CENTRALIZED AUTOMATIC REPORTING ON TRUNKS (CAROT)
CAROT 2/GENERIC 2
REPORT PROGRAM GENERATOR (RPG)

CONTENTS PAGE

1. INTRODUCTION

1.01 This section describes the report program generator (RPG) programming language used in CAROT 2, generic 2, software. Reference to this programming language will be via RPG and RPG/CAROT abbreviations. The purpose of RPG is to provide a high-level programming language capability which can be used to generate, with minimum effort, summary reports of information contained in the CAROT data base. RPG/CAROT is a subset of the full RPG and was developed specifically for CAROT applications.

1.02 Whenever this section is reissued, the reason for reissue will be presented in this paragraph.

1.03 Before reading this section, familiarity with the CAROT text editor and select program (Section 190-102-206) is recommended. Use of the text editor in program development is to provide the capability to input new RPG source programs and to easily make changes to existing source programs. The select program is used to access data to be used by the RPG program. The data should be accessed and stored in a peripheral or disc file prior to execution of an RPG program.

Printed in U.S.A.

1.04 There are many new terms and concepts in this section to be learned by those who have never been introduced to the RPG programming language. Because of this and the fact that many of these terms are interrelated, the following method of reading this section is suggested:

(1) Read Parts 2 and 3 for grasp of terminology and concepts.

(2) Read Part 4, in detail, to understand the sequence an RPG program goes through during execution.

(3) Reread, in detail, Parts 2 and 3.

(4) Read the remaining parts.

2. PROGRAMMING TERMINOLOGY

2.01 A program written in the RPG specification is not executable by the computer until it has been translated by another program called the RPG compiler. Programs written in RPG are called ***source programs.*** When a program is translated by the RPG compiler, tables which describe the program function are generated. The contents of these tables are not directly executable by the computer but do describe the function specified by the associated RPG source program. When an RPG program is to be executed, the RPG compiler references these tables for the functions to be performed. To avoid introduction of confusing terminology in subsequent discussions, these tables will be referred to as the ***object program which contains the object code.***

2.02 Input to RPG/CAROT will usually be from records accessed from the CAROT data base. Data from the data base is normally accessed via the select program. The select program can store the accessed data in a disc file or on a peripheral device for later input to RPG/CAROT. Nearly all records inputted to RPG/CAROT will be 8Ø characters in length since they normally originate in the CAROT data base. A record may contain specific information in certain positions within the record. The various positions can be designated a field and referred to with a ***field name.*** For example, the YG data base record contains the OFFICE Z name in columns 33 through 43. RPG/CAROT provides the ability to designate columns 33 through 43 as a field and to assign a field name such as ABC for referencing the contents of those columns.

2.03 In our example, columns 33 through 43 may not contain the same type of information for each record inputted. To distinguish between records, a way must be provided for the RPG program to identify the type of record. This is accomplished via ***record identification codes.*** For example, instruct RPG/CAROT to examine the first three columns of each record to determine record type. If /YG is in those columns, then the program can be instructed to treat columns 33 through 43 as an OFFICE Z field.

2.04 An RPG source program is normally prepared on five special work sheets which are designed to help organize different parts of the program for ease of entry. These five work sheets contain seven specifications which completely describe the program to be entered. These specifications are as follows:

Control Record Specification

This specification is used to specify general-type information such as the debug options, output device for listing the source program when the program is compiled, initial state of the user indicators (U1 through U8), and information pertaining to storage and execution.

File Description Specification

This specification contains a description of each input and output file associated with the program. Information such as file name, record length, and location of the file (disc, magnetic tape, etc) are entered.

File Extension Specification

This specification is used to define arrays; ie, name, number of elements, and length of each entry.

Line Counter Specification

This specification is used to identify information for line printer files. This information consists of the name of the file, the first line for printing, and the last line on which to print.

Input Specification

The input specification contains information pertaining to each input file. This information consists of the file name, number of records in the file, record identification codes and indicators, and a description of fields within the records.

Calculation Specification

This specification is used to describe all data manipulation, arithmetic, and branching operations required. Operations to be performed can be made dependent upon the state of various indicators.

Output Specification

This specification is used to describe the position of various fields in the output file. Indicators may be assigned to be tested before various data can be outputted by the object program.

2.05 RPG/CAROT has several types of ***indicators.*** An indicator is a software switch that can be set either on or off to condition various operations. At various points in the program, the programmer can specify that certain indicators are to be tested. If the test is satisfied, an operation or operations are performed. Indicators assigned to record types are turned off at the beginning of each program cycle. Each time the program reads a record, it turns on the indicator assigned to that record type. Then, all operations conditioned by this indicator to take place are performed.

2.06 There are seven types of indicators which are described as follows:

General Indicators (Ø1 through 99)

These indicators are entered as a pair of digits from Ø1 through 99. These are assigned to identify a specific record type in a file that contains two or more types. Whenever the program reads a record associated with a general indicator, it turns the indicator on and then performs all operations conditioned by the indicator.

Control-Level Indicators (L1 through L9)

These indicators are entered as L1 through L9. Control-level indicators are used to cause appropriate action to be taken when a ***control break*** occurs. A control break is a condition which occurs when the contents of a field in an input record differs from the previous record's field. For example, assume there is a file of cities associated by state and population, and it is desired to output by state the total population of the cities. Assume, also, that the cities associated with each state are consecutive and are not intermixed in the file. In this example, a control-level indicator could be assigned to the state field to indicate when the contents of the state field changes. A condition known as a control break will occur when the contents of the state field changes, and the assigned control-level indicator will turn on. The program could be arranged to keep a total of the population until a control break occurs, at which time, the total associated with that state could be outputted. Unlike general indicators, control-level indicators that are on at any point are turned off before the next record is read.

Last-Record Indicator (LR)

This indicator is entered as LR. The LR indicator is used to identify the last record to be read by the program. When this record is inputted, calculations conditioned by this indicator are performed, output completed, and then the program terminates.

Overflow Indicators (OA through OG, and OV)

These eight indicators are entered as OA through OG, and OV. Only one overflow indicator may be assigned to each printed output file named in the file description specification. The programmer specifies on the line counter specification which line on the printed page is to be the overflow (last) line. When the last line is encountered, the assigned overflow indicator turns on, conditioning certain parts of the program to output data such as totals for the page. A paper page in the HP 2767A printer contains 66 lines. This printer is mechanically set up to print only on 6Ø lines leaving a small margin at the top and bottom. Thus, the overflow line should be before 6Ø; for example, 58.

First-Page Indicator (1P)

This indicator is entered as 1P. It is turned on for conditioning the program to print first-page headings. After the headings are printed, this indicator can be used as a general indicator.

Halt Indicator (HØ through H9)

These indicators are entered as HØ through H9. When one of the H1 through H9 indicators turns on, a message will be outputted on the system console, and the program will halt at the end of the current detail cycle. The program turns the indicator on when a record type to which it is assigned is encountered.

When the HØ (debug) indicator turns on, the debug features selected on the control record specification are performed and then the program halts.

User-Indicators (U1 through U8)

The user-indicators are entered as U1 through U8. These indications may be used to condition various operations such as the general indicators. The state of the user-indicators before program execution may be specified on the control record specification.

3. WORK-SHEET PREPARATION

3.01 Before writing an RPG source program, the rules for making entries on the specification work sheets must be learned. Information that is applicable to all work sheets appears in Part 3A; while information unique to individual work sheets is described separately in Parts 3B through 3H.

3.02 Each part (3B through 3H) will reference a specific figure. References to specific parts of the figure will be via an item number. Each item number is associated with a callout in the figure. When numeric data (data consisting of numbers Ø through 9 to be used for arithmetic purposes) is to be entered, it should, unless otherwise noted, be right-justified in the field on the work sheet. Leading zeros need not be entered. Conversely, data that is not to be used for arithmetic purposes, such as the name of a file, should be entered left-justified. The blank fields on the work sheets are not used in the CAROT version of RPG, although they may be required or supported by other RPGs. Entries in these blank fields, except for comment lines, will result in the RPG/CAROT compiler issuing warning messages.

A. Common Entries—Reference Fig. 1

Items 1 through 5

3.03 These items contain spaces for entering the page number of the current sheet, the total number of sheets making up the program, the programmer's name, title of the program, and the current date.

Item 6—Sequence Number

3.04 Columns 1 through 5 may be used to assign an optional sequence number to the line. Leading zeros may be omitted. Sequence numbers are checked for valid integer values, but are not used by the RPG/CAROT compiler.

Items 7 and 8—Form Type

3.05 This field denotes the type of specification for which information is currently being entered. The entry is preprinted for convenience on each specification sheet as follows:

Column 6 Entry	Specification
H	Control record
F	File description
E	File extension
L	Line counter
I	Input
C	Calculation
O	Output

Item 9—Comment

3.06 An asterisk in column 6 (form type) or 7 indicates that the line contains comments or notations that are to be ignored by the RPG compiler. Comments are used by the programmer as an aid in understanding the program. If an asterisk is entered in column 7, do not enter one in column 6. Further, an asterisk in column 7 always indicates a comment even if a form type is entered in column 6.

Item 10—Program Name

3.07 The program name field appears only once on each work sheet. The program name field in the control record specification is used to identify the ***source listing*** and to provide a file name for the associated ***object program.*** This program name on the control record specification may contain up to six characters, beginning with a letter (A through Z) while the remaining characters may be letters, numerals (Ø through 9), or apostrophes ('). Embedded blanks are not allowed. If the program name field on the control record specification is left blank, the name RPGOBJ will be assigned by default.

3.08 The program name field on the other specifications is not assigned to any part of the object program and, other than appearing on the source listing, has no meaning. For this reason, the program name on these specifications may consist of any characters the programmer desires, or it may be left blank.

B. Control Record Specification—Reference Fig. 2

Item 1—Debug Options

3.09 If column 17 contains a 1, each input record accessed is outputted.

Item 2—Output Device for Source Listing

3.10 If a source listing output of the RPG program is desired, enter in columns 2Ø through 27 the key word representing the output service. If no source listing is desired, enter BB. Permissible key-word entries are as follows:

Key Word	Meaning
LP	Line printer
TE	User terminal
BB	Bit bucket (ie, no listing)

Item 3—External Indicators

3.11 The user indicators U1 through U8 may be conditioned when the RPG object program is executed. To turn on a user indicator, enter a "1" in the appropriate position in columns 28 through 35. To turn a user indicator off, leave the appropriate position blank.

Item 4—Execute and Save

3.12 An entry of "S" in column 45 is used to specify that the object code generated as a result of the compiling process is to be saved in a disc file. The name of the file is given in columns 75 through 8Ø of the control record specification. The reason for saving this code is to bypass the time required to compile the code each time the program is to be executed. In this way, a previously compiled program can be executed by using the object code saved in the file, thus eliminating the time spent for compiling. The time saved can range from a few seconds to several minutes, depending on the demand of the processor for other CAROT functions.

3.13 An entry of "E" in column 44 signifies that the previously compiled object code contained in the file specified in columns 75 through 8Ø of the control record specification is to be executed. This, as previously mentioned, bypasses the compile process. If an entry is made in column 44 or 45, the other should be left blank. If an "E" is entered in column 44, the control record specification should be the only record in the file. Thus, there are three files to consider:

(1) Object code file

(2) Source file containing the complete RPG program

(3) Source file containing a control record specification with an "E" entered in column 44.

3.14 Both source files are generated using the CAROT text editor while the object code file is generated by the RPG/CAROT compiler. A convention for naming these files should be established in order to keep track of their contents. A recommended convention is as follows:

Name—This is the name of the file containing the object code. This name is specified in columns 75 through 8Ø of the control record specification.

&Name—Begin the name of the file containing the RPG source program with an ampersand. This name is specified when saving a file using the text editor.

.Name—Begin the name of the file containing the control record specification which has an entry of "E" in column 44 with a period.

3.15 Examples of file names are OPSUM, **&**OPSUM, and .OPSUM. To compile and execute or just execute an RPG program, type **RUN:RPG** followed with a carriage return. Next, the computer outputs a message requesting the source file name;

enter **&Name** to compile and execute, or enter **.Name** to execute using previously compiled code.

C. File Description Specification—Reference Fig. 2

3.16 An entry for each input and output file must be made on this specification.

Item 5—File Name

3.17 A unique name must be assigned to each file used by the program. This name should be entered in columns 7 through 14. The file name may consist of one to eight characters. Only the first six characters are used, so they must be unique. The first character of a file name should begin with a letter (A through Z), and the remaining characters may be letters or numbers. Embedded blanks are not allowed.

Item 6—File Type

3.18 Enter I or O as follows:

Column 15	Meaning
I	Define the file as input data for processing.
O	Define the file as output data to be used to store processed data.

Item 7—File Designation

3.19 If the file is an input, it must be further defined as a primary or secondary file in column 16 as follows:

Column	Meaning
P	Primary file from which programs read their input. Only one primary file per program is allowed. During multifile processing, a program uses its primary file to control the order in which records from all files are processed. In programs that read records from only one file, that file is usually a primary file. With multifile processing and no record matching operations, the RPG program will process its primary file first.
S	Secondary file used in multiple operation. There can be any number of these files. The primary file is first processed followed by the secondary files, in the order in which they are specified on the file description specification sheet, until processing is complete.

3.20 If the file is an output, no entry is required in column 16.

Item 8—End of File

3.21 If the file is an input, column 17 can be used to specify whether the program can terminate (or not terminate) before all records are read from the file. No entry is required for output files. Column 17 entries are as follows:

Column 17	Meaning
E	The program cannot end until it reads all records from this file.
Blank	If this field contains "E" for any other files, the program can end whether or not it reads all records from this file. If this field is blank for all input files, the program cannot end until it reads all records from all of the input files.

3.22 Note that at least one file must be read. Normally, all input would be processed before terminating. However, if this is not required, enter an "E" in column 17 for those files that must be fully read, and leave this field blank for all other files.

Item 9—Record Format

3.23 Enter "F" (for fixed length) in column 19 if all records in the file are the same length. Conversely, enter "V" (for variable length) if the records in the file can be of varying lengths.

Item 10—Record Length

3.24 For fixed-length records, enter the length (1 through 9999) of each record in columns

24 through 27. If the records are variable lengths, enter the length of the longest record. All records accessed from the CAROT data base will be 8Ø characters in length. Normally, the length of output records to the line printer will be 8Ø characters.

Item 11—Overflow Indicator Assigned to Printer

3.25 An overflow indicator (OA through OG, and OV) may be entered in columns 33 through 34 for output files to the printer. This indicator will turn on when the overflow (last) line specified in the line counter specification is reached. Other statements can test this indicator to print total-type information at the bottom of the page before skipping to the top of the next page. Only one overflow indicator may be assigned to a file, and this indicator cannot be associated with any other file used by the program.

> ***Note:*** If an overflow indicator is assigned, a corresponding entry must be made in the line counter specification.

Item 12—Extension

3.26 If the file is an output to the printer and an overflow indicator has been assigned, an "L" should be entered in column 39 to indicate that a line counter specification exists which defines the length of the printed page. If no line counter specification exists leave column 39 blank.

Item 13—Device Name

3.27 Columns 4Ø through 46 are used for the entry of a key word which represents the storage location of the file. The following are valid entries:

Key-Word Entry in Columns 4Ø-46	Meaning
DI	Disc drive
LP	Line printer
PR	Paper tape reader
CR	Card reader
PU	Paper tape punch
TE	User terminal
MT	Magentic tape drive

Item 14—Append File

3.28 Column 66 is used to specify whether new records outputted to an existing file are to be written at the beginning of the file (overwriting any information already there), or are appended to the end of the file following any previously written information. Enter "A" in column 66 to append new records at the end of the file, or leave blank to write new records at the beginning of the file.

D. File Extension Specification—Reference Fig. 3

3.29 The purpose of this specification in RPG/CAROT is to define arrays. If the program does not contain an array, no entry on this specification is required.

Item 1—Array Name

3.30 Columns 27 through 32 are used to assign a name to an array. An array name consists of one to six characters beginning with a letter (A through Z) or @, $, or #. The remaining characters can be letters, digits, or @, $, or #. Embedded blanks are not allowed.

Item 2—Number of Elements

3.31 Enter the maximum number (1 through 9999) of elements (or entries) in the array in columns 36 through 39. If no entry is made in these columns, the number of elements will default to one.

Item 3—Length of Each Entry

3.32 Enter the length (1 through 9999) of each entry in the array in columns 4Ø through 42. If no entry is made, a default of six will occur.

Item 4—Decimal Position

3.33 Enter the number of positions (1 through 9) to the right of the decimal place in column 44. If the array is alphanumeric, leave column 44 blank.

Item 5—Comments

3.34 The programmer may enter a free-format comment in columns 58 through 74.

E. Line Counter Specification—Reference Fig. 3

3.35 An entry must be made for each printer output file which has an overflow indicator assigned.

Item 6—File Name

3.36 Columns 7 through 14 should contain the file name which references the line counter specification ("L" in column 39 of the file description specification).

Item 7—First Line on Page

3.37 Enter in columns 15 through 17 a line number (1 through 6Ø) that corresponds to the first line to be printed on each page.

> *Note:* The HP 2767A line printer prints only on 6Ø lines per page. This leaves a small margin at the top and bottom of each page. The first printed line on a page is normally designated line number 1.

Item 8—Last Line on Page

3.38 Enter in columns 2Ø through 22 the line number (1 through 6Ø) that corresponds to the last line to be printed on each page.

> *Note:* The HP 2767A line printer prints only on 6Ø lines per page. This leaves a small margin at the top and bottom of each page. The last line on the page is normally dependent on the number of lines in the total information. If two lines of total information are required, the last line should be designated line number 58.

F. Input Specification—Reference Fig. 4

3.39 An entry is required on this specification for each input file used by the RPG source program.

Item 1—File Name

3.40 Enter the name of each input file in columns 7 through 13 as it appears on the file description specification. Once the file name has been entered, it applies to all following lines until another file name appears in this field. Thus, if the input specification for a file requires more than one line, the file name does not have to be repeated. The file name may be repeated; however, all entries for this file must be described before describing those for another file.

Item 2—Group Sequence

3.41 A 2-digit (Ø1 through 99) group sequence number may be entered in columns 15 and 16. This number is entered if the RPG object program is to check, during execution, the sequence of different types of records within individual groups. Checking is done by comparing one or more specified record identification codes (see items 6 and 7) with the corresponding input data. This provides for processing individual records in a group in a specified order. For example, assume there is an input file composed of the customer name on one record and address on the following record. Assume that each customer name record contains "N" record identification code in column 1 while each address record contains an "A". Group sequence numbers could be assigned which would require the object program to check that an address always follows a name, and the next name follows the previous name's address. If the object program detects that the input does not follow this sequence, an error message will be outputted, and the program will terminate.

3.42 Group sequence numbers are only used to insure that within a group, an established sequence of record types exists. Group sequence numbers cannot be used to check the order of records with the same identification codes, or to check the order of records according to data field contents. Always assign the lowest sequence number, Ø1, to the first record type. Assign ascending numbers to the remaining records within the group.

3.43 If the file contains only one type of record or records of different types which need not be grouped in order, enter any two alphabetic characters in columns 15 and 16. In any case, ***this field must not be left blank.***

Item 3—Number of Records Per Type

3.44 If group sequencing has been specified in the previous field, enter the number of records (Ø through 9) of each type in column 17. If the number of records can vary, enter "N". Do not make an entry in this field if the specification line contains an AND or OR entry since the previous number of records entry applies. If group sequencing has not been specified, leave this field blank.

Item 4—Optional

3.45 If group sequence checking has been requested, this field may be used to indicate if the record type is optional and may or may not be present. This is done by entering the letter "O" in column 18. The entry of "O" in this column prevents a sequence error from occurring if the record of the designated type does not appear in the group. If this record must appear, leave column 18 blank.

Item 5—Record—Identifying Indicator

3.46 Any indicator described in Part 2 may be entered in columns 19 and 2Ø. The indicator will be turned on when the record type is accessed.

Item 6—Record Identification Codes

3.47 When an input contains more than one type of record and it is desired for the program to process each type in a different manner, a record identification code must be used to identify each record type. This code consists of a unique combination of characters that appears in certain positions in the input records. The specification of these codes will enable the object program to determine the type each time it reads a record. Record identification codes should not be entered if the input file contains only one type of record, or if all types are to be processed in the same manner.

3.48 Columns 21 through 41 define up to three characters which make up the record identification code. This field defines the character and position within each record type. A logical not operator ("N" in column 25) indicates the absence of a character as part of the record identification. Any letter, digit, or special character which occurs in the input record may appear in columns 27, 34, and 41 to indicate the record identification characters. More than three characters may be entered to make up the record identification code by entry of the logical operators AND and OR (item 7) in columns 14 through 16 for each succeeding line.

Item 7—Logical Operators AND and OR

3.49 The logical AND operator can be used to specify a record identification code that contains more than three characters. This is performed as follows:

(a) Enter the first three characters of the record identification code as previously described.

(b) Enter AND in columns 14 through 16 of the ***next*** line, and enter up to three additional character descriptions as previously described. Leave columns 17 through 2Ø blank.

(c) Continue with as many AND lines as needed to specify all the characters in the code.

3.50 The logical OR operator can be used to specify a record type by more than one record identification code. This is performed as follows:

(a) Enter the description of the first record identification using AND operators as needed.

(b) Enter OR in columns 14 through 15 of the ***next*** line, and begin describing the next code. Use the AND operator if more than three characters are to be described. Leave columns 16 through 18 blank.

(c) Continue with as many OR lines as needed to specify all codes that identify this type of record.

3.51 The logical OR operator can also be used to assign more than one name to an input record field. This will be described later.

Item 8—Field Description

3.52 The field description (columns 44 through 7Ø) is used to identify certain areas within individual records. This is accomplished by defining the FROM and TO column numbers which represent the position of the field in the input record, and then assigning a field name. The field name can be used in the calculations specification to reference the specified portion of the record. Each field should be described singly on a line beginning at least one line below the line containing the record description. In each line containing a field description, columns 7 through 43 should be left blank.

3.53 The FROM field position (columns 44 through 47) is a column number (1 through 9999) which indicates where the field begins. The TO field position (columns 48 through 51) indicates where the field ends.

3.54 If the field contains numeric data, enter in column 52 the number of decimal positions (Ø through 9) which appear to the right of the decimal point. If the field contains alphanumeric data, leave this field blank. The number of decimal positions should not exceed the maximum number of digits that can be contained in the field. If the input data contains a quantity with a decimal point, it overrides the specification in column 52.

3.55 The field name (columns 53 through 58) may be the name of the field, array, or an array item. This name may consist of up to six characters beginning with a letter (A through Z), while the remaining characters may be made up of alphanumeric or @, $, or # characters. Embedded blanks are not allowed. If an array item is specified, enter the array name, a comma, and an index. The index may be a number or the name of a field that contains a numeric quantity. The combination of array name, comma, and index is limited to six characters. Examples are TAB,I or J,6 or T,XX.

3.56 A name must be assigned to every field or array that the program uses. This is done so that any reference to this field or array in the program is made by this assigned name. By using an index, an individual item within an array can be referenced.

3.57 Assurance must be made that within each record type, the field names are unique. If the same name is assigned to more than one field within a record type, the RPG compiler will consider only the last described field. If this name has been defined elsewhere in the program, it must be associated with the same length, data format, and decimal position in each instance.

3.58 The same name may be assigned to fields of different record types if these fields are the same length and contain the same type of data. This is true even if the fields are in different positions in the input records. Each field must be described individually even though the same name is used. To eliminate duplicate coding, logical OR operators may be used to indicate that the fields named may appear in either record type. To do this, enter OR in columns 14 through 15 of the next line, and specify the record type on that line; enter the field description on the next line.

3.59 If an array is specified in the field name position and a name rather than a number is specified as the index, the index field must be defined as a numeric field with zero decimal positions. If the index value is to be read from the current input record, the index field on the record must be defined before attempting to use it. If the program establishes the index value through calculation operations, this must be done before the program can process the record to which the index applies.

3.60 A control-level indicator (L1 through L9) may be assigned in columns 59 through 6Ø. If a control-level indicator is not desired, leave this field blank.

3.61 When a control-level indicator is assigned to a field, the indicator identifies this field as a control field that the object program must check each time it reads a record. The program compares the control field data on this record with that on the previous record. If the control field contains identical information, the record belongs to the same control group as the previous group. But if the control field information differs, a control break occurs. In this case, the program turns on the indicator associated with this field and all control-level indicators of lower rank; then, the program performs all calculations and output operations associated with those indicators. This sequence allows the use of control-level indicators to condition the following:

- Operations to be performed when the first record of a control group is read
- Operations to be performed when all records within a control group have been read
- Operations to be output totals and subtotals.

3.62 Among the indicator designations L1 through L9, increasing digit numbers indicate increasing importance or higher level. For example, L9 ranks higher than L8, L8 higher than L7, etc. When a control break occurs, the assigned control-level indicator is turned on, and all those of lesser rank are turned on also. For example, if ROTL office field is assigned the control-level indicator L3 and the contents of the ROTL office field changed, then indicator L3 plus L2 and L1 will turn on. Thus, the importance of data in one control field to that in others should determine what indicator to assign to a field. Control-level indicators may be assigned in any sequence. For example, L5, L1, and L7 can be assigned in that order without wiring the remaining indicators.

3.63 The last-record (LR) indicator turns on when the object program reads the last record in the file. This is the highest level indicator available in RPG/CAROT. When it turns on, indicators L1 through L9 also turn on.

3.64 LØ is an additional control-level indicator that is always set on. It cannot be assigned to a control field, but it may be used to condition calculation operations.

3.65 A test may be performed of the data inputted in a field. This is done by assigning an indicator in an appropriate position in columns 65 through 7Ø. Any indicator described in Part 2 may be specified. Dependent on the position in which the indicator is entered, a test will be performed to determine if the data is positive (indicator in columns 65 through 66), negative (columns 67 through 68), zeros, or blank (columns 69 through 7Ø). The specified indicator will turn on if the test is affirmative; it will turn off if the test is not affirmative. Various indicators may be assigned for diverse reasons. For example, if a field should not contain a negative quantity, a halt indicator may be assigned. This will cause the program to terminate if a negative quantity was inputted in the field.

3.66 When a program tests a field for zeros, the indicator specified in columns 69 through 7Ø turns on if the field contain either zeros or blanks. However, if the program tests an alphanumeric field for blanks, the indicator turns on only if the field contains blanks, while zeros do not affect the indicator.

3.67 There are two important points to consider:

(1) Two or three field indications may be assigned to one input field. However, only the indicator which signals the result of the test turns on; the other turn off. This is true even if a control-level indicator is specified.

(2) When different field indicators are assigned to different record types, a field indicator turned on will remain on until another record of that type is read. Similarly, if a field indicator is assigned to more than one field within a single-record type, it will always reflect the status of the last field defined.

G. Calculation Specification—Reference Fig. 5

Item 1—Control Level

3.68 The ***control-level field*** (columns 7 and 8) is used for the entry of a control-level indicator (LØ through L9), last-record indicator (LR), or entry of a logical operator (AN and OR) to establish a relationship between lines.

3.69 Entry of a control-level or last-record indicator defines the lines as a ***total calculation*** which is to be done when the indicator turns on. These indicators normally turn on on occurrence of a control break but can be turned on via a special instruction on this specification.

3.70 If no indicator is specified, the line is defined as a ***detail calculation*** which will be performed every time the program reads a record, provided that the indicators appearing in the indicator field (columns 9 through 17) permit this. In an RPG/CAROT program, detail calculations must be requested before total calculations are requested.

3.71 The LØ indicator is always on. This indicator cannot be assigned to record types or control fields, but it can still be used to condition operations. The LØ indicator is normally used when a program's input records have no control fields, but it is desired to condition calculations and total output.

3.72 When a control-level indicator turns on, all indicators of lower level also turn on. Thus, if indicators L3, L2, and L1 are associated with operations in the program and L3 turns on because of a control brake, L2 and L1 will also turn on.

All operations conditioned by these three indicators will be performed with those associated with L3 first, then L2, and finally L1.

3.73 The LR indicator is normally used to condition operations to be performed only at the end of the program. This indicator as well as L1 through L9 indicators are turned on after the last input record is processed. Then, the total calculations conditioned by L1 through L9 are performed, and the program terminates.

3.74 Up to three indicators of any type may be entered in columns 9 through 17. By entering AN or OR in columns 7 through 8, a logical AND or OR relationship can be established between lines to specify more indicators. This is performed as follows:

(a) Begin the specification by entering any control-level indicator in columns 7 through 8, or leave blank. Enter one or more indicators in columns 9 through 17, and leave columns 18 through 59 blank.

(b) Enter the appropriate AN or OR relationship in columns 7 through 8 on the next line. Enter one or more indicators in columns 9 through 17. If this is not the last line of the specification, leave columns 18 through 59 blank.

(c) Continue entering AN and OR lines as needed. A maximum of seven AN and OR lines may be entered.

(d) On the last AN or OR line, enter the calculation operation and data fields to which all indicators in the total relationship applies.

> ***Note:*** At least one indicator must appear on each line separated by AN/OR.

The indicators on each line are in a logical AND relationship. The type of entry (detail, total, or last record) for the first of a group of logically related lines specifies the type for the entire group. For a logical AND relationship, the calculation is done if all indicator conditions are satisfied. For a logical OR relationship, the calculation is performed if the condition of any of the indicator sets separated by OR are satisfied.

Item 2—Indicators

3.75 Besides the control-level indicator in columns 7 through 8, up to three more indicators of any type may be entered in columns 9 through 17. More indicators may be specified by separating each line with a logical AND or OR relationship (see paragraph 3.74). A logical NOT operation which indicates that the indicator must be off to condition the operation may be specified. This is done by entering an "N" in the appropriate 9, 12, or 15 column.

3.76 In RPG/CAROT, all operations conditioned by the control-level indicators in the ***control-level field*** are performed before those conditioned by this type indicator in the ***indicators field.***

3.77 In general, indicators specified in the indicators field can be used as follows:

- Any indicator previously specified in the ***record indicator field*** (columns 19 through 2Ø) of the input specification can be used to condition an operation to be done only for a certain type of record.

- Any indicator previously specified in the ***record indicator field*** (columns 65 through 7Ø) of the input specification can be used to condition an operation to be done only after the contents of a field has met certain conditions.

- Any overflow indicator previously specified in the ***overflow indicators field*** (columns 33 through 34) of the file description specification can be used to condition operations when an overflow occurs.

- Any halt indicators previously specified in the ***field indicators field*** (columns 65 through 7Ø) of the input specification, can be used to terminate the program when a specified error condition occurs in the input data or in previous calculations.

- Any control-level indicator specified in the ***control-level field*** (columns 59 through 6Ø of the input specification, or in the ***resulting indicators field*** (columns 54 through 59) of the calculation specification, can be used to perform an operation on only the first record of a new control group at detail time. When this is done, do not enter the indicator in the ***control-level field*** (columns 7 through 8) of the calculations specifications.

- Any indicator specified in the ***resulting indicators field*** (columns 54 through 59) of the calculations specification can be used to condition operations on the basis of prior calculations.

- The last record indicator can be used to condition operations to be done at the end of the program. If this indicator is set on via an operation in the calculations specification, it should be used in the ***indicators field***. Otherwise, if it remains off until the last record is read, it can be entered in the ***control-level field*** (columns 7 through 8).

3.78 When a control-level indicator is entered in the ***indicators field*** and not in the ***control-level field***, the operation conditioned is performed only on the record that causes a control break of this or higher level.

3.79 In RPG/CAROT, all operations conditioned by control-level indicator in the ***control-level field*** are performed before those conditioned by this type of indicator in the ***indicator field.***

Item 3—Factors, Operation, and Result

3.80 These fields (columns 18 through 53) contain the actual calculation to be performed. Remember that the calculation will be performed only if the indicators in columns 7 through 17 are properly. conditioned.

3.81 Two fields, ***factor 1*** (columns 18 through 27), and ***factor 2*** (columns 33 through 42), are used to define the data or fields to be used in the calculation. Entries in factor 1 and factor 2 may consist of the following:

ENTRY	MEANING
Name of a field, array, array element, or file	The storage area containing the data to be used, or (in the case of a file name) the file to be used
Label	The label for a TAG, ENDSR, or GOTO operation
Alphanumeric or numerical literal	The actual data to be used
Blank	No operand

3.82 Entries in factor 1 and factor 2 will depend on the operations selected. Some calculations require entries in both fields, while others require an entry in only one of the fields or none at all. Entries in these fields whether alphanumeric or purely numeric should be left-justified.

3.83 Each field name or array name specified in factor 1 and factor 2 must have been previously defined elsewhere in the program.

3.84 A ***literal*** is the actual data to be used in the calculation, rather than the name of a field containing that data. An ***alphanumeric literal*** may consist of letters, digits, and/or special characters. A ***numeric literal*** may consist of only digits (Ø through 9), a decimal point, and a leading arithmetic sign (+ or -). Embedded blanks in a numeric literal are not allowed.

3.85 Alphanumeric literals may consist of any ASCII character, including blanks. Alphanumeric literals should be enclosed with quotation marks (“) to distinguish them from field names. For instance, the literal ABCD should be entered as “ABCD”. To use a quotation mark within a literal, enter two quotation marks. For example, to specify “PAGE”, enter “””PAGE””“. Alphanumeric literals cannot be used in arithmetic operations.

3.86 A numeric literal may consist of the digits Ø through 9, and an optional decimal point and sign. Some examples of valid numeric literals are as follows:

99.
-473.163
+2

3.87 In the operations field (columns 28 through 32), the following operations may be specified:

Operation Field	Meaning
ADD	Add factors 1 and 2 and place the sum in the result field.
COMP	Compare factor 1 with factor 2, and set the appropriate resulting indicators. No entry is allowed in the result field.
DIV	Divide factor 1 by factor 2, and place the quotient in the result field.
GOTO	Branch to the label specified in factor 2. No entry is allowed in the factor 1 or result field.
MOVEL	Move the characters in factor 2 to the field specified in the result field. No entry is allowed in the factor 1 field.
MULT	Multiply factor 1 by factor 2 and place the result in the result field.
SETOF	Turn off the indicators in the resulting indicators field. No entry is allowed in the factor 1 or 2, or result fields.
SETON	Turn on the indicators in the resulting indicators field. No entry is allowed in the factor 1 or 2, or result fields.
SUB	Subtract factor 2 from factor 1 and place the difference in the result field.
TAG	Assign a label (factor 1) which can be referred to by a GOTO instruction. No entry is allowed in the factor 2 field.
XFOOT	Add each item in the array specified in factor 2 and place the sum in the result field. No entry is allowed in the factor 2 field.
Z-ADD	Replace the result field with factor 2. Factor 2 is not allowed in the factor 1 field.
Z-SUB	Replace the result field with the negative of factor 2. Factor 2 is not altered. No entry is allowed in the factor 1 field.

3.88 The result field (columns 43 through 48) is used to specify any required result of the calculation. This field may contain the name of a field, array, or indexed array element. The entry in the result field may be defined elsewhere in the program, or a new name may be used. If a new name is specified, the length of the field must be established in the field length field (columns 49 through 51). If the name of a previously defined field or the name of an array is specified, the field length field may be left blank. If an entry is made, it must be the same length as that previously defined. The entry in the field length field must be right-justified and in the range of 1 through 256. Care should be exercised to insure that the result field is long enough to hold the largest result possible from the calculation. If the field is too small, significant digits will be truncated.

3.89 If the result field is numeric, enter the number of positions (Ø through 9) to the right of the decimal in the decimal positions field (column 52). Conversely, if the result field is alphanumeric, leave this field blank. If the result is numeric and previously defined, no entry is required; but if an entry is made, it should correspond to the same as that previously described. To indicate no decimal positions for numeric data, enter Ø. The number entered in column 52 should never exceed the total length of the field.

Item 4—Resulting Indicators

3.90 This field (columns 54 through 59) can be used to specify up to three indicators to be set depending on the result of the calculation performed, or the indicators can be unconditionally set via SETON or SETOF instructions. Indicators may be entered in three areas of the resulting indicators field. If the operation performed was a numeric calculation (such as ADD, SUB, COMP, DIV, MULT, XFOOT, Z-ADD, or Z-SUB), indicators may be entered to be turned on if factor 1 is greater than factor 2 (or result is positive), if factor 1 is less than factor 2 (or result is negative), or if factor 1 is equal to factor 2 (or result is zero). For example, if a subtract operation was specified and the result was negative, then any indicator entered in columns 56 through 57 will turn on and any indicator in the other two areas will turn off.

3.91 If a general indicator is entered in the resulting indicators field, it can be used to condition subsequent calculation or output operations. A halt indicator can be used to terminate the program when an unacceptable condition occurs. If any other indicator is entered, the RPG/CAROT operating cycle discussed in Part 4 should be thoroughly understood.

Item 5—Comments

3.92 A free-format comment may be entered in columns 6Ø through 74.

H. Output Specification—Reference Fig. 6

Item 1—File Name

3.93 Enter in columns 7 through 14 the name of the disc file or the printer output file whose records and fields are to be described. The file name may consist of from one to eight characters, beginning with a letter (A through Z). The remaining characters may be letters or digits. Embedded blanks are not allowed.

3.94 The file name entered must be assigned to an output file. It should not refer to an input file. The file name should also appear on the file description specification. All entries on the line containing the file name of all subsequent lines will apply to this file until a new file name appears in the file name field. Thus, if more than one record for this file is described, there is no need to repeat the file name on subsequent record description lines. If record descriptions are interspersed with other files, the file name should be entered on each line so that the RPG compiler can determine to which file the entries pertain.

Item 2—Type

3.95 In column 15, the programmer is required to define the record being described as one of the following three types:

Column 15	Meaning
H	Heading record
D	Detail record
T	Total record

3.96 ***Heading records*** identify output report or file information that generally remains the same from page to page such as the report title, column headings, and current date. These records may also contain a page number that is to be incremented for each page. Heading records are normally conditioned by the overflow indicators.

3.97 ***Detail records*** generally contain the most basic subject matter of the report. Usually, this information is closely related to the input records in that it is either read directly from those records or calculated by the program on the basis of the data furnished by the records. Detail records are normally conditioned by the general indicators.

3.98 ***Total records*** typically are used to output sums accumulated by adding and otherwise manipulating data from one or more fields in several detail records. Total records are normally conditioned by control breaks generated by the L1 through L9 and LR indicators, but these records can be conditioned by the overflow indicators to provide totals at the end of the page.

3.99 Within each type (H, D, or T), entries are processed in the order in which they are specified in the output specification. It is suggested that entries in the output specification be made in the following order: heading, detail, and then total. For all output files, it is permissible to define all heading records, then all details, and finally all total records.

3.100 When the assigned overflow indicator for this file turns on, the program prints a line on or below the overflow line. This signals the beginning of overflow processing. If the indicator is turned on while header, detail, or total information is being written, the indicate will turn off after the header and detail records of the ***next*** cycle are written. If an overflow indicator has been assigned to the file and has never been used to condition an output record, no special processing will take place when the overflow line is reached. If an overflow indicator has not been assigned to the file, the program will cause the printer to skip to the top of the next page whenever the overflow line is reached.

3.101 Whenever an assigned overflow indicator is on, the program will begin overflow processing. Normally, this occurs when the printer reaches the overflow line. During overflow processing, the following events occur:

(a) The program writes all detail records not conditioned by the overflow indicator remaining to be outputted during the present program cycle.

(b) The program next prints all total lines not conditioned by the overflow indicator remaining to be outputted during the present program cycle.

(c) The program then prints all records conditioned by the overflow indicator.

(d) And finally, if a skip to a new page has been specified, the program causes the line printer to advance.

3.102 If it is desired that the remaining detail and total lines on the page ***not*** be printed before the overflow records and advance to the next page, enter "F" in column 16. The printing of this record and all remaining detail and total records will be suppressed until the overflow records are printed. This will occur only if all indicators in columns 23 through 31 of this specification are satisfied and an overflow has actually occurred.

3.103 Overflow processing does not automatically advance the printer to the next page. This must be specified in columns 19 through 22 of the output specification.

Space

3.104 Columns 17 through 18 may be used to specify line spacing before and/or after output of the line. This field is actually composed of two subfields, before (column 17) and after (column 18), which determine when spacing is to occur. An entry may be made in either, both, or none of the subfields as follows:

Column 17 or 18 Entry	Meaning
1 or blank	Space one line
2	Space two lines
3	Space three lines

3.105 The HP 2767A printer only prints on 6Ø of the available 66 lines on a page. This always provides for a small margin at the top and bottom. When the program requests spacing beyond the overflow line but not onto a new page, and an overflow indicator has been assigned to the file, that indicator remains on until all overflow records are printed.

3.106 The space field is related to the skip field (columns 19 through 22, explained in the following paragraph). The skip field permits moving the carriage ahead several lines at once without proceeding line by line. If both spacing and skipping ar specified on the same line of the output specification, these operations will be done in the following order:

(1) Skip before printing.

(2) Space before printing.

(3) Skip after printing.

(4) Space after printing.

Skip

3.107 Columns 19 through 22 provide for skipping to a particular line number before and/or after outputting the record. Like the space field, the skip field is also composed of a before subfield (columns 19 through 2Ø) and an after subfield (columns 21 trough 22) which are used to determine when skipping occurs. A line counter specification must be provided before entries in this field can be executed by the program. The entry ranges from Ø1 through 99 and indicates the line number to which the operator will skip. If no skipping is desired, leave the field blank.

Note: Do not specify skipping when the output device is not the printer. Only spacing is allowed for the line printer.

3.108 Normally, programmers request skipping when a new page is required or when many lines are to be spanned. If a request is made to the current position of the printer, no skipping is performed. If a request is made beyond the overflow line (but not onto a new page), the overflow indicator turns on and remains on until all overflow lines are printed. Skipping to a line beyond the form-length line defined for this line on the line center specification should not be requested.

Item 4—Output Indicators

3.109 Columns 23 through 31 are used to assign one or more indicators which are used to determine when information on each line is to be outputted. Up to three indicators may be specified to condition a field output. More than three indicators may be used to condition the entire record output by using logical AND and OR operators to combine lines (see Item 5). Any indicator or no indicator may be assigned. If ,no indicator is assigned, the information on the line will be outputted for each record read.

3.110 A NOT column is associated with each indicator. If it is desired that an indicator in the off state should condition the record, enter "N" in the NOT column associated with the indicator.

Item 5—AND and OR

3.111 AND and OR entries (columns 14 through 16) are used to condition an entire record output, and ***cannot*** be used to condition fields within records.

3.112 To condition a record output with more than three indicators, use the logical AND operator as follows:

(1) Specify the first three indicators in columns 23 through 31 of the line containing the record description.

(2) Enter AND in columns 14 through 16 of the ***next*** line, and up to three additional indicators in columns 23 through 31 of this line. Leave columns 17 through 22 blank.

(3) Continue with as many AND lines as needed to specify all indicators required.

3.113 To condition a record output with one or more alternative indicators, use the logical OR operator as follows:

(1) Specify up to three indicators in columns 23 through 31 of the line containing the record description.

(2) Enter OR in columns 14 through 15 of the ***next*** line, and up to three additional indicators for the next set of conditions in columns 23 through 31 of this line. Leave columns 17 through 22 blank.

(3) Continue with as many OR lines as needed to specify all indicators required.

3.114 When more than one indicator is entered on a line, those indicators are AND-related logically. The AND and OR operators which combine lines may be used in any combination to condition the output record.

Item 6—Field Name

3.115 In columns 32 through 37, enter the name of the field, array, or array element to be outputted. This field name should be previously defined in the file description, file extension, input, or calculation specifications, and entered exactly as previously defined. When a field name is entered, leave columns 7 through 22 of this line blank. Do not enter a name in this field if a constant (Item 7) is entered in columns 45 through 7Ø.

End Positions

3.116 Fields within records to be outputted may be listed on the output specification in any order. The sequence of outputting fields is determined by the end position field (columns 4Ø through 43). This entry can range from 1 through 9999 and indicates the column number where the field ends.

Edit Code

3.117 This field is used to provide editing of the data the program outputs. These codes are divided into two groups: simple edit codes and complex edit codes. The simple edit codes are summarized as follows:

Column 38 Entry	Meaning
X	Removes plus sign from units position of field before field data is written. This code does not remove a minus sign nor suppress leading zeros. No decimal points are printed.
Y	Insert slash marks into fields ranging from three to six digits in length as follows:

Number of Digits in Field	Format
3	nn/n
4	nn/nn
5	nn/nn/n
6	nn/nn/nn

Z	Suppress leading zeros, ignore (don't print) any decimal point specified for field, and remove arithmetic sign from unit position.

3.118 The complex edit codes (1 through 4, A through D, J through M) insert various punctuation marks in the data fields specified. Their effect is summarized in Table A.

3.119 Examples of the effect of specifying a complex edit code is provided in Table B.

3.120 It should be remembered that editing adds extra characters to the output. Thus, when specifying the end position for an edited field, allow for the extra editing characters.

Blank After

3.121 An entry of "B" in column 39 is used to reset the contents of a numeric field to zero, or the content of an alphanumeric field to blanks. Leave column 39 blank if the contents of the field are ***not*** to be reset.

3.122 The blank after field is typically used when accumulating totals for several individual control groups. After finding and printing the totals for one group, the total field is reset to zero. Then, when totals are accumulated for the next control group, the new totals will not be added to those of the previous group. The field will reset immediately after it is printed.

3.123 The blank after feature cannot be used in descriptions of constants or in the special date fields—UDATE, UDAY, UMONTH, or UYEAR.

3.124 When the blank after feature is specified, any indicators used to test the field for zeros or blanks during processing of input or calculation specifications are set on to show that the field is now blank. Only the first indicator used in such a test is set on.

Item 7—Constant Word

3.125 Columns 45 through 7Ø are used to specify a constant to be outputted. This entry consists of 1 through 24 characters (letters, digits, or special characters) surrounded with quotation marks. The field name (columns 32 through 37) should be left blank.

3.126 A constant is a group of characters that remains the same each time it is outputted. Constants may appear as report titles, page or column headings, or record-identifying information. Constants are often used when specifying heading records.

4. RPG/CAROT OBJECT PROGRAM OPERATING CYCLE

4.01 Every RPG/CAROT object program executes the same general sequence of operations each time it processes a record. This sequence is called the RPG/CAROT object program operating cycle. Although this cycle cannot actually be observed, knowledge of its details will help in organizing source code and debugging programs more effectively.

4.02 Before the object program begins the cycle, it performs various initialization and housekeeping operations described in Step 1. Next, the program enters the operating cycle (Steps 2 through 8), repeating the operating cycle once for every record processed. Figure 7 briefly illustrates the operating cycle, while Fig. 8 provides a flowchart which describes the cycle in detail. A brief discussion of each step follows:

Step **Performance**

1 ***Precycle Initialization and Housekeeping.*** The object program performs certain preliminary functions, that are done only once, before the cycle actually begins. During these operations, the program opens (prepares for processing) all files to be used. It aligns the first page for any printer files and writes all first-page headings (those records conditioned by the 1P indicator). The program next reads a record from each file used for input and identifies its type. At this point, the program is ready to begin the operating cycle.

2 ***Heading and Detail-Time Output.*** This is the beginning of the operating cycle. The program outputs all other heading and detail records whose conditions are satisfied. The program next tests the status of all halt indicators (halts if any are on), and turns off all record-identifying and control-level indicators that are on at this point. It turns off any overflow indicators that were on before the last detail calculation (in Step 8). It also tests the LR indicator for end-of-program status and transfers to Step 6 if this indicator is on.

3 ***Input Record Operation.*** If this is not the first pass through the cycle, the program reads the next input record from the last file processed and identifies its type, checking for an end-of-file indication and proper sequence. This step is skipped on the first pass through the operating cycle.

4 ***Record Selection.*** If the program uses only one file for input, the next record from that file is selected for processing. If the program uses more than one file for input, the next record is selected sequentially from the list in the order in which they appear on the field description specification.

5 ***Control Break Test.*** The program tests for a control break and turns on the appropriate control-level indicator and all lower-level indicators if a break occurs.

6 ***Total Time Operations.*** If this is the first record with control-level fields, the program skips total-time operations. However, if this is not the first record and does contain control-level fields, the program performs total-time calculations and output (including total-time overflow output). Total-time operations are all those calculations with LØ though L9 or LR indicator entries in the control-level field on the calculation specification and output operations of record-type "T" on the output specification. These operations are done ***after*** each control break occurs or after the last input record is read, but ***before*** the information on the input record selected in Step 4 is actually made available for processing. The program also sets all specified resulting indicators; and if the last record indicator is on, the program terminates.

7 ***Data Movement.*** The program moves the data from the record inputted into work areas for processing.

8 ***Detail-Time Calculations.*** The program performs detail-time calculations for the specified record, and returns to Step 2. Detail-time calculations are those calculations ***not*** conditioned by control-level indicators in the control-level field

on the calculation specification. Note that these calculations and the detail-time output described in Step 2 are both done ***after*** information from the record selected in Step 4 becomes available. At this point, the program also turns on any resulting indicators affected by these calculations.

5. ERROR MESSAGES

5.01 A list of error codes and their associated meaning which can occur when compiling source code into object code is provided in Table C.

5.02 A list of error codes and their associated meaning which can occur when executing an object program is provided in Table D.

6. EXAMPLE OF A PROGRAM GENERATION PROCESS

6.01 This part describes the process of generating an example RPG program. This example program, TRSUM, generates the daily office summary report. Study this example and refer back to the material previously described.

A. Defining the Problem

6.02 The first step in generating an RPG program is to define the problem. This definition consists of what data is to be processed and how it is to be processed to generate a report. In our example, TRSUM uses the data found in the file MNDATA which will be created using the select program. Dependent on the type of data in MNDATA, this program generates either the daily summary or a cumulative summary.

B. Printer Spacing Chart

6.03 The Printer Spacing Chart is used to arrange the data outputted for the summary report. This chart provides a grid that allows the user to plan exactly where each character will appear in each line outputted. On this chart, headings and total and detail outputs can be identifed. To indicate constants, enter each constant exactly as it is to appear when outputted. To indicate variable fields, use characters such as Xs, Ys, and Zs (or any other character) to identify the field's maximum length, and then write the field name in parentheses under these characters. Insert periods, commas, etc, as they are to appear when the report is generated. Figure 9 is an example printer chart which illustrates how TRSUM is to be programmed to arrange the outputted report.

C. Entry of Information on Work Sheets

6.04 Now that the problem has been defined and it has been decided how the output is to be arranged, the program can be coded and entered on the work sheets. The example program, TRSUM, is contained on eight work sheets in Fig. 10. Refer to Parts 2, 3, and 4 for details of the actual entries. If a large number of comments are to be entered using this program, a special work sheet (comment sheet) is provided in Part 8. Figure 11 is a computer-generated source listing of TRSUM with a number of comments which aid in understanding the program.

D. Inputting a Source Program Using the Text Editor

6.05 The CAROT text editor is used to input RPG source programs. Refer to Section 190-102-206 for details. When inputting an RPG program, various information needs to be entered in certain column positions. To aid in this input requirement, the text editor provides **COL** and **TAB** commands. The **COL** command results in the outputting of a line of numbers which correspond to column numbers. Thus, with this command, the user can space over to the desired column position and enter the data. The **TAB** command can be used to set tab points much like those associated with typewriters. With this command, the user can set the tab points to various column positions and, when inputting data, input the tab character to move to the next field.

6.06 When the source program has been completely entered, a last line should be entered. This line should contain the sequence number followed by the letter "G", which is entered in column 6. This is used to indicate the end of the source program to the RPG compiler.

6.07 Since a file name is associated with the source program inputted via the text editor and another file name is associated with the RPG object program, a convention should be established to identify the type of file. A suggested convention is to first name the object program, then provide the same name to the RPG source file; however, the name of the associated file should begin with an ampersand (&). For example, name the object file TRSUM and the associated text file &TRSUM.

E. Compiling Source Code Into Object Code

6.08 To compile an RPG source program, type **RUN:RPG** followed with a carriage return. Next, a message will appear requesting entry of the name of the source file to be compiled. Enter this file as requested followed with a carriage return. This results in the source program being compiled into object code. If no errors occur while compiling, the object program will be executed. If errors occur during compiling or execution, messages will be outputted describing the illegal operation. A list of possible error messages are provided in Tables C and D.

F. Debugging

6.09 If an error occurs while compiling or executing, consult the control record specification for the debug options available.

G. Execute Object Program

6.10 Once a program has been compiled and the object program has been saved in a disc file, a special source file can be set up for the purpose of executing the associated object program. This bypasses the compiling process, noticeably reducing processor time devoted to the RPG program. To provide this capability, use the CAROT text editor to set up a source file containing the control record specification with an entry of "E" in column 44. Enter a second record in this file that has a "G" in column 6. This last record indicates the end of the RPG source file. Name this file the same as the object file, however, begin the name with a period. This name beginning with a period is part of the suggested convention for naming files by type.

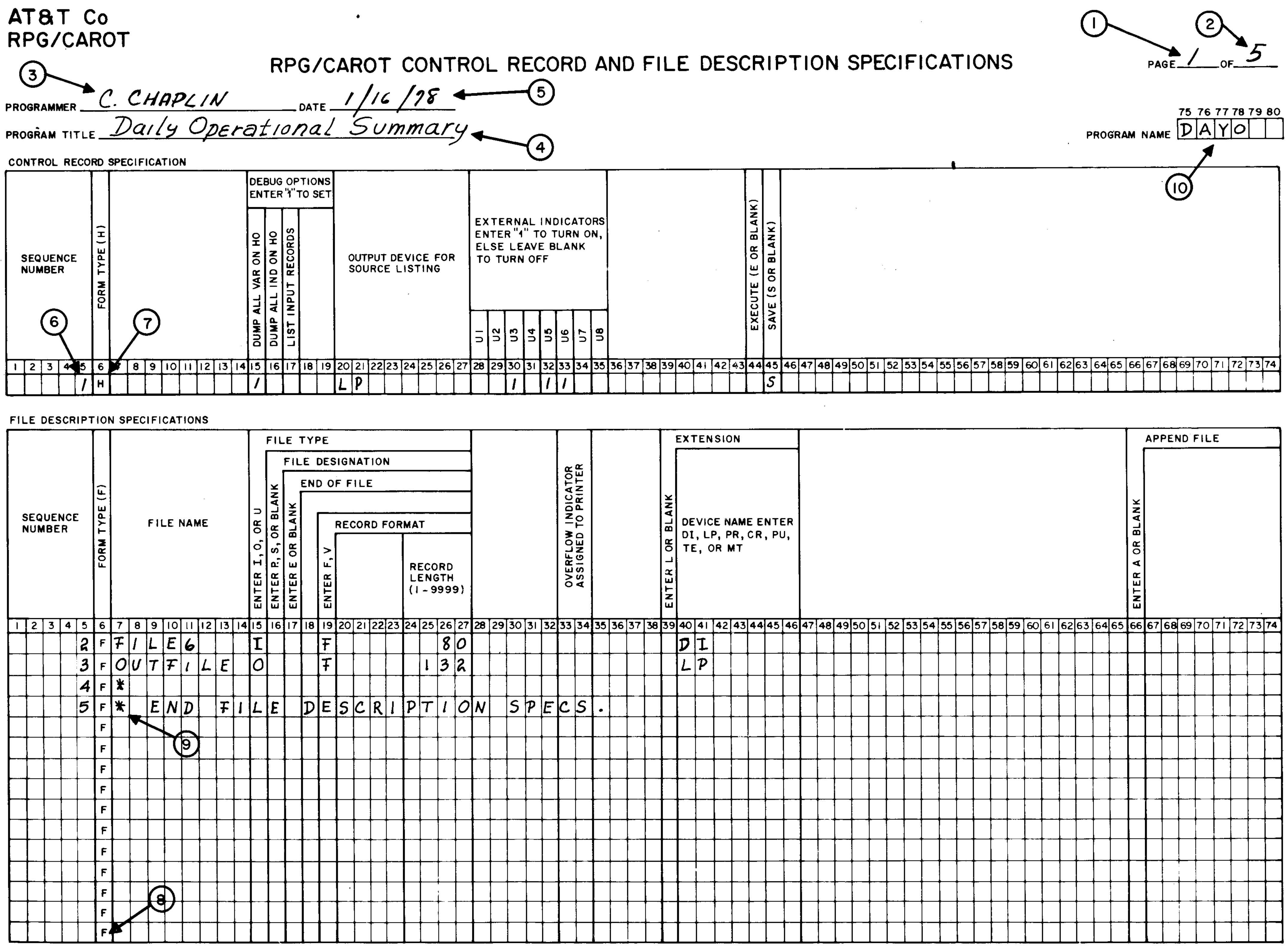
6.11 After this file has been set up, the object program can be executed by typing **RUN:RPG** followed with a carriage return. Next a message will appear requesting entry of the source program name. Enter the name which begins with a period. The RPG object program named in columns 75 through 8Ø of the control record specification will be executed. Figure 12 is an example of the output generated by the example program—TRSUM.

7. REFERENCES

SECTION	DESCRIPTION
190-102-203	CAROT 2, Generic 2, Data Base Description and Input File Preparation
190-102-206	Data Base Manipulation using the Select Program and Text Editor

8. RPG/CAROT BLANK WORK SHEETS

8.01 This part contains blank work sheets for use in generating RPG/CAROT programs (see Fig. 13, 14, 15, 16, 17, 18, and 19).



AT&T Co
RPG/CAROT

RPG/CAROT CONTROL RECORD AND FILE DESCRIPTION SPECIFICATIONS

PAGE 1 OF 5

PROGRAMMER C. CHAPLIN DATE 1/16/78

PROGRAM TITLE Daily Operational Summary

PROGRAM NAME (75 76 77 78 79 80) DAYO

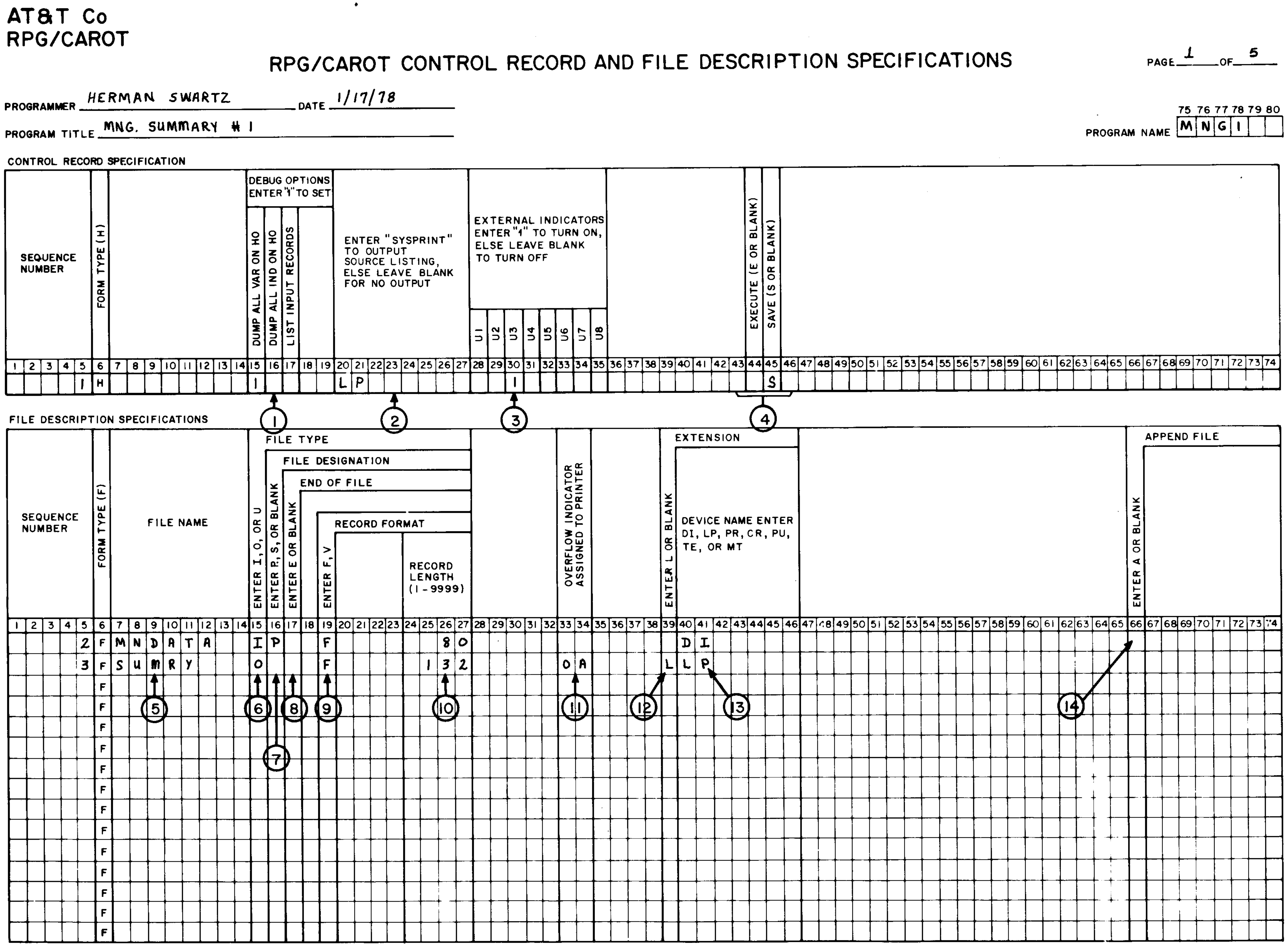
CONTROL RECORD SPECIFICATION

SEQUENCE NUMBER (1-5)	FORM TYPE (H) (6)	DEBUG OPTIONS ENTER "1" TO SET: DUMP ALL VAR ON HO (15)	DUMP ALL IND ON HO (16)	LIST INPUT RECORDS (17)	OUTPUT DEVICE FOR SOURCE LISTING (20-21)	EXTERNAL INDICATORS ENTER "1" TO TURN ON, ELSE LEAVE BLANK TO TURN OFF: U1	U2	U3	U4	U5	U6	U7	U8	EXECUTE (E OR BLANK) (44)	SAVE (S OR BLANK) (45)
1	H	1			LP			1		1	1				S

FILE DESCRIPTION SPECIFICATIONS

SEQUENCE NUMBER	FORM TYPE (F)	FILE NAME	FILE TYPE: ENTER I, O, OR U	FILE DESIGNATION: ENTER P, S, OR BLANK	END OF FILE: ENTER E OR BLANK	RECORD FORMAT: ENTER F, V	RECORD LENGTH (1-9999)	OVERFLOW INDICATOR ASSIGNED TO PRINTER	EXTENSION: ENTER L OR BLANK	DEVICE NAME ENTER DI, LP, PR, CR, PU, TE, OR MT	APPEND FILE: ENTER A OR BLANK
2	F	FILE6	I			F	80			DI	
3	F	OUTFILE	O			F	132			LP	
4	F	*									
5	F	* END FILE DESCRIPTION SPECS.									
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										

Fig. 1—Common Entries on All RPG/CAROT Work Sheets



AT&T Co
RPG/CAROT

RPG/CAROT CONTROL RECORD AND FILE DESCRIPTION SPECIFICATIONS

PAGE 1 OF 5

PROGRAMMER HERMAN SWARTZ DATE 1/17/78

PROGRAM TITLE MNG. SUMMARY # 1

PROGRAM NAME (75 76 77 78 79 80) M N G 1

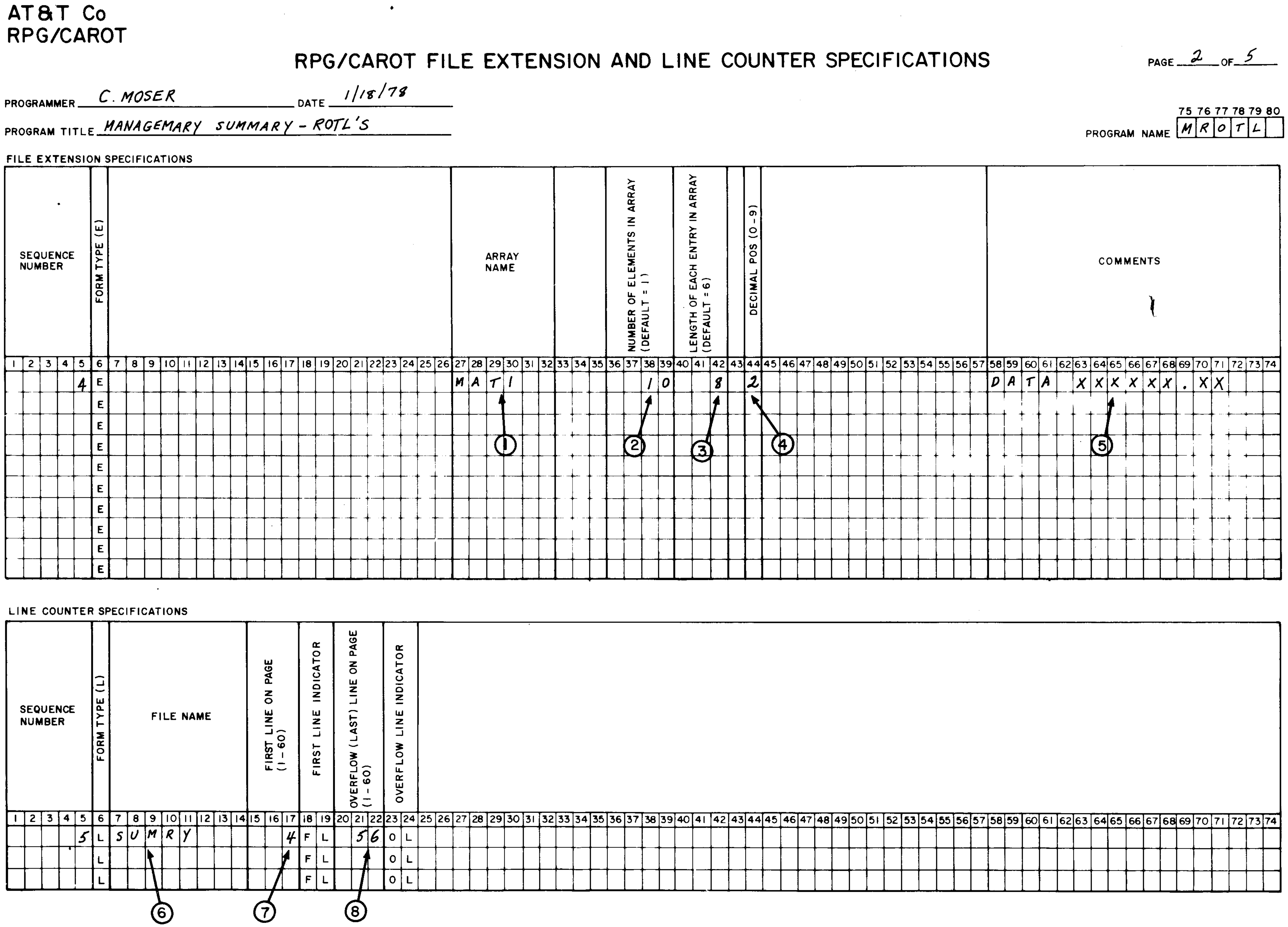
CONTROL RECORD SPECIFICATION

SEQUENCE NUMBER (1-5)	FORM TYPE (H) (6)	DEBUG OPTIONS ENTER "1" TO SET: DUMP ALL VAR ON HO (15)	DUMP ALL IND ON HO (16)	LIST INPUT RECORDS (17)	ENTER "SYSPRINT" TO OUTPUT SOURCE LISTING, ELSE LEAVE BLANK FOR NO OUTPUT (20-27)	EXTERNAL INDICATORS ENTER "1" TO TURN ON, ELSE LEAVE BLANK TO TURN OFF: U1 (28)	U2 (29)	U3 (30)	U4 (31)	U5 (32)	U6 (33)	U7 (34)	U8 (35)	EXECUTE (E OR BLANK) (44)	SAVE (S OR BLANK) (45)
1	H	1			LP			1							S

FILE DESCRIPTION SPECIFICATIONS

SEQUENCE NUMBER (1-5)	FORM TYPE (F) (6)	FILE NAME (7-14)	FILE TYPE: ENTER I, O, OR U (15)	FILE DESIGNATION: ENTER P, S, OR BLANK (16)	END OF FILE: ENTER E OR BLANK (17)	RECORD FORMAT: ENTER F, V (19)	RECORD LENGTH (1-9999) (24-27)	OVERFLOW INDICATOR ASSIGNED TO PRINTER (33-34)	EXTENSION: ENTER L OR BLANK (39)	DEVICE NAME ENTER DI, LP, PR, CR, PU, TE, OR MT (40-46)	APPEND FILE: ENTER A OR BLANK (66)
2	F	MNDATA	I	P		F	80			DI	
3	F	SUMRY	O			F	132	OA	L	LP	
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										
	F										

Fig. 2—Example of a Completed Work Sheet (Forms H and F)



AT&T Co
RPG/CAROT

RPG/CAROT FILE EXTENSION AND LINE COUNTER SPECIFICATIONS

PAGE 2 OF 5

PROGRAMMER C. MOSER DATE 1/18/78

PROGRAM TITLE MANAGEMARY SUMMARY - ROTL'S

PROGRAM NAME (75 76 77 78 79 80) M R O T L

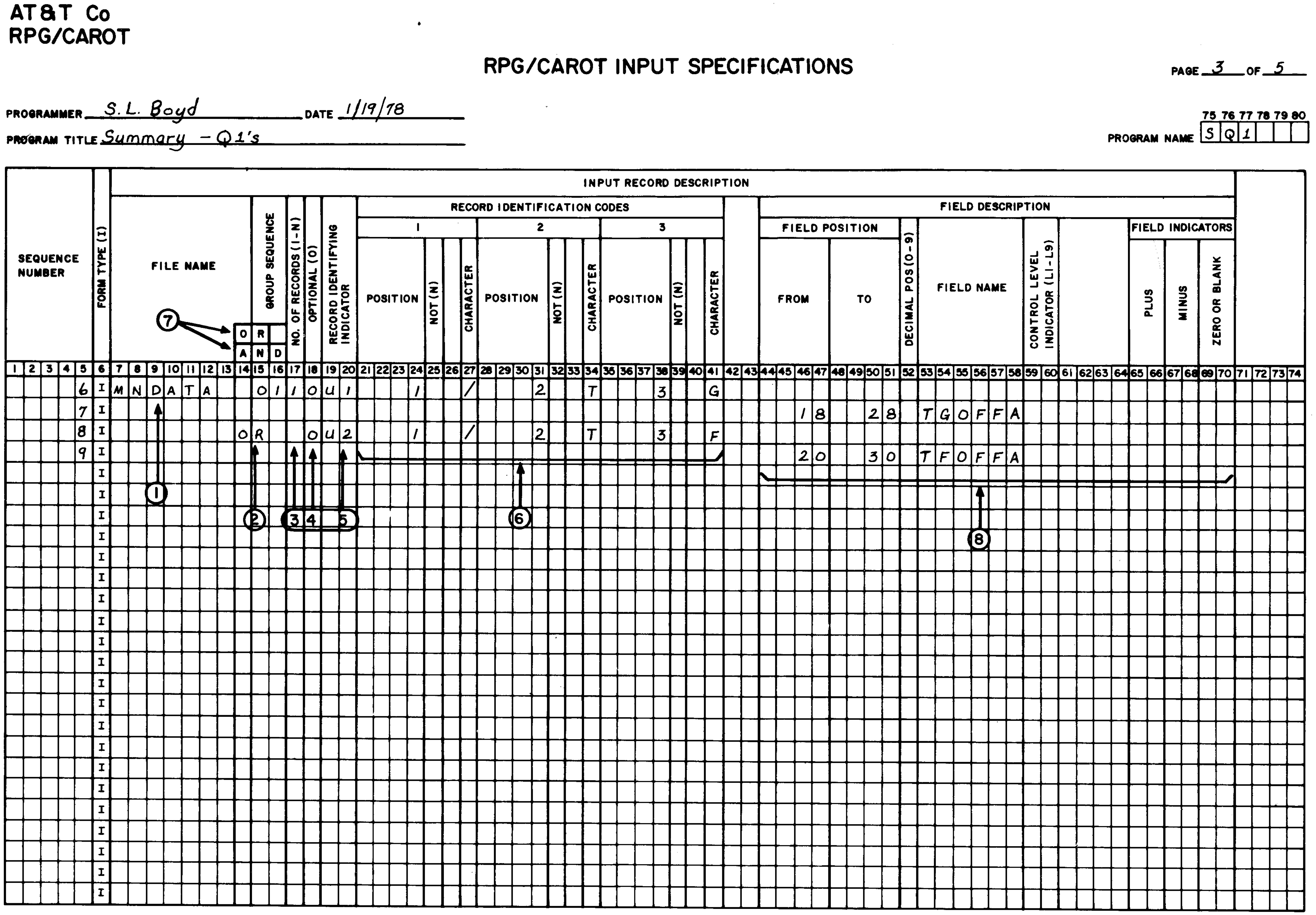
FILE EXTENSION SPECIFICATIONS

SEQUENCE NUMBER (1-5)	FORM TYPE (E) (6)	(7-26)	ARRAY NAME (27-32)	(33-35)	NUMBER OF ELEMENTS IN ARRAY (DEFAULT = 1) (36-39)	LENGTH OF EACH ENTRY IN ARRAY (DEFAULT = 6) (40-42)	(43)	DECIMAL POS (0 - 9) (44)	(45-57)	COMMENTS (58-74)
4	E		MAT1		10	8		2		DATA XXXXXX.XX
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									

LINE COUNTER SPECIFICATIONS

SEQUENCE NUMBER (1-5)	FORM TYPE (L) (6)	FILE NAME (7-14)	FIRST LINE ON PAGE (1 - 60) (15-17)	FIRST LINE INDICATOR (18-19)	OVERFLOW (LAST) LINE ON PAGE (1 - 60) (20-22)	OVERFLOW LINE INDICATOR (23-24)	(25-74)
5	L	SUMRY	4	FL	56	OL	
	L			FL		OL	
	L			FL		OL	

Fig. 3—Example of a Completed Work Sheet (Forms E and L)



AT&T Co
RPG/CAROT

RPG/CAROT INPUT SPECIFICATIONS

PAGE 3 OF 5

PROGRAMMER S. L. Boyd DATE 1/19/78

PROGRAM TITLE Summary – Q1's

PROGRAM NAME (75 76 77 78 79 80) S Q 1

		INPUT RECORD DESCRIPTION																											
							RECORD IDENTIFICATION CODES													FIELD DESCRIPTION									
							1				2				3					FIELD POSITION						FIELD INDICATORS			
SEQUENCE NUMBER	FORM TYPE (I)	FILE NAME	GROUP SEQUENCE OR AND	NO. OF RECORDS (1-N)	OPTIONAL (O)	RECORD IDENTIFYING INDICATOR	POSITION	NOT (N)		CHARACTER	POSITION	NOT (N)		CHARACTER	POSITION	NOT (N)		CHARACTER		FROM	TO	DECIMAL POS (0-9)	FIELD NAME	CONTROL LEVEL INDICATOR (L1-L9)		PLUS	MINUS	ZERO OR BLANK	
1 2 3 4 5	6	7 8 9 10 11 12 13	14 15 16	17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38	39	40	41	42 43	44 45 46 47	48 49 50 51	52	53 54 55 56 57 58	59 60	61 62 63 64	65 66	67 68	69 70	71 72 73 74
6	I	MNDATA	01	1	O	U1	1			/	2			T	3			G											
7	I																			18	28		TGOFFA						
8	I		OR		O	U2	1			/	2			T	3			F											
9	I																			20	30		TFOFFA						
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												

Fig. 4—Example of a Completed Work Sheet (Form I)

AT&T Co
RPG/CAROT

RPG/CAROT CALCULATION SPECIFICATIONS

PAGE 4 OF 5

PROGRAMMER J. HALL DATE 1-18-78

PROGRAM TITLE DAILY OPERATIONAL SUMMARY

75 76 77 78 79 80
PROGRAM NAME DAYOP

SEQUENCE NUMBER	FORM TYPE (C)	CONTROL LEVEL (LO-L9, LR, AN/OR)	INDICATORS (NOT, AND, NOT, AND, NOT)	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD NAME	RESULT FIELD LENGTH	DECIMAL POS (0-9)	HALF ADJUST (H)	RESULTING INDICATORS: ARITHMETIC PLUS / COMPARE 1>2 / LOOKUP HIGH	MINUS / 1<2 / LOW	ZERO / 1=2 / EQUAL	COMMENTS
1 2 3 4 5	6	7 8	9 10 11 12 13 14 15 16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
10	c	*												
11	c	*	THE FOLLOWING ARE DONE AFTER EACH OPER. RECORD (MO)											
12	c	*												
13	c		07	OSCHD	COMP	0.							14	I=0 THEN I 14
14	c		07	BUSY	ADD	HANDD	NOTDON	60						
15	c	*												
16	c	LR		TOTSCH	DIV	100.	SCHFAC							
17	c	*												
18	c		03 N04 05											
19	c	AN	06	ETA	ADD	ZETA	RES3	10	0					
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													
	c													

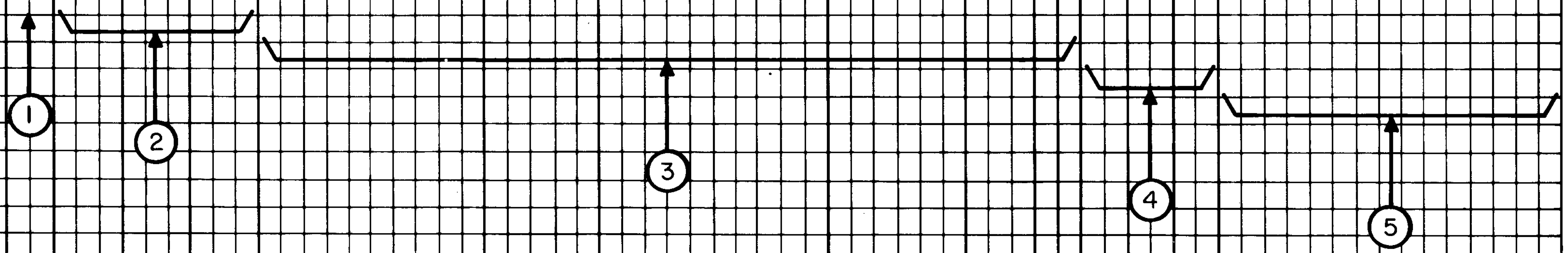
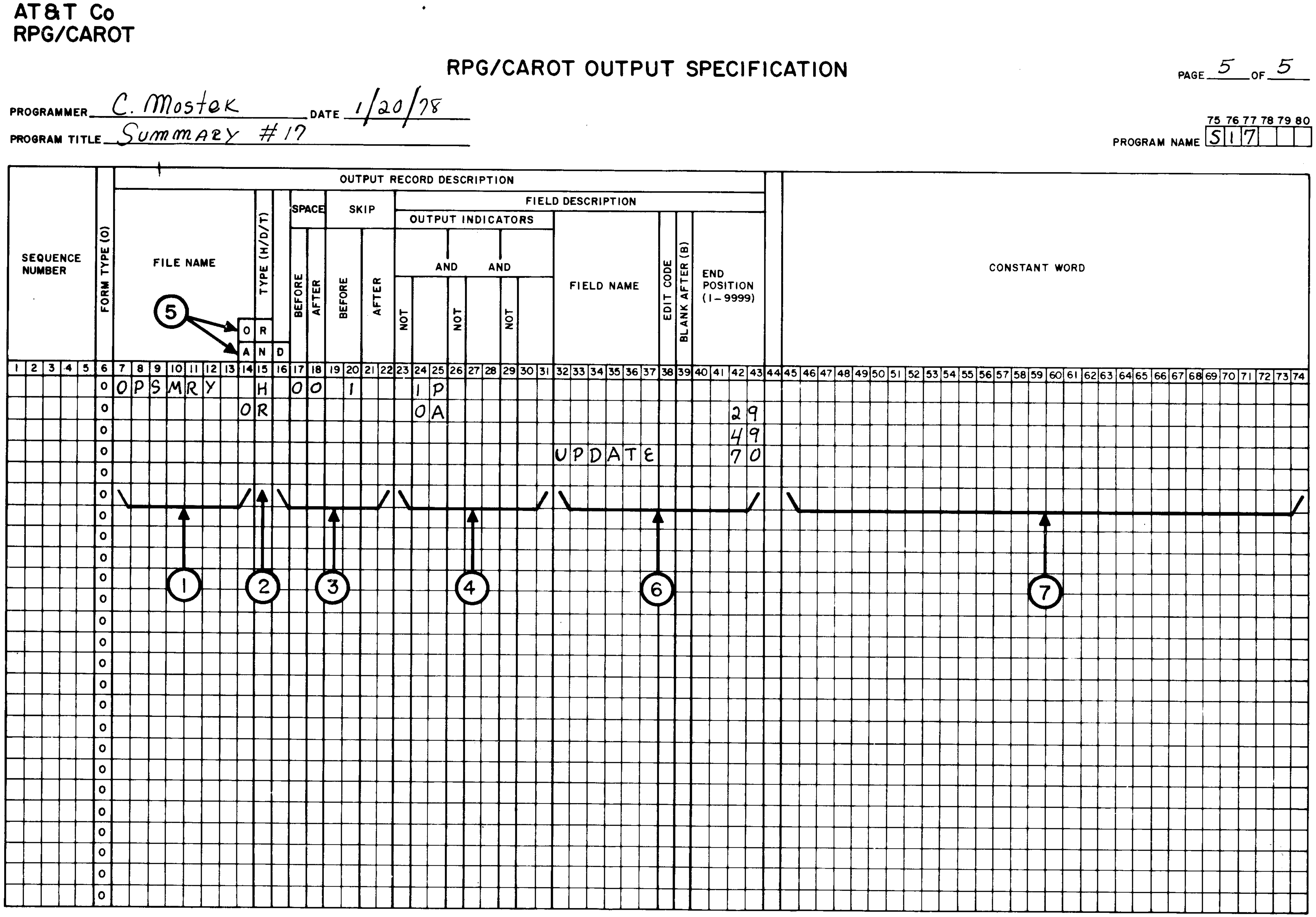


Fig. 5—Example of a Completed Work Sheet (Form C)

AT&T Co
RPG/CAROT



RPG/CAROT OUTPUT SPECIFICATION

PAGE 5 OF 5

PROGRAMMER C. Mostek DATE 1/20/78

PROGRAM TITLE Summary #17

PROGRAM NAME (75–80) S17

OUTPUT RECORD DESCRIPTION

SEQUENCE NUMBER	FORM TYPE (O)	FILE NAME	AND / OR	TYPE (H/D/T)	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS (NOT / AND NOT / AND NOT)	FIELD NAME	EDIT CODE	BLANK AFTER (B)	END POSITION (1–9999)	CONSTANT WORD
	O	OPSMRY		H	0	0	1		1P					
	O		OR						OA				29	
	O												49	
	O									UPDATE			70	
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													
	O													

Fig. 6—Example of a Completed Work Sheet (Form O)

TABLE A

COMPLEX EDIT CODES

EDIT CODE (COLUMN 38)	COMMAS	DECIMAL POINT	SIGN FOR NEGATIVE BALANCE			ZERO SUPPRESS	PRINTOUT ON ZERO BALANCE
			NO SIGN	CR	–(MINUS)		
1	Yes	Yes	No Sign			Yes	.00 or 0
2	Yes	Yes	No Sign			Yes	Blanks
3		Yes	No Sign			Yes	.00 or 0
4		Yes	No Sign			Yes	Blanks
A	Yes	Yes		CR		Yes	.00 or 0
B	Yes	Yes		CR		Yes	Blanks
C		Yes		CR		Yes	.00 or 0
D		Yes		CR		Yes	Blanks
J	Yes	Yes			—	Yes	.00 or 0
K	Yes	Yes			—	Yes	Blanks
L		Yes			—	Yes	.00 or 0
M		Yes			—	Yes	Blanks

TABLE B

COMPLEX EDIT CODE EFFECTS

EDIT CODE (COLUMN 38)	FIELD CHARACTERISTICS				
	1769532	02	00	000	041345
	FIELD LENGTH AND DIGITS				
	POSITIVE NUMBER TWO DECIMAL POSITIONS	NEGATIVE NUMBER TWO DECIMAL POSITIONS	ZERO–TWO DECIMAL POSITIONS	ZERO–NO DECIMAL POSITIONS	POSITIVE NUMBER THREE DECIMAL POSITIONS
1	17,695.32	.02	.00	0	41.345
2	17,695.32	.02			41.345
3	17,695.32	.02	.00	0	41.345
4	17,695.32	.02			41.345
A	17,695.32	.02CR	.00	0	41.345
B	17,695.32	.02CR			41.345
C	17,695.32	.02CR	.00	0	41.345
D	17,695.32	.02CR			41.345
J	17,695.32	.02—	.00	0	41.345
K	17,695.32	.02—			41.345
L	17,695.32	.02—			41.345
M	17,695.32	.02—			41.345

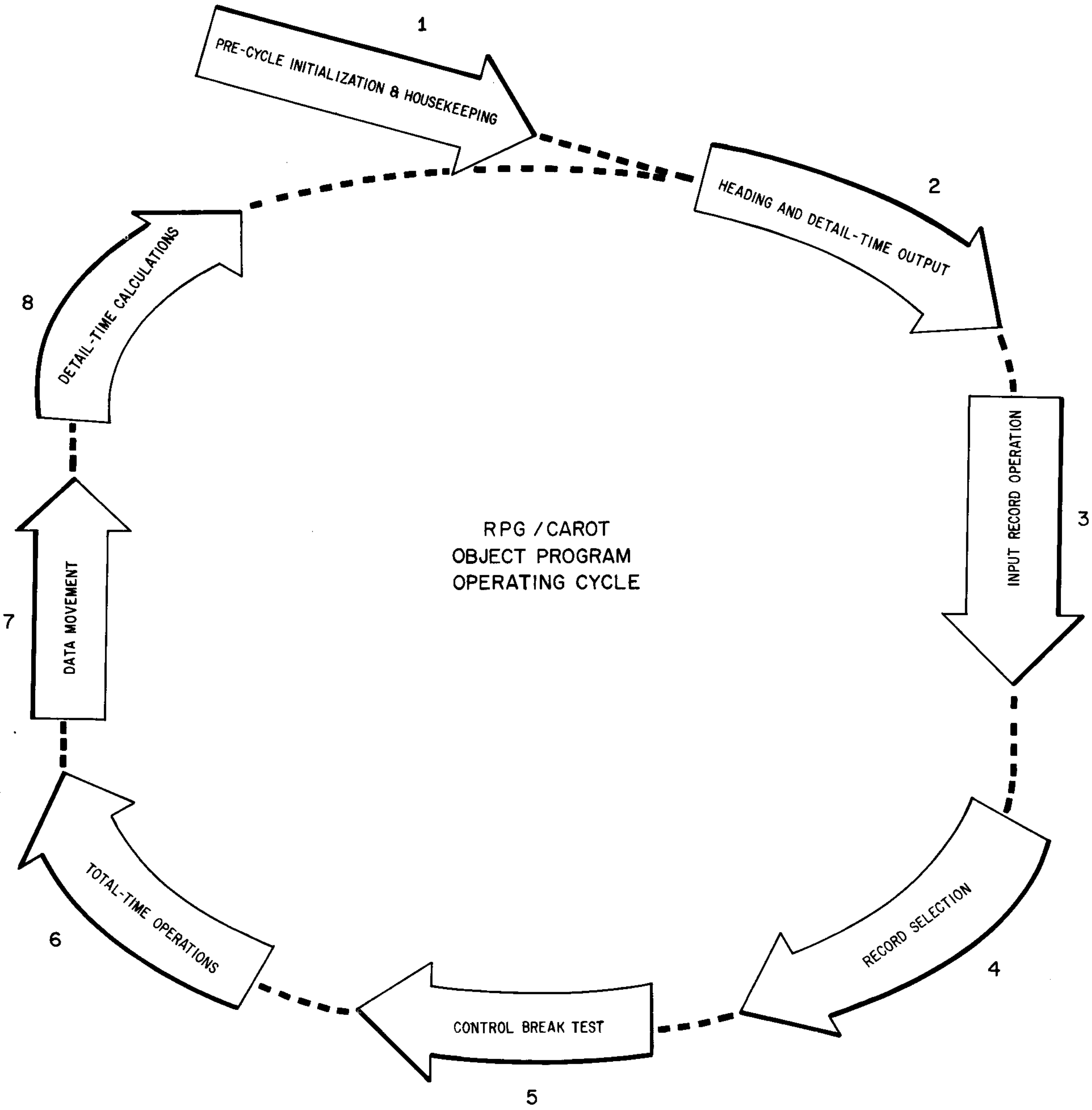


Fig. 7—RPG/CAROT Object Program Operating Cycle

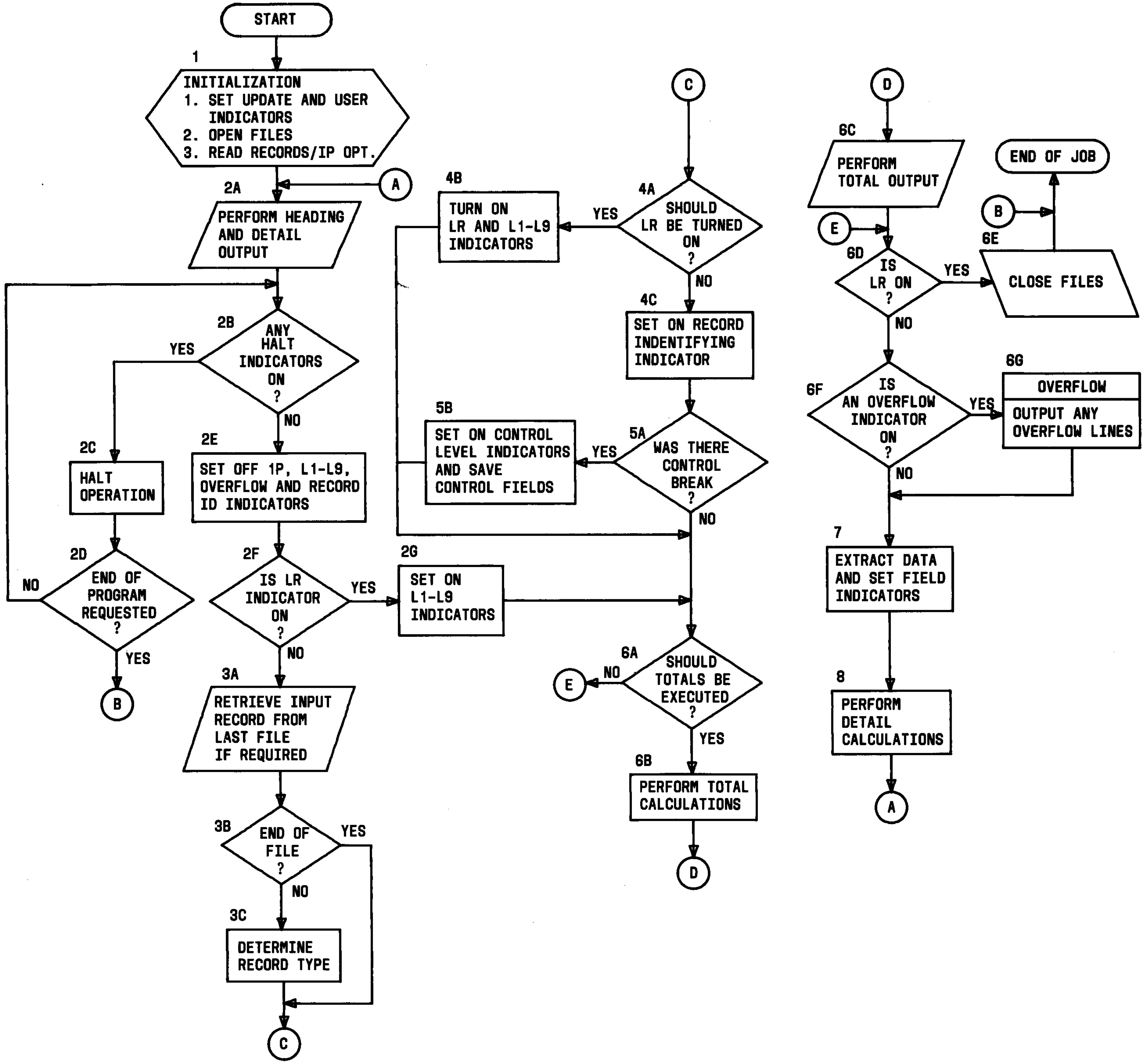

START
1
INITIALIZATION
1. SET UPDATE AND USER INDICATORS
2. OPEN FILES
3. READ RECORDS/IP OPT.
A
2A
PERFORM HEADING AND DETAIL OUTPUT
2B
ANY HALT INDICATORS ON ?
YES
NO
2C
HALT OPERATION
2D
END OF PROGRAM REQUESTED ?
NO
YES
B
2E
SET OFF 1P, L1-L9, OVERFLOW AND RECORD ID INDICATORS
2F
IS LR INDICATOR ON ?
YES
NO
2G
SET ON L1-L9 INDICATORS
3A
RETRIEVE INPUT RECORD FROM LAST FILE IF REQUIRED
3B
END OF FILE ?
YES
NO
3C
DETERMINE RECORD TYPE
C
C
4A
SHOULD LR BE TURNED ON ?
YES
NO
4B
TURN ON LR AND L1-L9 INDICATORS
4C
SET ON RECORD INDENTIFYING INDICATOR
5A
WAS THERE CONTROL BREAK ?
YES
NO
5B
SET ON CONTROL LEVEL INDICATORS AND SAVE CONTROL FIELDS
6A
SHOULD TOTALS BE EXECUTED ?
NO
E
YES
6B
PERFORM TOTAL CALCULATIONS
D
D
6C
PERFORM TOTAL OUTPUT
E
6D
IS LR ON ?
YES
NO
6E
CLOSE FILES
B
END OF JOB
6F
IS AN OVERFLOW INDICATOR ON ?
YES
NO
6G
OVERFLOW
OUTPUT ANY OVERFLOW LINES
7
EXTRACT DATA AND SET FIELD INDICATORS
8
PERFORM DETAIL CALCULATIONS
A

Fig. 8—Object Program Logic Flowchart

TABLE C

SOURCE PROGRAM ERRORS

ERROR CODE	MEANING
1	Invalid form type specified. Specification will not be processed.
2	Invalid variable or file name given.
3	Invalid file type specified.
4	Invalid file designation given. Secondary is assumed.
5	Invalid EOF entry. Blank is assumed.
6	No sequence checking is implemented. Entry is ignored.
7	Variable length records not implemented. Fixed length is assumed.
8	No random file processing is implemented. Entry is ignored.
9	Invalid overflow indicator specified. Blank is assumed.
10	Invalid extension code specified. Blank is assumed.
11	File specified was already specified. First specification used.
12	Program is too large. Available memory has been exceeded. Program must be reduced in size.
13	File name specified is not a valid file name. Check definition of valid file name.
14	Only first line and overflow lines have been inplemented. Check manual for valid entry on this specification record.
16	Sequence numbers must be in ascending order. This sequence number has been ignored.
17	"N" or "1" must be specified for numeric sequence. "N" is assumed.
18	Option entry must be the letter O or a space. If no entry, space is assumed.
19	Entry must be blank. Entry is ignored.
20	File name was either not specified on file specification record or not specified as an output file.
21	Invalid entry in space before, space after, skip before, or skip after field. Simple integer expected.
22	Invalid indicator specified. No indicator is assumed.
23	Invalid entry.
24	Invalid field name found. Either undefined or ill-formed.
25	Invalid edit code specified. Blank is assumed.
26	Invalid blank after code specified. Blank is assumed.
27	End position specified is after the end of record. End of record is assumed.
28	No valid field name or constant field was found.
29	Invalid entry found. Integer or blank required.
30	Invalid device specified. Program marked nonexecutable.
31	Invalid indicator specified. No indicator is assumed.
32	Invalid operation specified. Specification record is ignored.
33	Simple positive integer expected. One is assumed.
34	Invalid entry.
35	Unexpected end of file found. Processing halted.
36	Invalid or previously defined TAG found. Specification record is ignored.
37	Not all labels referenced by GOTO operations were defined.
38	Invalid control level specified. L0 is assumed.
39	Invalid operand or result found. Specification ignored.
40	Attempted to redefine a variable. The first definition is used.
41	Undefined variable used. The variable is assumed to be numeric, with a length = 10, and no decimal places.

TABLE C (Contd)

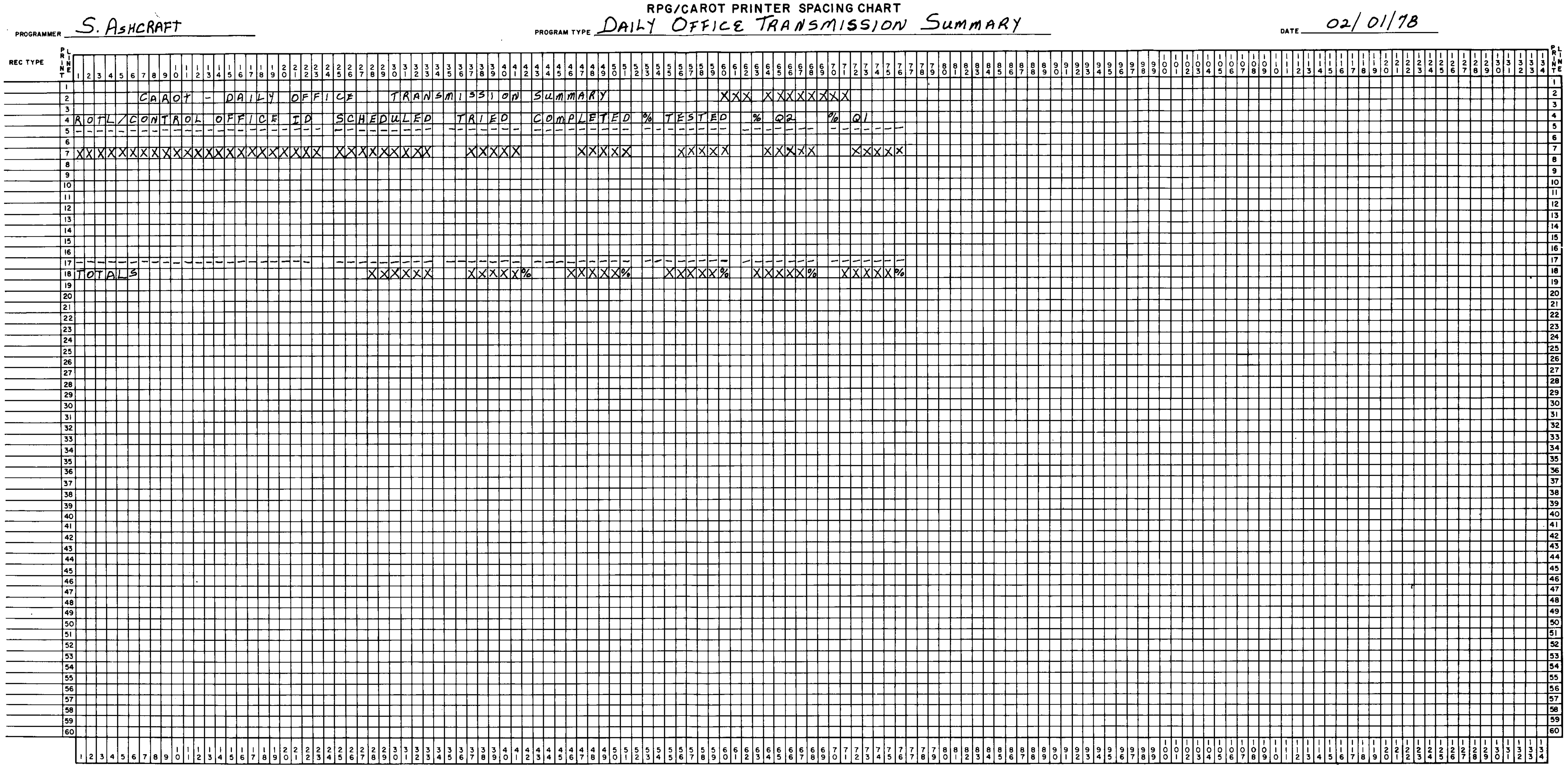
SOURCE PROGRAM ERRORS

ERROR CODE	MEANING
42	No valid input records were specified. The program is marked nonexecutable.
43	Invalid group sequence specified.
44	Invalid record identification found. No record identification code associated with this record.
45	Invalid record type found. Detail is assumed.
46	No line counter specification made for this file.
47	Invalid index specified. Either the index was invalid or it was used with a simple variable.
48	Invalid constant or literal specified. Entry is ignored.
49	Invalid record length was specified. Record length = 80 is assumed.
50	Invalid first line or overflow line specified. Overflow line = 59 and first line = 1 is assumed.
51	Invalid number of entries for specified array found. Defaulted to 80 elements.
52	Invalid size specified. Size defaulted to 7.
53	Invalid number of decimal positions specified. Number of decimal positions defaulted to 0.
54	No control card specified. Program was marked nonexecutable.
55	No device specified for system output. User's terminal is assumed.
56	Invalid entry. Entry must be either 0 or 1.
57	Invalid device specified for system output. User's terminal is assumed.
58	Invalid program name specified. Program name defaulted to RPGOBJ.
59	Invalid fetch overflow indicator found. Blank is assumed.
60	Invalid entry. "N" is assumed.
61	Invalid entry. Blank is assumed.
63	Either beginning position or end position specified is greater than record length. Invalid entry defaulted to 1.
64	Invalid index used. Index defaulted to 1.
65	Input/output device is of incorrect type; ie, line printer can be used as input device. User's terminal is assumed.
66	Invalid entry. Output files are not designated "S" or "P".
67	Only one primary file may be specified for input. First file so designated is used; all subsequent files are assumed to be secondary.
68	Blank after entry must be either "B" or blank. Entry defaulted to blank.
69	Line counter specifications may be assigned only for output files which are to be output on the line printer.
70	Only first line and overflow line are implemented in RPG/CAROT.
71	Invalid overflow line specified. Entry defaulted to 59.

TABLE D

OBJECT PROGRAM EXECUTION ERRORS

ERROR CODE	MEANING
101	File could not be opened. Program halted.
102	Available memory has been exceeded. Program halted.
103	Edited field too large for allocated space. Field is truncated.
104	"Y" edit code requires 3- to 6-digit field for input and 5 to 8 spaces for output.
106	Invalid subscript found. Subscript was either undefined or out of range.
107	Program attempted to write past the end of the record.
108	TAG variable was used as an output variable.
109	Alphanumeric variable was used where numeric variable was required.
110	Result of operation invalid due to overflow, underflow, or invalid operands.
111	Invalid sequence for input records.
112	Variable specified must be either numeric or alphanumeric.
113	Invalid character found in input field.
114	Either logical unit down or unexpected EOF found.
115	Error occurred during attempted read operation.
116	Invalid data found on input record.
117	Error occurred during write operation.
118	WARNING — Input record does not match any set of record identification codes.
119	File could not be accessed. Program halted.
120	Field to be edited exceeded 200 characters which is the system maximum.
121	Numeric variable was used where alpanumeric variable is expected.
122	Execution was deleted due to fatal error in source program.



RPG/CAROT PRINTER SPACING CHART

PROGRAMMER S. ASHCRAFT

PROGRAM TYPE DAILY OFFICE TRANSMISSION SUMMARY

DATE 02/01/78

```
 2        CAROT - DAILY OFFICE   TRANSMISSION SUMMARY          XXX XXXXXXXX
 4 ROTL/CONTROL OFFICE ID  SCHEDULED   TRIED  COMPLETED % TESTED  % Q2   % Q1
 5 ----------------------- ---------   -----  --------- -------- -----  ------
 7 XXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXX   XXXXX      XXXXX   XXXXX  XXXXX  XXXXX
17 ----------------------- ---------   -----  --------- -------- -----  ------
18 TOTALS                     XXXXXX   XXXXX%   XXXXX%  XXXXX%  XXXXX%  XXXXX%
```

Fig. 9—Printer Spacing Chart for TRSUM

AT&T Co
RPG/CAROT

RPG/CAROT CONTROL RECORD AND FILE DESCRIPTION SPECIFICATIONS

PAGE 1 OF 8

PROGRAMMER S. ASHCRAFT DATE 02/01/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75 76 77 78 79 80) T R S U M

CONTROL RECORD SPECIFICATION

SEQUENCE NUMBER (1-5)	FORM TYPE (H) (6)	(7-14)	DEBUG OPTIONS ENTER "1" TO SET: DUMP ALL VAR ON HO (15)	DUMP ALL IND ON HO (16)	LIST INPUT RECORDS (17)	(18-19)	OUTPUT DEVICE FOR SOURCE LISTING (20-27)	EXTERNAL INDICATORS ENTER "1" TO TURN ON, ELSE LEAVE BLANK TO TURN OFF: U1	U2	U3	U4	U5	U6	U7	U8	(36-43)	EXECUTE (E OR BLANK) (44)	SAVE (S OR BLANK) (45)	(46-74)
1	H						LP											S	

FILE DESCRIPTION SPECIFICATIONS

SEQUENCE NUMBER (1-5)	FORM TYPE (F) (6)	FILE NAME (7-14)	FILE TYPE: ENTER I, O, OR U (15)	FILE DESIGNATION: ENTER P, S, OR BLANK (16)	END OF FILE: ENTER E OR BLANK (17)	(18)	RECORD FORMAT: ENTER F, V (19)	(20-23)	RECORD LENGTH (1-9999) (24-27)	(28-32)	OVERFLOW INDICATOR ASSIGNED TO PRINTER (33-34)	(35-38)	EXTENSION: ENTER L OR BLANK (39)	DEVICE NAME ENTER DI, LP, PR, CR, PU, TE, OR MT (40-46)	(47-65)	APPEND FILE: ENTER A OR BLANK (66)	(67-74)
2	F	MNDATA	I	P			F		8Ø					DI			
3	F	TRSMRY	O				F		8Ø		OA			LP			
	F																
	F																
	F																
	F																
	F																
	F																
	F																
	F																
	F																
	F																
	F																
	F																
	F																

Fig. 10—Work Sheet for TRSUM (Sheet 1 of 8)

AT&T Co
RPG/CAROT

RPG/CAROT FILE EXTENSION AND LINE COUNTER SPECIFICATIONS

PAGE 2 OF 8

PROGRAMMER S. ASHCRAFT DATE 02/01/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75 76 77 78 79 80) T R S U M

FILE EXTENSION SPECIFICATIONS

SEQUENCE NUMBER (1-5)	FORM TYPE (E) (6)	(7-26)	ARRAY NAME (27-32)	(33-35)	NUMBER OF ELEMENTS IN ARRAY (DEFAULT = 1) (36-39)	LENGTH OF EACH ENTRY IN ARRAY (DEFAULT = 6) (40-42)	(43)	DECIMAL POS (0-9) (44)	(45-57)	COMMENTS (58-74)
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									

LINE COUNTER SPECIFICATIONS

SEQUENCE NUMBER (1-5)	FORM TYPE (L) (6)	FILE NAME (7-14)	FIRST LINE ON PAGE (1-60) (15-17)	(18-19)	OVERFLOW (LAST) LINE ON PAGE (1-60) (20-22)	(23-74)
4	L	TRSMRY	1		57	
	L					
	L					

Fig. 10—Work Sheet for TRSUM (Sheet 2 of 8)

AT&T Co
RPG/CAROT

RPG/CAROT INPUT SPECIFICATIONS

PAGE 3 OF 8

PROGRAMMER S. ASHCRAFT DATE 02/01/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75 76 77 78 79 80) TRSUM

INPUT RECORD DESCRIPTION

SEQUENCE NUMBER (1-5)	FORM TYPE (I) (6)	FILE NAME (7-14)	GROUP SEQUENCE (15-16)	NO. OF RECORDS (1-N) (17)	OPTIONAL (O) (18)	RECORD IDENTIFYING INDICATOR (19-20)	RIC 1 POSITION (21-24)	RIC 1 NOT (N) (25)	(26)	RIC 1 CHARACTER (27)	RIC 2 POSITION (28-31)	RIC 2 NOT (N) (32)	(33)	RIC 2 CHARACTER (34)	RIC 3 POSITION (35-38)	RIC 3 NOT (N) (39)	(40)	RIC 3 CHARACTER (41)	(42-43)	FIELD POSITION FROM (44-47)	FIELD POSITION TO (48-51)	DECIMAL POS (0-9) (52)	FIELD NAME (53-58)	CONTROL LEVEL INDICATOR (L1-L9) (59-60)	(61-64)	FIELD INDICATORS PLUS (65-66)	MINUS (67-68)	ZERO OR BLANK (69-70)	(71-74)
5	I	MNDATA	AA				2			C	3			T															
6	I		Ø1	1		Ø5	2			R	3			C															
7	I																			4	26		OFFICE						
8	I		Ø3	N		Ø6	2			M	3			T															
9	I																			4	9	Ø	SCHD						
1Ø	I																			11	16	Ø	TEST						
11	I																			18	23	Ø	Q1L						
12	I																			25	3Ø	Ø	Q1N						
13	I																			32	37	Ø	Q2L						
14	I																			39	44	Ø	Q2N						
15	I																			46	51	Ø	CHRN						
16	I																			53	58	Ø	BUSY						
17	I																			6Ø	65	Ø	HANDD						
18	I																			67	72	Ø	OTHERS						
19	I		Ø4	N		Ø7	2			M	3			0															
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												
	I																												

Fig. 10—Work Sheet for TRSUM (Sheet 3 of 8)

AT&T Co
RPG/CAROT

RPG/CAROT CALCULATION SPECIFICATIONS

PAGE 4 OF 8

PROGRAMMER S. ASHCRAFT DATE Ø2/Ø1/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75 76 77 78 79 80) TRSUM

SEQUENCE NUMBER (1-5)	FORM TYPE (C) (6)	CONTROL LEVEL (LO-L9, LR, AN/OR) (7-8)	INDICATORS NOT (9-11)	AND NOT (12-14)	AND NOT (15-17)	FACTOR 1 (18-27)	OPERATION (28-32)	FACTOR 2 (33-42)	RESULT FIELD NAME (43-48)	LENGTH (49-51)	DECIMAL POS (0-9) (52)	HALF ADJUST (H) (53)	PLUS / 1>2 / HIGH (54-55)	MINUS / 1<2 / LOW (56-57)	ZERO / 1=2 / EQUAL (58-59)	COMMENTS (60-74)
2Ø	C		Ø6			SCHD	COMP	Ø							14	
21	C		Ø6	N14		Q2L	ADD	Q2N	Q2S	6	Ø					
22	C		Ø6	N14		Q1L	ADD	Q1N	Q1S	6	Ø					
23	C		Ø7	N14		BUSY	ADD	HANDD	NOTDON	6	Ø					
24	C		Ø7	N14		NOTDON	ADD	OTHERS	NOTDON							
25	C		Ø7	N14		SCHD -	DIV	1ØØ.	SCHFAC	7	2					
26	C		Ø7	N14		TEST	DIV	1ØØ	TSTFAC	7	2					
27	C		Ø7	N14		TEST	DIV	SCHFAC	TESTPC	5	1					
28	C		Ø7	N14		SCHD	SUB	TEST	NOTRIE	6	Ø					
29	C		Ø7	N14		NOTRIE	SUB	NOTDON	NOTRIE					15		
3Ø	C	*														
31	C		Ø7	N14		Q2S	DIV	TSTFAC	Q2PC	5	1					
32	C		Ø7	N14		Q1S	DIV	TSTFAC	Q1PC	5	1					
33	C	*	SET NEGATIVE # OF TRIES TO Ø													
34	C	*														
35	C		15				Z-ADD	Ø.	NOTRIE							
36	C		Ø7	N14		NOTRIE	DIV	SCHFAC	TRYPC	5	1					
37	C		Ø7	N14		NOTDON	DIV	SCHFAC	DONPC	5	1					
38	C		Ø7	N14		TOTSCH	ADD	SCHD	TOTSCH	6	Ø					
39	C		Ø7	N14		TOTTST	ADD	TEST	TOTTST	6	Ø					
4Ø	C		Ø7	N14		TOTTRY	ADD	NOTRIE	TOTTRY	6	Ø					
41	C		Ø7	N14		TOTDON	ADD	NOTDON	TOTDON	6	Ø					
42	C		Ø7	N14		TOTQ2S	ADD	Q2S	TOTQ2S	6	Ø					
43	C		Ø7	N14		TOTQ1S	ADD	Q1S	TOTQ1S	6	Ø					
	C															

Fig. 10—Work Sheet for TRSUM (Sheet 4 of 8)

AT&T Co
RPG/CAROT

RPG/CAROT CALCULATION SPECIFICATIONS

PAGE 5 OF 8

PROGRAMMER S. ASHCRAFT DATE 02/01/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75–80) TRSUM

SEQUENCE NUMBER	FORM TYPE (C)	CONTROL LEVEL (L0-L9, LR, AN/OR)	INDICATORS (NOT / AND NOT / AND NOT)	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD NAME	RESULT FIELD LENGTH	DECIMAL POS (0-9)	HALF ADJUST (H)	RESULTING INDICATORS: ARITHMETIC PLUS / MINUS / ZERO; COMPARE 1>2 / 1<2 / 1=2; LOOKUP (FACTOR 2) IS HIGH / LOW / EQUAL	COMMENTS
44	C	*										
45	C	*	TOTAL TIME CALCULATIONS									
46	C	*										
47	C	LR		TOTSCH	DIV	100.	SCHFAC					
48	C	LR		TOTTST	DIV	100.	TSTFAC					
49	C	LR		TOTTST	DIV	SCHFAC	TESTPC					
50	C	LR		TOTDON	DIV	SCHFAC	DONPC					
51	C	LR		TOTTRY	DIV	SCHFAC	TRYPC					
52	C	LR		TOTQ2S	DIV	TSTFAC	Q2PC					
53	C	LR		TOTQ1S	DIV	TSTFAC	Q1PC					
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											
	C											

Fig. 10—Work Sheet for TRSUM (Sheet 5 of 8)

AT&T Co
RPG/CAROT

RPG/CAROT OUTPUT SPECIFICATION

PAGE 6 OF 8

PROGRAMMER S. ASHCRAFT DATE 02/01/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75 76 77 78 79 80) TRSUM

SEQUENCE NUMBER (1–5)	FORM TYPE (O) (6)	FILE NAME (7–13)	AND / OR (14–16)	TYPE (H/D/T) (15)	SPACE BEFORE (17)	SPACE AFTER (18)	SKIP BEFORE (19–20)	SKIP AFTER (21–22)	OUTPUT INDICATORS NOT / (23–25)	AND NOT / (26–28)	AND NOT / (29–31)	FIELD NAME (32–37)	EDIT CODE (38)	BLANK AFTER (B) (39)	END POSITION (1–9999) (40–43)	CONSTANT WORD (45–74)
54	O	*														
55	O	* NOW BEGIN OUTPUT														
56	O	*														
57	O	* OUTPUT HEADINGS														
58	O	*														
59	O	TRSMRY		H	Ø	Ø	1		1P							
60	O		OR						OA							
61	O														29	"CAROT - DAILY OFFICE "
62	O														5Ø	"TRANSMISSION SUMMARY"
63	O											UDATE			71	
64	O			H	Ø	1			1P							
65	O		OR						OA							
66	O														48	"% NOT % NOT"
67	O			H	Ø	Ø			1P							
68	O		OR						OA							
69	O														24	"ROTL/CONTROL OFFICE ID "
70	O														47	"SCHEDULED TRIED COMPL"
71	O														70	"ETED % TESTED % Q2 "
72	O														73	" % Q1"
73	O			H	Ø	2			1P							
74	O		OR						OA							
75	O														24	"----------------------- "
76	O														48	"------------------------"
77	O														70	"----------------------"
78	O														76	"------"

Fig. 10—Work Sheet for TRSUM (Sheet 6 of 8)

AT&T Co
RPG/CAROT

RPG/CAROT OUTPUT SPECIFICATION

PAGE 7 OF 8

PROGRAMMER S. ASHCRAFT DATE 02/01/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75 76 77 78 79 80) TRSUM

SEQUENCE NUMBER	FORM TYPE (O)	FILE NAME	TYPE (H/D/T)	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS	FIELD NAME	EDIT CODE	BLANK AFTER (B)	END POSITION (1-9999)	CONSTANT WORD
79	O	*											
8Ø	O	* ONLY OUTPUT THE FOLLOWING FOR THE MO RECORD											
81	O	*											
82	O		D	Ø	1		Ø7						
83	O								OFFICE			23	
84	O					N	14		SCHD	1		33	
85	O					N	14		TRYPC	1		41	
86	O					N	14		DONPC	1		51	
87	O					N	14		TESTPC	1		6Ø	
88	O					N	14		Q2PC	1		68	
89	O					N	14		Q1PC	1		76	
9Ø	O						14					50	"**** NOT SCHEDULED ****"
91	O	*											
92	O	* OUTPUT TOTALS											
93	O	*											
94	O		T	Ø	1		LR						
95	O											24	"---------------------- "
96	O											48	"------------------------"
97	O											7Ø	"----------------------"
98	O											76	"------"
99	O		T	Ø	1		LR						
1ØØ	O											7	"TOTALS"
1Ø1	O								TOTSCH	1		33	
1Ø2	O								TRYPC	1		41	
	O												

Fig. 10—Work Sheet for TRSUM (Sheet 7 of 8)

AT&T Co
RPG/CAROT

RPG/CAROT OUTPUT SPECIFICATION

PAGE 8 OF 8

PROGRAMMER S. ASHCRAFT DATE 02/01/78

PROGRAM TITLE TRANSMISSION SUMMARY REPORT

PROGRAM NAME (75 76 77 78 79 80) T R S U M

OUTPUT RECORD DESCRIPTION

SEQUENCE NUMBER (1–5)	FORM TYPE (O) (6)	FILE NAME (7–13)	OR / AND (14–16)	TYPE (H/D/T)	SPACE BEFORE (17)	SPACE AFTER (18)	SKIP BEFORE (19–20)	SKIP AFTER (21–22)	OUTPUT INDICATORS NOT (23–25)	AND NOT (26–28)	AND NOT (29–31)	FIELD NAME (32–37)	EDIT CODE (38)	BLANK AFTER (B) (39)	END POSITION (1–9999) (40–43)	CONSTANT WORD (45–74)
103	O											DONPC	1		51	
104	O											TESTPC	1		60	
105	O											Q2PC	1		68	
106	O											Q1PC	1		76	
107	O														42	"%"
108	O														51	"%"
109	O														60	"%"
100	O														68	"%"
111	O														76	"%"
112	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															
	O															

Fig. 10—Work Sheet for TRSUM (Sheet 8 of 8)

```
STRSUM- IS ON - CR ØØØØ4  TYPE    Ø4  USING   ØØ32  BLKS  R=  Ø148
DATE; Ø2 -14 -78  TIME: 12 :42  AM

12345H          1  BB                       S
     *
     **  @TRSUM IS A REPORT GENERATES THE DAILY OFFICE TRANSMISSION
     *  SUMMARY REPORT.@TRSUM  USES THE DATA FOUND IN THE FILE
     *  MNDATA WHICH WAS CREATED BY SELEC.  THIS DATA IS
     *  EITHER CUMULATIVE OR THE DAILY MANAGEMENT SUMMARY.
     *  DEPENDING ON WHAT IS IN THE FILE MNDATA  $SUMMARY EITHER GIVES
     *  THE DAILY SUMMARY OR A CUMULATIVE SUMMARY.
     *
     *  MNDATA IS THE FILE CREATED BY SELEC WITH THE LATEST
     *  MANAGEMENT SUMMARY INFORMATION
     *
     *  DEFINITION OF VARIABLES
     *
     *  SCHD       -  # OF TRUNKS SCHEDULED TO BE TESTED
     *  TEST       -  # OF TRUNKS TESTED
     *  Q1L        -  # OF TRUNKS WHICH ARE Q1 ON NOISE  TEST
     *  Q1N        -  # OF TRUNKS WHICH ARE Q1 ON NOISE TEST
     *  Q2L        -  # OF TRUNKS WHICH ARE Q2 ON LEVEL TEST
     *  Q2N        -  # OF TRUNKS WHICH ARE Q2 ON NOISE TEST
     *  Q1S        -  TOTAL OF Q1L + Q1N
     *  Q2S        -  TOTAL OF Q2L + Q2N
     *  CHRN       -  # OF TRUNKS WHICH ARE CHRONIC
     *  BUSY       -  # OF TRUNKS WHICH ARE BUSY
     *  HANDD      -  # OF TRUNKS WHICH ARE HIGH AND DRY
     *  OTHERS     -  # OF TRUNKS NOT TESTED WHICH DON'T IN ANY
     *                OF THE ABOVE CATEGORIES
     *  SCHFAC     -  # OF TRUNKS SCHEDULED/ 1ØØ
     *  TOTSCH     -  TOTAL # OF TRUNKS SCHEDULED
     *  TOTTST     -  TOTAL # OF TRUNKS TESTED
     *  TOTTRY     -  TOTAL # OF TRUNKS NOT TRIED
     *  TOTDON     -  TOTAL # OF TRUNKS NOT COMPLETED
     *  TRYPC      -  % TRIED
     *  DONPC      -  % NOT COMPLETED
     *  TESTPC     -  % TESTED
     *  FAILPC     -  % WHICH FAILED OPERATIONAL TESTING
     *  Q2PC       -  Q2S/TEST*1ØØ
     *  Q2PC       -  Q1S/TEST*1ØØ
     *
     * INPUT IS THE DISK FILE MNDATA CREATED BY SELEC
     * OUTPUT IS THE USER'S TERMINAL
     *
     FMNDATA  IP  F      8Ø            DI
     FTRSMRY  O   F      8Ø     OA     LP
     * NEXT LINE IS THE OVERFLOW INDICATOR SPECIFICATION AND SHOULD
     * BE INCLUDED IF OUTPUT IS TO GO TO LINE PRINTER
     LTRSMRY    1FL 57OL
     IMNDATA  AA       2  C   3  T
     I        Ø11 Ø5   2  R   3  C
     I                                          4  26 OFFICE
     I        Ø3N Ø6   2  M   3  T
     I                                          4   9ØSCHD
     I                                         11  16ØTEST
     I                                         18  23ØQ1L
     I                                         25  3ØØQ1N
     I                                         32  37ØQ2L
     I                                         39  44ØQ2N
```

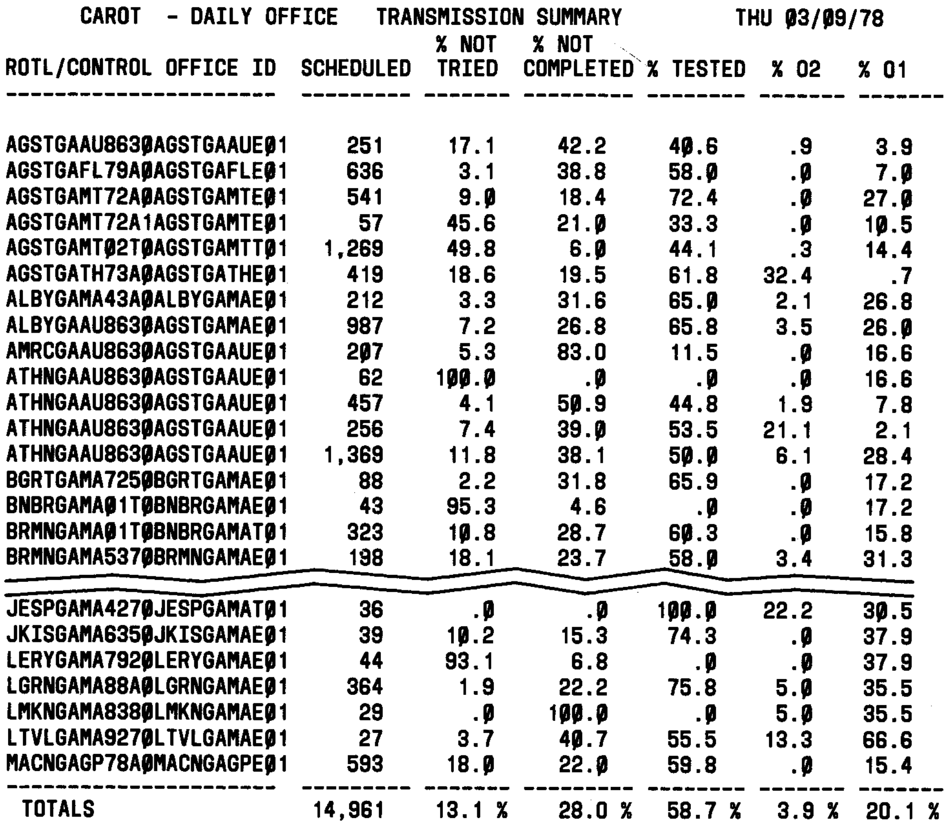
Fig. 11—Source Listing for TRSUM (Sheet 1 of 3)

```
ØØ58     I                                       46   51ØCHRN
ØØ59     I                                       53   58ØBUSY
ØØ6Ø     I                                       6Ø   65ØHANDD
ØØ61     I                                       67   72ØOTHERS
ØØ62     I        Ø4N Ø7   2  M   3  0
ØØ63     *
ØØ64     * BEGIN CALCULATIONS
ØØ65     *
ØØ66     *
ØØ67     * THE FOLLOWING CALCULATIONS ARE DONE AFTER EACH OPER RECORD (MO)
ØØ68     *
ØØ69     C   Ø6      SCHD      COMP Ø.                        14
ØØ7Ø     C   Ø6N14   Q2L       ADD  Q2N       Q2S     6Ø
ØØ71     C   Ø6N14   Q1L       ADD  Q1N       Q1S     6Ø
ØØ72     C   Ø7N14   BUSY      ADD  HANDD     NOTDON  6Ø
ØØ73     C   Ø7N14   NOTDON    ADD  OTHERS    NOTDON
ØØ74     C   Ø7N14   SCHD      DIV  1ØØ.      SCHFAC  72
ØØ75     C   Ø7N14   TEST      DIV  1ØØ.      TSTFAC  72
ØØ76     C   Ø7N14   TEST      DIV  SCHFAC    TESTPC  51
ØØ77     C   Ø7N14   SCHD      SUB  TEST      NOTRIE  6Ø
ØØ78     C   Ø7N14   NOTRIE    SUB  NOTDON    NOTRIE        15
ØØ79     *
ØØ8Ø     C   Ø7N14   Q2S       DIV  TSTFAC    Q2PC    51
ØØ81     C   Ø7N14   Q1S       DIV  TSTFAC    Q1PC    51
ØØ82     * SET NEGATIVE # OF NO TRIES TO Ø
ØØ83     *
ØØ84     C   15                Z-ADDØ.        NOTRIE
ØØ85     C   Ø7N14   NOTRIE    DIV  SCHFAC    TRYPC   51
ØØ86     C   Ø7N14   NOTDON    DIV  SCHFAC    DONPC   51
ØØ87     C   Ø7N14   TOTSCH    ADD  SCHD      TOTSCH  6Ø
ØØ88     C   Ø7N14   TOTTST    ADD  TEST      TOTTST  6Ø
ØØ89     C   Ø7N14   TOTTRY    ADD  NOTRIE    TOTTRY  6Ø
ØØ9Ø     C   Ø7N14   TOTDON    ADD  NOTDON    TOTDON  6Ø
ØØ91     C   Ø7N14   TOTQ2S    ADD  Q2S       TOTQ2S  6Ø
ØØ92     C   Ø7N14   TOTQ1S    ADD  Q1S       TOTQ1S  6Ø
ØØ93     *
ØØ94     * TOTAL TIME CALCULATIONS
ØØ95     *
ØØ96     CLR         TOTSCH    DIV  1ØØ.      SCHFAC
ØØ97     CLR         TOTTST    DIV  1ØØ.      TSTFAC
ØØ98     CLR         TOTTST    DIV  SCHFAC    TESTPC
ØØ99     CLR         TOTDON    DIV  SCHFAC    DONPC
Ø1ØØ     CLR         TOTTRY    DIV  SCHFAC    TRYPC
Ø1Ø1     CLR         TOTQ2S    DIV  TSTFAC    Q2PC
Ø1Ø2     CLR         TOTQ1S    DIV  TSTFAC    Q1PC
Ø1Ø3     *
Ø1Ø4     * NOW BEGIN OUTPUT
Ø1Ø5     *
Ø1Ø6     * OUTPUT HEADINGS
Ø1Ø7     *
Ø1Ø8     OTRSMRY  H ØØ 1    1P
Ø1Ø9     O       OR         OA
Ø11Ø     O                                   29 "CAROT  - DAILY OFFICE  "
Ø111     O                                   5Ø "TRANSMISSION SUMMARY"
Ø112     O                         UDATE     71
Ø113     O        H Ø1      1P
Ø114     O       OR         OA
Ø115     O                                   48 "% NOT   % NOT"
Ø116     O        H ØØ      1P
Ø117     O       OR         OA
```

Fig. 11—Source Listing for TRSUM (Sheet 2 of 3)

```
Ø118       O                                   24 "ROTL/CONTROL OFFICE ID  "
Ø119       O                                   47 "SCHEDULED  TRIED  COMPL"
Ø12Ø       O                                   7Ø "ETED % TESTED  % Q2    "
Ø121       O                                   73 " % Q1"
Ø122       O        H Ø2    1P
Ø123       O       OR       OA
Ø124       O                                   24"-----------------------  "
Ø125       O                                   48 "--------- ------- ------"
Ø126       O                                   7Ø "--- -------- ------- -"
Ø127       O                                   76 "------"
Ø128       *
Ø129       * ONLY OUTPUT THE FOLLOWING THE MO RECORD
Ø13Ø       *
Ø131       O        D Ø1    Ø7
Ø132       O                        OFFICE     23
Ø133       O               N14      SCHD  1    33
Ø134       O               N14      TRYPC 1    41
Ø135       O               N14      DONPC 1    51
Ø136       O               N14      TESTPC1    6Ø
Ø137       O               N14      Q2PC  1    68
Ø138       O               N14      Q1PC  1    76
Ø139       O                14                 5Ø "**** NOT SCHEDULED  ****"
Ø14Ø       *
Ø141       * OUTPUT TOTALS
Ø142       *
Ø143       O        T Ø1    LR
Ø144       O                                   24 "----------------------  "
Ø145       O                                   48 "--------- ------- ------"
Ø146       O                                   7Ø "--- -------- ------- -"
Ø147       O                                   76 "------"
Ø148       O        T Ø1    LR
Ø149       O                                    7 "TOTALS"
Ø15Ø       O                        TOTSCH1    33
Ø151       O                        TRYPC 1    41
Ø152       O                        DONPC 1    51
Ø153       O                        TESTPC1    6Ø
Ø154       O                        Q2PC  1    68
Ø155       O                        Q1PC  1    76
Ø156       O                                   42 "%"
Ø157       O                                   51 "%"
Ø158       O                                   6Ø "%"
Ø159       O                                   68 "%"
Ø16Ø       O                                   76 "%"
Ø161       * G RECORD INDICATES THE END OF THE SOURCE PROGRAM
Ø162       G
```

Fig. 11—Source Listing for TRSUM (Sheet 3 of 3)

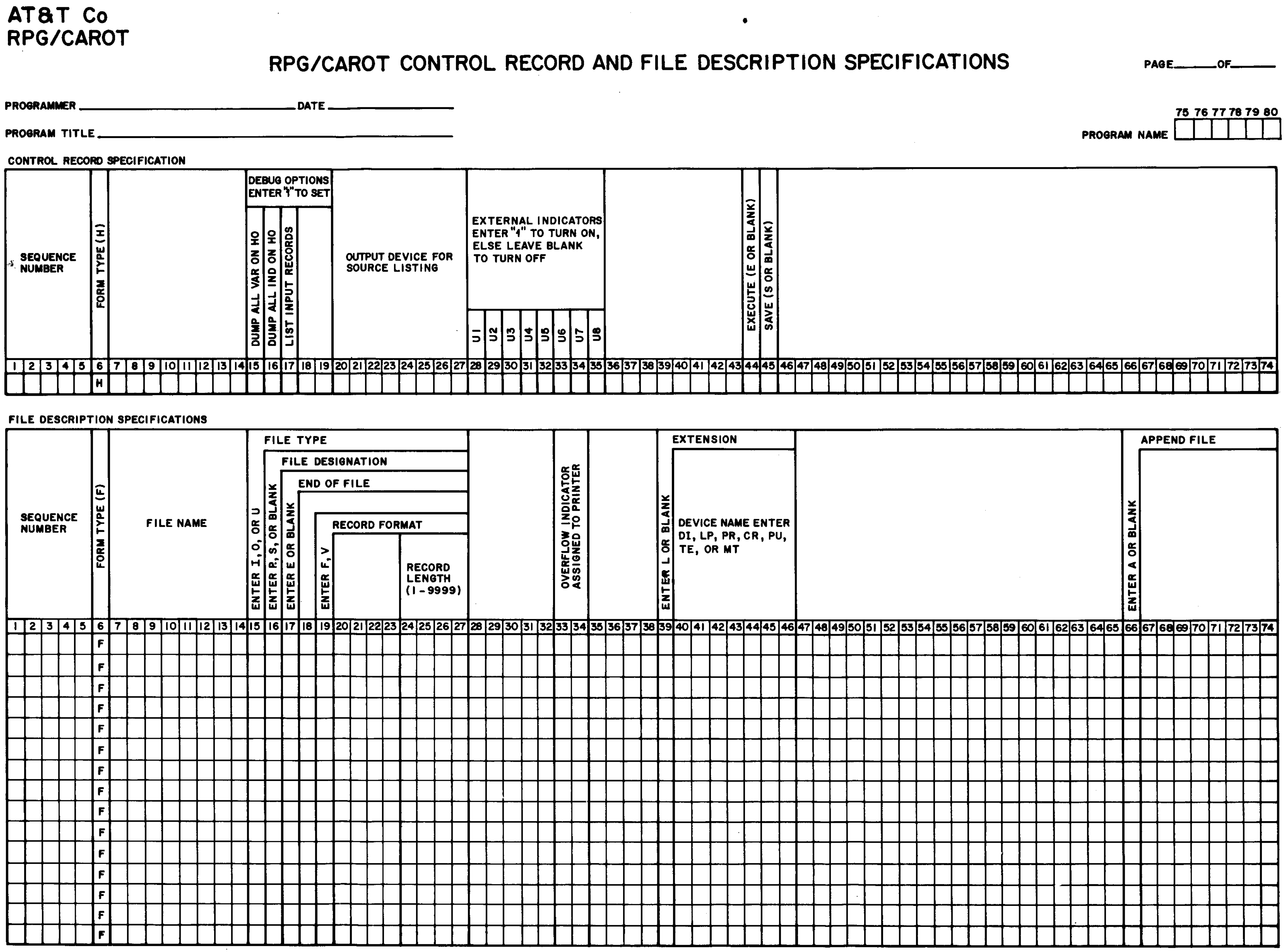


```
      CAROT  - DAILY OFFICE   TRANSMISSION SUMMARY        THU Ø3/Ø9/78
                                 % NOT    % NOT
ROTL/CONTROL OFFICE ID  SCHEDULED TRIED   COMPLETED % TESTED  % 02    % 01
----------------------  --------- ------- --------- -------- ------- -------

AGSTGAAU863ØAGSTGAAUEØ1     251     17.1      42.2     4Ø.6      .9     3.9
AGSTGAFL79AØAGSTGAFLEØ1     636      3.1      38.8     58.Ø      .Ø     7.Ø
AGSTGAMT72AØAGSTGAMTEØ1     541      9.Ø      18.4     72.4      .Ø    27.Ø
AGSTGAMT72A1AGSTGAMTEØ1      57     45.6      21.Ø     33.3      .Ø    1Ø.5
AGSTGAMTØ2TØAGSTGAMTTØ1   1,269     49.8       6.Ø     44.1      .3    14.4
AGSTGATH73AØAGSTGATHEØ1     419     18.6      19.5     61.8    32.4      .7
ALBYGAMA43AØALBYGAMAEØ1     212      3.3      31.6     65.Ø     2.1    26.8
ALBYGAAU863ØAGSTGAMAEØ1     987      7.2      26.8     65.8     3.5    26.Ø
AMRCGAAU863ØAGSTGAAUEØ1     2Ø7      5.3      83.0     11.5      .Ø    16.6
ATHNGAAU863ØAGSTGAAUEØ1      62    1ØØ.Ø        .Ø       .Ø      .Ø    16.6
ATHNGAAU863ØAGSTGAAUEØ1     457      4.1      5Ø.9     44.8     1.9     7.8
ATHNGAAU863ØAGSTGAAUEØ1     256      7.4      39.Ø     53.5    21.1     2.1
ATHNGAAU863ØAGSTGAAUEØ1   1,369     11.8      38.1     5Ø.Ø     6.1    28.4
BGRTGAMA725ØBGRTGAMAEØ1      88      2.2      31.8     65.9      .Ø    17.2
BNBRGAMAØ1TØBNBRGAMAEØ1      43     95.3       4.6       .Ø      .Ø    17.2
BRMNGAMAØ1TØBNBRGAMATØ1     323     1Ø.8      28.7     6Ø.3      .Ø    15.8
BRMNGAMA537ØBRMNGAMAEØ1     198     18.1      23.7     58.Ø     3.4    31.3

JESPGAMA427ØJESPGAMATØ1      36       .Ø        .Ø    1ØØ.Ø    22.2    3Ø.5
JKISGAMA635ØJKISGAMAEØ1      39     1Ø.2      15.3     74.3      .Ø    37.9
LERYGAMA792ØLERYGAMAEØ1      44     93.1       6.8       .Ø      .Ø    37.9
LGRNGAMA88AØLGRNGAMAEØ1     364      1.9      22.2     75.8     5.Ø    35.5
LMKNGAMA838ØLMKNGAMAEØ1      29       .Ø     1ØØ.Ø       .Ø     5.Ø    35.5
LTVLGAMA927ØLTVLGAMAEØ1      27      3.7      4Ø.7     55.5    13.3    66.6
MACNGAGP78AØMACNGAGPEØ1     593     18.Ø      22.Ø     59.8      .Ø    15.4
----------------------  --------- ------- --------- -------- ------- -------
  TOTALS                  14,961    13.1 %    28.0 %   58.7 %   3.9 %  20.1 %
```

Fig. 12—Output Generated by TRSUM



AT&T Co
RPG/CAROT

RPG/CAROT CONTROL RECORD AND FILE DESCRIPTION SPECIFICATIONS

PAGE_____OF_____

PROGRAMMER ____________________ DATE __________

PROGRAM TITLE ______________________________

75 76 77 78 79 80
PROGRAM NAME □□□□□□

CONTROL RECORD SPECIFICATION

SEQUENCE NUMBER	FORM TYPE (H)		DEBUG OPTIONS ENTER "1" TO SET				OUTPUT DEVICE FOR SOURCE LISTING	EXTERNAL INDICATORS ENTER "1" TO TURN ON, ELSE LEAVE BLANK TO TURN OFF									EXECUTE (E OR BLANK)	SAVE (S OR BLANK)	
			DUMP ALL VAR ON HO	DUMP ALL IND ON HO	LIST INPUT RECORDS			U1	U2	U3	U4	U5	U6	U7	U8				
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15	16	17	18 19	20 21 22 23 24 25 26 27	28	29	30	31	32	33	34	35	36 37 38 39 40 41 42 43	44	45	46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
	H																		

FILE DESCRIPTION SPECIFICATIONS

SEQUENCE NUMBER	FORM TYPE (F)	FILE NAME	FILE TYPE	FILE DESIGNATION	END OF FILE		RECORD FORMAT				OVERFLOW INDICATOR ASSIGNED TO PRINTER		EXTENSION			APPEND FILE
			ENTER I, O, OR U	ENTER R, S, OR BLANK	ENTER E OR BLANK		ENTER F, V		RECORD LENGTH (1-9999)				ENTER L OR BLANK	DEVICE NAME ENTER DI, LP, PR, CR, PU, TE, OR MT		ENTER A OR BLANK
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15	16	17	18	19	20 21 22 23	24 25 26 27	28 29 30 31 32	33 34	35 36 37 38	39	40 41 42 43 44 45 46	47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65	66 67 68 69 70 71 72 73 74
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															
	F															

Fig. 13—Blank Control Record and File Description Specification Work Sheet

AT&T Co
RPG/CAROT

RPG/CAROT FILE EXTENSION AND LINE COUNTER SPECIFICATIONS

PAGE_____OF_____

PROGRAMMER____________________ DATE__________

PROGRAM TITLE____________________________

PROGRAM NAME 75 76 77 78 79 80 □□□□□□

FILE EXTENSION SPECIFICATIONS

SEQUENCE NUMBER	FORM TYPE (E)		ARRAY NAME		NUMBER OF ELEMENTS IN ARRAY (DEFAULT = 1)	LENGTH OF EACH ENTRY IN ARRAY (DEFAULT = 6)		DECIMAL POS (0-9)		COMMENTS
1 2 3 4 5	6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	27 28 29 30 31 32	33 34 35	36 37 38 39	40 41 42	43	44	45 46 47 48 49 50 51 52 53 54 55 56 57	58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									
	E									

LINE COUNTER SPECIFICATIONS

SEQUENCE NUMBER	FORM TYPE (L)	FILE NAME	FIRST LINE ON PAGE (1-60)	FIRST LINE INDICATOR	OVERFLOW (LAST) LINE ON PAGE (1-60)	OVERFLOW LINE INDICATOR	
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15 16 17	18 19	20 21 22	23 24	25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
	L			F L		O L	
	L			F L		O L	
	L			F L		O L	

Fig. 14—Blank File Extension and Line Counter Specification Work Sheet

AT&T Co
RPG/CAROT

RPG/CAROT INPUT SPECIFICATIONS

PAGE ____ OF ____

PROGRAMMER ______________________ DATE __________

PROGRAM TITLE ______________________

PROGRAM NAME 75 76 77 78 79 80 □□□□□□

		INPUT RECORD DESCRIPTION																							
							RECORD IDENTIFICATION CODES											FIELD DESCRIPTION							
							1				2				3				FIELD POSITION						FIELD INDICATORS
SEQUENCE NUMBER	FORM TYPE (I)	FILE NAME	OR AND	GROUP SEQUENCE	NO. OF RECORDS (1-N)	OPTIONAL (O)	RECORD IDENTIFYING INDICATOR	POSITION	NOT (N)		CHARACTER	POSITION	NOT (N)		CHARACTER	POSITION	NOT (N)		CHARACTER		FROM	TO	DECIMAL POS (0-9)	FIELD NAME	CONTROL LEVEL INDICATOR (L1-L9)		PLUS	MINUS	ZERO OR BLANK	
1 2 3 4 5	6	7 8 9 10 11 12 13	14	15 16	17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38	39	40	41	42 43	44 45 46 47	48 49 50 51	52	53 54 55 56 57 58	59 60	61 62 63 64	65 66	67 68	69 70	71 72 73 74
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													
	I																													

Fig. 15—Blank Input Specification Work Sheet

AT&T Co
RPG/CAROT

RPG/CAROT CALCULATION SPECIFICATIONS

PAGE______OF______

PROGRAMMER________________________ DATE ________________

PROGRAM TITLE____

PROGRAM NAME □□□□□□ (75 76 77 78 79 80)

SEQUENCE NUMBER	FORM TYPE (C)	CONTROL LEVEL (LO-L9, LR, AN/OR)	INDICATORS			FACTOR I	OPERATION	FACTOR 2	RESULT FIELD		DECIMAL POS (0-9)	HALF ADJUST (H)	RESULTING INDICATORS			COMMENTS
				AND	AND				NAME	LENGTH			ARITHMETIC PLUS	MINUS	ZERO	
													COMPARE 1>2	1<2	1=2	
			NOT	NOT	NOT								LOOKUP (FACTOR 2) IS HIGH	LOW	EQUAL	
1 2 3 4 5	6	7 8	9 10 11	12 13 14	15 16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															
	C															

Fig. 16—Blank Calculation Specification Work Sheet

AT&T Co
RPG/CAROT

RPG/CAROT OUTPUT SPECIFICATION

PAGE______OF______

PROGRAMMER____________________________DATE______________

PROGRAM TITLE____

PROGRAM NAME □□□□□□ (75 76 77 78 79 80)

		OUTPUT RECORD DESCRIPTION																			
						SPACE		SKIP		FIELD DESCRIPTION											
										OUTPUT INDICATORS											
											AND		AND								
SEQUENCE NUMBER	FORM TYPE (O)	FILE NAME	O A	TYPE (H/D/T) R N	D	BEFORE	AFTER	BEFORE	AFTER	NOT		NOT		NOT		FIELD NAME	EDIT CODE	BLANK AFTER (B)	END POSITION (1-9999)		CONSTANT WORD
1 2 3 4 5	6	7 8 9 10 11 12 13	14	15	16	17	18	19 20	21 22	23	24 25	26	27 28	29	30 31	32 33 34 35 36 37	38	39	40 41 42 43	44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				
	O																				

Fig. 17—Blank Output Specification Work Sheet

AT&T Co
RPG/CAROT

COMMENT SHEET

PAGE______OF______

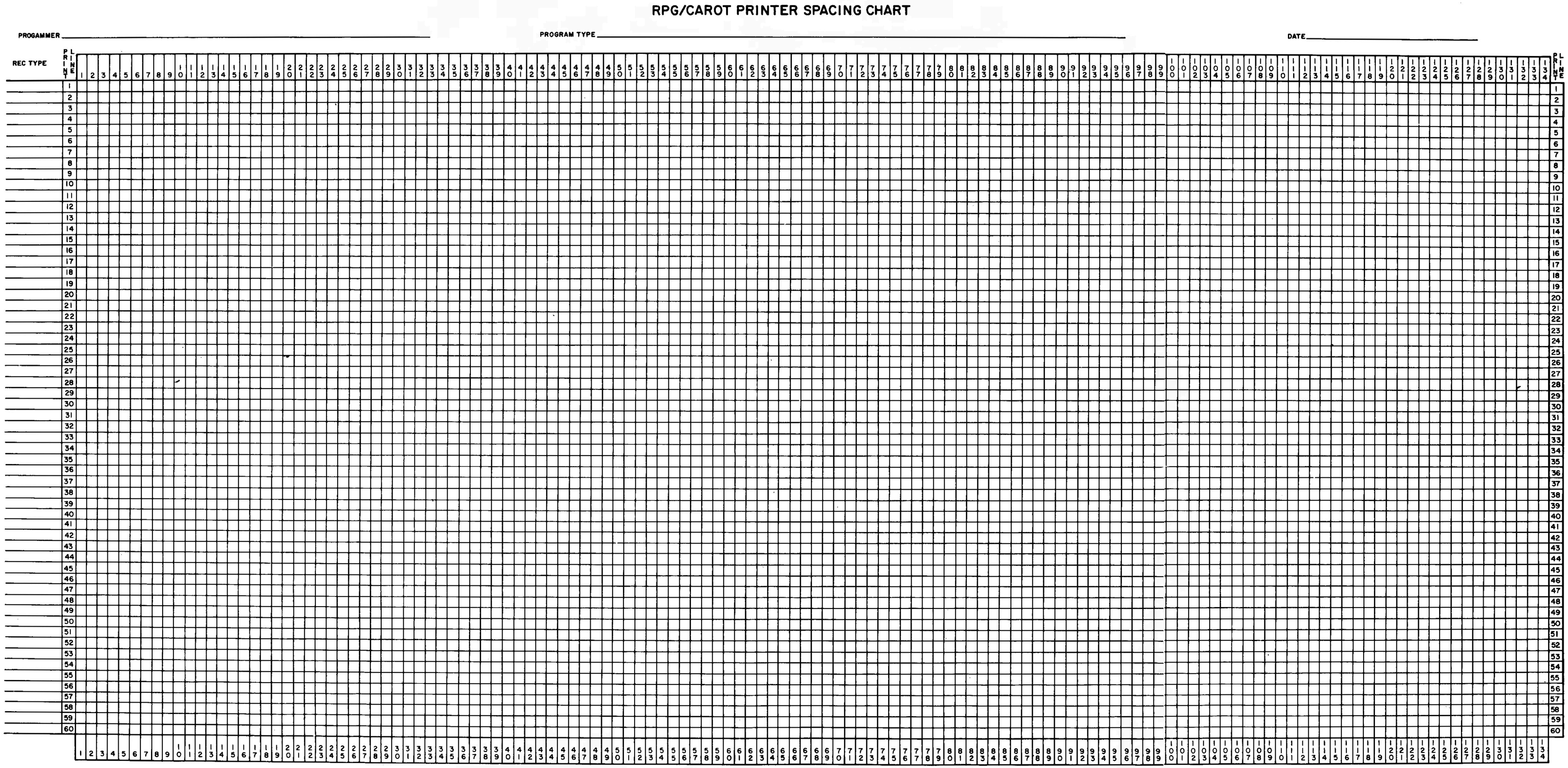
PROGRAMMER______________________________DATE______________

PROGRAM TITLE____

	75	76	77	78	79	80
PROGRAM NAME						

SEQUENCE NUMBER						COMMENT																																																																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				
					*																																																																				

Fig. 18—Blank Comment Sheet



RPG/CAROT PRINTER SPACING CHART

PROGAMMER ____________________ PROGRAM TYPE ____________________ DATE ____________________

REC TYPE

PRINT LINE

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60

Fig. 19—Blank Printer Spacing Chart