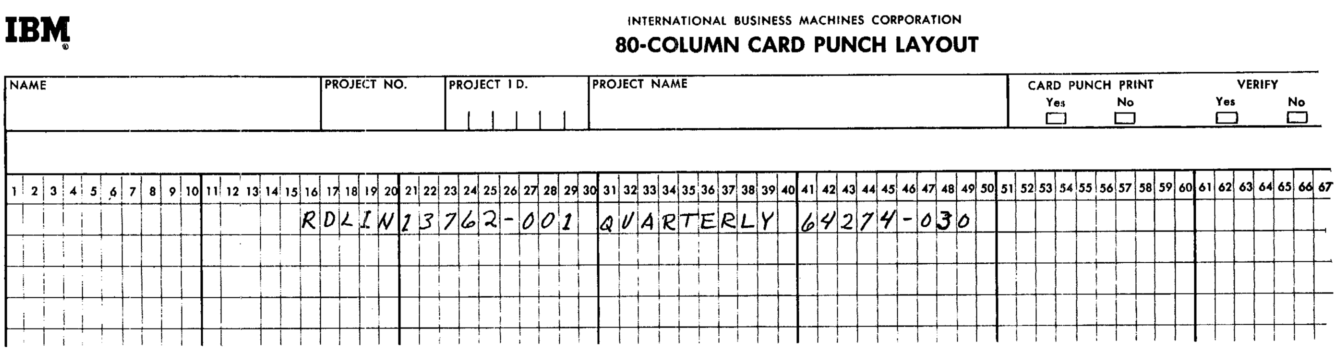

Figure 40. Input Tape RDLIN Card

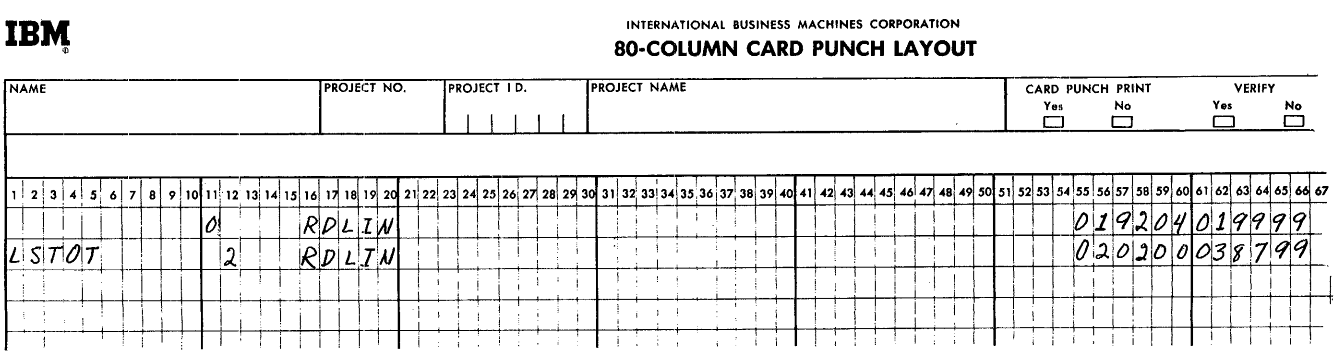

Figure 41. Output Disk RDLIN Cards

Control Cards

The control card specifications for the sample Sort 63 application are as follows:

Control Card 1

Punch	*Columns*	*Indicate*
2	1	Input media
D	2	Output media
005	3-5	Input blocking factor (BI)
005	6-8	Output blocking factor (BO)
	9-10	Number of sectors per input block (BIS)
04	11-12	Number of sectors per output block (BOS)
0080	13-16	Input record length (L)
1	17	Machine size (MS)
0	18	Ascending Sequence
0	19	Deletions by class in phase 1
1	20	Selections by class in phase 1
	21	Character controlling selection or deletion
0049	22-25	Position of controlling character in record
0	26	Card additions
0	27	Deletion by control data in phase 4
0000	28-31	Number of low-order characters to be deleted from each record
0	32	Type of card input or additions
01	33-34	Number of tape reels in the input file
1	35	Sequence check and hash total
0	36	Record marks in the output file
0	37	Restart run
	38-42	Starting address of phase-1 user area
	43-47	Starting address of phase-4 user area
	48-52	Starting address of phase-6 user area
0	53	Output desired
	54	Location of phase-4 output.
	55-66	Blank
0	67	Input disk labels
0	68	Output disk labels
1	69	Input tape labels
0	70	Output tape labels
	71-78	Blank
1	79	Control card number
	80	Blank

Control Card 2

Punch	*Columns*	*Indicate*
01	1-2	Total number of control-data fields per record.
008	3-5	Total number of control-data characters per record
0011	6-9	Control-data field 1 location
008	10-12	Control-data field 1 length
	13-75	Blank because only one control-data field is to be used.
	76-78	Blank
2	79	Control card number
	80	Blank

Glossary

To clarify the meaning of terms and abbreviations used in this publication, the following definitions are given. Standard terms are defined in *Glossary for Information Processing,* Form C20-8089.

Autocoder Control Cards. Cards that specify operations to be performed by the Autocoder system.

Autocoder System. A set of disk-resident programs that can assemble and/or execute user programs. The Autocoder system also maintains and uses the Autocoder library.

BI. Input blocking factor.

BIS. Number of sectors required for an input block.

Block. A group of records handled as one unit.

Blocking. Combining two or more records into a block.

Blocking Factor. Number of records per block.

BO. Output blocking factor.

BOS. Number of sectors required for an output block.

Control Data. Those characters in each record that determines the position of the records or its address in the sorted output.

Generation Job. Those operations relating to the generation of a generalized sort object program.

Job. A series of operations performed by a machine-language program.

Parameter Cards. Cards that supply the generative sort program (Sort 6) with the general characteristics of the sort application.

Phase. A section of a machine-language program.

RDLIN (read label information) cards. Cards that supply the sort object program with label information and/or limits of disk files.

Sort Control Cards. Cards that supply the sort object program with the specific characteristics of the sort application.

Sort Job. Those operations relating to the sorting of card, disk, or tape files.

Stack. A set of one or more jobs that are to be performed during the same machine run.

Index

E026602MTC

READER'S COMMENT FORM

Sort Programs for IBM 1401-1311 and 1460-1311
Specifications and Operating Procedures
Generative Program: Sort 6
Object Programs: Sorts 61, 62, 63, and 64

Form C24-1420-5

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. Comments and suggestions become the property of IBM.

	Yes	No
• Does this publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

- What is your occupation? ____________________.
- How do you use this publication?

As an introduction to the subject?	☐	As an instructor in a class?	☐
For advanced knowledge of the subject?	☐	As a student in a class?	☐
For information about operating procedures?	☐	As a reference manual?	☐

Other ____________________.

- Please give specific page and line references with your comments when appropriate. If you wish a reply, be sure to include your name and address.

COMMENTS:

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

C24-1420-5

fold

fold

FIRST CLASS
PERMIT NO. 387
ROCHESTER, MINN.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
Systems Development Division
Development Laboratory
Rochester, Minnesota 55901

Attention: Programming Publications, Dept. 425

fold

fold

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

1401-1311 and 1460-1311 Printed in U.S.A. C24-1420-5

1401-1311 and 1460-1311 Printed in U.S.A. C24-1420-5

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]