GC21-7514-1

Systems

Introduction to RPG II

IBM

GC21-7514-1

Systems Introduction to RPG II

Second Edition (December 1971)

This is a major revision of, and obsoletes, GC21-7514-0. Information about printers with carriage control tapes has been added. Changes are indicated by a vertical line at the left of each change.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

Address comments concerning this publication to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901.

Preface

This book is intended for persons who want to learn computer programming using the RPG II language. The book is designed to be used by a person with no previous knowledge of computers and programming, or by a person who already knows a programming language but wants to learn about RPG II.

The first chapter describes, in general terms, how a computer operates and the things a programmer must do to make the computer work. The information presented should answer such questions as:

- What are the parts of a computer?
- What is a computer program?
- What is a programming language?
- What is an RPG II program?
- How is an RPG II program run on a computer?

The second chapter describes the RPG II program cycle and the RPG II specifications a programmer must write to do a particular task. The material in this chapter is arranged to provide a gradual development of concepts, proceeding from the simple to the more complex. Thus, it is important to read the material in sequence. Sample jobs are used to illustrate the concepts presented.

The third chapter explains an RPG II programmer's job more fully. It shows, by means of a sample job, the things you must do from the start of a job to its completion.

After reading this book, the reader should not expect to be able to write complex RPG II programs. However, he should have gained enough background knowledge so that he can readily learn more detailed information—either from reference manuals, classes, or IBM personnel—which is required for writing programs for his computer.

Contents

Chapter 1. Basic Computer and Programming Concepts

PARTS OF A COMPUTER

Computers differ widely in appearance, usually consisting of several connected units. Regardless of their size or shape, however, all computers have common features.

Computers operate electronically. If you were to look inside the units of a computer, you would see thousands of circuits and wires. Fortunately, you don't have to understand the purpose of each circuit or wire. As a programmer, all you need to know are the purposes of the main parts of the computer: the input devices, the output devices, and the processing unit. Note that some devices are used for both input and output.

Input Devices

Data you give a computer to work with is called *input.* The device used for getting that data into the computer is called an *input device.* Several kinds of input devices are used: card readers, disk units, and keyboards. Card readers, of course read cards containing data in the form of punched holes; disk units read data recorded in the form of magnetic dots on disks; keyboards, which operate like a typewriter, transfer data directly into the computer.

Output Devices

Data produced by a computer is called *output.* The device that produces the output is called an *output device.* Several kinds of output devices are used: card punches, printers, and disk units. Card punches place data in cards in the form of holes; printers print data on paper; disk units record data in the form of magnetic dots on disk.

Processing Unit

The main part of a computer is the processing unit. The processing unit can be divided into three sections–storage, control, and arithmetic/logic–according to the special function each performs.

Storage

Storage is the computer's memory area. This area is divided into many storage positions which the computer uses to electronically store information. The actual number of positions in storage depends upon the size of the storage unit. Each storage position has an identifying number called an *address.*

A storage address serves the same purpose as a house address. Information is sent to and from these locations. The information can be easily retrieved using the address where the information is stored.

Arithmetic/Logic

Calculations (such as add, subtract, multiply, and divide) are performed in the arithmetic/logic section. When your instructions tell the computer to do an operation such as add, the information to be added is transferred from storage to the arithmetic/logic section. The operation is then performed and the result is sent back to storage.

Control

The control section is the computer's decision maker. It retrieves instructions from storage, determines what has to be done, and directs other units or devices to perform the required operations.

PROGRAMS AND PROGRAMMING LANGUAGES

Computers do only what you tell them. When you give a computer instructions, however, it might seem as though the computer requires more than you would need to do the same job. But remember, a computer cannot think: it requires explicit instructions, even for those things you would do almost without thinking.

When you are to do a job yourself, you need three basic things:

- Information to work with (input).
- Instructions telling you how to work with (process) the information.
- Additional instructions describing the expected results (output).

In computer terms, input is what you put into the computer, processing is what the computer does with the input, and output is the result of processing. Every job you run on a computer has these three parts. You write instructions to describe what you want the computer to do with each part. These instructions are called a *program.*

To communicate with the computer, you must use the computer's language, or one that can be translated into that language. The computer's language is called *machine language.* It consists of letters, numbers, and symbols that, when properly arranged, have a specific meaning to the computer and, when interpreted by the computer, cause it to perform a desired function.

Since machine language is so very different from our own language, it is extremely difficult to use it to write a program. For this reason, programming languages have been created. A programming language allows the programmer to use familiar words and symbols to write instructions.

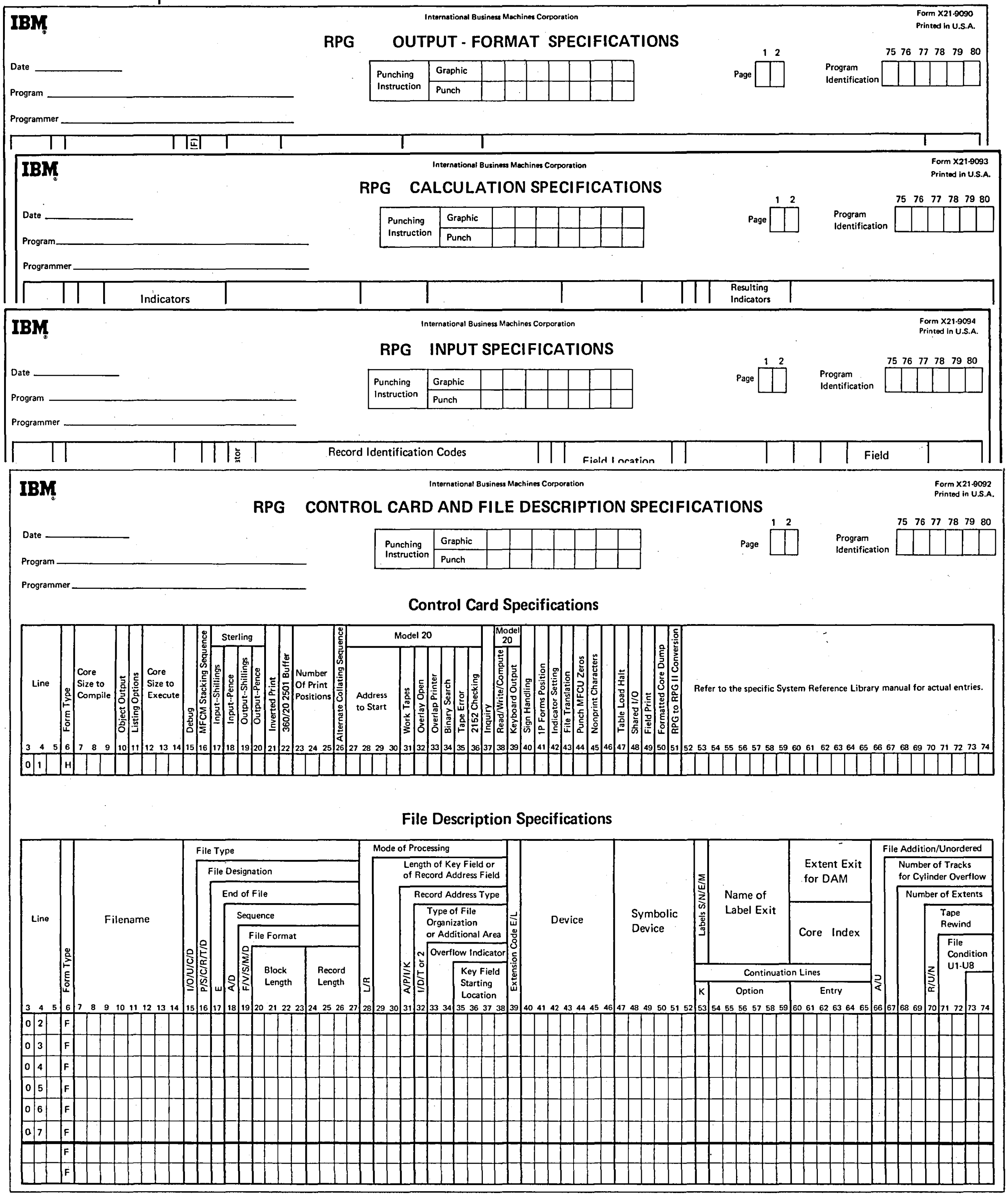
The RPG II programming language is composed of letters, numbers, and symbols which you put together to form an instruction (express a thought). When creating instructions in the RPG II language, you must follow certain rules just as you would when constructing a sentence in English. You will learn about these rules in the second part of this manual.

The set of instructions you write is called a *source program.* The source program is translated by the computer, resulting in a machine language program called the *object program.* It is the object program that you use to do a job. In fact, you can use it over and over to do the same job.

Source Programs

The instructions you write for any program must describe the input, processing, and output requirements of the job. For example, one instruction might direct the computer to read a punched card, another might specify the adding of two numbers, and another may tell the computer to print a line on the printer. Since all jobs are not the same, you provide a different set of instructions (program) for each job.

To write the instructions, you fill out RPG II specification sheets (see Figure 1). These sheets have been specially designed to help you write instructions according to the rules of the RPG II language. The act of writing instructions on these sheets is called *coding*; the entries you make on the sheets are called *specifications.*



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

(F)

IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

Indicators

Resulting Indicators

IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

ator

Record Identification Codes

Field Location

Field

IBM

International Business Machines Corporation

Form X21-9092
Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

Control Card Specifications

Line	Form Type	Core Size to Compile	Object Output	Listing Options	Core Size to Execute	Debug	MFCM Stacking Sequence	Sterling Input-Shillings	Sterling Input-Pence	Sterling Output-Shillings	Sterling Output-Pence	Inverted Print	360/20 2501 Buffer	Number Of Print Positions	Alternate Collating Sequence	Model 20 Address to Start	Model 20 Work Tapes	Model 20 Overlay Open	Model 20 Overlap Printer	Model 20 Binary Search	Model 20 Tape Error	Model 20 2152 Checking	Inquiry	Model 20 Read/Write/Compute	Model 20 Keyboard Output	Sign Handling	1P Forms Position	Indicator Setting	File Translation	Punch MFCU Zeros	Nonprint Characters		Table Load Halt	Shared I/O	Field Print	Formatted Core Dump	RPG to RPG II Conversion	Refer to the specific System Reference Library manual for actual entries.
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
0 1	H																																					

File Description Specifications

Line	Form Type	Filename	File Type I/O/U/C/D	File Designation P/S/C/R/T/D	End of File E	Sequence A/D	File Format F/V/S/M/D	Block Length	Record Length	Mode of Processing L/R	Length of Key Field or of Record Address Field	Record Address Type A/P/I/K	Type of File Organization or Additional Area I/D/T or 2	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device	Symbolic Device	Labels S/N/E/M Continuation Lines K	Name of Label Exit Option	Extent Exit for DAM Core Index Entry	File Addition/Unordered A/U	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind R/U/N	File Condition U1-U8	
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
0 2	F																										
0 3	F																										
0 4	F																										
0 5	F																										
0 6	F																										
0 7	F																										
	F																										
	F																										

Figure 1. RPG II Specification Sheets

To describe the input, processing, and output requirements of your job, you supply different information on each sheet. For example, you have to describe what your input data is like and specify the device (such as a card reader or a disk unit) that will read it. You also have to describe how the input data is to be processed. This includes specifying what type of operations (add, subtract, etc.) must be performed upon the data. Finally you specify what kind of output you want (printed report; punched cards), what information must be included in the output, how that information should be arranged, and which device will produce the output.

After you've coded the specifications sheets, the next step is to get the coded information into the computer. The computer can't read the coded sheets, so you must put the specifications into a form the computer can read. Depending on your system, you could enter the specifications on punched cards or through a keyboard.

Source Programs to Object Programs

As we said earlier, the computer understands only machine language. It cannot use a programming language like RPG II directly. Any program you write in RPG II must be translated into machine language. The translator is a computer program called a *compiler.* The RPG II Compiler program is available from IBM.

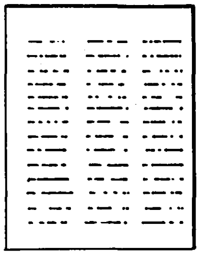
The compiler translates your RPG II specifications (source program) into machine language (object program). The translating it does is called *compilation.* Essentially, the compiler performs three functions during compilation:

1. Determines what machine instructions are necessary for the computer to perform the job described by your RPG II specifications.

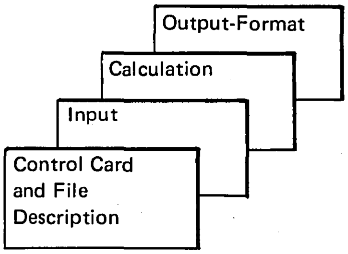
2. Translates your RPG II specifications into a machine language program.

3. Assigns storage locations to program instructions and data.

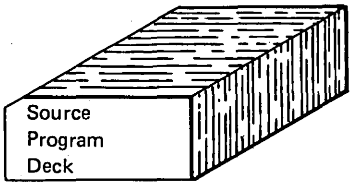
SUMMARY



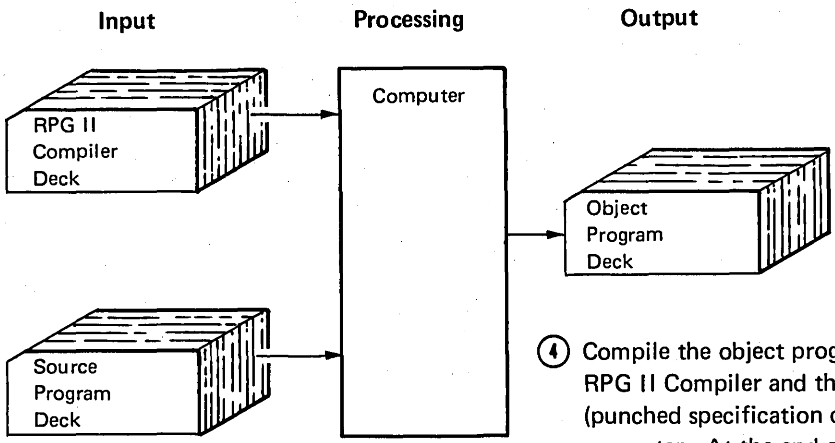
① Determine the requirements of your job. Define the input and required output. Also decide what processing must be done in order to get the proper results.



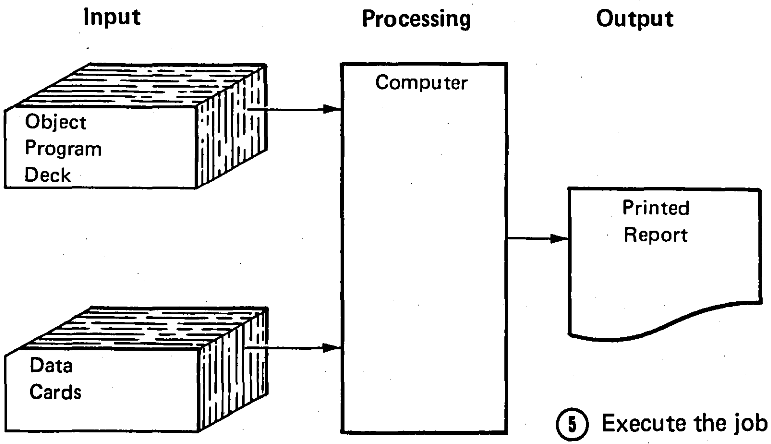
② Write the source program by describing your job on the RPG II specification sheets.



③ Punch the specifications on cards.



④ Compile the object program. Place the RPG II Compiler and the source deck (punched specification cards) into the computer. At the end of the run, a machine language program (object program) will be punched on blank cards.



⑤ Execute the job. Place the object program deck and the data cards into the computer. At the end of the run, output, such as a printed report, is produced.

RPG II Program Cycle

When you do any job, you must do it in a particular order. The computer must also do its job in a particular order. This logical order for the computer program is supplied by the RPG II Compiler.

The logic the compiler supplies is called a *program cycle* (see Figure 2). The object program goes through this cycle of operations every time a record is processed. Depending on your specifications, the object program may or may not use a particular operation in the cycle. However, the program still goes through the complete program cycle every time. Since one program cycle is needed for each record read, many program cycles are required for every job.

Note: The program cycle shown gives the general order of the operations. There may be minor variations between this cycle and the detailed cycle discussed in the reference manual applicable to your system.

You do not need to memorize the program cycle. The cycle is only shown at this time to give you an idea of the cycle of the operations. The operations will be discussed in greater detail later.

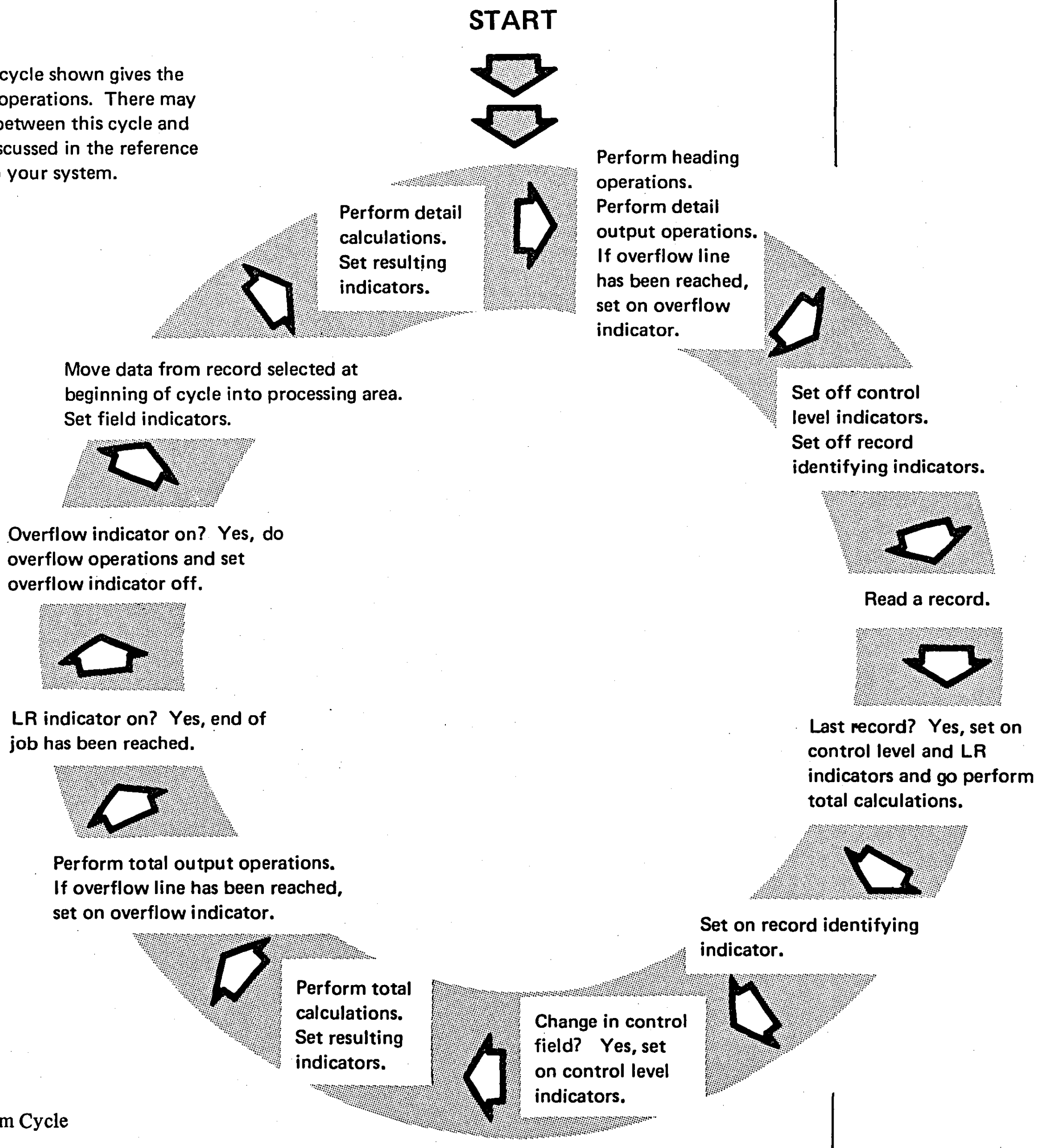

START
Perform heading operations.
Perform detail output operations.
If overflow line has been reached, set on overflow indicator.
Set off control level indicators.
Set off record identifying indicators.
Read a record.
Last record? Yes, set on control level and LR indicators and go perform total calculations.
Set on record identifying indicator.
Change in control field? Yes, set on control level indicators.
Perform total calculations.
Set resulting indicators.
Perform total output operations.
If overflow line has been reached, set on overflow indicator.
LR indicator on? Yes, end of job has been reached.
Overflow indicator on? Yes, do overflow operations and set overflow indicator off.
Move data from record selected at beginning of cycle into processing area.
Set field indicators.
Perform detail calculations.
Set resulting indicators.

Figure 2. Program Cycle

It is important that you, the programmer, know the order of the operations in the RPG II program cycle. This enables you to write specifications that will make correct use of the cycle. By knowing the order in which the operations in the cycle are performed, you can organize your program correctly.

In this chapter, the operations in the RPG II program cycle are explained a few at a time. You will learn:

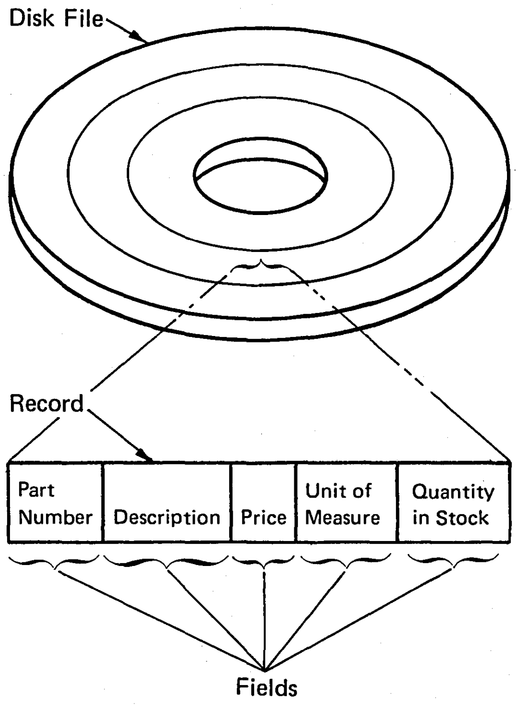
- Which operations are used for a particular function.

- Which RPG II specifications you must write to use the function.

Data Processing Terms and Programming Aids

In the discussion of RPG II, you will find reference to data processing terms and programming aids, which are described in the following illustrations.

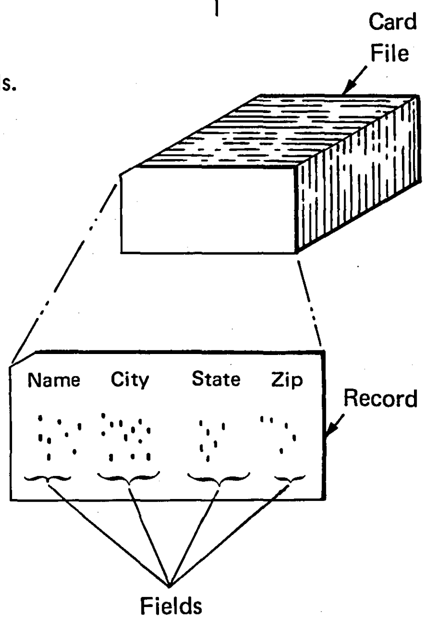
BASIC PROGRAMMING TERMS

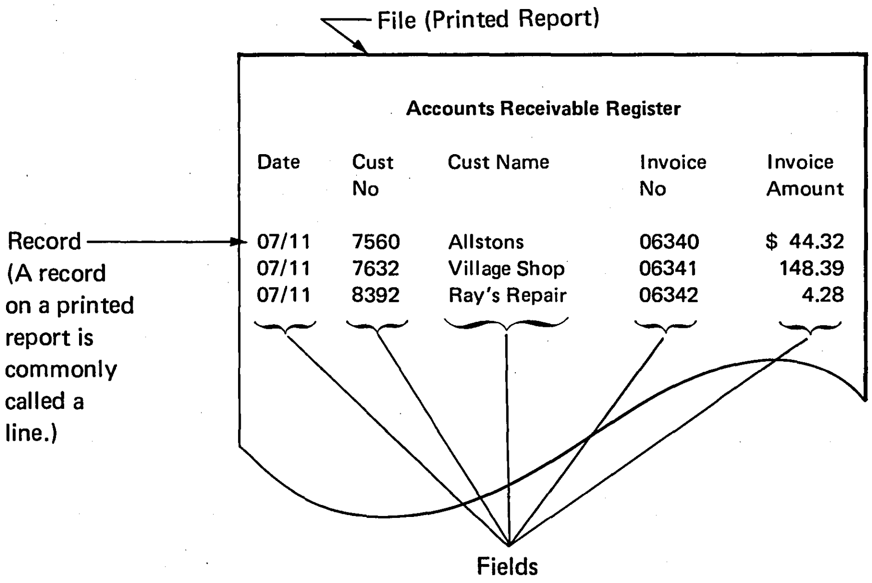


Field: An area on a record reserved and used for a particular item of information.

Record: A group of related fields.

File: A group of related records.

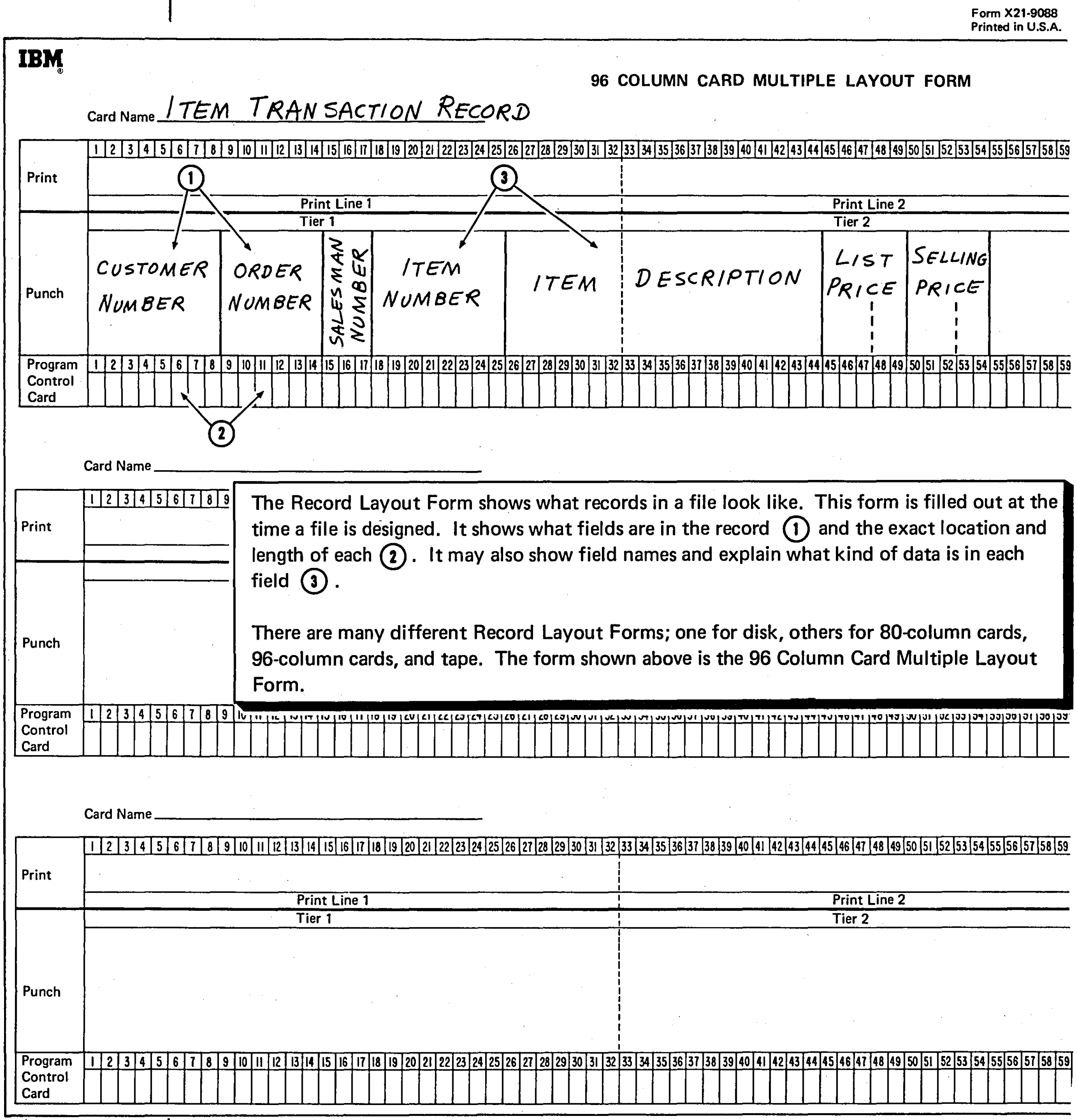




Accounts Receivable Register

Date	Cust No	Cust Name	Invoice No	Invoice Amount
07/11	7560	Allstons	06340	$ 44.32
07/11	7632	Village Shop	06341	148.39
07/11	8392	Ray's Repair	06342	4.28

Record Layout Form



Form X21-9088
Printed in U.S.A.

IBM

96 COLUMN CARD MULTIPLE LAYOUT FORM

Card Name ITEM TRANSACTION RECORD

	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
Print	①			③				
	Print Line 1				Print Line 2			
	Tier 1				Tier 2			
Punch	CUSTOMER NUMBER	ORDER NUMBER	SALESMAN NUMBER	ITEM NUMBER	ITEM DESCRIPTION	LIST PRICE	SELLING PRICE	
Program Control Card	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

②

Card Name

Print

Punch

Program Control Card

The Record Layout Form shows what records in a file look like. This form is filled out at the time a file is designed. It shows what fields are in the record ① and the exact location and length of each ②. It may also show field names and explain what kind of data is in each field ③.

There are many different Record Layout Forms; one for disk, others for 80-column cards, 96-column cards, and tape. The form shown above is the 96 Column Card Multiple Layout Form.

Card Name

Print

Print Line 1 | Print Line 2

Tier 1 | Tier 2

Punch

Program Control Card

Printer Spacing Chart

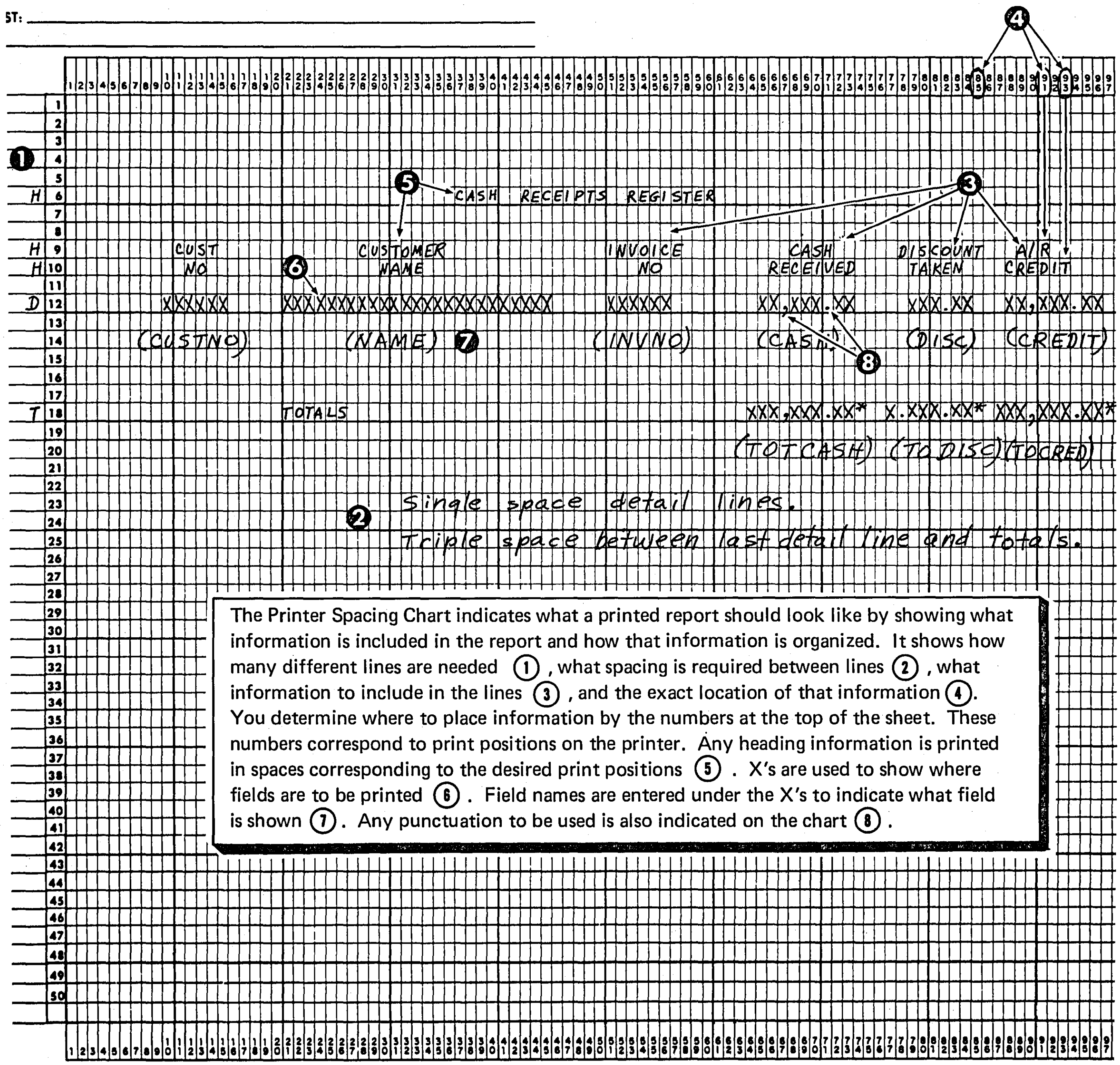
Form GX20-1776

ЭG. ID. ______________________ PAGE ________

:HARACTERS PER INCH, 6 LINES PER VERTICAL INCH) DATE ________

ST: ______________________

Fold back at dotted line.



The Printer Spacing Chart indicates what a printed report should look like by showing what information is included in the report and how that information is organized. It shows how many different lines are needed (1), what spacing is required between lines (2), what information to include in the lines (3), and the exact location of that information (4). You determine where to place information by the numbers at the top of the sheet. These numbers correspond to print positions on the printer. Any heading information is printed in spaces corresponding to the desired print positions (5). X's are used to show where fields are to be printed (6). Field names are entered under the X's to indicate what field is shown (7). Any punctuation to be used is also indicated on the chart (8).

Fold back at dotted line.

Writing Specifications For Input And Output Operations

One of the simplest jobs you can do on a computer is read information from an input record, such as a card, then put that same information out, such as in the form of a printed report. No calculations are done.

PROGRAM CYCLE OPERATIONS

To do this simple job, the computer uses only the three most basic operations in the RPG II program cycle. Figure 3 shows these operations.

Notice that two operations are concerned with the basic requirements of a job: input (read a record) and output (detail output). The third operation is the movement of data inside the computer.

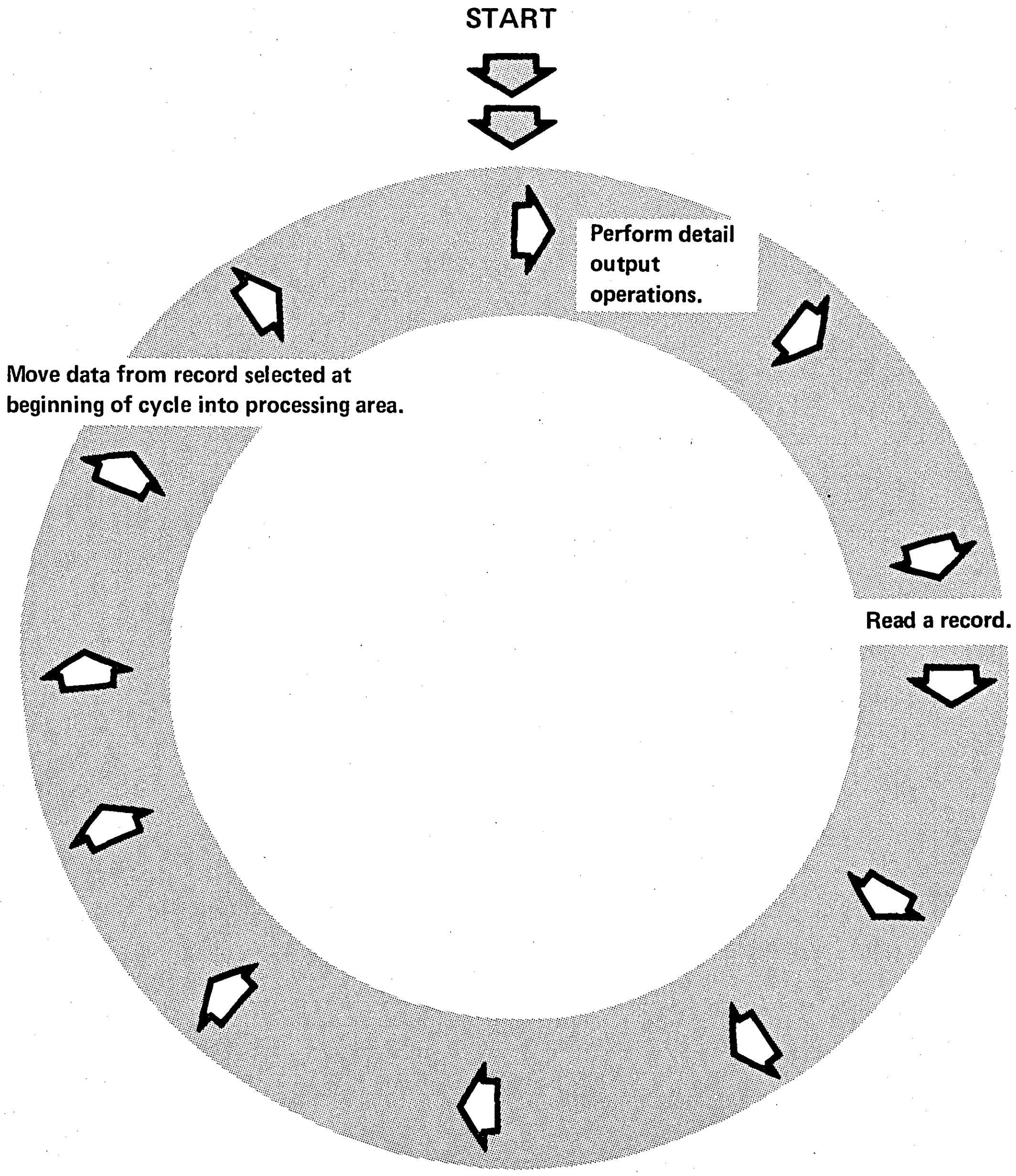

START
Perform detail output operations.
Move data from record selected at beginning of cycle into processing area.
Read a record.

Figure 3. Three Basic Operations in the RPG II Program Cycle

Data read by an input device must be transferred to the computer's processing unit before it can be used. Moving data is a mandatory operation done for every job. Because this operation is mandatory and is done exactly the same for every job, the compiler automatically supplies instructions to do it.

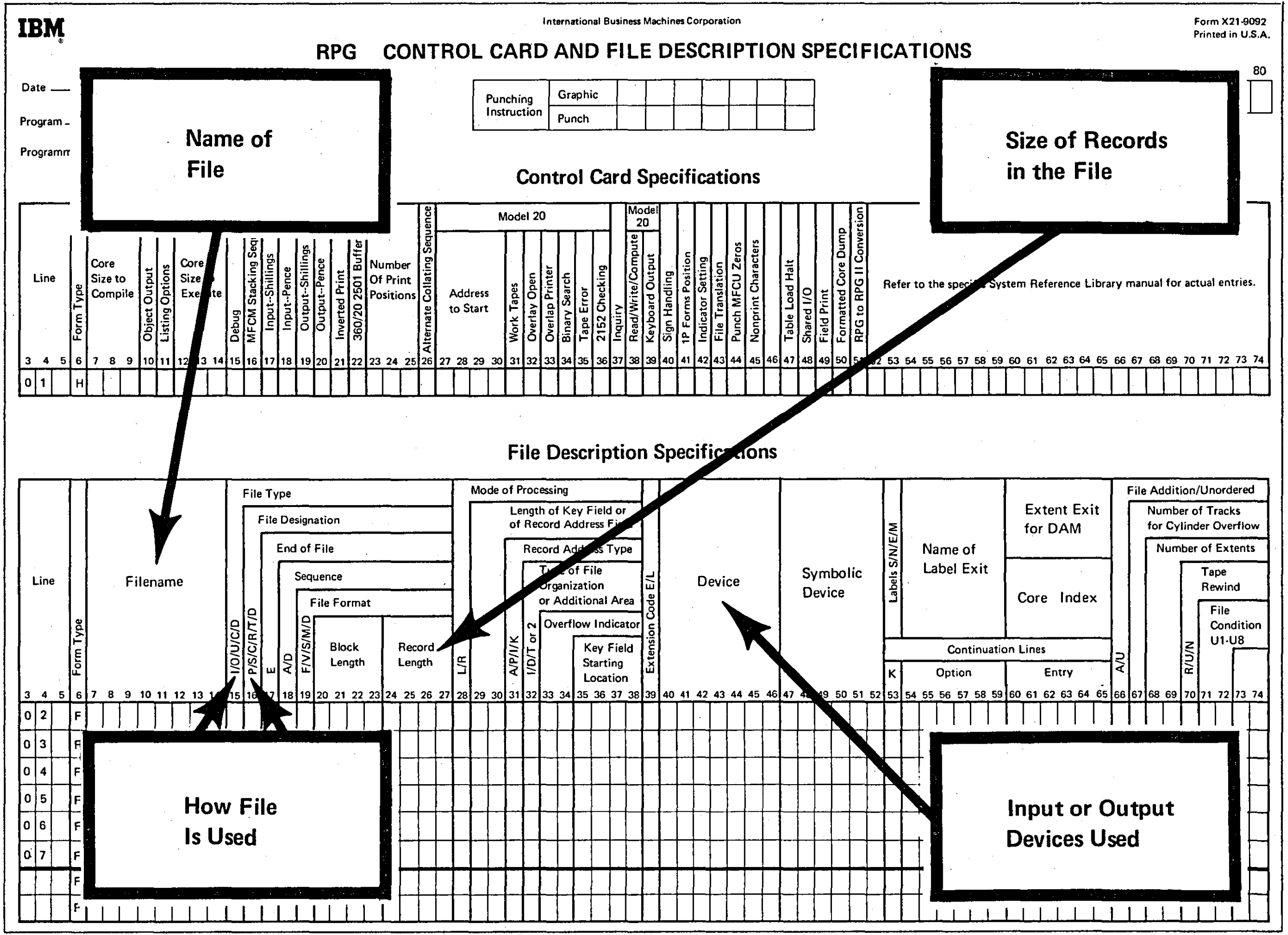
When the program is executed, the program cycle is repeated over and over. All three operations are used for every record in the input file. The term *detail output*, in the cycle operation, means that the specified output operations are performed for every input record.

It may seem strange that detail output comes before a record is read. This occurs, however, so that headings can be printed on a report. If a job (such as this one) does not print headings, no information is printed during the first cycle.

To make proper use of these cycle operations, your specifications must describe the records in the input file and specify how the output records should be created. You must also indicate what devices are used in the job.

DESCRIBING THE FILES

The File Description sheet is used to describe all the files used by your program. This information includes the name of the file, the device used with the file, and how the file is to be used. You fill out the indicated columns on the bottom half of the sheet:



IBM

International Business Machines Corporation

Form X21-9092
Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date
Program
Programmer

Punching Instruction — Graphic / Punch

80

Control Card Specifications

Line | Form Type | Core Size to Compile | Object Output | Listing Options | Core Size to Execute | Debug | MFCM Stacking Sequence | Input-Shillings | Input-Pence | Output-Shillings | Output-Pence | Inverted Print | 360/20 2501 Buffer | Number Of Print Positions | Alternate Collating Sequence | Model 20: Address to Start, Work Tapes, Overlay Open, Overlap Printer, Binary Search, Tape Error, 2152 Checking | Inquiry | Model 20: Read/Write/Compute | Keyboard Output | Sign Handling | 1P Forms Position | Indicator Setting | File Translation | Punch MFCU Zeros | Nonprint Characters | Table Load Halt | Shared I/O | Field Print | Formatted Core Dump | RPG to RPG II Conversion

Refer to the specific System Reference Library manual for actual entries.

0 1 H

File Description Specifications

Line | Form Type | Filename | File Type (I/O/U/C/D) | File Designation (P/S/C/R/T/D) | End of File (E) | Sequence (A/D) | File Format (F/V/S/M/D) | Block Length | Record Length | Mode of Processing (L/R) | Length of Key Field or of Record Address Field | Record Address Type (A/P/I/K) | Type of File Organization or Additional Area (I/D/T or 2) | Overflow Indicator | Key Field Starting Location | Extension Code E/L | Device | Symbolic Device | Labels S/N/E/M | Name of Label Exit | Extent Exit for DAM | Core Index | Continuation Lines: K, Option, Entry | A/U | File Addition/Unordered | Number of Tracks for Cylinder Overflow | Number of Extents | Tape Rewind (R/U/N) | File Condition U1-U8

0 2 F
0 3 F
0 4 F
0 5 F
0 6 F
0 7 F
F
F

Name of File

Size of Records in the File

How File Is Used

Input or Output Devices Used

You must describe, on a separate line, every file used in your job. Many simple jobs require only one input and one output file. In the first jobs we discuss, therefore, we will use only one input and one output file.

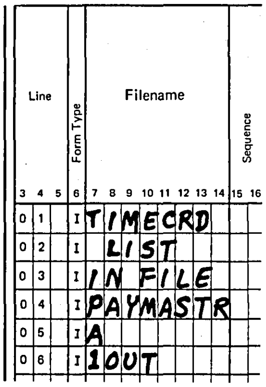
File Names

Every file used in a job must be named. The name provides you and the compiler a means of identifying the file. During compilation, the compiler associates the file name with other characteristics of the file. Thus, you can refer to that file by name throughout your program and the compiler knows exactly which file you are referring to.

The compiler, however, recognizes file names only if they conform to these rules:

- A file name must be 1-8 characters long.
- The first character of a file name must be alphabetic. (The letters A-Z and the @, $ and # signs are considered alphabetic characters.) The remaining characters in the name can be either alphabetic or numeric.
- Blanks must not appear between characters in the file name.
- No two files used in the same program can have the same name. (Because some RPG II Compilers use only the first seven letters of an 8-letter file name, be certain, when using these systems, to make the first seven letters unique; for example, TRANSACT and TRANFILE, not TRANFILA and TRANFILB.)
- The file name must begin in column 7 on the specification sheet.

Which Names Are Valid?



Line 3 4 5	Form Type 6	Filename 7 8 9 10 11 12 13 14	Sequence 15 16	
0 1	I	TIMECRD		← Valid filename.
0 2	I	LIST (begins in column 8)		← Invalid filename. Name must start in column 7.
0 3	I	IN FILE		← Invalid filename. A blank must not be used between letters.
0 4	I	PAYMASTR		← Valid filename.
0 5	I	A		← Valid filename.
0 6	I	1OUT		← Invalid filename. The first character of the name must be an alphabetic character.

It is a good practice to assign meaningful file names. Meaningful names indicate something about the file, such as the type of records in the file or the use of the file. Because file names can be no longer than eight characters, abbreviations may be necessary. But these too can be meaningful. For example, the abbreviation CUSTCHG might be assigned to an input file consisting of records for all customers having charge accounts.

Device Designation

You must also specify which devices your job will use (card reader/punch; disk; keyboard; printer; data recorder). The ones you use, of course, depend upon the system you have, the devices you have, and your job.

To indicate the device used for the file you named, enter the RPG II code name for that device in columns 40-46. The name must begin in column 40.

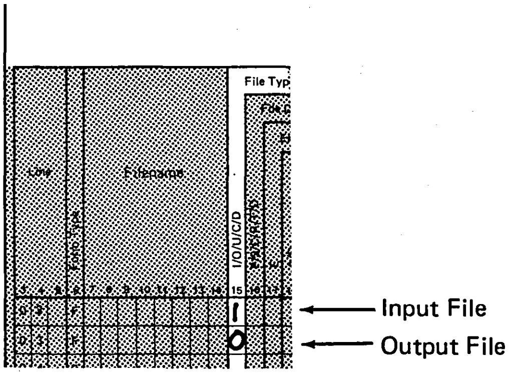
Note: In the examples in this manual, you will see device names or abbreviations of device names rather than actual RPG II code names for devices. Code names differ from system to system and from device to device.

During compilation, the compiler associates the file name with the device name. When you use the same file name in the rest of your program, the computer will know which device to use.

File Use

You must also describe how each file and its associated device is used in a program. Files can be used as either input or output. If records are read from a file, the file is an input file. If a new file is created during the job, the new file is an output file. Printed reports are the most common type of output files, but card and disk files can also be output files.

You specify file use by placing either an I (input) or O (output) in column 15:

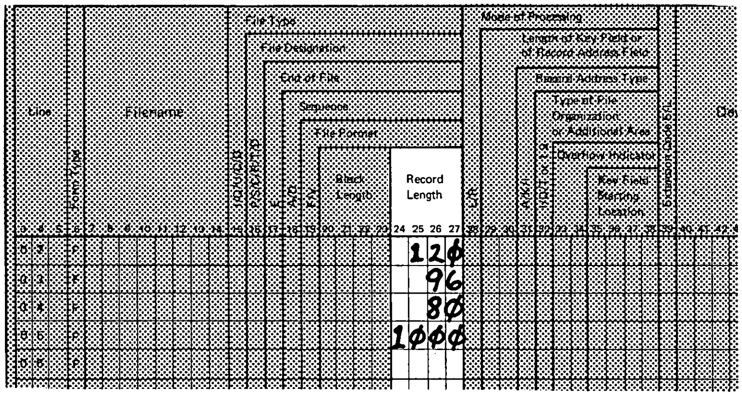


File Designation

Column 16 is used to explain more about the use of input files. An input file can be either primary or secondary. In this book, we are discussing the use of only one input file. When only one input file is used, it is always a *primary* file and you place a P in column 16.

Record Size

When describing files, you must specify the length (in characters) of records in the file. Record length is entered in columns 24-27. Enter the length so the last digit is in column 27:



Line	Form Type	Record Length
02	F	12Ø
03	F	96
04	F	8Ø
05	F	1ØØØ
06	F	

The record length specification does two things:

1. It tells the compiler how much storage space to set aside for a record (input or output).

2. It specifies how many characters must be read to get a complete input record.

Record size for card files is easy to determine. It is either 80 or 96 depending upon the size of cards you have. Maybe not all your cards have information in all columns, but all columns must be read to get an entire record. Blanks are placed in storage positions corresponding to unpunched card columns.

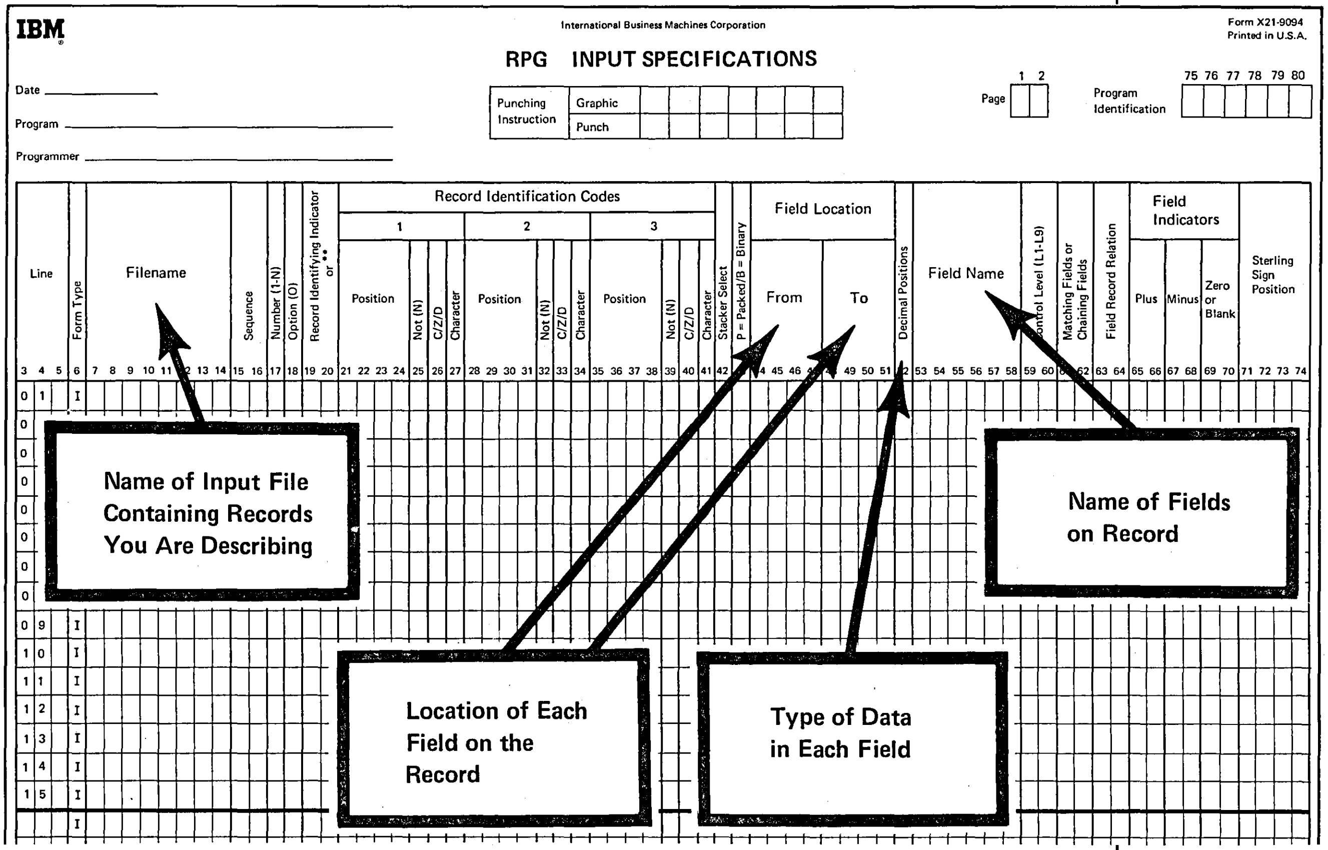
The size of records (lines) on the printer is also easy to determine. Printed records are limited by the size of your printer (the number of print positions in a line).

You may, if you wish, specify a record size smaller than actual printer size. If you do this, make certain that none of the lines to be printed are longer than the length you specify; otherwise, some information will be lost because the storage area reserved for the output record is only as large as the specified record size.

Records on disk files can be any size. The maximum size is limited only by the capability of the device. When you use one of these files for a job, make sure you enter the correct record size; that is, the one established at the time the file was created.

DESCRIBING INPUT RECORDS

Besides describing files, you must describe the records in each file. The compiler needs this information to create an object program that will read records properly. Input records are described on the Input sheet. Information needed to describe the record in a file includes the name of the file containing the record, the name of each field in the record, and where each field is located on the record. You fill out the indicated columns to describe fields and data in the record:



Name of Input File Containing Records You Are Describing

Name of Fields on Record

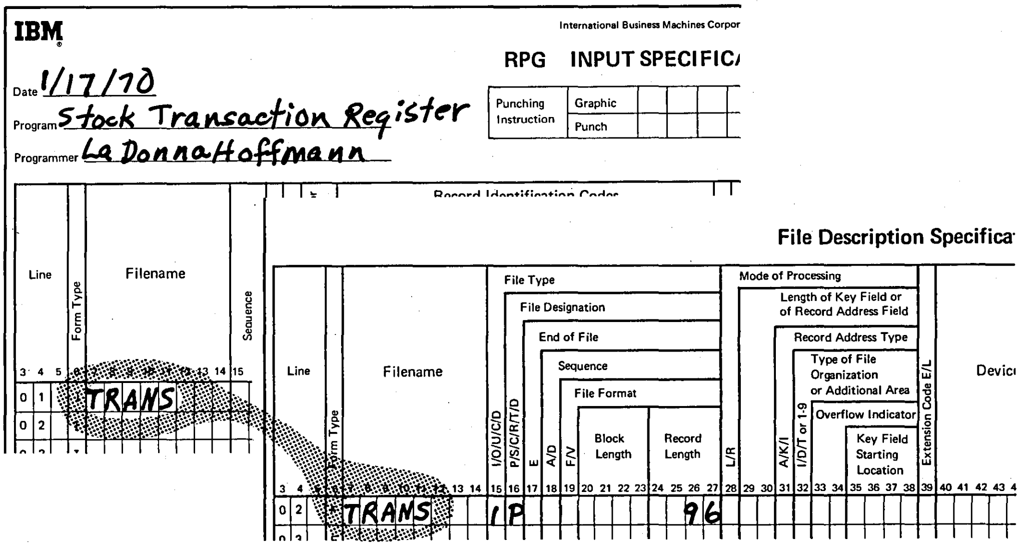
Location of Each Field on the Record

Type of Data in Each Field

To help in describing input records, you can use the Record Layout Form described in the beginning of this chapter. This form shows the location and length of all fields in the record.

File Names

To tell the compiler which records you are describing, enter the name of the file containing them in columns 7-14. The name must be the same (and spelled exactly the same) as the one you assigned to the input file on the File Description sheet:


IBM
International Business Machines Corpor
RPG INPUT SPECIFIC
Punching Instruction
Graphic
Punch
Line
Form Type
Filename
Sequence
File Description Specifica
File Type
File Designation
End of File
Sequence
File Format
Block Length
Record Length
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
Extension Code E/L
Devic

Date 1/17/70

Program Stock Transaction Register

Programmer La Donna Hoffmann

Line	Form Type	Filename
0 1	I	TRANS
0 2	I	

Line	Form Type	Filename	File Type	File Designation	Record Length
0 2	F	TRANS	I	P	96

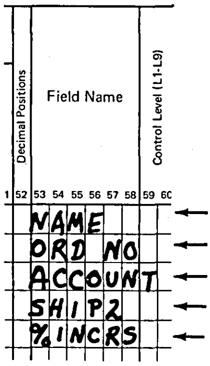
Field Names

To identify individual fields in the record, you must give each field a unique name. From information you placed on the File Description sheet, the compiler determines the size of the storage area for each input record. The field names you supply on the Input sheet tell the compiler to divide this storage area into smaller sections so each can be addressed separately.

The rules for forming field names are as follows:

- The field name must be from 1-6 characters long.
- The first character must be alphabetic. Remaining characters can be either alphabetic or numeric.
- Blanks must not be placed between characters in a field name.
- The field name must begin in column 53 on the Input sheet.

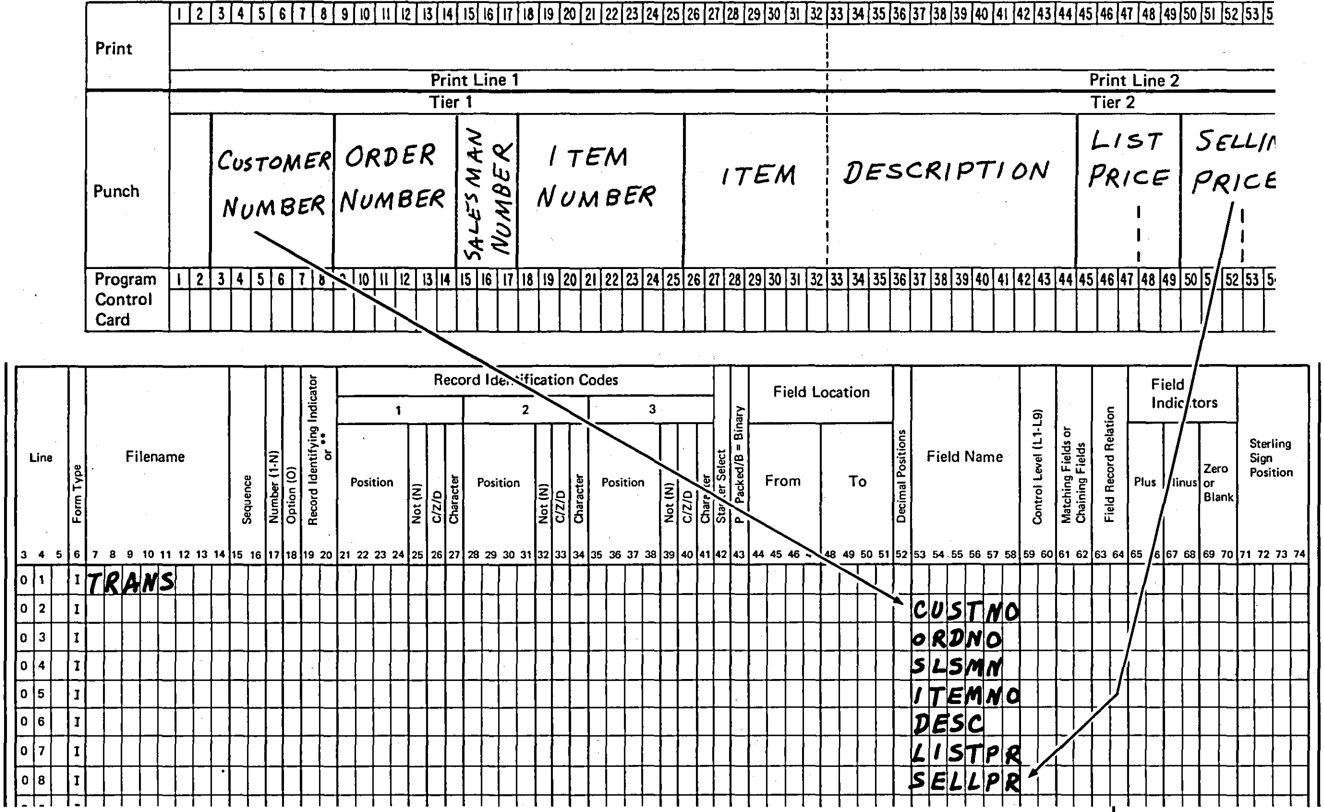
Which Names Are Valid?



Decimal Positions	Field Name	Control Level (L1-L9)	
1 52	53 54 55 56 57 58	59 60	
	NAME		←Valid name.
	ORD NO		←Invalid name. A blank cannot be used between letters in the name.
	ACCOUN	T	←Invalid name. A field name can be no longer than six characters.
	SHIP2		←Valid name.
	%INCRS		←Invalid name. The name must start with an alphabetic character (A-Z, #, $, @).

It is a good practice to assign meaningful field names. For example, a field containing customer numbers would be more meaningful if it were called CUSTNO rather than FIELDA. CUSTNO indicates something about the data in the field.

Enter field names one line below the file name, using a separate line for each field:

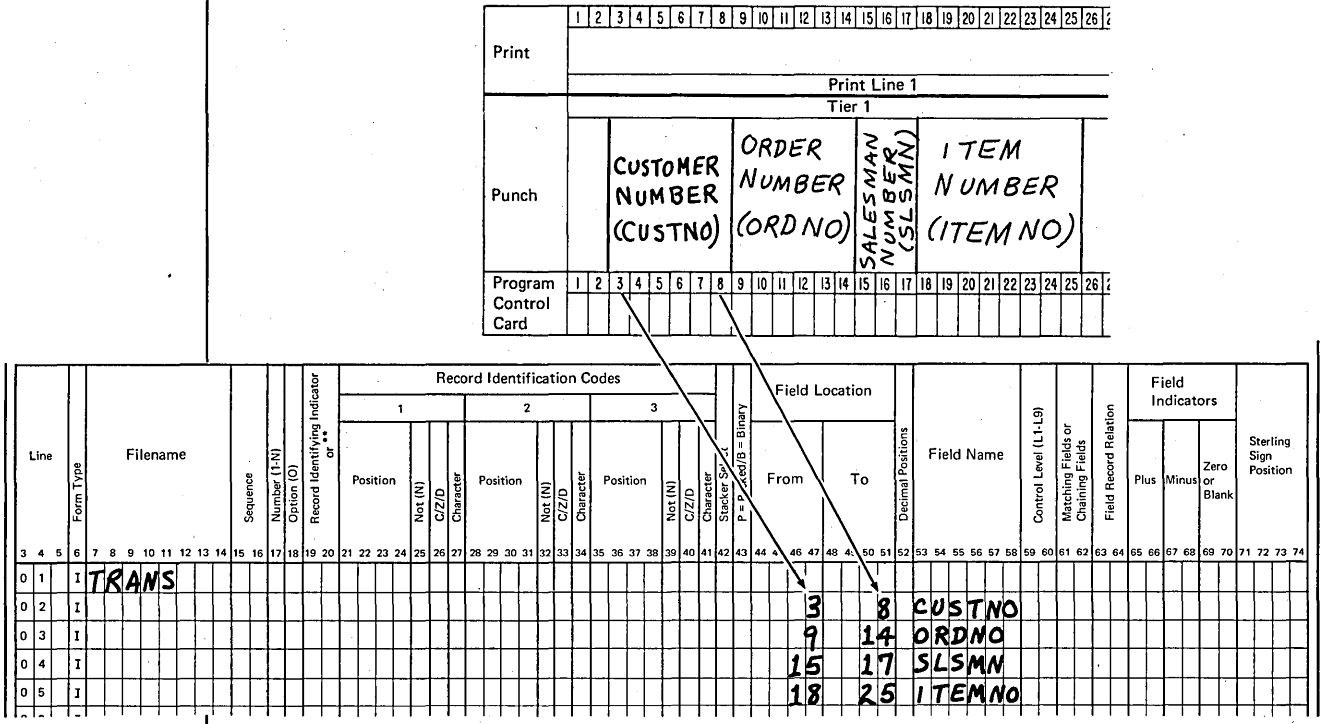


Line	Form Type	Filename	Field Name
01	I	TRANS	
02	I		CUSTNO
03	I		ORDNO
04	I		SLSMN
05	I		ITEMNO
06	I		DESC
07	I		LISTPR
08	I		SELLPR

Be sure to name every field that contains information necessary for your job. If you need all fields on the record, name them all; if you need only a few, name only those you will need. The entire record is read, of course, regardless of how many fields you are using from that record. However, only information in the fields you name can be used by the program for output.

Field Location

After you assign a field name, you must tell where the field is located in the record. This enables the compiler to associate the field name with the right information. To describe the field location, you specify the position in the record at which the field begins and ends. Starting position is specified in columns 44-47 (From); ending position is specified in columns 48-52 (To). When a field is only one character long, starting and ending position entries are the same. Field location entries can be easily determined from the Record Layout Form:



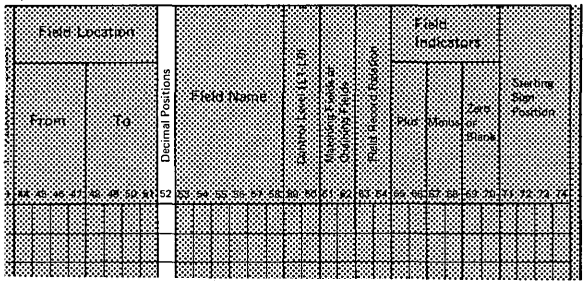
Line	Form Type	Filename	Field Location From	Field Location To	Field Name
01	I	TRANS			
02	I		3	8	CUSTNO
03	I		9	14	ORDNO
04	I		15	17	SLSMN
05	I		18	25	ITEMNO

The compiler also determines field length from the To and From entries. The compiler needs field length to determine how many storage positions to allow for each field. If you specify a field length of 6, the compiler allows six positions in storage for the field.

Type of Data

To complete your description of the input fields, the compiler checks column 52. This column indicates whether data in each field is alphameric or numeric. A numeric field can contain only numbers; an alphameric field can contain numbers, letters, and special characters.

If column 52 is blank, the compiler assumes the field is alphameric. For numeric fields, column 52 must contain an entry. This entry indicates the number of decimal positions (digits to the right of the decimal point) in the field:



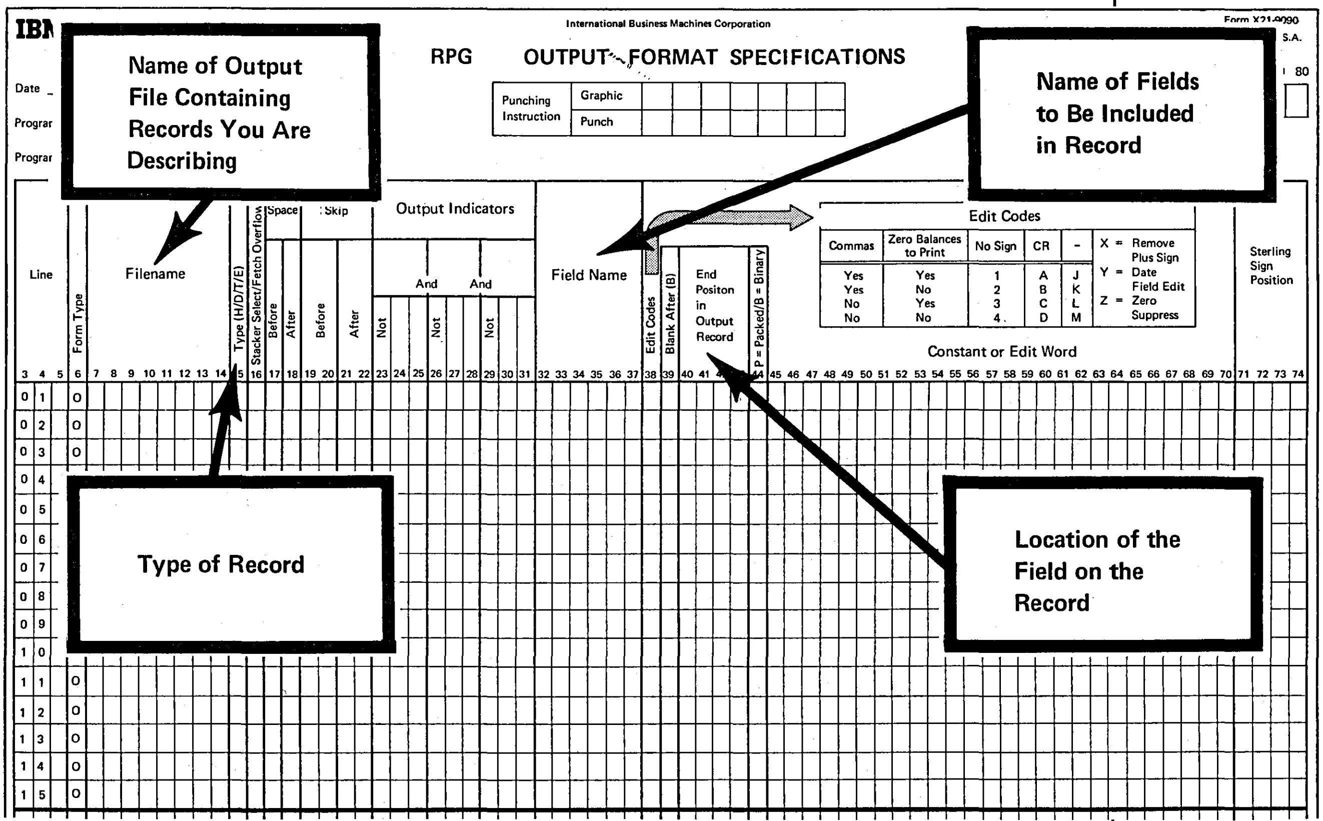
Leave column 52 blank for alphameric fields. Enter 0-9 to indicate number of decimal positions in numeric field.

Although you do not include decimal points in fields on input records, you must consider them if you want correct output data. By specifying to the compiler the number of decimal positions you know to be in a numeric field, you provide the information necessary to produce an object program that will handle numeric data with decimals.

Remember, any field used in an arithmetic operation (add, subtract, multiply, or divide) must be specified as numeric.

DESCRIBING OUTPUT RECORDS

Output records are described on the Output-Format sheet. Information needed includes the name of the file containing the output record, the name of each field in the record, and where each field is to be placed in the record. You fill in the indicated columns to describe how the output records should look:



Name of Output File Containing Records You Are Describing

Name of Fields to Be Included in Record

Type of Record

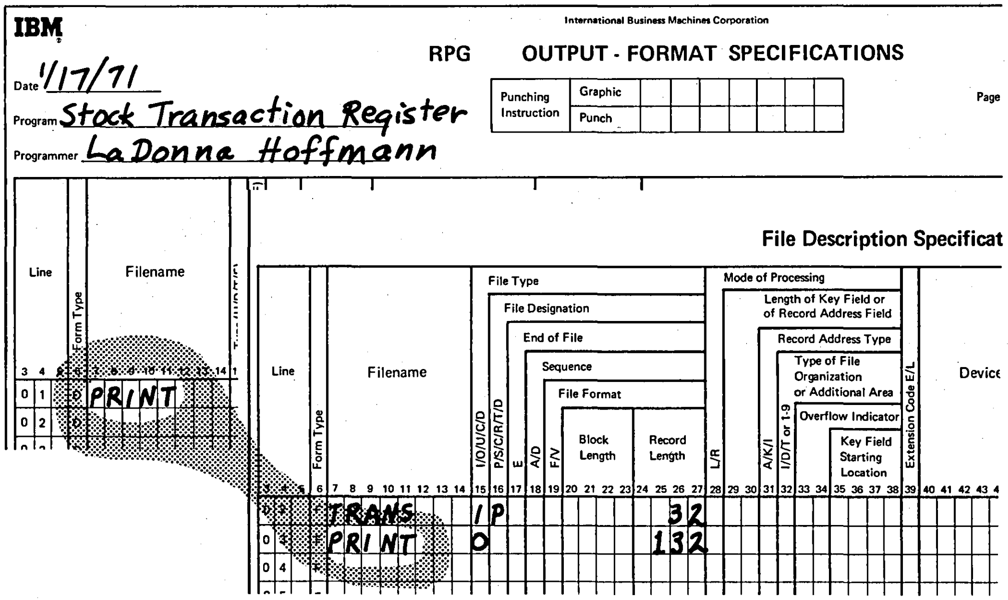
Location of the Field on the Record

If the output is a printed report, you make additional entries describing the format of the report; that is, entries indicating the spacing and punctuation you want. These entries are discussed later under *Printed Reports.*

The Printer Spacing Chart and Record Layout Forms are useful when you are writing Output-Format specifications. The Record Layout Form shows the organization of fields on a card or disk record; the Printer Spacing Chart shows the format of printed records.

File Names

You indicate the output record you want created and the device you want to create the record by entering an output file name in columns 7-14. Make sure the name you enter is the same (and spelled exactly the same) as the name you entered on the File Description sheet for the output file:



IBM

International Business Machines Corporation

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/17/71

Program Stock Transaction Register

Programmer LaDonna Hoffmann

Punching Instruction: Graphic / Punch

Page

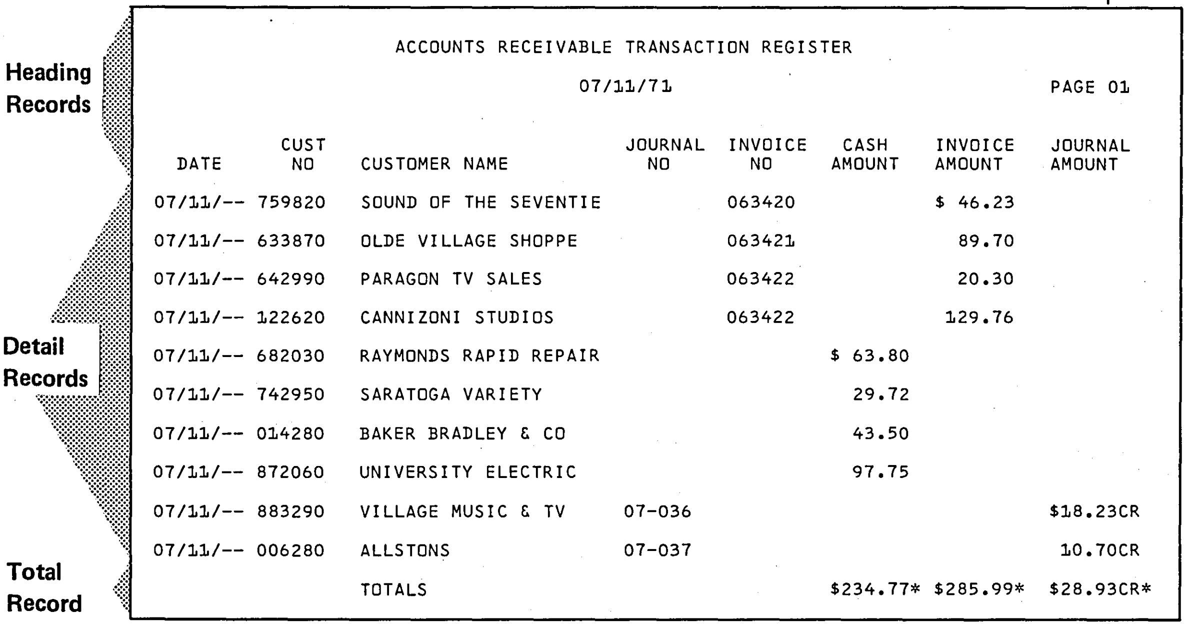
Line	Form Type	Filename
3 4 5	6	7 8 9 10 11 12 13 14
0 1	O	PRINT
0 2	O	

File Description Specificat

Line	Form Type	Filename	File Type (I/O/U/C/D)	File Designation (P/S/C/R/T/D)	Record Length
3 4 5	6	7 8 9 10 11 12 13 14	15	16	24 25 26 27
0 2	F	TRANS	I	P	32
0 3	F	PRINT	O		132
0 4	F				

Record Type

Three different types of records can be specified on the Output-Format sheet: heading, detail, and total. You usually find all three types in a printed report. When output consists of card or disk files, however, you normally have only detail and total records:


Heading Records
Detail Records
Total Record

```
                    ACCOUNTS RECEIVABLE TRANSACTION REGISTER

                                   07/11/71                                PAGE 01

                CUST                                 JOURNAL  INVOICE   CASH     INVOICE    JOURNAL
    DATE         NO     CUSTOMER NAME                  NO       NO      AMOUNT   AMOUNT     AMOUNT

07/11/--  759820   SOUND OF THE SEVENTIE                 063420             $ 46.23

07/11/--  633870   OLDE VILLAGE SHOPPE                   063421               89.70

07/11/--  642990   PARAGON TV SALES                      063422               20.30

07/11/--  122620   CANNIZONI STUDIOS                     063422              129.76

07/11/--  682030   RAYMONDS RAPID REPAIR                          $ 63.80

07/11/--  742950   SARATOGA VARIETY                                 29.72

07/11/--  014280   BAKER BRADLEY & CO                               43.50

07/11/--  872060   UNIVERSITY ELECTRIC                              97.75

07/11/--  883290   VILLAGE MUSIC & TV     07-036                                       $18.23CR

07/11/--  006280   ALLSTONS               07-037                                        10.70CR

                   TOTALS                                         $234.77* $285.99*  $28.93CR*
```

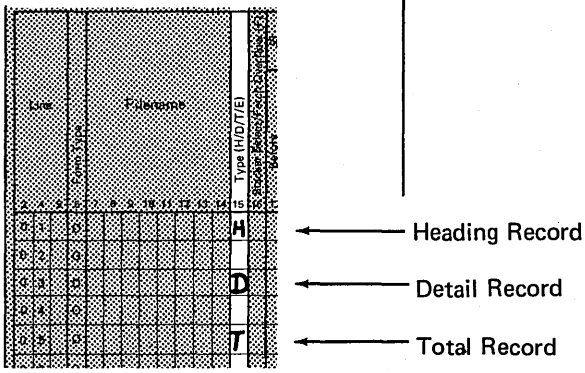
Heading records are printed at the top of a page. They include report titles, column headings, or any other information needed to identify the kinds of information found in the report.

Detail records contain information about an individual item. Information in a detail record is often taken directly from an input record.

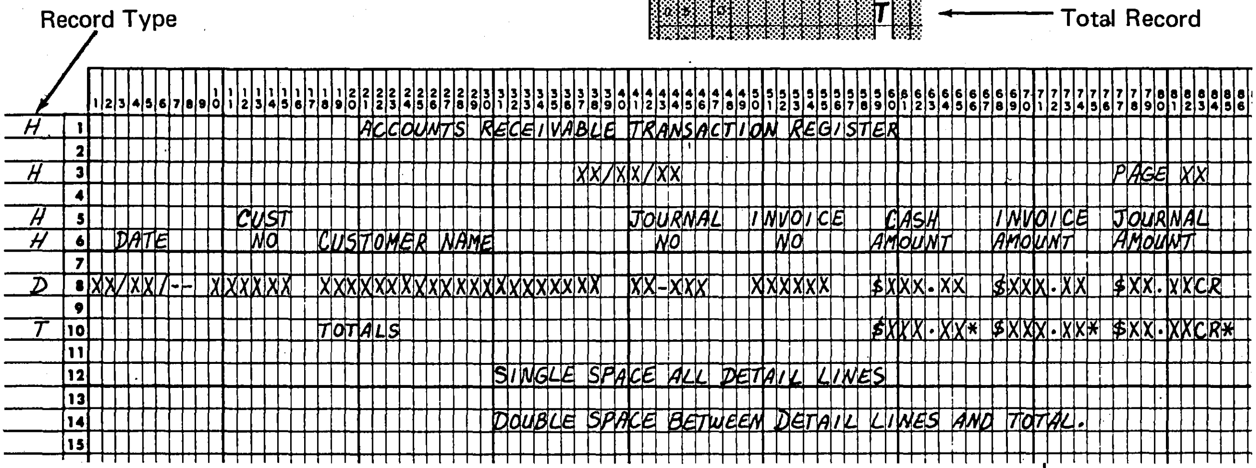
Total records are written after a group of detail records. They usually contain data that is the result of calculations on information in a group of detail records.

To specify record type, place an appropriate entry in column 15. If the Printer Spacing Chart was properly filled out, you can refer to it to determine record type:

These entries specify the record type:



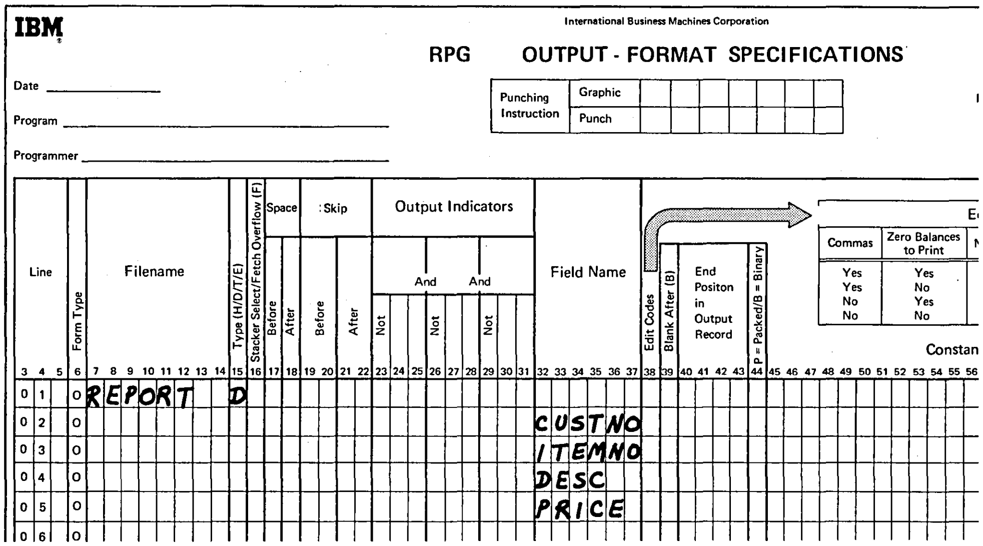
Refer to the Printer Spacing Chart to determine the record type:


Record Type

```
H   1                      ACCOUNTS RECEIVABLE TRANSACTION REGISTER
    2
H   3                                   XX/XX/XX                                PAGE XX
    4
H   5              CUST                          JOURNAL   INVOICE    CASH      INVOICE   JOURNAL
H   6    DATE       NO    CUSTOMER NAME            NO        NO      AMOUNT     AMOUNT    AMOUNT
    7
D   8 XX/XX/--  XXXXXX    XXXXXXXXXXXXXXXXXXXXX  XX-XXX    XXXXXX   $XXX.XX    $XXX.XX   $XX.XXCR
    9
T  10                     TOTALS                                  $XXX.XX*   $XXX.XX*  $XX.XXCR*
   11
   12                                    SINGLE SPACE ALL DETAIL LINES
   13
   14                                    DOUBLE SPACE BETWEEN DETAIL LINES AND TOTAL.
   15
```

Field Names

To specify the information to be placed in each output record, you must name each field to be included. You specify these fields on separate lines of the Output-Format sheet in columns 32-37. Begin the list of fields one line below the file name:



IBM

International Business Machines Corporation

RPG OUTPUT - FORMAT SPECIFICATIONS

Date

Program

Programmer

Punching Instruction	Graphic	
	Punch	

Commas	Zero Balances to Print
Yes	Yes
Yes	No
No	Yes
No	No

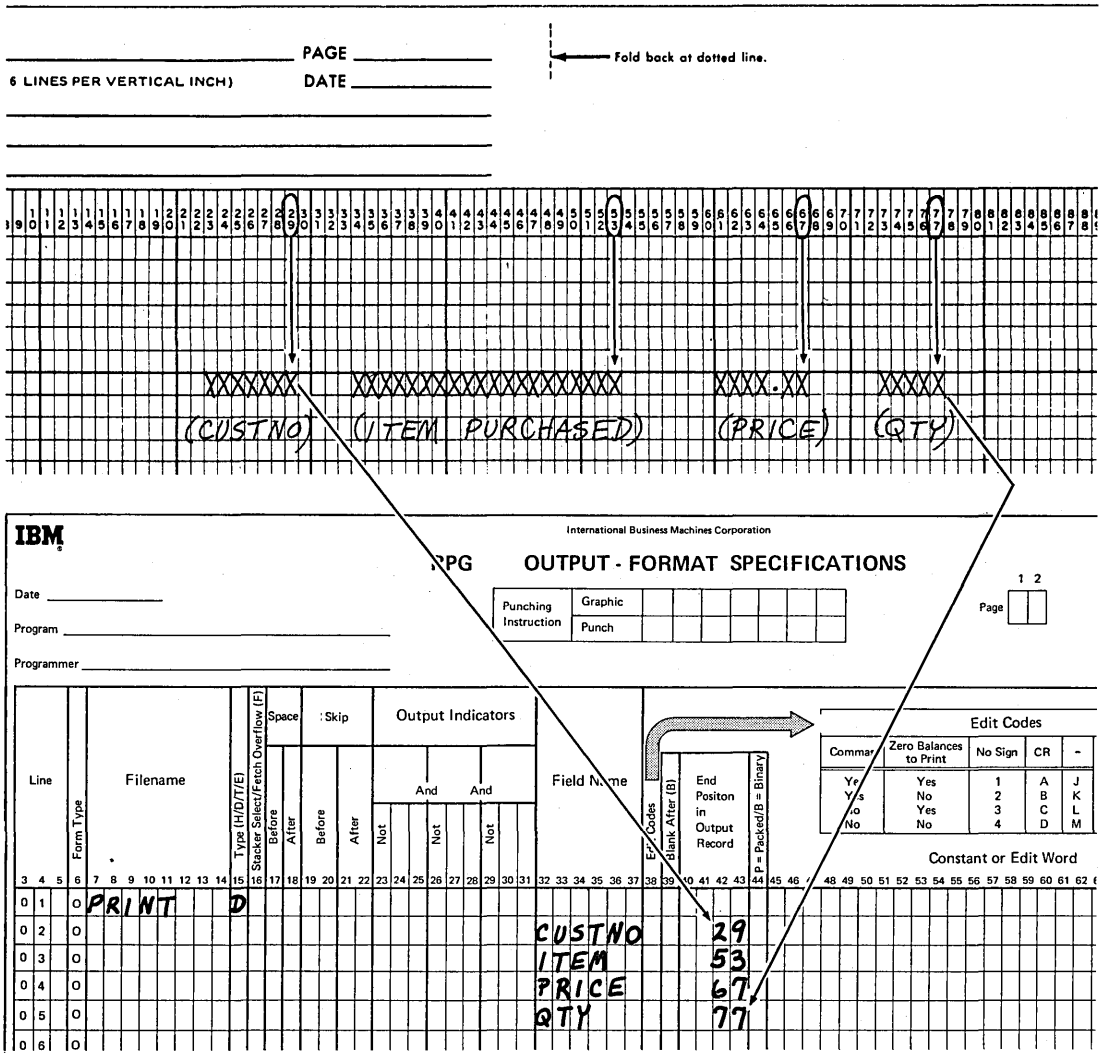
Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators Not		Output Indicators And Not		Output Indicators And Not		Field Name	Edit Codes	Blank After (B)	End Positon in Output Record	P = Packed/B = Binary	Constan
3 4 5	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
0 1	O	REPORT	D																	
0 2	O														CUSTNO					
0 3	O														ITEMNO					
0 4	O														DESC					
0 5	O														PRICE					
0 6	O																			

When listing the fields, make sure you enter a name that was previously given to a field (for example, a field named on the Input sheet). If the name you enter on the Output-Format sheet is not one previously used, the compiler won't know what information you're referring to.

Field names are used to create the output record in the output storage area. Information is placed in the storage area one field at a time in the order you list them on the Output-Format sheet.

Field Location

To specify where you want fields placed in the output records, you make an entry in columns 40-43 (End Position in Output Record). This entry must be the exact position where the last character in a field should be placed in the output storage area at the time the record is being created. The entry is easy to determine if you use a Printer Spacing Chart or Record Layout Form:



Line	Form Type	Filename	Type (H/D/T/E)	Field Name	End Position in Output Record
01	O	PRINT	D		
02	O			CUSTNO	29
03	O			ITEM	53
04	O			PRICE	67
05	O			QTY	77
06	O				

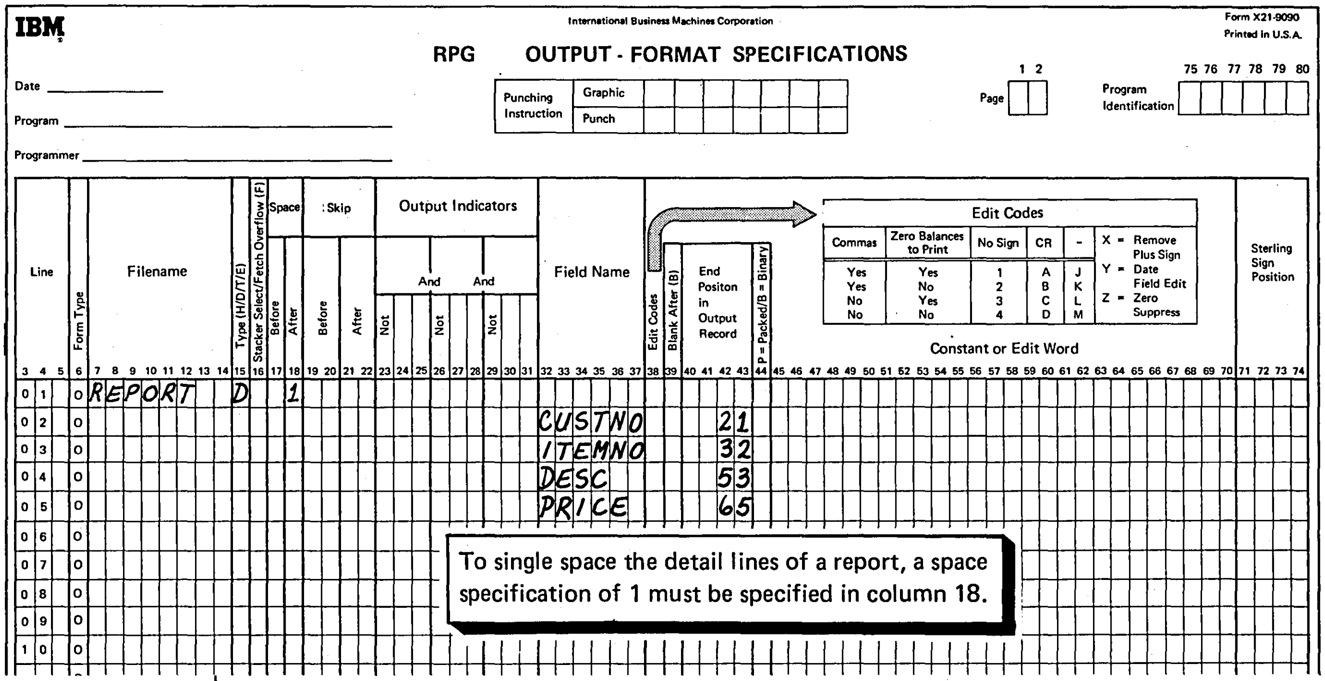
PRINTED REPORTS

When your output is a printed report, additional entries are needed on the Output-Format sheet to make the report easy to read. Information must be neatly arranged in rows and columns with adequate space between items in a line and between lines.

Spacing

Your field location entries (columns 40-43) control space between fields, but to control spaces between lines you code columns 17-18 (Space).

You can have the printer single, double, or triple space between lines by entering the number 1, 2, or 3 in the appropriate columns. If you enter the number in Space Before (column 17), the printer spaces before printing the line; if you enter the number in Space After (column 18), the printer spaces after printing the line. You can enter numbers in both columns 17-18 if you wish. As many as six spaces (three before printing and three after) can be made between lines.



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date

Program

Programmer

Punching Instruction	Graphic	Punch

Page 1 2

Program Identification 75 76 77 78 79 80

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Space After (18)	Field Name (32-37)	End Position in Output Record (40-43)
0 1	O	REPORT	D	1		
0 2	O				CUSTNO	21
0 3	O				ITEMNO	32
0 4	O				DESC	53
0 5	O				PRICE	65
0 6	O					
0 7	O					
0 8	O					
0 9	O					
1 0	O					

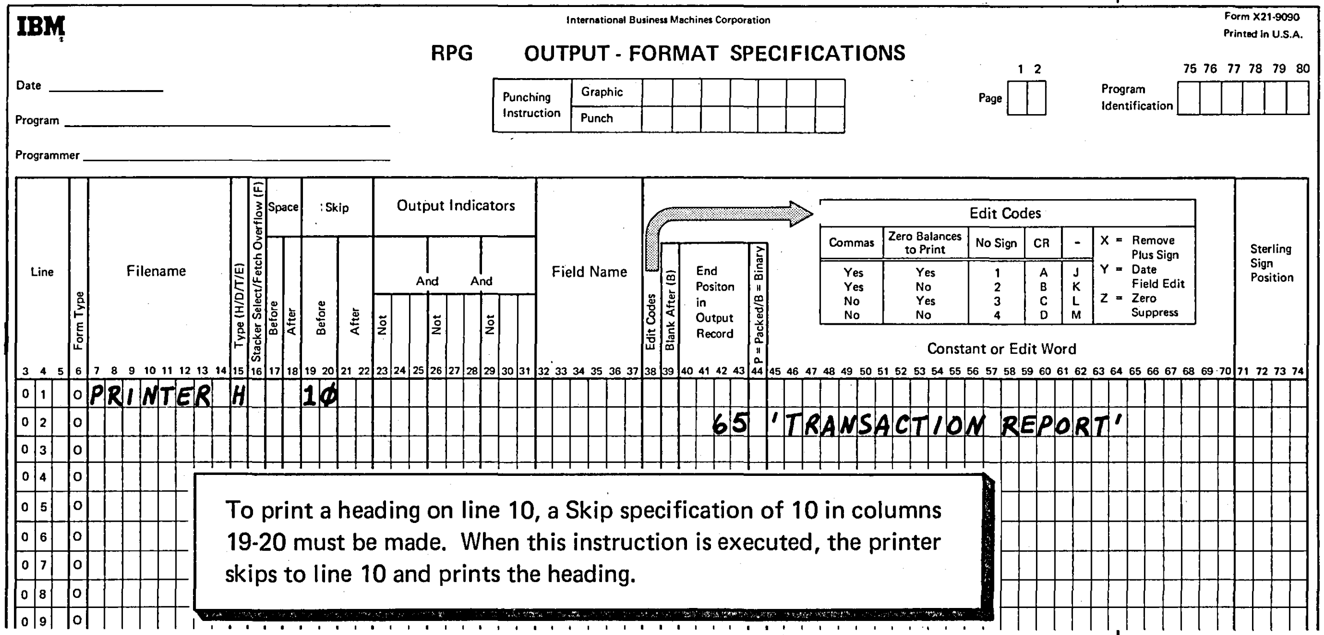
To single space the detail lines of a report, a space specification of 1 must be specified in column 18.

Skipping

You can use Skip entries in columns 19-22 to control spacing between lines on a page and to control printing the first and last lines on a page. A skip can be made before or after a line is printed. You indicate this by coding the skip in either columns 19-20 (Before) or 21-22 (After). The entry you place in these columns depends on the type of printer you have.

Printer with Tapeless Carriage Control

The Skip entry for this type of printer can be any line number (1-112) on the computer paper. (The standard 11-inch computer paper has 66 lines per page when 6 lines are printed per inch.) A Skip entry in columns 19-20 indicates the line to which the printer skips before it prints the next line:



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date ____________

Program ____________

Programmer ____________

Punching Instruction: Graphic / Punch

Page (1 2)

Program Identification (75 76 77 78 79 80)

Edit Codes					
Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
0 1	O	PRINTER	H				1Ø									
0 2	O												65		'TRANSACTION REPORT'	
0 3	O															
0 4	O															
0 5	O															
0 6	O															
0 7	O															
0 8	O															
0 9	O															

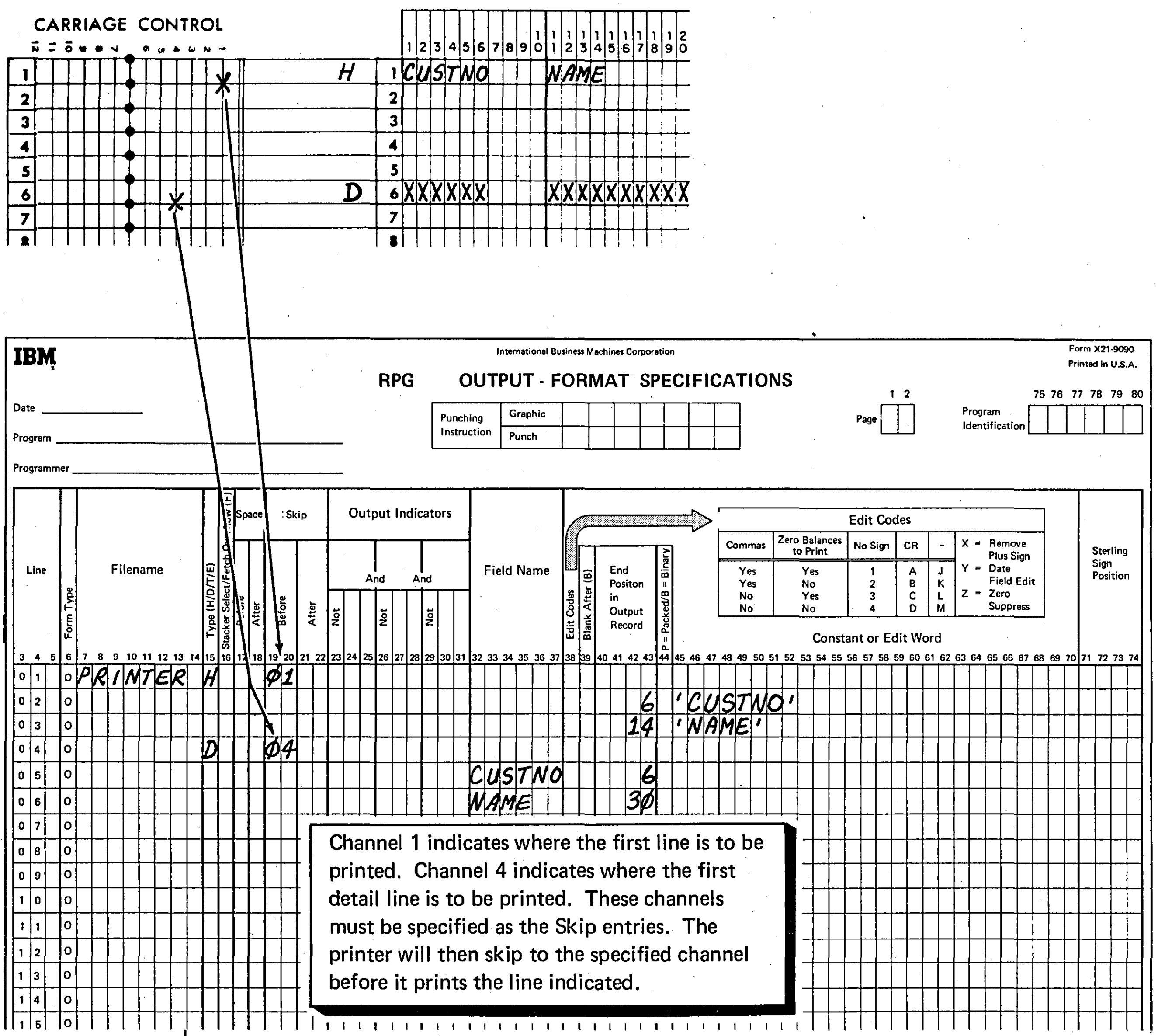
To print a heading on line 10, a Skip specification of 10 in columns 19-20 must be made. When this instruction is executed, the printer skips to line 10 and prints the heading.

Printer paper is not rolled backward. If the printer is on line 50 when the program issues a skip instruction to line 10, the printer skips to line 10 on the next page.

Printer with Carriage Control Tape

The Skip entry for this type of printer is the channel number punched in the carriage control tape. A channel-1 punch should indicate the first line to be printed on a page; a channel-12 punch should indicate the last line to be printed. Any of the other channels can be used for skipping by placing a punch in the appropriate line of the tape. For example, if you have a 2-punch on line 20 of the tape and specify 02 as the Skip entry, the printer will skip to line 20 when the program issues the skip instruction.

To indicate the line where you want to skip, punch one of the channels 1-12 on the carriage tape. Then specify the channel in columns 19-20 if you want the printer to skip before printing the line. Specify the channel in columns 21-22 if you want the printer to skip after printing the line:



RPG OUTPUT - FORMAT SPECIFICATIONS

Line	Form Type	Filename	Type (H/D/T/E)	Skip Before	Field Name	End Position in Output Record	Constant or Edit Word
01	O	PRINTER	H	Ø1			
02	O					6	'CUSTNO'
03	O					14	'NAME'
04	O		D	Ø4			
05	O				CUSTNO	6	
06	O				NAME	3Ø	

Channel 1 indicates where the first line is to be printed. Channel 4 indicates where the first detail line is to be printed. These channels must be specified as the Skip entries. The printer will then skip to the specified channel before it prints the line indicated.

Editing

Editing is a means of punctuating numeric fields by adding decimal points, commas, and negative signs. It can also consist of suppressing leading zeros (in the number 00149, 00 are called leading zeros).

When a numeric field is read into storage, it contains no decimal point or commas; when an unedited numeric field is printed, it appears exactly as it is in storage. A large number printed without commas or decimals is hard to read. Furthermore, an unedited field may not be meaningful when printed out because of the way the computer keeps track of negative numbers.

The computer uses the last digit in a numeric field to indicate sign (plus or minus). If the field is minus, the computer combines a minus sign with the last digit. When a negative number is printed out unedited, the combination of digit and sign appears as a letter. For example, minus 6439 prints as 643R. On the other hand, a positive field has no sign (a numeric field that does not have a negative sign is assumed to be positive). A positive field, therefore, prints normally. Positive 6439 prints as 6439.

The compiler can provide instructions to edit in a number of ways. All you have to do is enter an edit code in column 38 of the Output-Format sheet. Many codes are available, each indicating a different type of editing. Figure 4 shows the codes and the editing done for each. Figure 5 shows some examples of editing.

Note: When you edit a field, you often add characters to it. When printed, the edited fields require more space than they did on input records or in storage. When specifying end position for an edited field, always take into account the spaces needed for the punctuation that will be added. The Printer Spacing Chart shows the amount of space needed for the edited field.

Edit Code	Commas	Decimal Point	Sign For Negative Balance			Zero Suppress	Print Out 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
X*							
Y**						Yes	
Z***						Yes	

* The X code removes the plus sign of the field.

** The Y code is used for date fields. It suppresses only the leftmost zero and puts slashes in a three to six digit field according to the following pattern:

nn/n
nn/nn
nn/nn/n
nn/nn/nn

*** The Z code removes signs and suppresses zeros.

Figure 4. The edit codes shown in the first column are used in column 38 of the Output-Format sheet to punctuate the field named on the same line. Only numeric fields can be edited. The decimal point is automatically inserted in the correct position.

Edit Codes

Field Length and Digits	1769532	02	00	000	041345
Field Characteristics	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	17695.32	.02	.00	0	41.345
4	17695.32	.02			41.345
A	17,695.32	.02CR	.00	0	41.345
B	17,695.32	.02CR			41.345
C	17695.32	.02CR	.00	0	41.345
D	17695.32	.02CR			41.345
J	17,695.32	.02–	.00	0	41.345
K	17,695.32	.02–			41.345
L	17695.32	.02–	.00	0	41.345
M	17695.32	.02–			41.345
X	1769532	0K	00	000	041345
Y	Must be used with a 3 to 6-digit field.			0/0	4/13/45
Z	1769532	2			41345

Figure 5. The table above shows the effect of editing on five different fields. It illustrates what will be printed out by using each edit code on the fields.

Job 1: Printing A Simple Report Using The Three Basic Cycle Operations

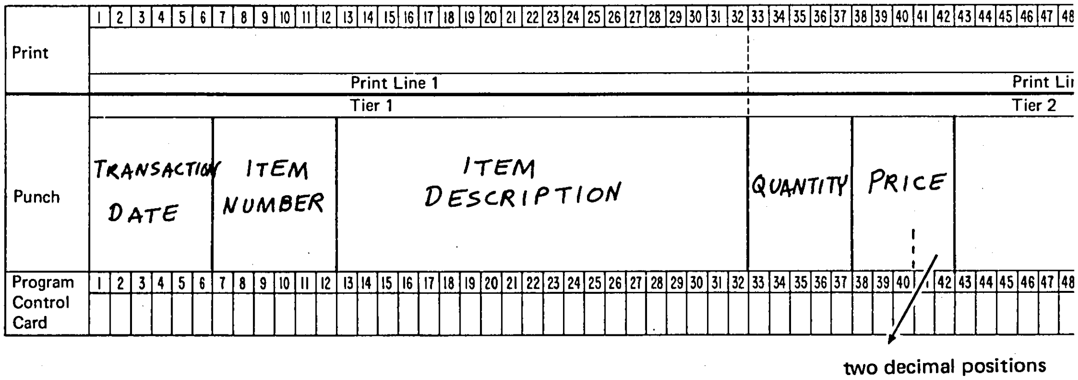
JOB DEFINITION

Print a report listing all items sold during a week. The selling of an item is known as a transaction, so the report is titled *Transaction Register.*

During the week, a transaction file is created. At the end of each day, transaction records are punched in cards from information obtained from order forms received during the day. To get the printed transaction report, you list the information from all input records on the printed report.

JOB REQUIREMENTS

Input: Sales transaction file consisting of 96-column cards. The format of the input records is shown on this Record Layout Form:

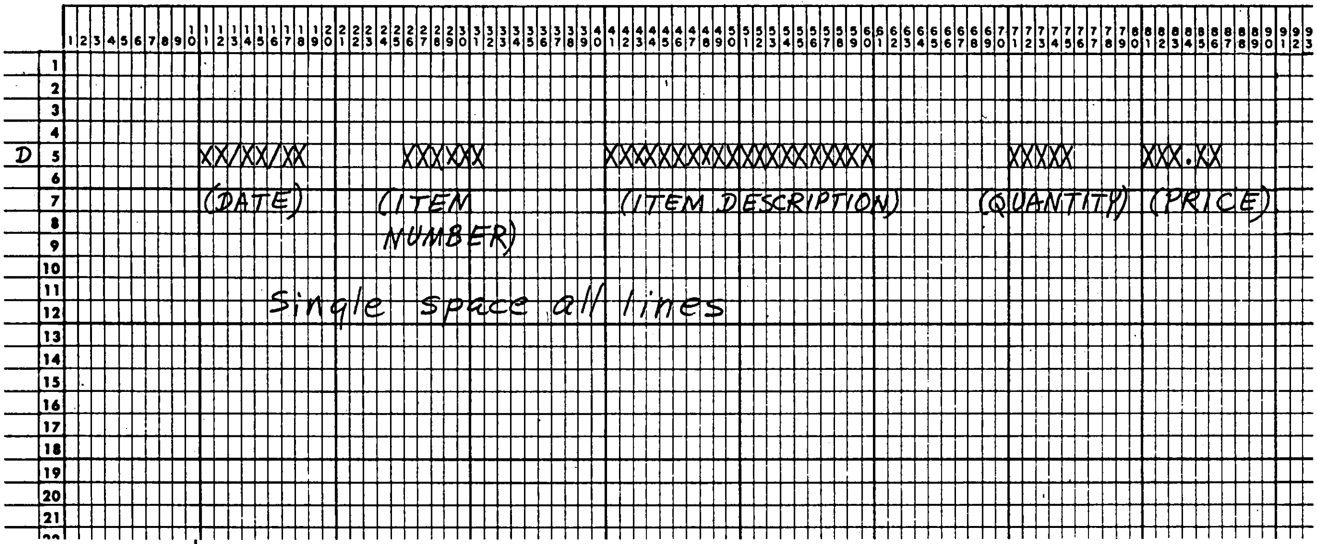


Print
Print Line 1
Print Li
Tier 1
Tier 2
Punch
TRANSACTION DATE
ITEM NUMBER
ITEM DESCRIPTION
QUANTITY
PRICE
Program Control Card
two decimal positions

Output: A Transaction register printed on a 96-position printer:

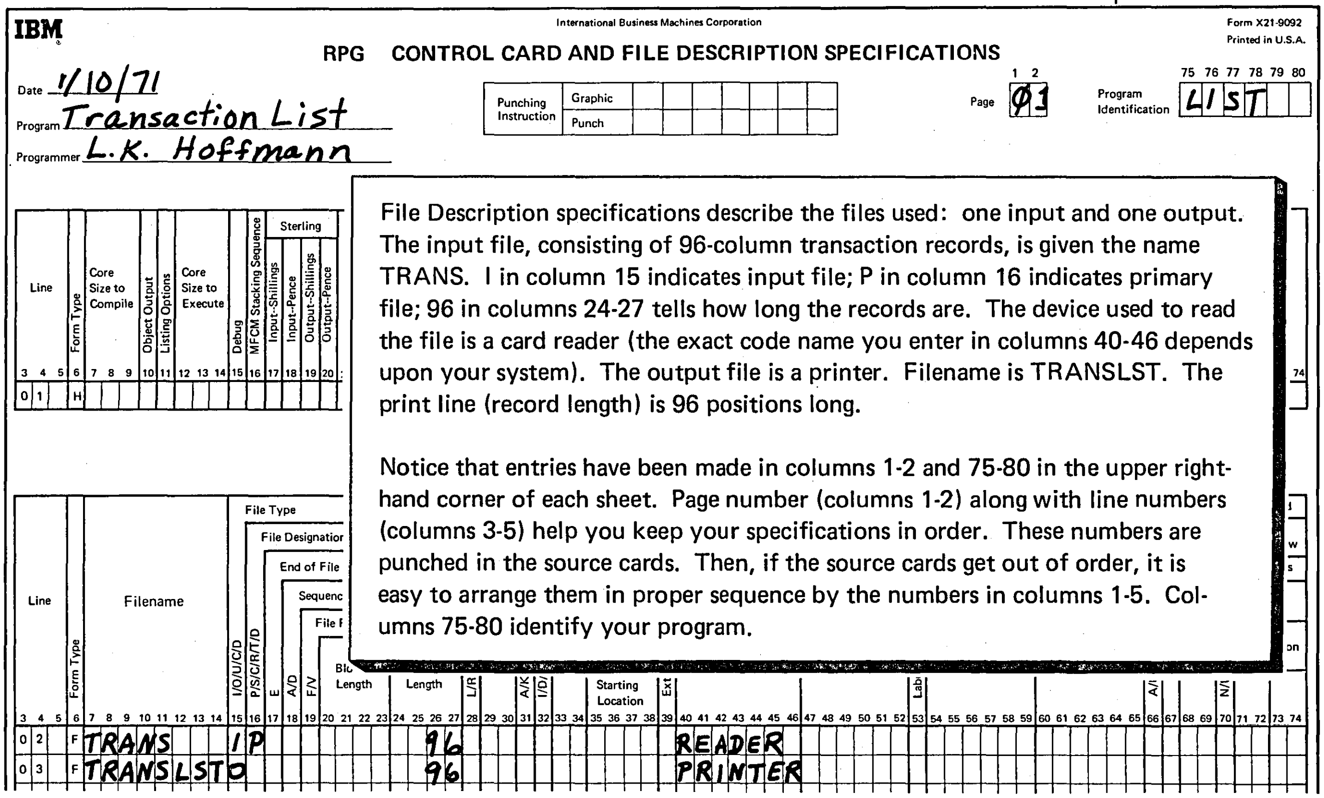
```
07/23/70      413010        CH001 BOX 100A FLUSH            10       4.90
07/23/70      412146        CH148 BREAKER 15A              100        .89
07/23/70      411116        1500 TWIN SOCKET B             500       1.12
07/24/70      503029        MOTOR 1/2 HP 60 CYC              2     146.78
07/24/70      317802        TERMINAL CLIP                  100       5.12
07/24/70      326917        TERMINAL BAR                   100       4.12
07/24/70      411121        1506 SOCKT ADAPT BRN           400        .19
07/24/70      412997        CH173 BREAKER 30A               60       1.15
07/24/70      413088        CH176 BREAKER 60A               40       1.15
07/24/70      411174        C151 SIL SWITCH BRN            200       1.16
07/24/70      413090        CH005 BR BOX 150A               10       4.98
07/24/70      718326        FC803 FUSE 15A                 200        .32
```

This Printer Spacing Chart shows how the report is formatted:



D
XX/XX/XX
XXXXXX
XXXXXXXXXXXXXXXXXXXX
XXXXX
XXX.XX
(DATE)
(ITEM NUMBER)
(ITEM DESCRIPTION)
(QUANTITY)
(PRICE)
Single space all lines

JOB SPECIFICATIONS



IBM

International Business Machines Corporation

Form X21-9092
Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71

Program Transaction List

Programmer L.K. Hoffmann

Punching Instruction: Graphic / Punch

Page (1 2): Ø1

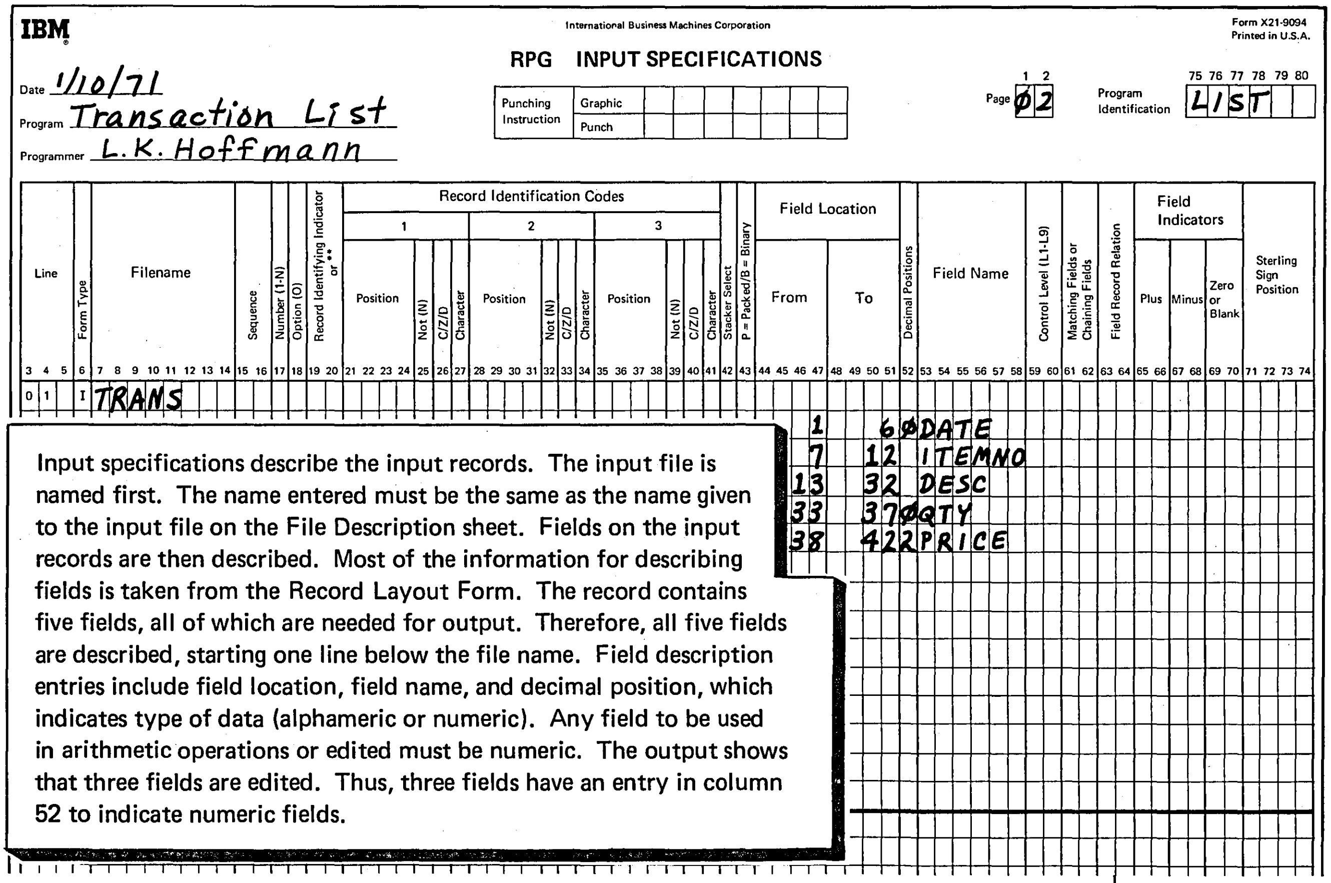
Program Identification (75 76 77 78 79 80): LIST

Line (3-5)	Form Type (6)
01	H

File Description specifications describe the files used: one input and one output. The input file, consisting of 96-column transaction records, is given the name TRANS. I in column 15 indicates input file; P in column 16 indicates primary file; 96 in columns 24-27 tells how long the records are. The device used to read the file is a card reader (the exact code name you enter in columns 40-46 depends upon your system). The output file is a printer. Filename is TRANSLST. The print line (record length) is 96 positions long.

Notice that entries have been made in columns 1-2 and 75-80 in the upper right-hand corner of each sheet. Page number (columns 1-2) along with line numbers (columns 3-5) help you keep your specifications in order. These numbers are punched in the source cards. Then, if the source cards get out of order, it is easy to arrange them in proper sequence by the numbers in columns 1-5. Columns 75-80 identify your program.

Line (3-5)	Form Type (6)	Filename (7-14)	I/O/U/C/D (15)	P/S/C/R/T/D (16)	Length (24-27)	(40-46)
02	F	TRANS	I	P	96	READER
03	F	TRANSLST	O		96	PRINTER



IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date 1/10/71

Program Transaction List

Programmer L.K. Hoffmann

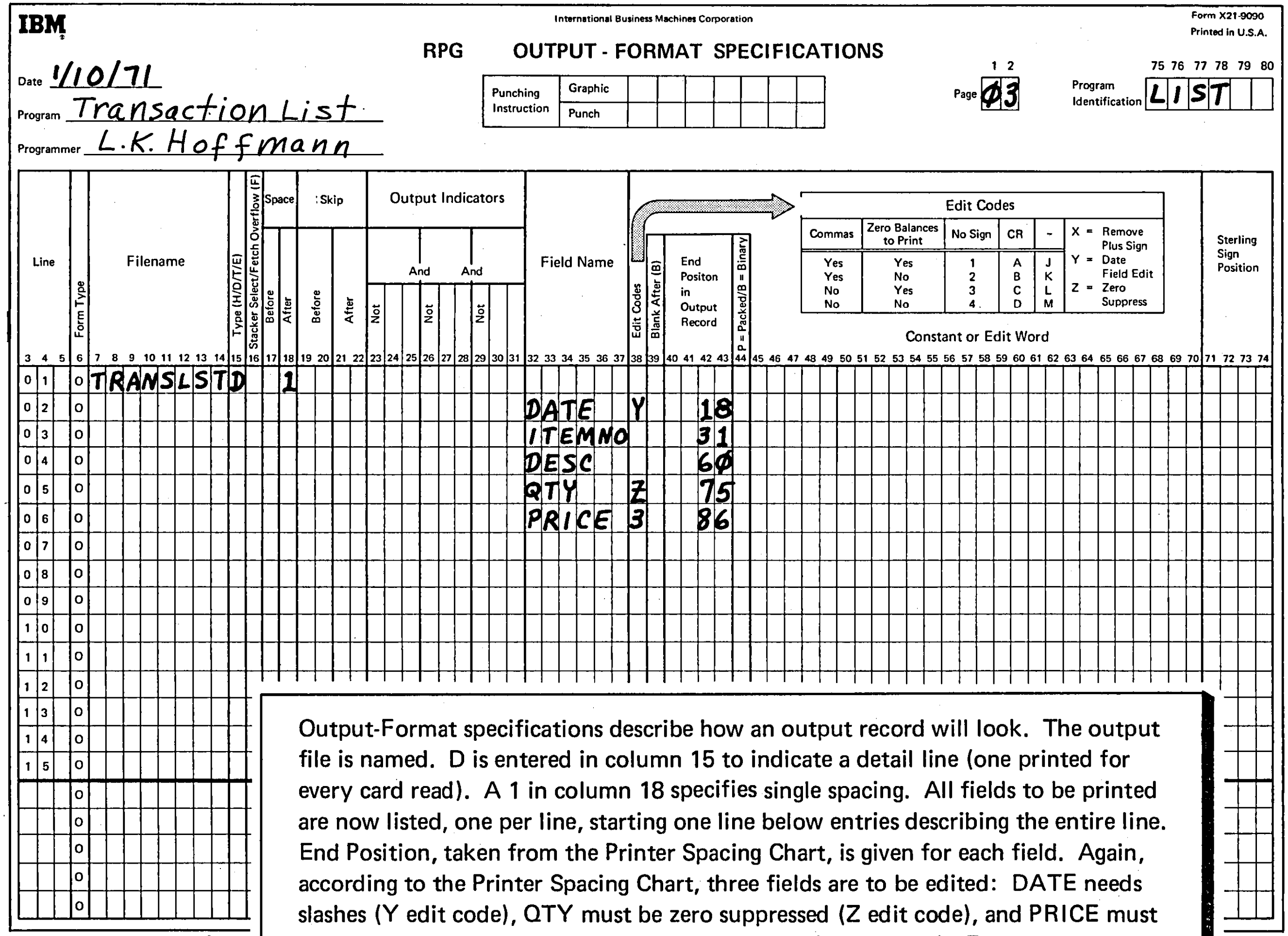
Punching Instruction: Graphic / Punch

Page (1 2): Ø2

Program Identification (75 76 77 78 79 80): LIST

Line (3-5)	Form Type (6)	Filename (7-14)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)
01	I	TRANS				
			1	6	Ø	DATE
			7	12		ITEMNO
			13	32		DESC
			33	37	Ø	QTY
			38	42	2	PRICE

Input specifications describe the input records. The input file is named first. The name entered must be the same as the name given to the input file on the File Description sheet. Fields on the input records are then described. Most of the information for describing fields is taken from the Record Layout Form. The record contains five fields, all of which are needed for output. Therefore, all five fields are described, starting one line below the file name. Field description entries include field location, field name, and decimal position, which indicates type of data (alphameric or numeric). Any field to be used in arithmetic operations or edited must be numeric. The output shows that three fields are edited. Thus, three fields have an entry in column 52 to indicate numeric fields.



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71
Program Transaction List
Programmer L.K. Hoffmann

Punching Instruction	Graphic	
	Punch	

Page Ø3 (1 2)

Program Identification LIST (75 76 77 78 79 80)

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Space After (18)	Field Name (32-37)	Edit Codes (38)	End Position in Output Record (40-43)
01	O	TRANSLST	D	1			
02	O				DATE	Y	18
03	O				ITEMNO		31
04	O				DESC		6Ø
05	O				QTY	Z	75
06	O				PRICE	3	86
07	O						
08	O						
09	O						
10	O						
11	O						
12	O						
13	O						
14	O						
15	O						
	O						
	O						
	O						
	O						
	O						

Output-Format specifications describe how an output record will look. The output file is named. D is entered in column 15 to indicate a detail line (one printed for every card read). A 1 in column 18 specifies single spacing. All fields to be printed are now listed, one per line, starting one line below entries describing the entire line. End Position, taken from the Printer Spacing Chart, is given for each field. Again, according to the Printer Spacing Chart, three fields are to be edited: DATE needs slashes (Y edit code), QTY must be zero suppressed (Z edit code), and PRICE must be zero suppressed and punctuated with decimals (edit code 3). Edit code 3 was chosen for the PRICE field instead of edit code 2 because PRICE, being a 5-position field with two decimals (xxx.xx), needs no commas.

Writing Specifications For Calculation Operations

Most jobs require some processing. In RPG II, processing can include calculating, comparing, moving, or changing data. In this discussion we'll consider only calculating; that is, adding, subtracting, multiplying, and dividing.

PROGRAM CYCLE OPERATIONS

When you specify a calculation operation, you are adding one more operation to the basic program cycle: the detail calculation operation (see Figure 6).

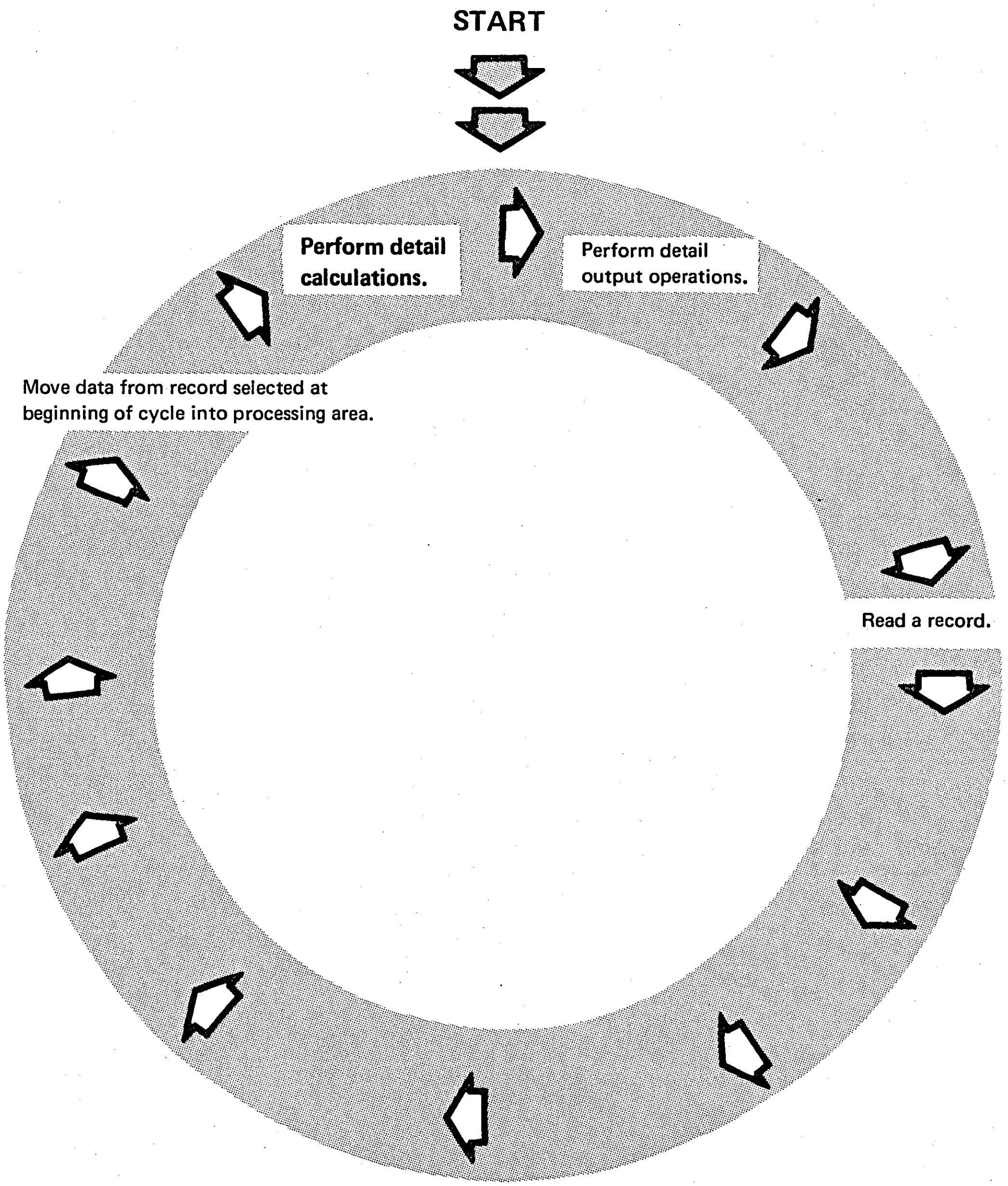
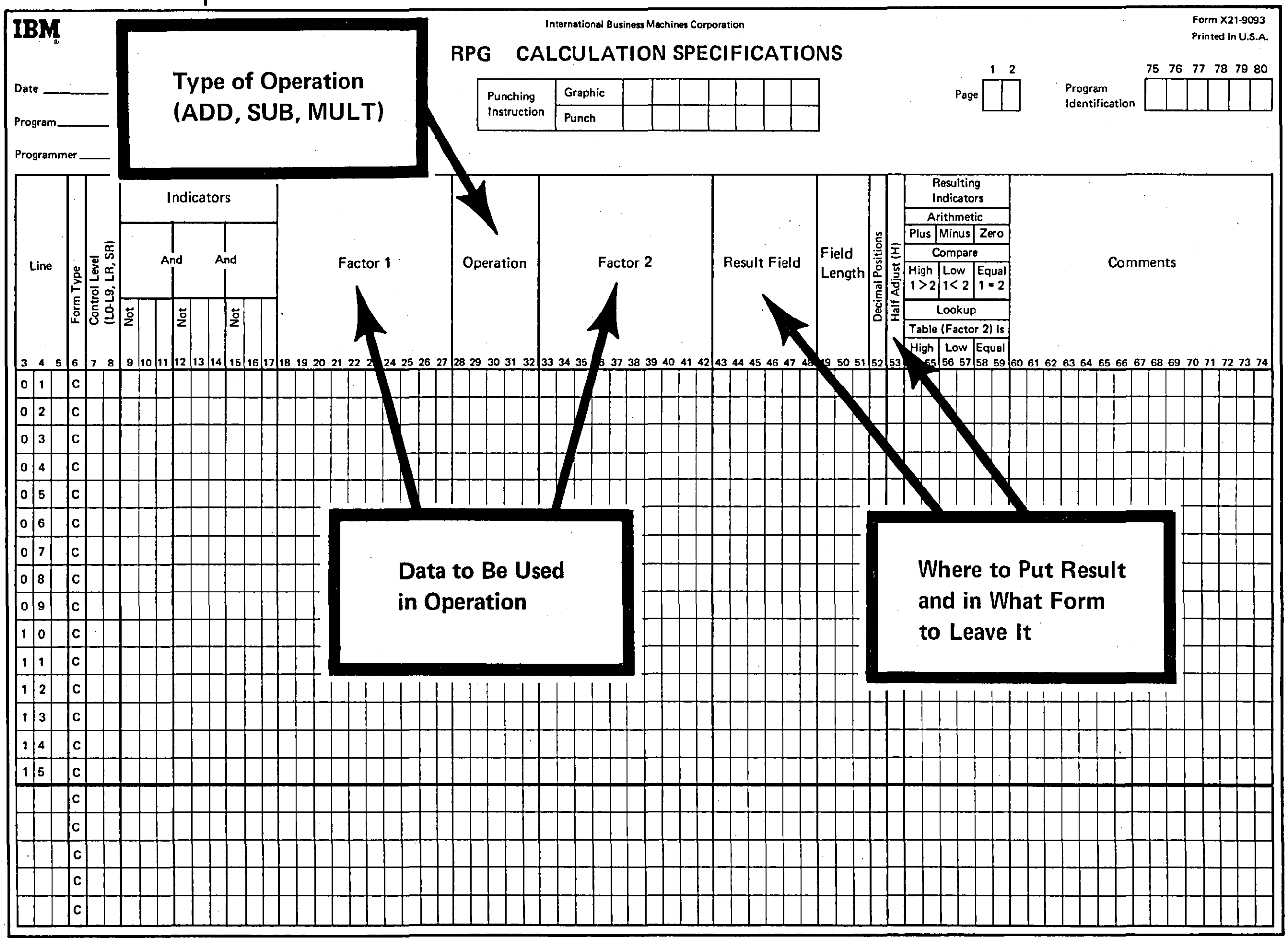

START
Perform detail calculations.
Perform detail output operations.
Move data from record selected at beginning of cycle into processing area.
Read a record.

Figure 6. This is a basic program cycle showing the addition of detail operations.
Because this is a detail operation, it is performed during every cycle for every record read.

The Calculation sheet is used to describe the operations you want performed. Information needed includes the type of operation to be done, the field or constant to be used in the calculation, and where the result of the calculation is to be placed. You fill out the indicated columns:



IBM

International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Form X21-9093
Printed in U.S.A.

Date ______
Program ______
Programmer ______

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3 4 5) | Form Type (6) | Control Level (L0-L9, LR, SR) (7 8) | Indicators: Not, And Not, And Not (9–17) | Factor 1 (18–27) | Operation (28–32) | Factor 2 (33–42) | Result Field (43–48) | Field Length (49–51) | Decimal Positions (52) | Half Adjust (H) (53) | Resulting Indicators: Arithmetic (Plus, Minus, Zero); Compare (High 1>2, Low 1<2, Equal 1=2); Lookup Table (Factor 2) is (High, Low, Equal) (54–59) | Comments (60–74)

Lines 01 C, 02 C, 03 C, 04 C, 05 C, 06 C, 07 C, 08 C, 09 C, 10 C, 11 C, 12 C, 13 C, 14 C, 15 C, C, C, C, C, C

Type of Operation (ADD, SUB, MULT)

Data to Be Used in Operation

Where to Put Result and in What Form to Leave It

Specify one operation per line. In each program cycle, processing steps are done in the order you specified on this sheet. If calculations must be done in a particular order, you must list the operations in that order.

DESCRIBING TYPE OF OPERATION

To indicate the type of operation, you enter one of the following operation codes in columns 28-32 on the Calculation sheet:

ADD (add)

SUB (subtract)

MULT (multiply)

DIV (divide)

DESCRIBING DATA TO BE USED

After you have specified the type of operation, you must identify the data to be used. If you specified ADD, for example, you must tell the system what to add. You do this by naming the fields to be used in columns 18-27 (Factor 1) and 33-42 (Factor 2).

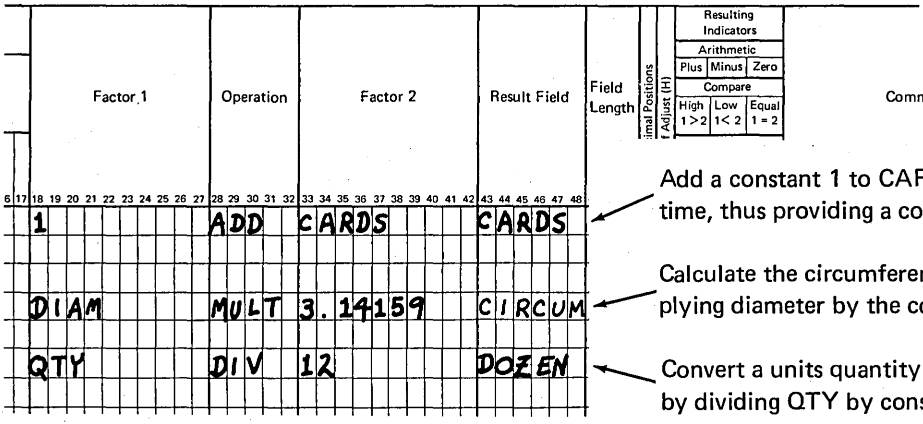
Instead of naming a field in Factor 1 or Factor 2, you can enter a *constant;* that is, the actual data instead of the name of a field containing the data:

500	Constant (actual data)
AMOUNT	Name of a field containing data

Constants can be either numeric or alphameric, but for now we'll discuss only numeric constants. The rules for using numeric constants are as follows:

- Constants can be up to ten numeric digits (0-9).
- Constants can have a sign and decimal point. The sign, if used, must be the leftmost character. The decimal point, if used, must be shown as part of the constant (4.12).
- The first character of the constant must be placed in the leftmost column of the Factor field.
- Constants cannot contain blanks.

The contents of a field can change during execution of a program, but constants do not. If you want to add, multiply, subtract, or divide the same number during every program cycle, you can use a constant:



	Factor 1	Operation	Factor 2	Result Field	Field Length	imal Positions	f Adjust (H)	Resulting Indicators Arithmetic Plus Compare High 1>2	Minus Low 1< 2	Zero Equal 1 = 2	Comn
6 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							
	1	ADD	CARDS	CARDS							
	DIAM	MULT	3.14159	CIRCUM							
	QTY	DIV	12	DOZEN							

Add a constant 1 to CARDS at detail calculation time, thus providing a count of the records processed.

Calculate the circumference of a circle by multiplying diameter by the constant 3.14159.

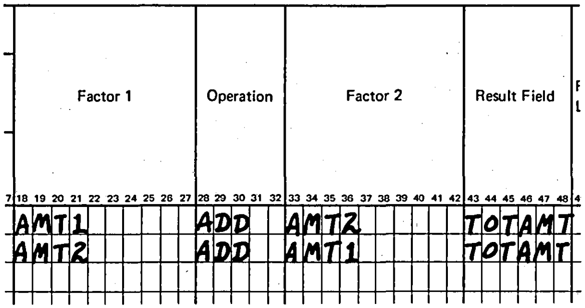
Convert a units quantity to a dozens quantity by dividing QTY by constant 12.

To the compiler, a constant is like a field name. During compilation, the compiler checks Factor 1 and Factor 2 for constants. If there are any, the compiler assigns a storage location for the constant and gives instructions to the computer to put the appropriate constant in that location at the beginning of job execution.

When you enter the fields in Factor 1 and Factor 2, be sure to consider their order because specified operation may have an affect on the result:

ADD

Factor 2 is added to Factor 1 and the sum placed in the Result Field.

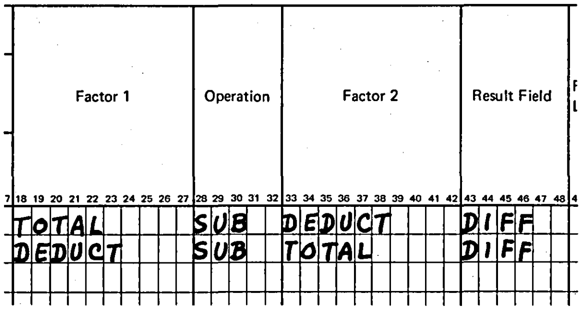


Factor 1 18-27	Operation 28-32	Factor 2 33-42	Result Field 43-48
AMT1	ADD	AMT2	TOTAMT
AMT2	ADD	AMT1	TOTAMT

Either line adds the two amount fields. The order of the fields makes no difference in addition.

SUBTRACT

Factor 2 is subtracted from Factor 1 and the difference placed in the Result Field.

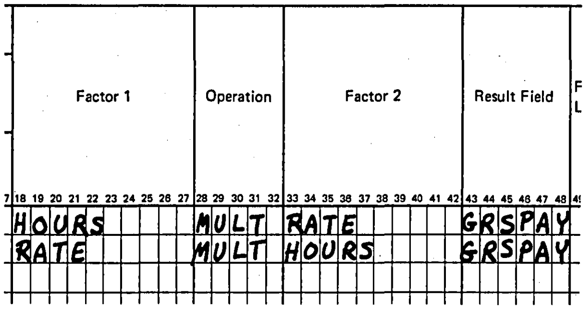


Factor 1 18-27	Operation 28-32	Factor 2 33-42	Result Field 43-48
TOTAL	SUB	DEDUCT	DIFF
DEDUCT	SUB	TOTAL	DIFF

The order of subtract operations is important. The bottom line would not produce the desired result.

MULTIPLY

Factor 1 is multiplied by Factor 2 and the product placed in the Result Field.

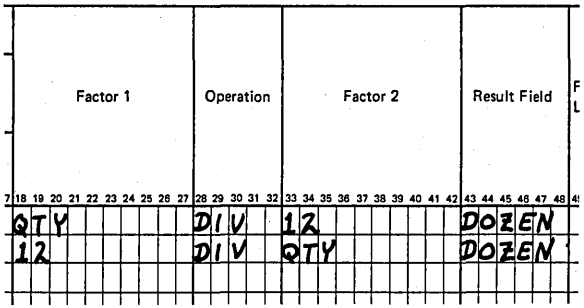


Factor 1 18-27	Operation 28-32	Factor 2 33-42	Result Field 43-48
HOURS	MULT	RATE	GRSPAY
RATE	MULT	HOURS	GRSPAY

Either line multiplies the hours and rate to obtain the gross pay. The order of the fields makes no difference in multiplication.

DIVIDE

Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero.



Factor 1 18-27	Operation 28-32	Factor 2 33-42	Result Field 43-48
QTY	DIV	12	DOZEN
12	DIV	QTY	DOZEN

The order of the fields in divide operations is important. The bottom line will not convert a units quantity to dozens.

DESCRIBING THE RESULT FIELD

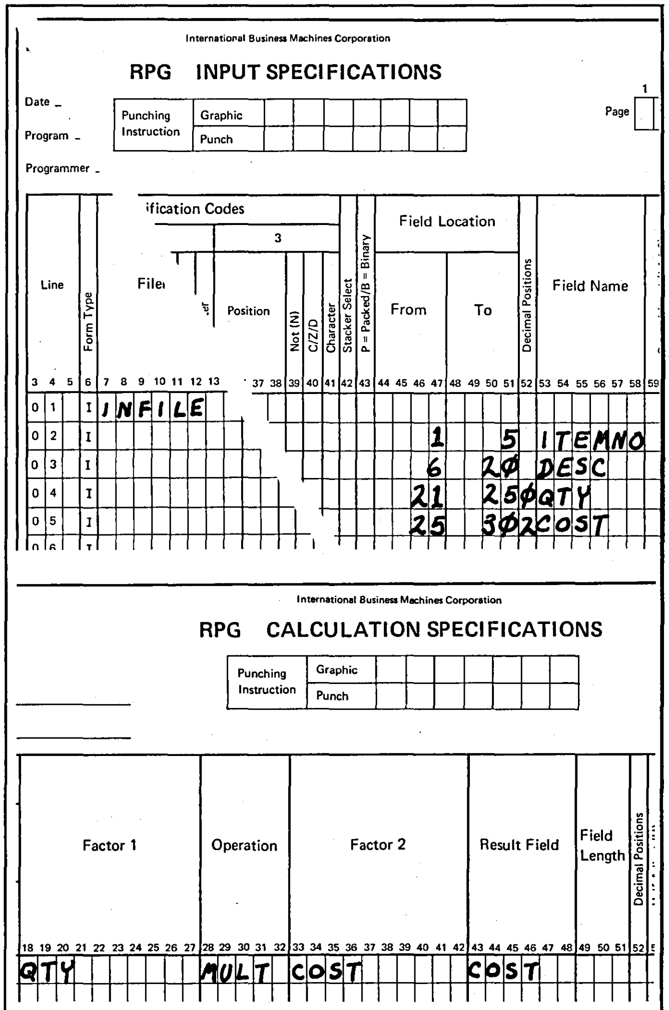
You must specify where you want the result of a calculation stored by naming that field in columns 43-48 (Result Field). The name you enter in the Result Field can be the name of a field already defined on the Input sheet or a new field.

You would need to name a new result field in these two situations:

1. No input field is available. When data is placed in a storage location, it destroys any previous data in that location. Consequently, when the result of a calculation is stored in a result field, it destroys what was in that field. If you need all information from the input record in detail output and also need a result field, you must name a new field.

2. No input field is large enough. You cannot change the length of an input field by specifying a field length on the Calculation sheet that is different than the one you specified on the Input sheet. If you need a larger result field than any available input field, you have to specify a new field and give it a different name and length.

If you name a new field, you must specify field length (columns 49-51) and decimal position (column 52) so the compiler can assign adequate storage for the new field:



International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction: Graphic / Punch

Page 1

Line	Form Type	Filename	Field Location From	Field Location To	Decimal Positions	Field Name
01	I	INFILE				
02	I		1	5		ITEMNO
03	I		6	2Ø		DESC
04	I		21	25	Ø	QTY
05	I		25	3Ø	2	COST

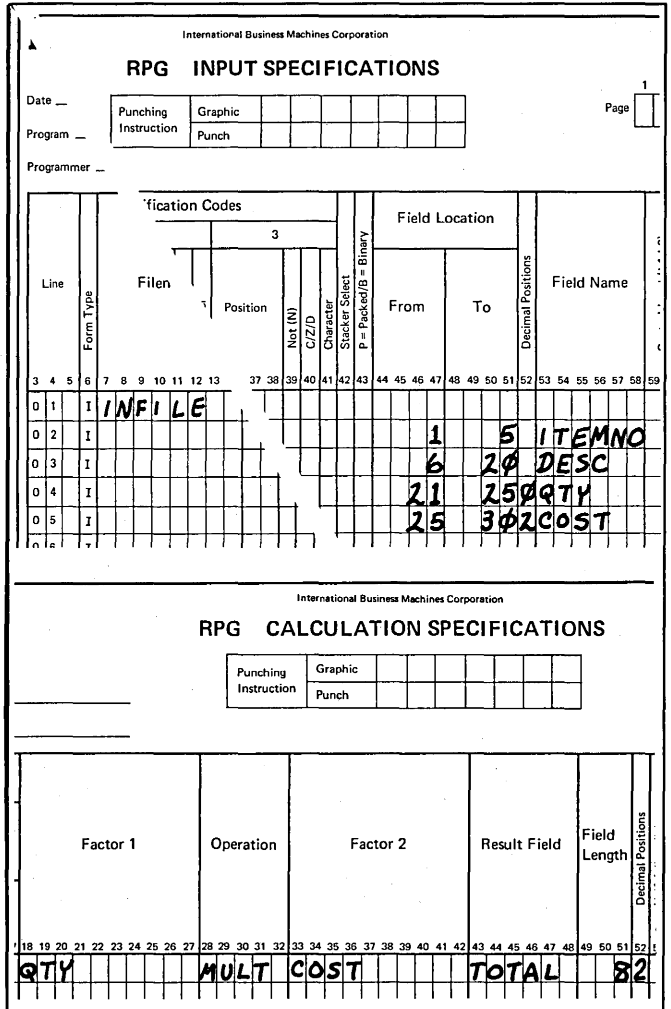
International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Punching Instruction: Graphic / Punch

Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions
QTY	MULT	COST	COST		

The result field COST is not a new field because it is already defined by Input specifications. Field length and decimal position entries are not needed because the compiler already has this information and has set aside storage space for the field.



International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction: Graphic / Punch

Page 1

Line	Form Type	Filename	Field Location From	Field Location To	Decimal Positions	Field Name
01	I	INFILE				
02	I		1	5		ITEMNO
03	I		6	2Ø		DESC
04	I		21	25	Ø	QTY
05	I		25	3Ø	2	COST

International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Punching Instruction: Graphic / Punch

Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions
QTY	MULT	COST	TOTAL	8	2

The result field TOTAL is a new field because it is not defined on the Input sheet. Field length and decimal position entries are needed so that the compiler can set aside a storage area for this field.

Result Field Length

When you name a result field, make sure you specify one large enough to hold the results. Always consider the length of the fields involved in the operations. For example, if you are adding a two-position field to a three-position field, you must determine the largest result you could possibly have:

$$\begin{array}{r} 999 \\ \underline{99} \\ 1098 \end{array}$$

Because there are four digits in this result, you would specify at least 4 as the result field length.

If this calculation would occur many times in your program, as in a running total, you would probably need a result field length larger than 4. It is up to you to determine the decimal positions needed; failure to specify a large enough result field can mean a loss of data.

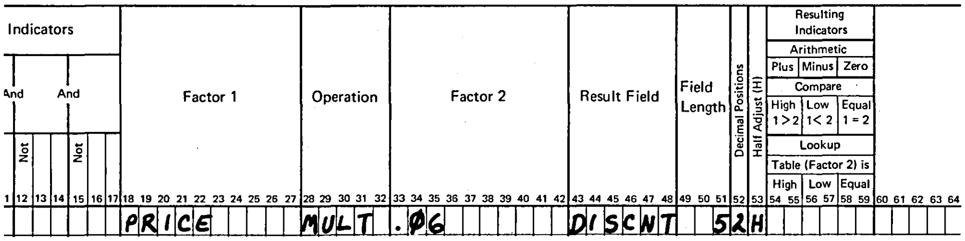
Decimal Positions

For a new result field, be certain to place an entry in column 52. If the new field contains no decimal positions, enter a zero. Remember, this entry indicates types of field (numeric or alphameric) as well as decimal position. If the result field is not specified as numeric by an entry in column 52, the compiler will not provide instructions for the operation.

Half-Adjusting Results (Rounding)

In RPG II, rounding results is called *half-adjusting:* when the number to the right of the last numeral you want to keep is greater than 4, 1 is added to the last numeral. The number 3.14159 rounded to four decimal positions becomes 3.1416. The same number rounded to two decimal positions is 3.14.

To half-adjust any calculation result, you place an H in column 53 of the Calculation sheet on the same line as the field to be half-adjusted:



Indicators And And Not Not	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Arithmetic: Plus Minus Zero Compare: High 1>2 Low 1<2 Equal 1=2 Lookup: Table (Factor 2) is High Low Equal	
1 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
	PRICE	MULT	.Ø6	DISCNT	5	2	H		

In this example, DISCNT is half-adjusted. The entry in column 52 (Decimal Positions) indicates the number of digits to be retained after half-adjusting is completed. In this case, two digits are required. The multiplication and half-adjusting would be done like this:

74.98	← Assumed value of PRICE.
x .06	← Constant representing 6% discount rate.
4.4988	← Result which must be half-adjusted to 2 places.
1	← 1 is added to 9 because 8 is greater than 4.
4.50~~88~~	← Slashed digits are dropped since only two decimal positions are required.

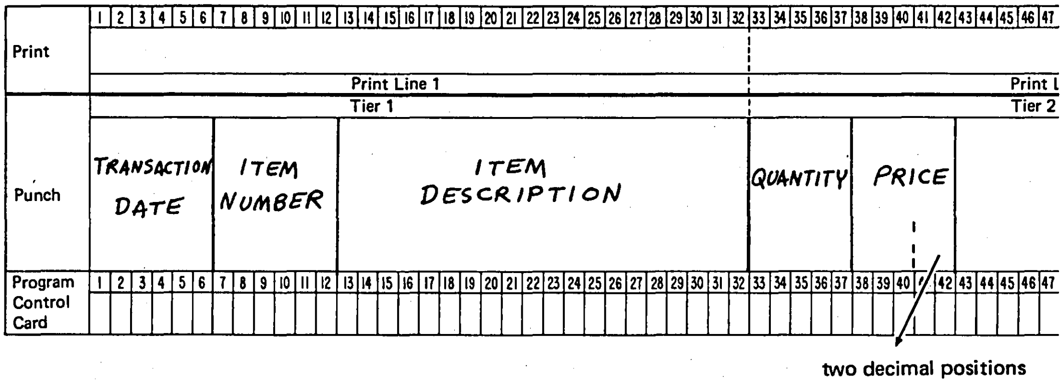
Job 2: Doing Simple Calculations

JOB DEFINITION

Print a report listing all sales transactions for a week. This report is similar to the report created in Job 1. The only difference is the addition of the last column on the report which is the extended cost per item. Extended cost (quantity sold times item price) is not found on the input record and must, therefore, be calculated.

JOB REQUIREMENTS

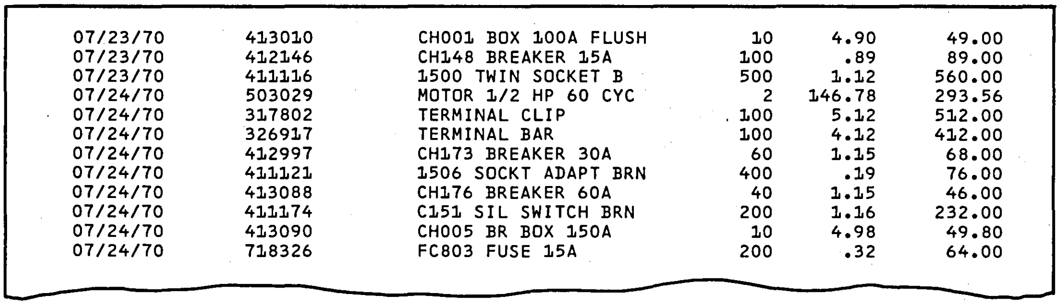
Input: Sales transaction file consisting of 96-column cards. The format of the input records is shown on this Record Layout Form:



	1–6	7–12	13–32	33–37	38–42	43–47
Print	Print Line 1			Print L		
Punch	Tier 1			Tier 2		
	TRANSACTION DATE	ITEM NUMBER	ITEM DESCRIPTION	QUANTITY	PRICE	
Program Control Card						

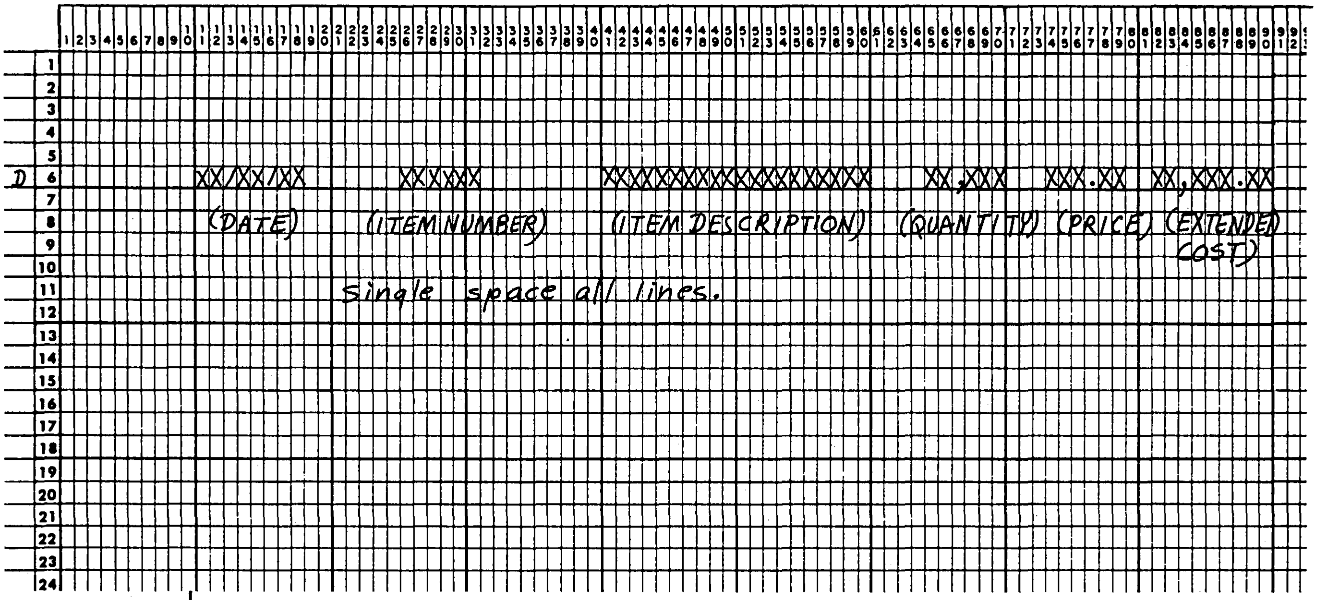
two decimal positions

Output: A Transaction register printed on a 96-position printer:



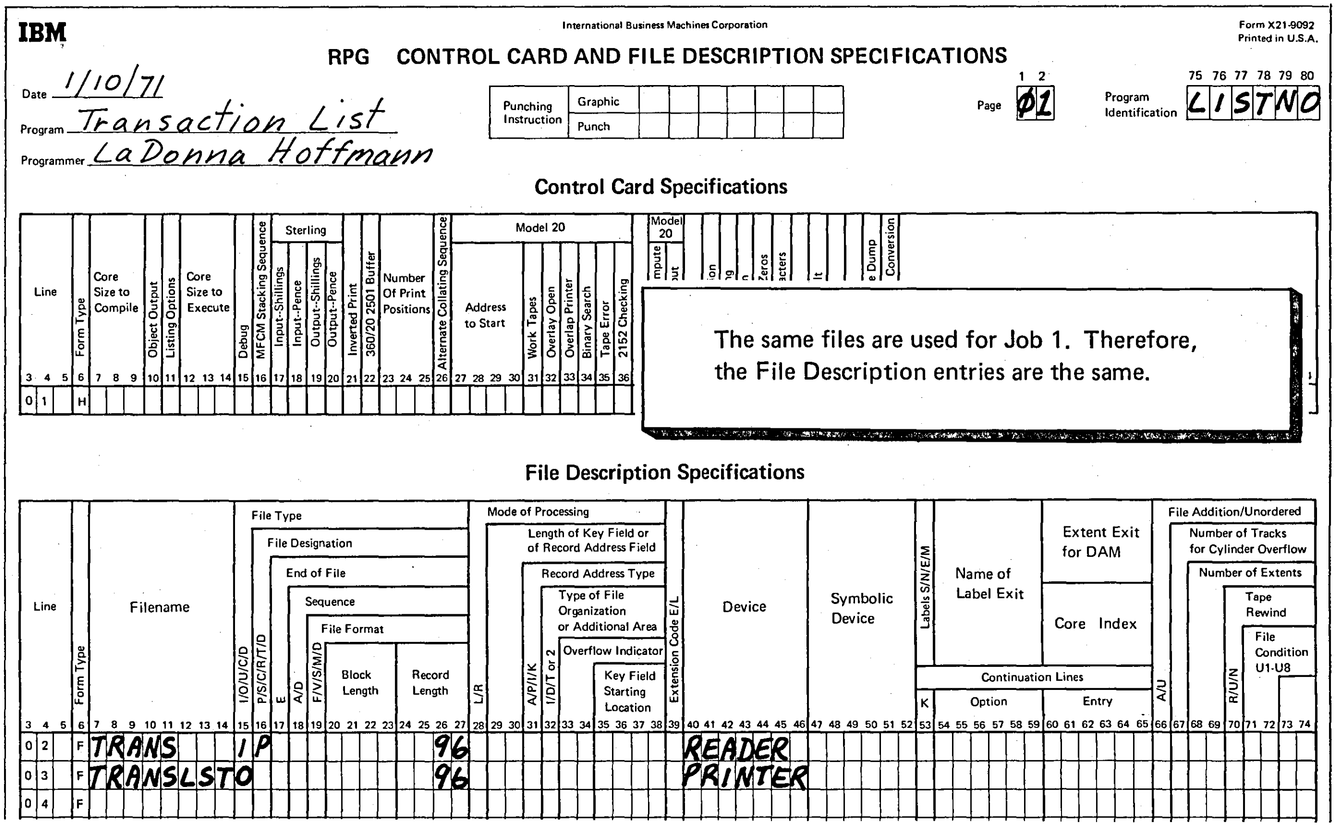
```
07/23/70      413010      CH001 BOX 100A FLUSH        10     4.90      49.00
07/23/70      412146      CH148 BREAKER 15A          100      .89      89.00
07/23/70      411116      1500 TWIN SOCKET B         500     1.12     560.00
07/24/70      503029      MOTOR 1/2 HP 60 CYC          2   146.78     293.56
07/24/70      317802      TERMINAL CLIP              100     5.12     512.00
07/24/70      326917      TERMINAL BAR               100     4.12     412.00
07/24/70      412997      CH173 BREAKER 30A           60     1.15      68.00
07/24/70      411121      1506 SOCKT ADAPT BRN       400      .19      76.00
07/24/70      413088      CH176 BREAKER 60A           40     1.15      46.00
07/24/70      411174      C151 SIL SWITCH BRN        200     1.16     232.00
07/24/70      413090      CH005 BR BOX 150A           10     4.98      49.80
07/24/70      718326      FC803 FUSE 15A             200      .32      64.00
```

This Printer Spacing Chart shows how the report is formatted:



```
D  6   XX/XX/XX      XXXXXX        XXXXXXXXXXXXXXXXXXXX    XX,XXX    XXX.XX    XX,XXX.XX
   8   (DATE)       (ITEM NUMBER)  (ITEM DESCRIPTION)    (QUANTITY) (PRICE)  (EXTENDED
   9                                                                           COST)
  11              single space all lines.
```

JOB SPECIFICATIONS



IBM

International Business Machines Corporation

Form X21-9092
Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71

Program Transaction List

Programmer LaDonna Hoffmann

Punching Instruction: Graphic / Punch

Page Ø1

Program Identification LISTNO

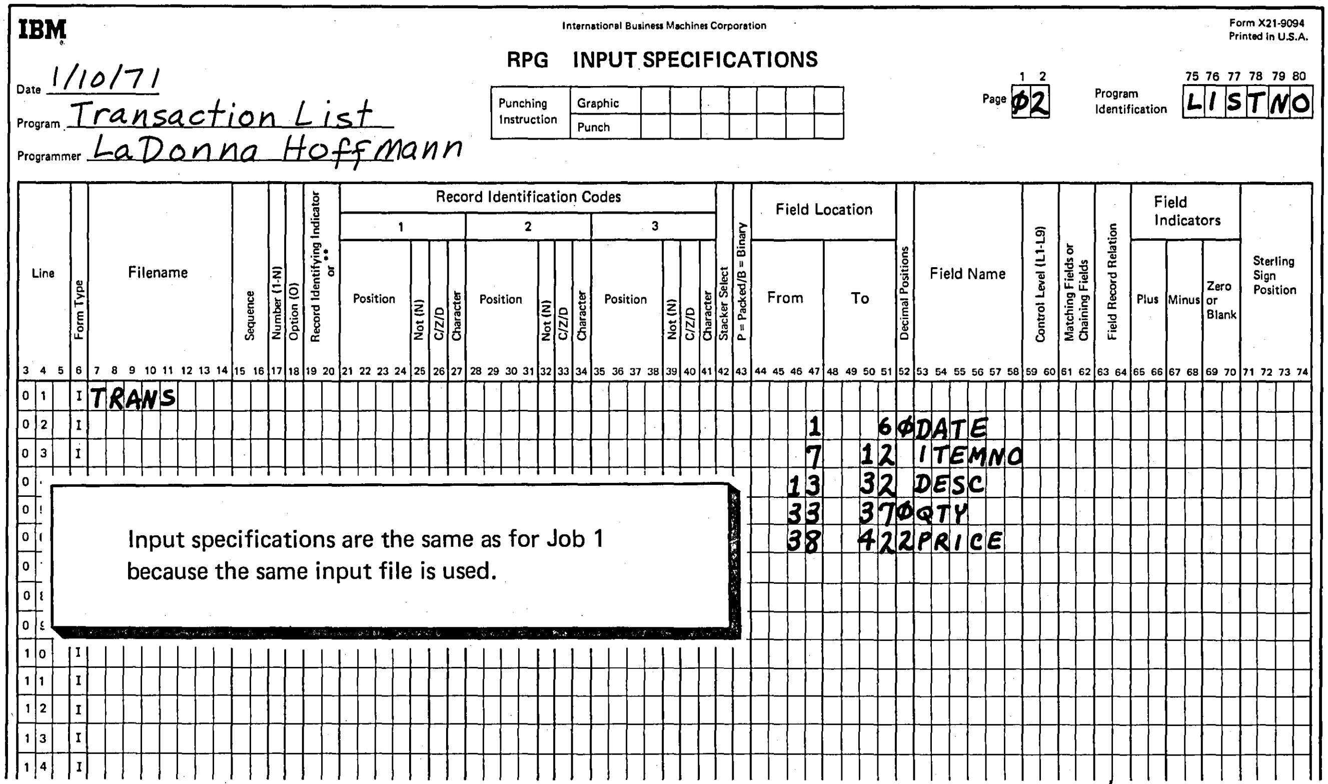
Control Card Specifications

Line	Form Type
01	H

The same files are used for Job 1. Therefore, the File Description entries are the same.

File Description Specifications

Line	Form Type	Filename	I/O/U/C/D	P/S/C/R/T/D	Record Length	Device
02	F	TRANS	I	P	96	READER
03	F	TRANSLST	O		96	PRINTER
04	F					



IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date 1/10/71

Program Transaction List

Programmer LaDonna Hoffmann

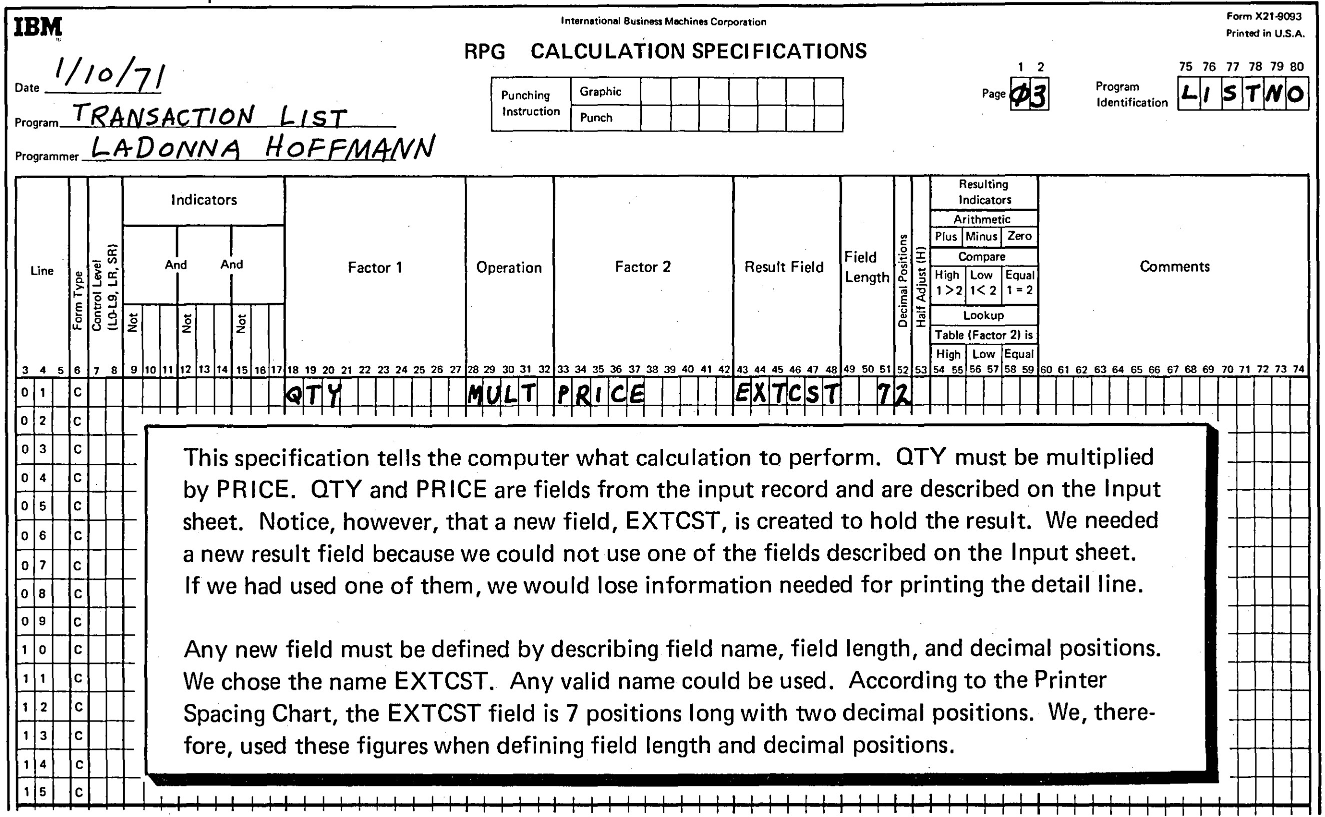
Punching Instruction: Graphic / Punch

Page Ø2

Program Identification LISTNO

Line	Form Type	Filename	Field Location From	Field Location To	Decimal Positions	Field Name
01	I	TRANS				
02	I		1	6	Ø	DATE
03	I		7	12		ITEMNO
			13	32		DESC
			33	37	Ø	QTY
			38	42	2	PRICE
10	I					
11	I					
12	I					
13	I					
14	I					

Input specifications are the same as for Job 1 because the same input file is used.



IBM — International Business Machines Corporation — Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date 1/10/71

Program TRANSACTION LIST

Programmer LADONNA HOFFMANN

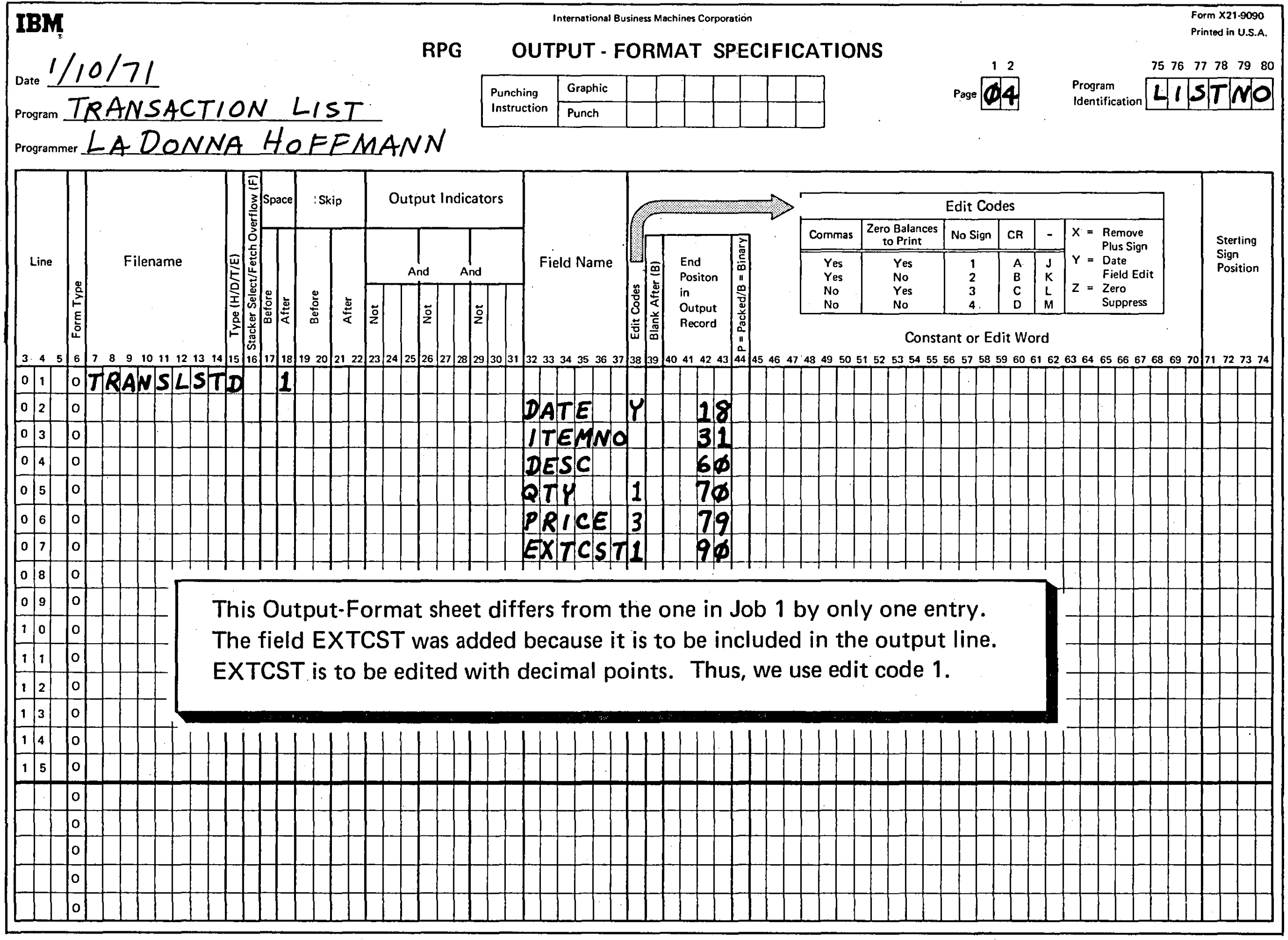
Punching Instruction: Graphic / Punch

Page Ø3 — Program Identification LISTNO

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators (Not / And Not / And Not)	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C			QTY	MULT	PRICE	EXTCST	7	2			
02	C											
03	C											
04	C											
05	C											
06	C											
07	C											
08	C											
09	C											
10	C											
11	C											
12	C											
13	C											
14	C											
15	C											

This specification tells the computer what calculation to perform. QTY must be multiplied by PRICE. QTY and PRICE are fields from the input record and are described on the Input sheet. Notice, however, that a new field, EXTCST, is created to hold the result. We needed a new result field because we could not use one of the fields described on the Input sheet. If we had used one of them, we would lose information needed for printing the detail line.

Any new field must be defined by describing field name, field length, and decimal positions. We chose the name EXTCST. Any valid name could be used. According to the Printer Spacing Chart, the EXTCST field is 7 positions long with two decimal positions. We, therefore, used these figures when defining field length and decimal positions.



IBM — International Business Machines Corporation — Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71

Program TRANSACTION LIST

Programmer LADONNA HOFFMANN

Punching Instruction: Graphic / Punch

Page Ø4 — Program Identification LISTNO

Edit Codes	Commas	Zero Balances to Print	No Sign	CR	-
	Yes	Yes	1	A	J
	Yes	No	2	B	K
	No	Yes	3	C	L
	No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Packed/B = Binary	Constant or Edit Word	Sterling Sign Position
01	O	TRANSLST	D			1										
02	O									DATE	Y		18			
03	O									ITEMNO			31			
04	O									DESC			6Ø			
05	O									QTY	1		7Ø			
06	O									PRICE	3		79			
07	O									EXTCST	1		9Ø			
08	O															
09	O															
10	O															
11	O															
12	O															
13	O															
14	O															
15	O															

This Output-Format sheet differs from the one in Job 1 by only one entry. The field EXTCST was added because it is to be included in the output line. EXTCST is to be edited with decimal points. Thus, we use edit code 1.

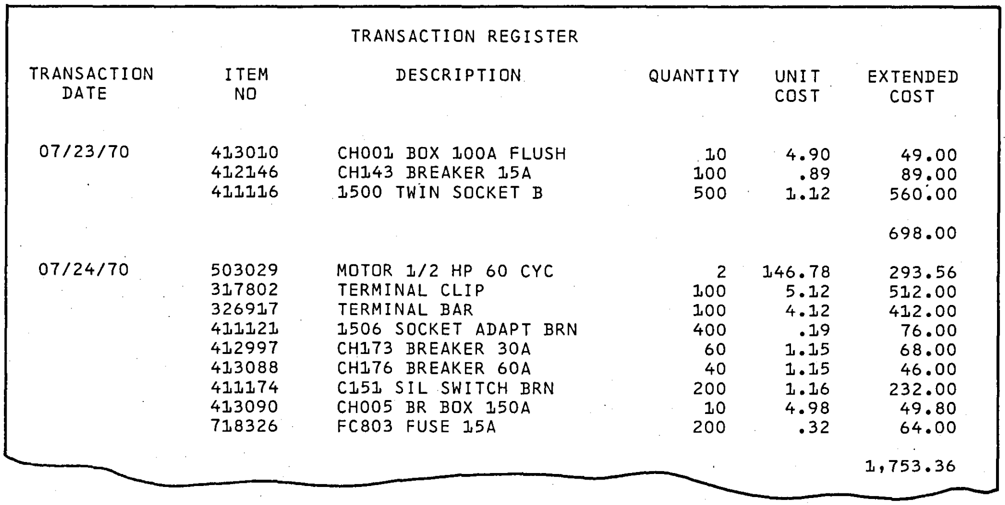
Writing Specifications For Indicators

So far you've learned how to use an RPG II program cycle for producing simple reports. However, actual business reports would be more complex. They would include more information, have page and column headings, and probably include subtotals and final totals.

A report like that shown in Figure 7 would require printing four different lines: report heading, column headings, detail lines, and total lines. Some of these lines must be printed only at certain times: headings would be printed only at the top of the page and totals only after all detail lines are printed. To produce the report correctly, you must use *indicators* to specify when you want certain things done.

To you, indicators are two numbers or alphabetic characters you specify on the specification sheets. To the object program, indicators are like switches, located in the computer. They mean one thing when on; another when off. You can use several types of indicators; each type signals something different.

You must know which program cycle operations are done when indicators are used because it is the only way you can use indicators correctly. In this section, indicators are discussed one at a time. You will learn when to use indicators, how to specify them, and which program cycle operations are associated with each.



```
                                  TRANSACTION REGISTER

TRANSACTION        ITEM             DESCRIPTION               QUANTITY   UNIT      EXTENDED
   DATE             NO                                                    COST        COST

 07/23/70          413010      CH001 BOX 100A FLUSH               10       4.90       49.00
                   412146      CH143 BREAKER 15A                 100        .89       89.00
                   411116      1500 TWIN SOCKET B                500       1.12      560.00

                                                                                     698.00

 07/24/70          503029      MOTOR 1/2 HP 60 CYC                 2     146.78      293.56
                   317802      TERMINAL CLIP                     100       5.12      512.00
                   326917      TERMINAL BAR                      100       4.12      412.00
                   411121      1506 SOCKET ADAPT BRN             400        .19       76.00
                   412997      CH173 BREAKER 30A                  60       1.15       68.00
                   413088      CH176 BREAKER 60A                  40       1.15       46.00
                   411174      C151 SIL SWITCH BRN               200       1.16      232.00
                   413090      CH005 BR BOX 150A                  10       4.98       49.80
                   718326      FC803 FUSE 15A                    200        .32       64.00

                                                                                   1,753.36
```

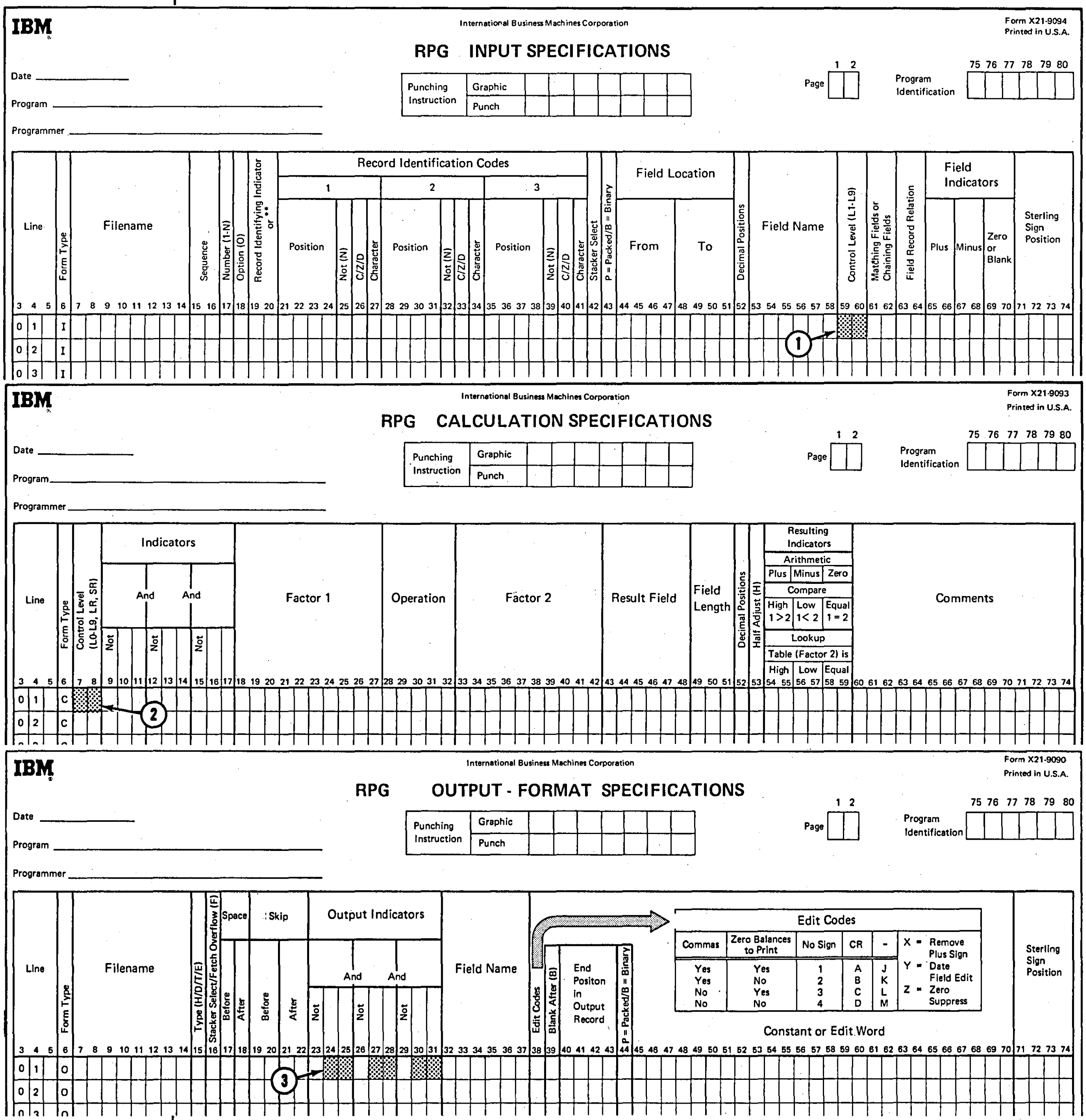
Figure 7. This report is similar to those shown before, but note the addition of headings and totals.

CONTROL LEVEL INDICATORS

Control level indicators are used when you want to calculate and print totals. Nine different indicators can be used (L1 through L9), allowing as many as nine different totals in the same program. The control level indicators tell the program two things:

1. When totals should be calculated.

2. Which calculations and output operations are total operations.



Control level indicators are specified at (1) to tell the computer when total operations are to be done. They are used at (2) and (3) to tell which operations are total operations.

An input field specified on the Input sheet determines when totals should be calculated and printed. This input field is called a *control field.* Whenever the contents of the control field changes, a *control break* occurs. A control break turns on the control level indicator assigned to the control field, then all calculation and output operations (total operations) conditioned by the same control level indicator are done.

Program Cycle Operations

Figure 8 shows the program cycle operations associated with control level indicators. The computer can do calculations and output operations at two different times in one cycle: at detail time and at total time. Total operations are not done in every cycle; they are done during the cycle in which the control field changes.

After a record is read, the program determines whether the control field in the record just read is different than the control field in the previous record. If it is, a control break occurs and the control level indicator you specified is set on. When the indicator is on, it means that all records in the control group have been read and total operations can be performed. Control level indicators are then set off before the next record is read.

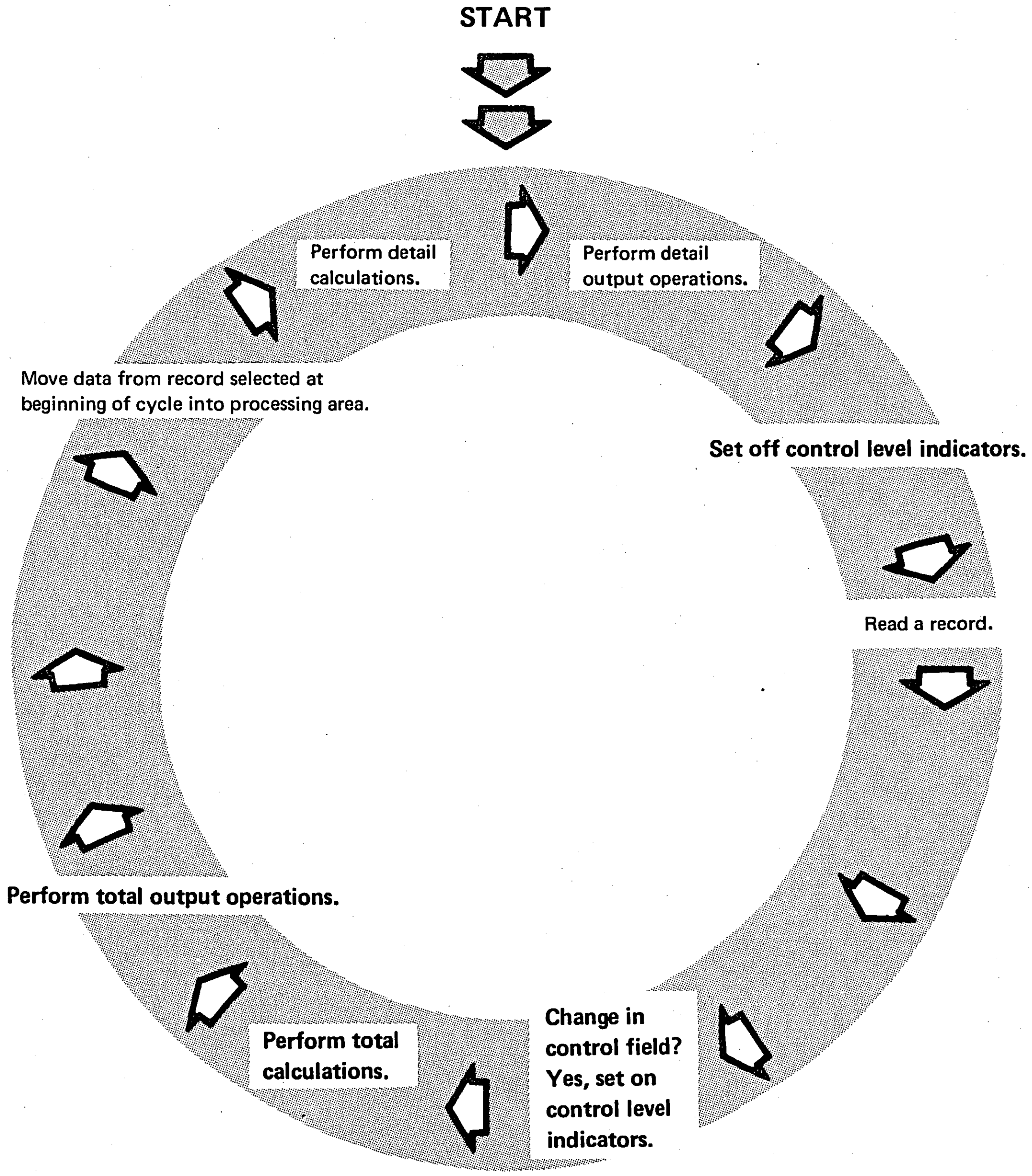
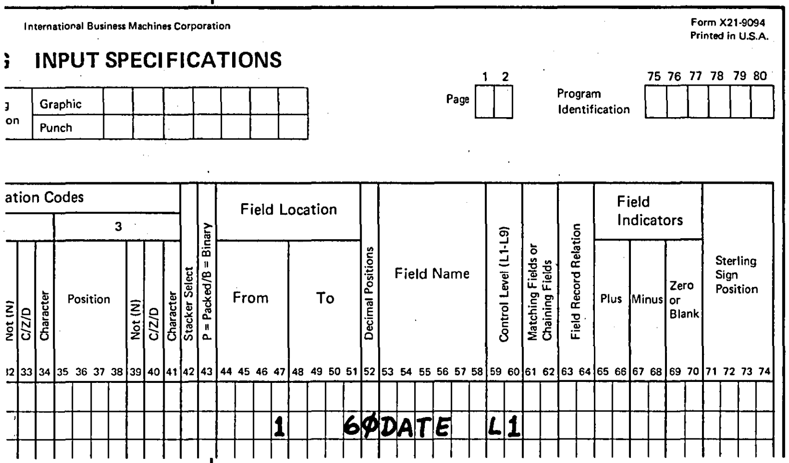

START
Perform detail output operations.
Set off control level indicators.
Read a record.
Change in control field? Yes, set on control level indicators.
Perform total calculations.
Perform total output operations.
Move data from record selected at beginning of cycle into processing area.
Perform detail calculations.

Figure 8. Program Cycle Operations for the Control Level Indicators

Detail operations for the record that caused the control break are done only after total operations for previous records. If the record that caused the control break was processed before the total operations were done, information from that record would be included with information from records in the previous group. The totals from the previous group would then be wrong.

RPG II Specifications

To specify a field as a control field, you assign a control level indicator (L1-L9) to an input field in columns 59-60 on the Input sheet:



International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

INPUT SPECIFICATIONS

Graphic
Punch

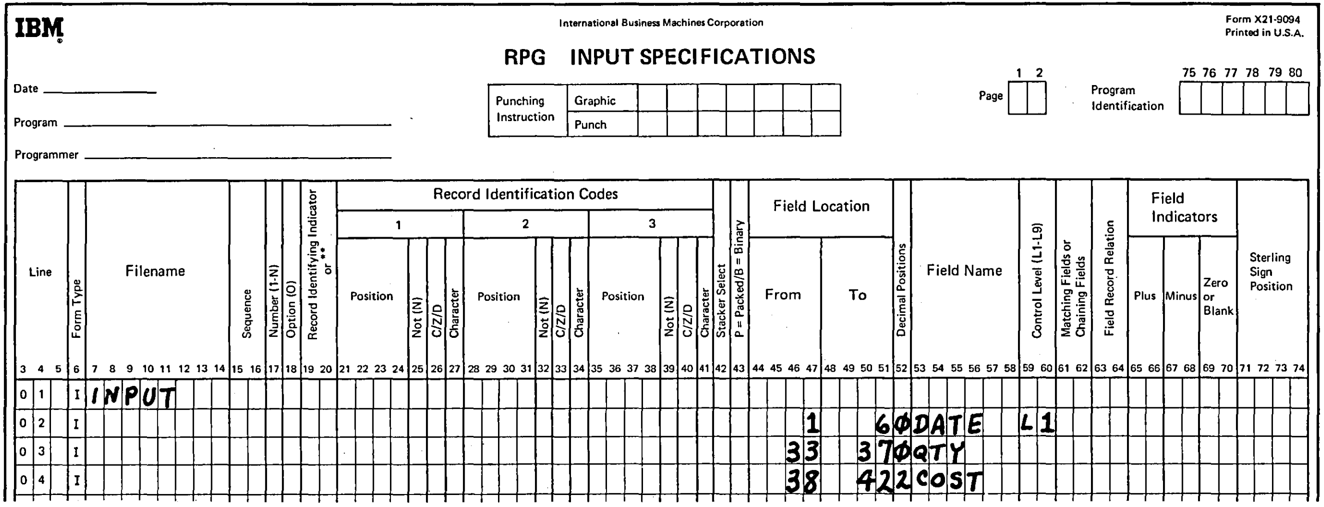
Page 1 2

Program Identification 75 76 77 78 79 80

Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields or Chaining Fields (61-62)	Field Record Relation (63-64)	Field Indicators Plus (65-66)	Field Indicators Minus (67-68)	Field Indicators Zero or Blank (69-70)	Sterling Sign Position (71-74)
1	6Ø		DATE	L1						

L1, assigned on the same specification line as the date field, tells the computer to use DATE as the control field.

To specify which operations are total operations, you assign the same control level indicator in columns 7-8 on the Calculation sheet and in columns 24-25, 27-28, or 30-31 on the Output-Format sheet:

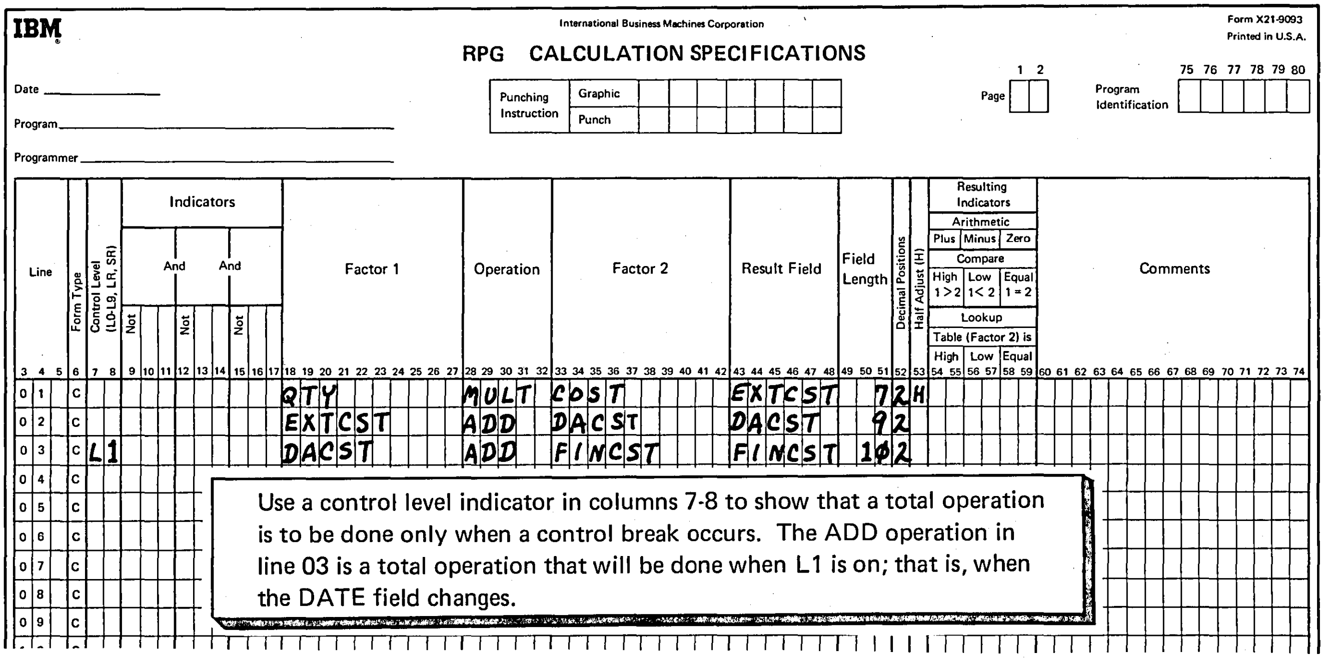


IBM — International Business Machines Corporation — Form X21-9094, Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date ____ Program ____ Programmer ____ Punching Instruction: Graphic / Punch. Page (1 2). Program Identification (75 76 77 78 79 80)

Line	Form Type	Filename	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	INPUT					
02	I		1	6	Ø	DATE	L1
03	I		33	37	Ø	QTY	
04	I		38	42	2	COST	



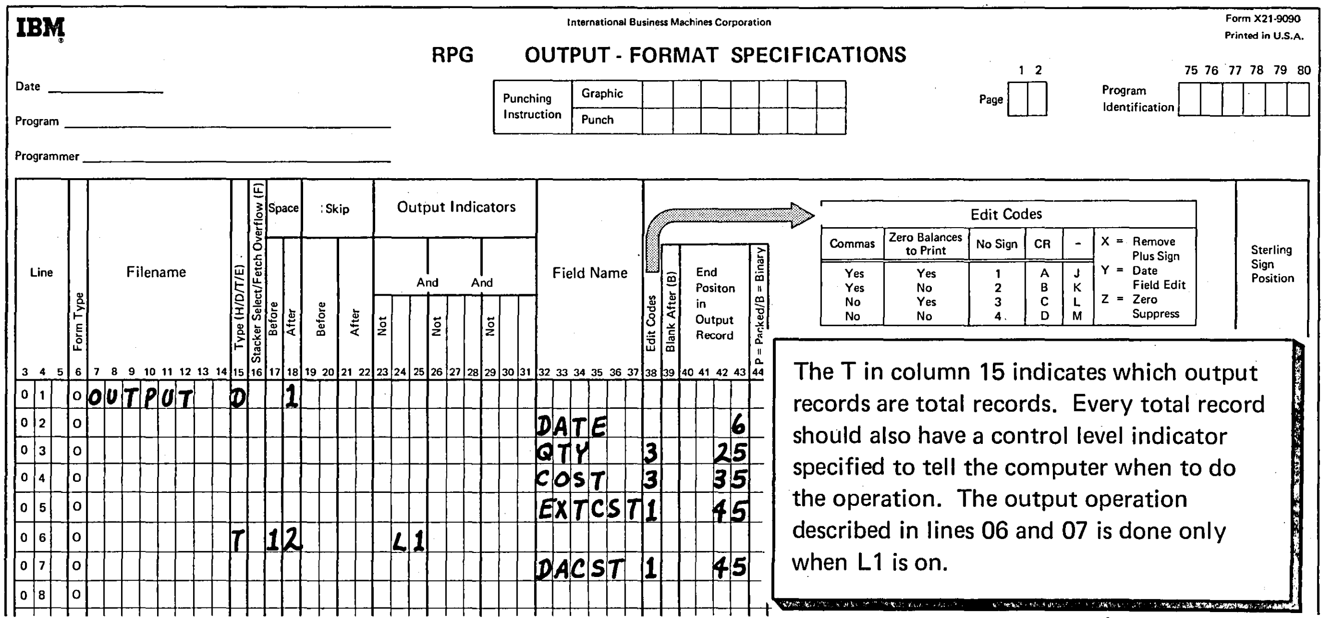
IBM — International Business Machines Corporation — Form X21-9093, Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date ____ Program ____ Programmer ____ Punching Instruction: Graphic / Punch. Page (1 2). Program Identification (75 76 77 78 79 80)

Line	Form Type	Control Level (L0-L9, LR, SR)	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)
01	C		QTY	MULT	COST	EXTCST	7	2	H
02	C		EXTCST	ADD	DACST	DACST	9	2	
03	C	L1	DACST	ADD	FINCST	FINCST	1Ø	2	
04	C								
05	C								
06	C								
07	C								
08	C								
09	C								

Use a control level indicator in columns 7-8 to show that a total operation is to be done only when a control break occurs. The ADD operation in line 03 is a total operation that will be done when L1 is on; that is, when the DATE field changes.



IBM — International Business Machines Corporation — Form X21-9090, Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

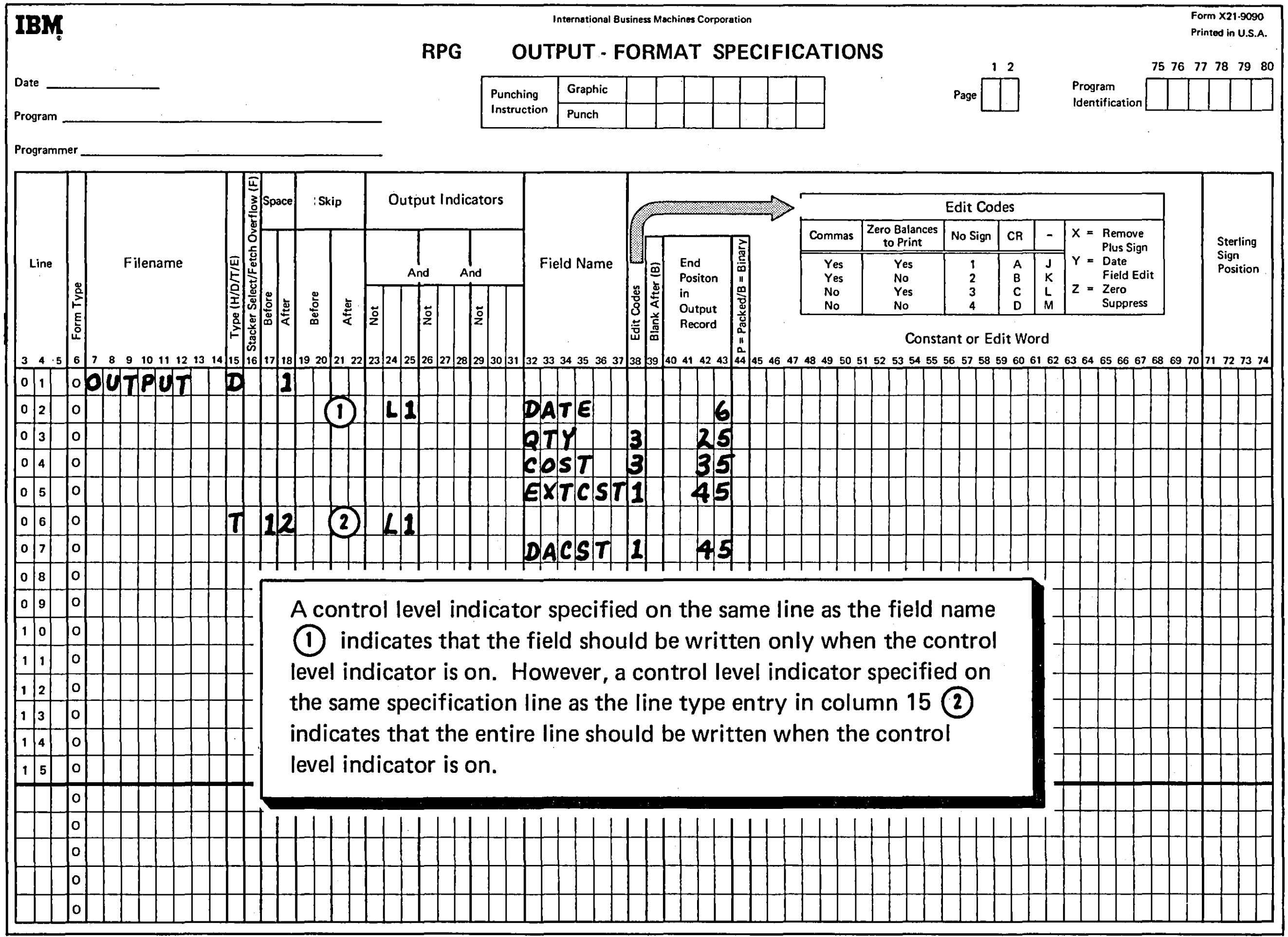
Date ____ Program ____ Programmer ____ Punching Instruction: Graphic / Punch. Page (1 2). Program Identification (75 76 77 78 79 80)

Edit Codes: Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Output Indicators	Field Name	Edit Codes	End Position in Output Record
01	O	OUTPUT	D		1				
02	O						DATE		6
03	O						QTY	3	25
04	O						COST	3	35
05	O						EXTCST	1	45
06	O		T	1	2	L1			
07	O						DACST	1	45
08	O								

The T in column 15 indicates which output records are total records. Every total record should also have a control level indicator specified to tell the computer when to do the operation. The output operation described in lines 06 and 07 is done only when L1 is on.

You can specify up to three different indicators on a line on the Output-Format sheet. If you are using only one indicator, you can enter it in any one of the three positions. The control level indicators specified on this sheet can be used to condition an entire output record or only certain fields in the output record:



IBM

International Business Machines Corporation

RPG OUTPUT - FORMAT SPECIFICATIONS

Form X21-9090
Printed in U.S.A.

Date

Program

Programmer

Punching Instruction: Graphic / Punch

Page (1 2)

Program Identification (75 76 77 78 79 80)

Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After (B)	End Position in Output Record
01	O	OUTPUT	D		1							
02	O						①	L1	DATE			6
03	O								QTY	3		25
04	O								COST	3		35
05	O								EXTCST	1		45
06	O		T	1	2		②	L1				
07	O								DACST	1		45

A control level indicator specified on the same line as the field name ① indicates that the field should be written only when the control level indicator is on. However, a control level indicator specified on the same specification line as the line type entry in column 15 ② indicates that the entire line should be written when the control level indicator is on.

Using the Blank-After Specification

In RPG II, you can set fields in storage to blanks (in the case of alphameric fields) or zeros (in the case of numeric fields) after they have been written out. You do this by entering a B in column 39 of the Output-Format sheet.

This is a particularly useful feature when you are doing total operations. It allows you to use the same field over and over for accumulating and printing totals. For example, you could use a numeric field to accumulate totals for a particular group of records. After the totals are accumulated and printed for that group, you can use the same numeric field to accumulate the totals for the next group of records. To do this, place a B in column 39 for the total field. If you don't place a B in column 39, the totals for the second group of records would be added to the totals for the first group of records.

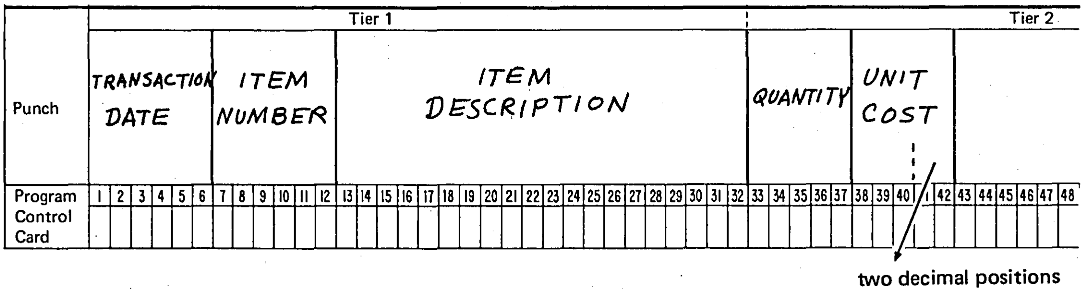
Job 3: Using Control Level Indicators To Calculate And Print Totals

JOB DEFINITION

Print a weekly sales transaction report that lists all daily transactions and gives the total sales for each day. This report is similar to the reports produced in Jobs 1 and 2. All items sold each day are listed. Item number, item description, quantity sold, unit cost, and extended cost (quantity times unit cost) are included for each item. The date is printed only for the first transaction encountered that has a new date. The total sales amount for a day is printed after all transactions for that day have been recorded.

JOB REQUIREMENTS

Input: Sales transaction file consisting of 96-column cards. Cards are arranged in ascending order by date. The format of the input records is shown on this Record Layout Form:



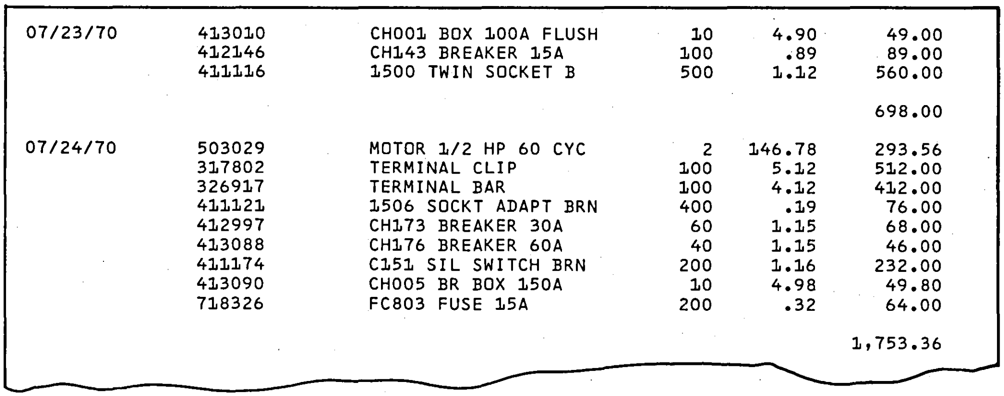
	Tier 1					Tier 2
Punch	TRANSACTION DATE	ITEM NUMBER	ITEM DESCRIPTION	QUANTITY	UNIT COST	
Program Control Card	1 2 3 4 5 6	7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32	33 34 35 36 37	38 39 40 41 42	43 44 45 46 47 48

two decimal positions

Processing:

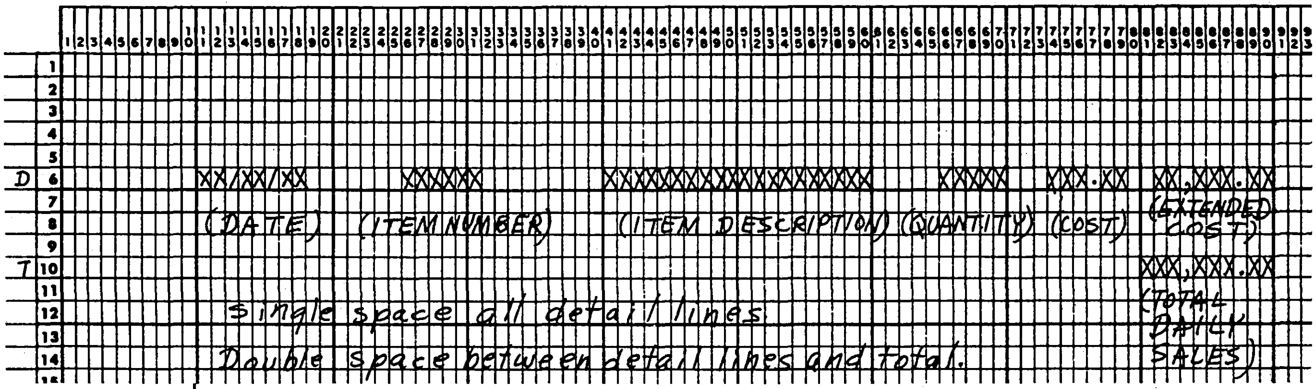
- Multiply quantity times unit cost to find extended cost.
- Find total of all item sales per day.

Output: A Transaction Register printed on a 96-position printer:



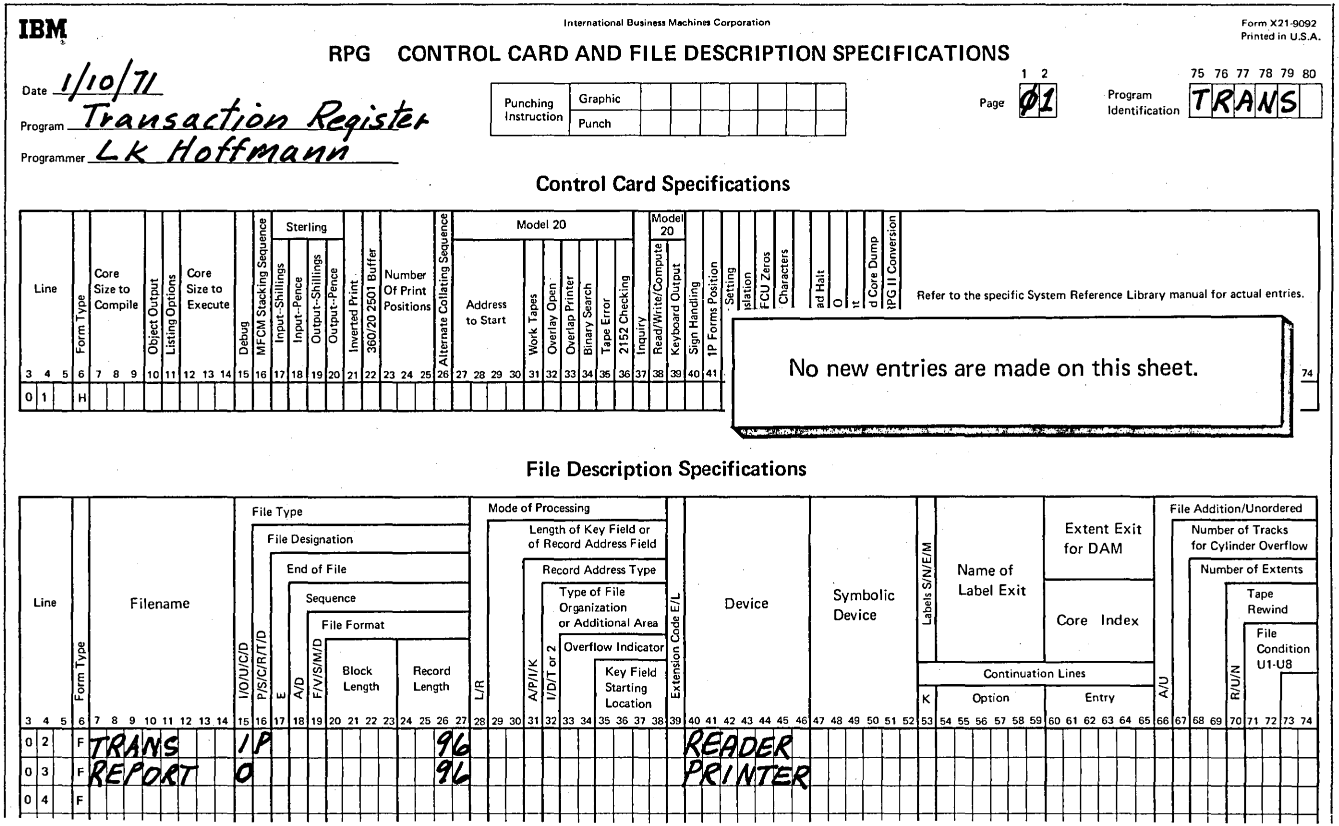
```
07/23/70        413010          CH001 BOX 100A FLUSH       10     4.90      49.00
                412146          CH143 BREAKER 15A         100      .89      89.00
                411116          1500 TWIN SOCKET B        500     1.12     560.00

                                                                           698.00

07/24/70        503029          MOTOR 1/2 HP 60 CYC         2   146.78     293.56
                317802          TERMINAL CLIP             100     5.12     512.00
                326917          TERMINAL BAR              100     4.12     412.00
                411121          1506 SOCKT ADAPT BRN      400      .19      76.00
                412997          CH173 BREAKER 30A          60     1.15      68.00
                413088          CH176 BREAKER 60A          40     1.15      46.00
                411174          C151 SIL SWITCH BRN       200     1.16     232.00
                413090          CH005 BR BOX 150A          10     4.98      49.80
                718326          FC803 FUSE 15A            200      .32      64.00

                                                                         1,753.36
```

This Printer Spacing Chart shows how the report is formatted:



```
   1
   2
   3
   4
   5
D  6          XX/XX/XX     XXXXXX        XXXXXXXXXXXXXXXXXXX    XXXXX   XXX.XX  XX,XXX.XX
   7                                                                             (EXTENDED
   8          (DATE)   (ITEM NUMBER)     (ITEM DESCRIPTION)  (QUANTITY)  (COST)    COST)
   9
T 10                                                                             XXX,XXX.XX
  11                                                                              (TOTAL
  12            single space all detail lines                                      DAILY
  13                                                                               SALES)
  14          Double space between detail lines and total.
  15
```

JOB SPECIFICATIONS



IBM International Business Machines Corporation Form X21-9092 Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71
Program Transaction Register
Programmer LK Hoffmann
Page Ø1
Program Identification TRANS

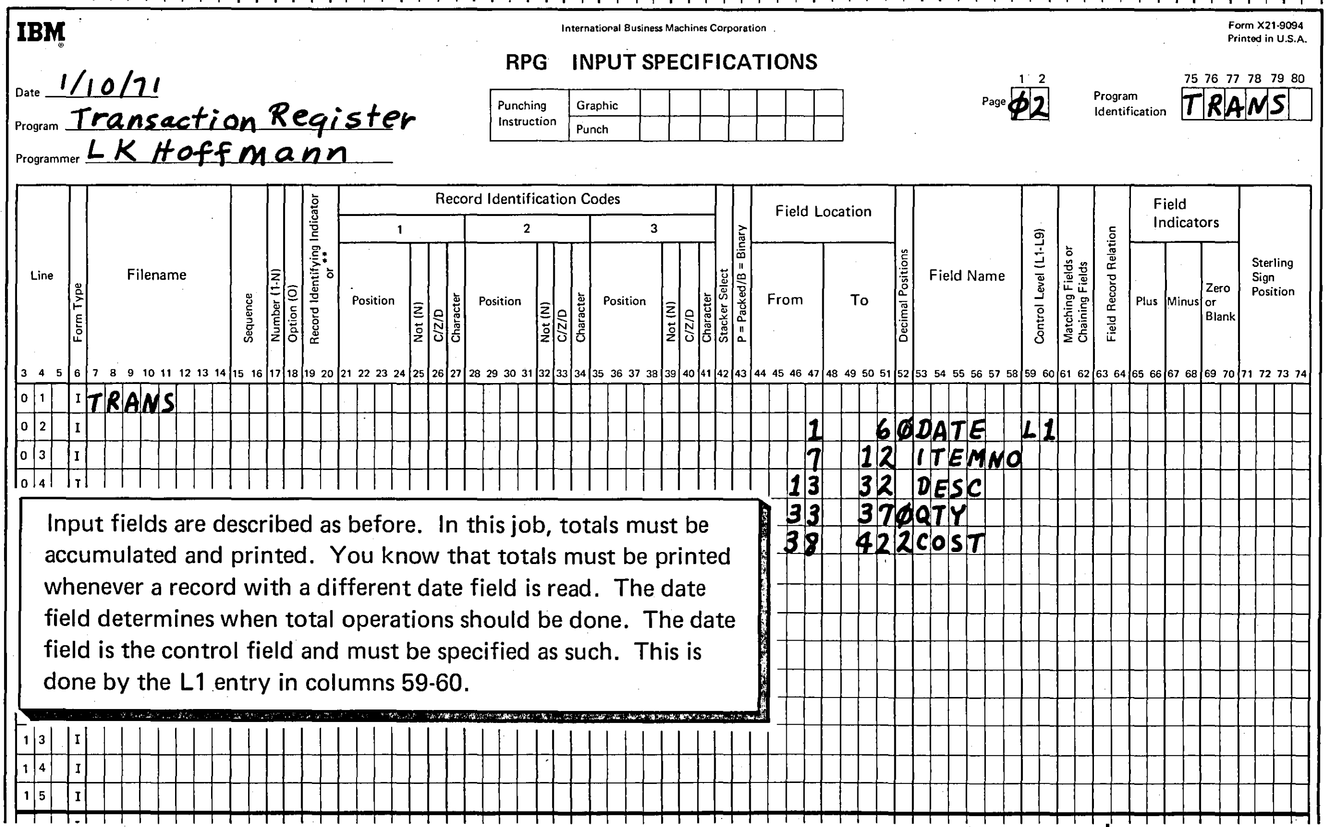
Control Card Specifications

Line	Form Type
01	H

No new entries are made on this sheet.

File Description Specifications

Line	Form Type	Filename	File Type	File Designation	Record Length	Device
02	F	TRANS	I	P	96	READER
03	F	REPORT	O		96	PRINTER
04	F					



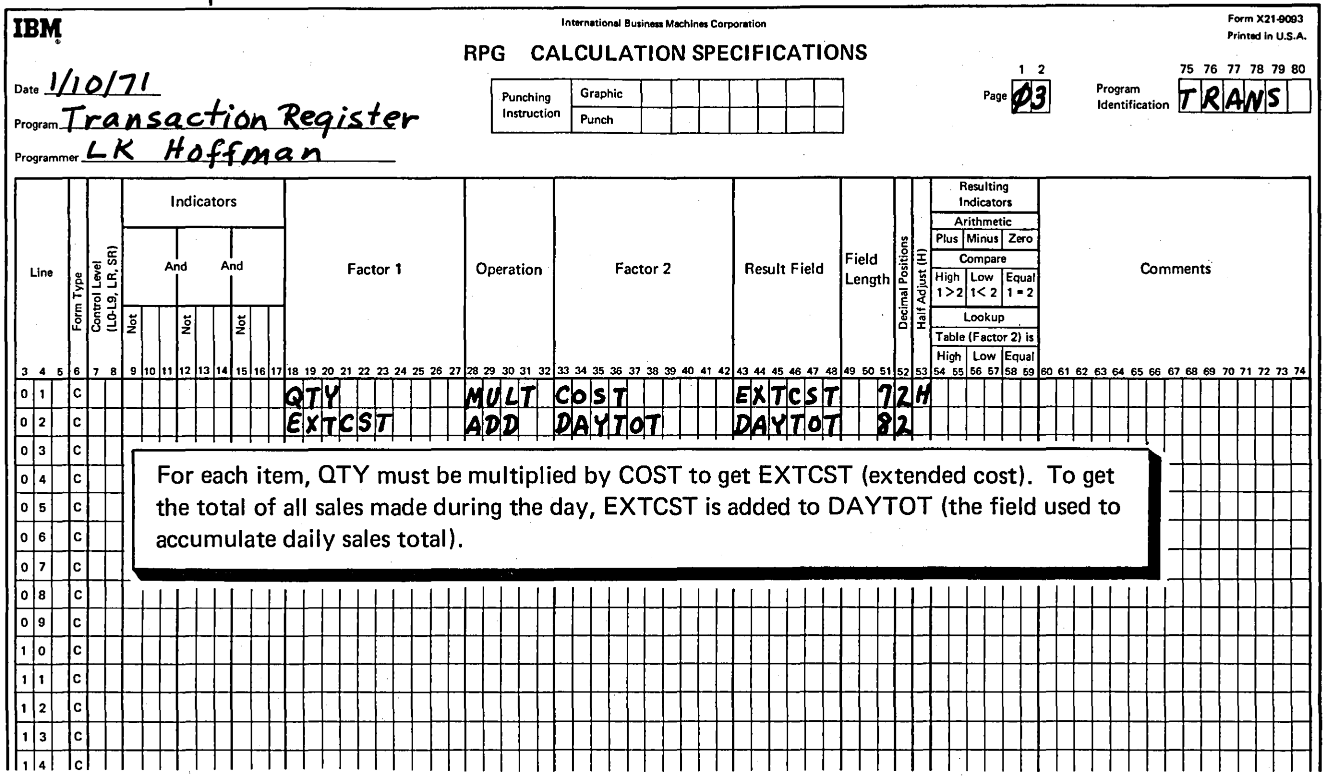
IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date 1/10/71
Program Transaction Register
Programmer LK Hoffmann
Page Ø2
Program Identification TRANS

Line	Form Type	Filename	From	To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	TRANS					
02	I		1	6	Ø	DATE	L1
03	I		7	12		ITEMNO	
04	I		13	32		DESC	
			33	37	Ø	QTY	
			38	42	2	COST	
13	I						
14	I						
15	I						

Input fields are described as before. In this job, totals must be accumulated and printed. You know that totals must be printed whenever a record with a different date field is read. The date field determines when total operations should be done. The date field is the control field and must be specified as such. This is done by the L1 entry in columns 59-60.



IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date 1/10/71

Program Transaction Register

Programmer LK Hoffman

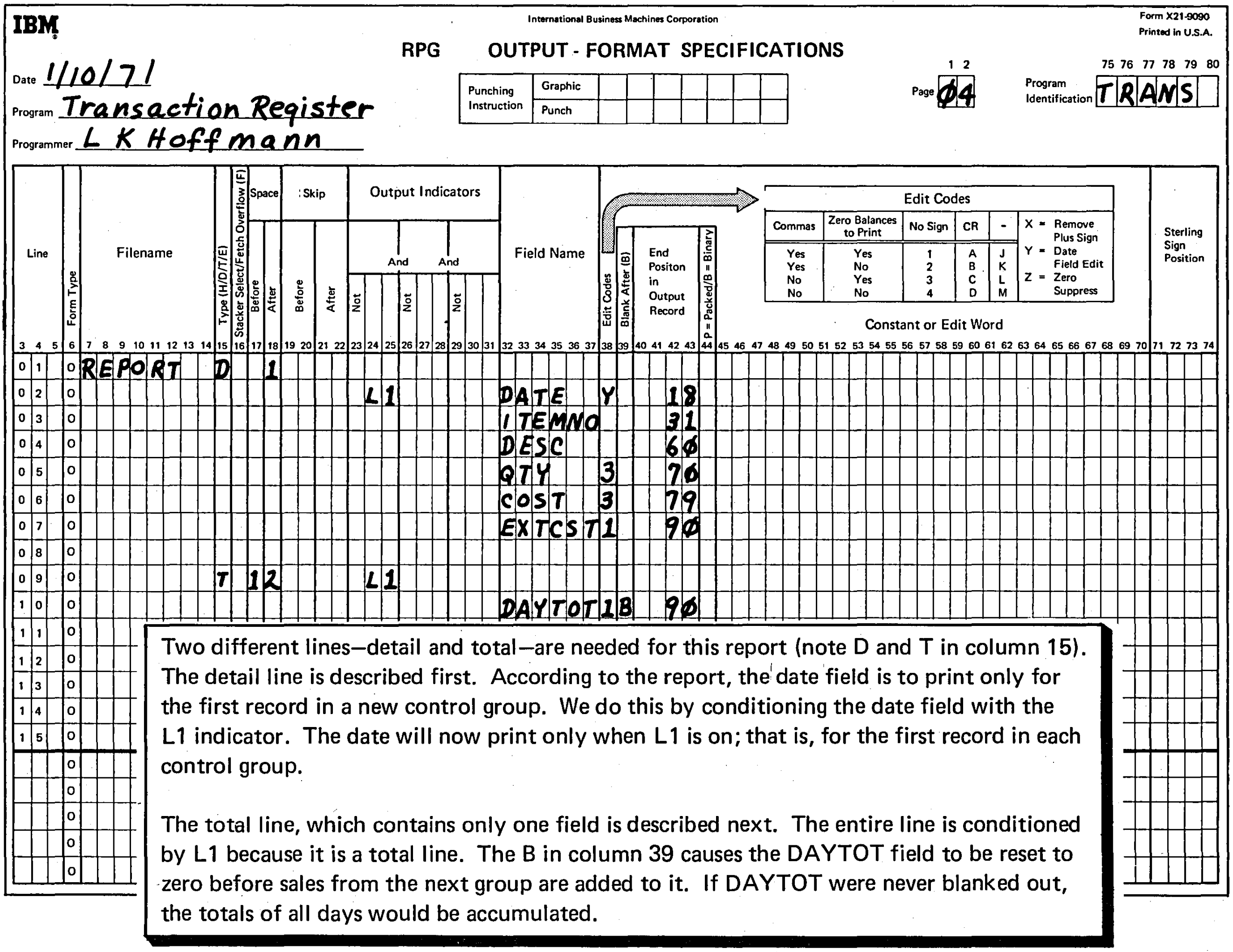
Punching Instruction: Graphic / Punch

Page ∅3

Program Identification TRANS

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators (Not / And Not / And Not)	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C			QTY	MULT	COST	EXTCST	7	2	H		
02	C			EXTCST	ADD	DAYTOT	DAYTOT	8	2			
03	C											
04	C											
05	C											
06	C											
07	C											
08	C											
09	C											
10	C											
11	C											
12	C											
13	C											
14	C											

For each item, QTY must be multiplied by COST to get EXTCST (extended cost). To get the total of all sales made during the day, EXTCST is added to DAYTOT (the field used to accumulate daily sales total).



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71

Program Transaction Register

Programmer L K Hoffmann

Punching Instruction: Graphic / Punch

Page ∅4

Program Identification TRANS

Edit Codes: Commas Yes, Zero Balances to Print Yes: 1 A J; Yes, No: 2 B K; No, Yes: 3 C L; No, No: 4 D M. X = Remove Plus Sign; Y = Date Field Edit; Z = Zero Suppress

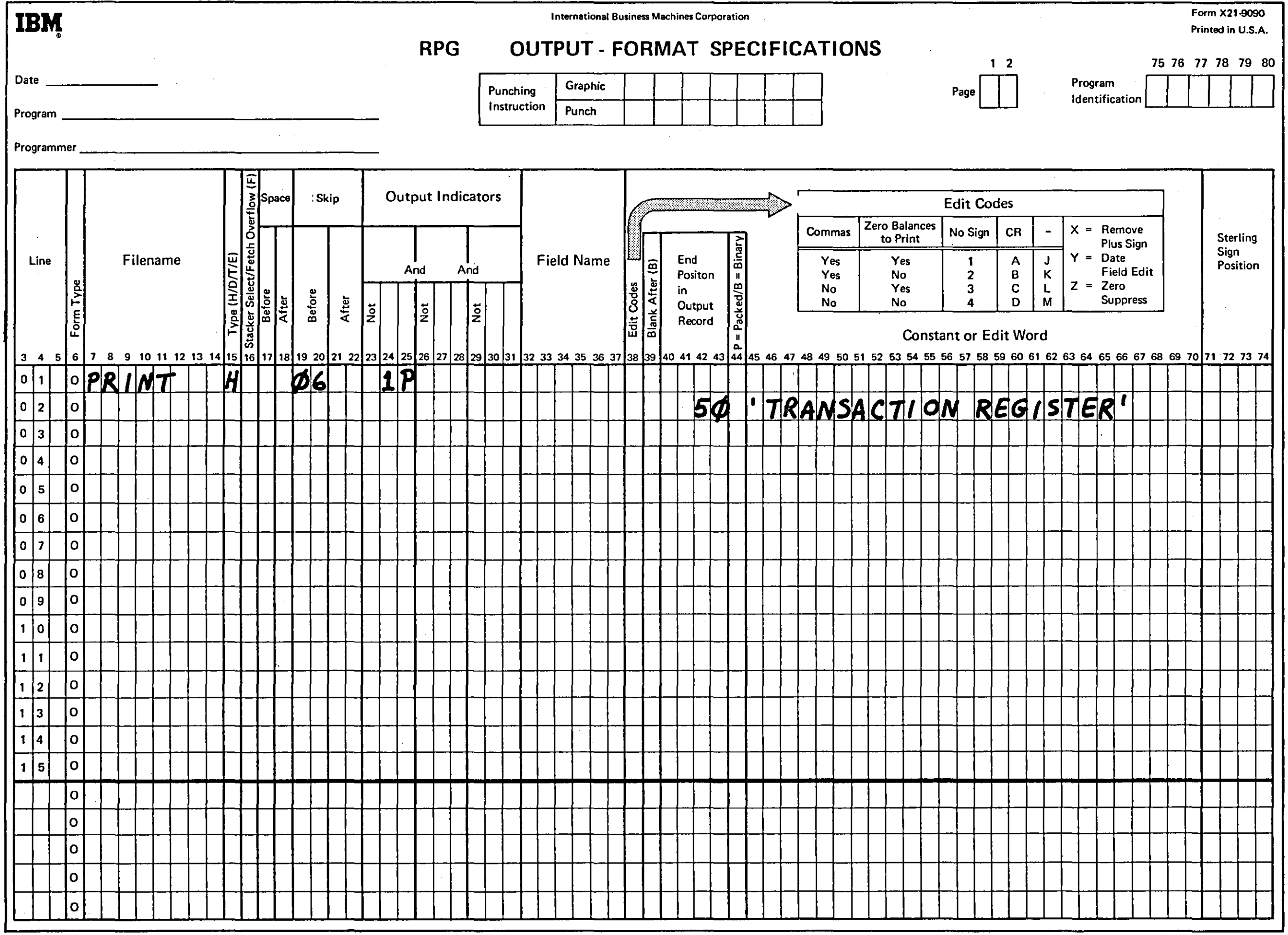
Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Packed/B = Binary	Constant or Edit Word	Sterling Sign Position
01	O	REPORT	D			1										
02	O								L1	DATE	Y		18			
03	O									ITEMNO			31			
04	O									DESC			6∅			
05	O									QTY	3		7∅			
06	O									COST	3		79			
07	O									EXTCST	1		9∅			
08	O															
09	O		T		1	2			L1							
10	O									DAYTOT	1	B	9∅			
11	O															
12	O															
13	O															
14	O															
15	O															
	O															
	O															
	O															
	O															
	O															

Two different lines—detail and total—are needed for this report (note D and T in column 15). The detail line is described first. According to the report, the date field is to print only for the first record in a new control group. We do this by conditioning the date field with the L1 indicator. The date will now print only when L1 is on; that is, for the first record in each control group.

The total line, which contains only one field is described next. The entire line is conditioned by L1 because it is a total line. The B in column 39 causes the DAYTOT field to be reset to zero before sales from the next group are added to it. If DAYTOT were never blanked out, the totals of all days would be accumulated.

FIRST PAGE INDICATOR

The first page (1P) indicator is used on the Output-Format sheet to specify the headings you want printed only on the first page of a report. Headings, usually printed at the top of the page, include such things as report titles or column names:



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date ______________

Program ______________

Programmer ______________

Punching Instruction: Graphic | Punch

Page (1 2)

Program Identification (75 76 77 78 79 80)

Edit Codes					
Commas	Zero Balances to Print	No Sign	CR	-	X = Remove Plus Sign Y = Date Field Edit Z = Zero Suppress
Yes	Yes	1	A	J	
Yes	No	2	B	K	
No	Yes	3	C	L	
No	No	4	D	M	

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators Not (23)	(24-25)	And Not (26)	(27-28)	And Not (29)	(30-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Positon in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
01	O	PRINT	H				Ø6			1P											
02	O																	5Ø		'TRANSACTION REGISTER'	
03	O																				
04	O																				
05	O																				
06	O																				
07	O																				
08	O																				
09	O																				
10	O																				
11	O																				
12	O																				
13	O																				
14	O																				
15	O																				
	O																				
	O																				
	O																				
	O																				
	O																				

Program Cycle Operations

One operation in the program cycle is concerned with the 1P indicator (see Figure 9). The 1P indicator is automatically set on at the beginning of every job, so the first operation taken by the computer is to print any output record conditioned by 1P. After this is done, the first record is read and the program cycle operations are executed in order.

Headings conditioned by 1P are printed only once–at the beginning of the job on the first page of the report. Any heading records that are not conditioned by 1P are handled in the same way as detail records. This means that they will be printed along with detail records in every cycle.

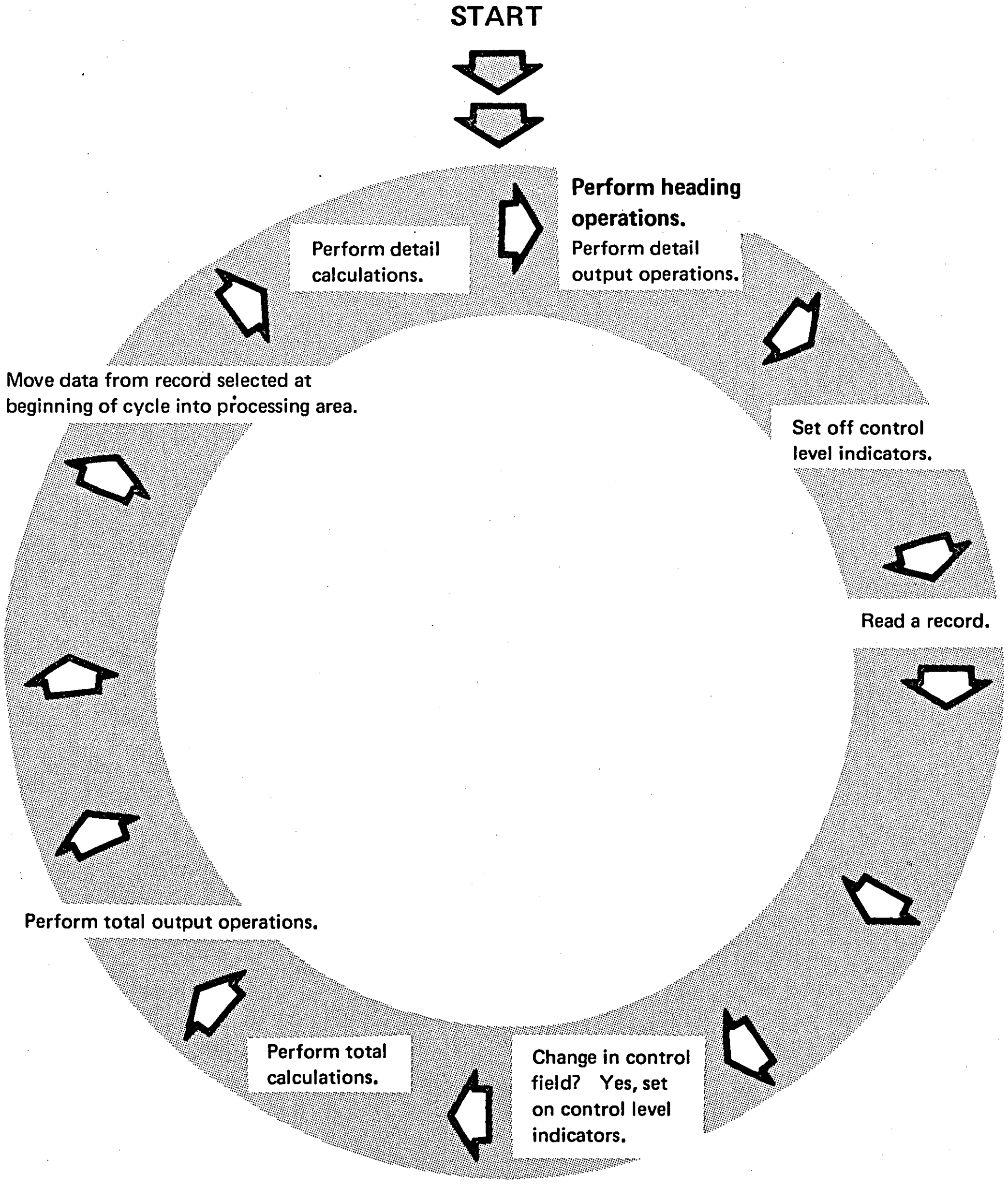
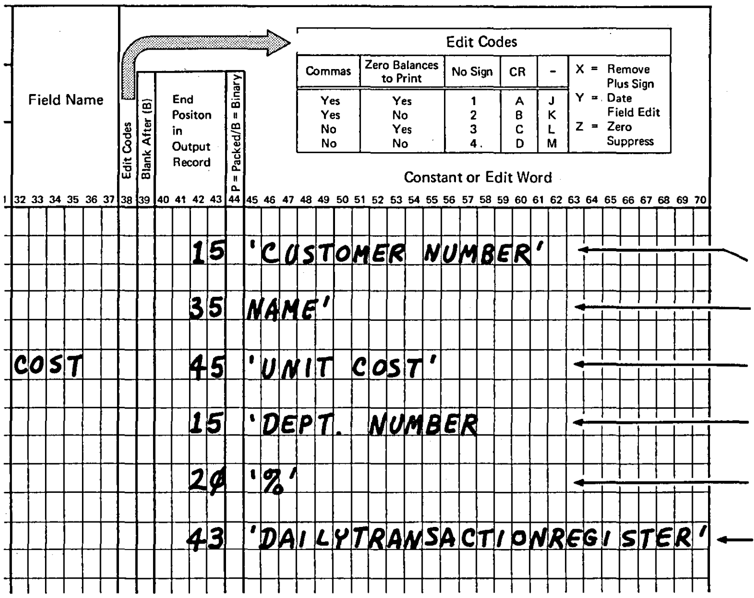

START
Perform heading operations.
Perform detail output operations.
Set off control level indicators.
Read a record.
Change in control field? Yes, set on control level indicators.
Perform total calculations.
Perform total output operations.
Move data from record selected at beginning of cycle into processing area.
Perform detail calculations.

Figure 9. The first operation in the first program cycle concerns output operations conditioned by the first page (1P) indicator.

Heading information to be printed on the first page of a report is specified by using constants (actual information instead of field names). Constants for headings must be specified according to these rules:

- Constants must be entered in columns 45-70 of the Output-Format sheet.
- Constants can contain either alphameric or numeric characters.
- Constants must be enclosed in single quotes. (The beginning quote is always entered in column 45.)
- No field name can be used on the same line as a constant.
- An end-position entry must be entered for every constant.

Which Constants Are Valid?



Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4.	D	M	

Field Name 32-37	Edit Codes 38	Blank After (B) 39	End Positon in Output Record 40-43	P = Packed/B = Binary 44	Constant or Edit Word 45-70
			15		'CUSTOMER NUMBER'
			35		NAME'
COST			45		'UNIT COST'
			15		'DEPT. NUMBER
			2Ø		'%'
			43		'DAILYTRANSACTIONREGISTER'

Valid constant.

Invalid constant. The constant must be preceded by a quote in column 45.

Invalid constant. A field name cannot be specified on the same line as a constant.

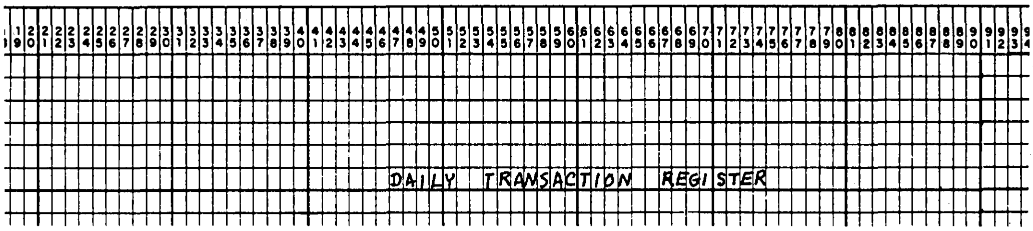
Invalid constant. The last character in the constant must be followed by a quote.

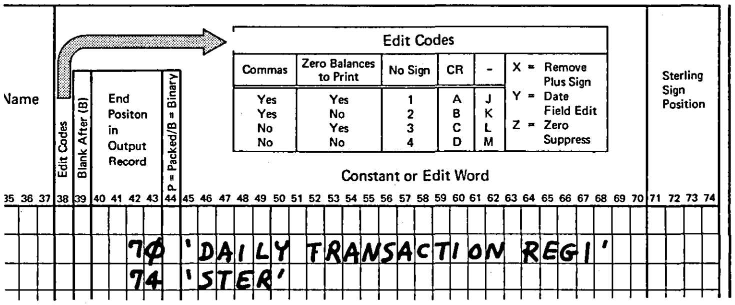
Valid constant.

Valid constant. Although the constant is specified correctly, it will be unreadable when printed because no spaces were left between words.

Headings too long to specify on one line of the Output-Format sheet can be split and placed on separate lines. You must, however, give an end position for each part.

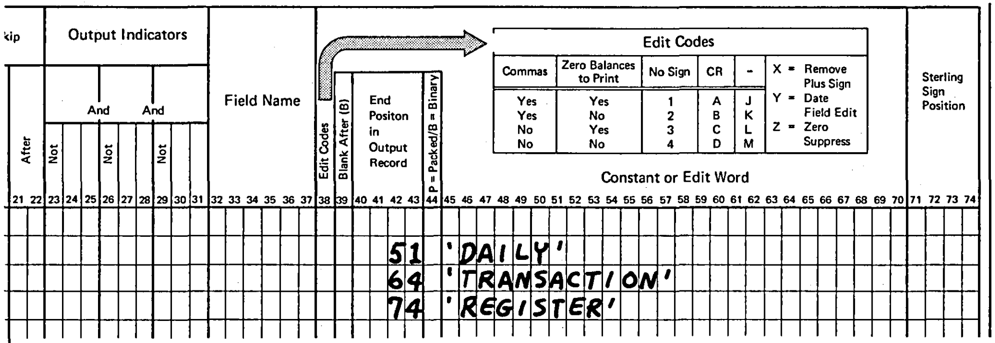
The heading shown in the Printer Spacing Chart takes 28 positions. A constant containing a maximum of 24 characters can be specified on one specification line. Since the entire heading cannot be specified on one line, it must be broken into parts. The examples given show three different ways to specify this heading:

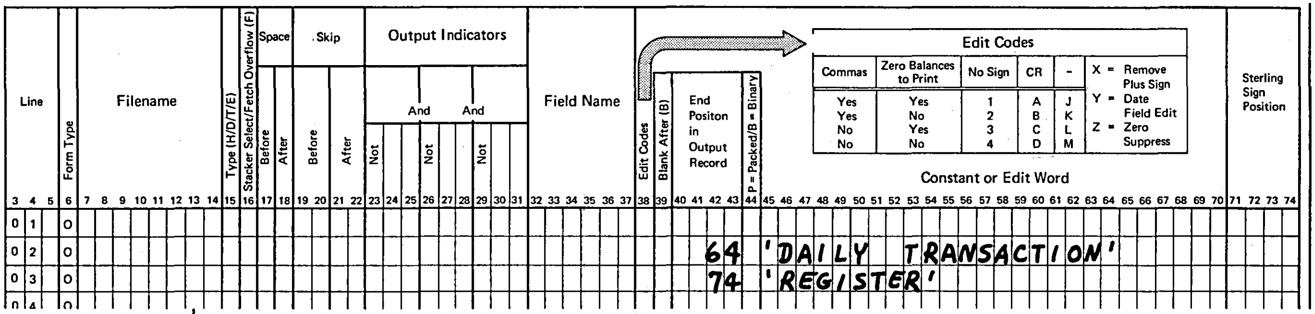

DAILY TRANSACTION REGISTER


Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

End Positon in Output Record (40-43)	Constant or Edit Word (45-70)
7Ø	'DAILY TRANSACTION REGI'
74	'STER'


End Positon in Output Record (40-43)	Constant or Edit Word (45-70)
51	'DAILY'
64	'TRANSACTION'
74	'REGISTER'


Line	Form Type	End Positon in Output Record (40-43)	Constant or Edit Word (45-70)
01	O		
02	O	64	'DAILY TRANSACTION'
03	O	74	'REGISTER'
04	O		

Heading lines should be specified first on the Output-Format sheet. The best method is to specify your record types in this order: heading, detail, and total.

OVERFLOW INDICATORS

You use overflow indicators to:

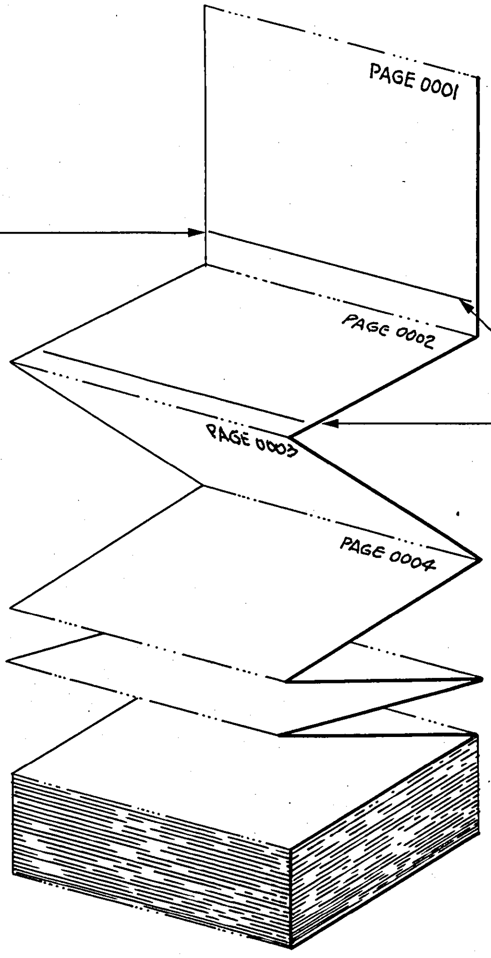
1. Print headings on every page but the first page of a report (the 1P indicator allows headings to be printed on the first page).

2. Control where printing begins and ends on a page.

3. Advance forms from one page to the next (provided a skip specification is also used).

To understand how overflow indicators work, you must know how the concept of overflow is defined in RPG II:

- Overflow–Lines that remain to be printed after a page is full.
- Overflow handling–Advancing forms to a new page after the last line has been printed on the current page.
- Overflow line–The last line to be printed on a page.
- Overflow page–The new page to be printed when overflow occurs.



The last line to be printed on a page is known as the *overflow line.*

Overflow occurs when the overflow line is printed.

Printers use continuous forms: a series of pages divided by perforations. Overflow handling refers to the means of advancing forms from one page to the next.

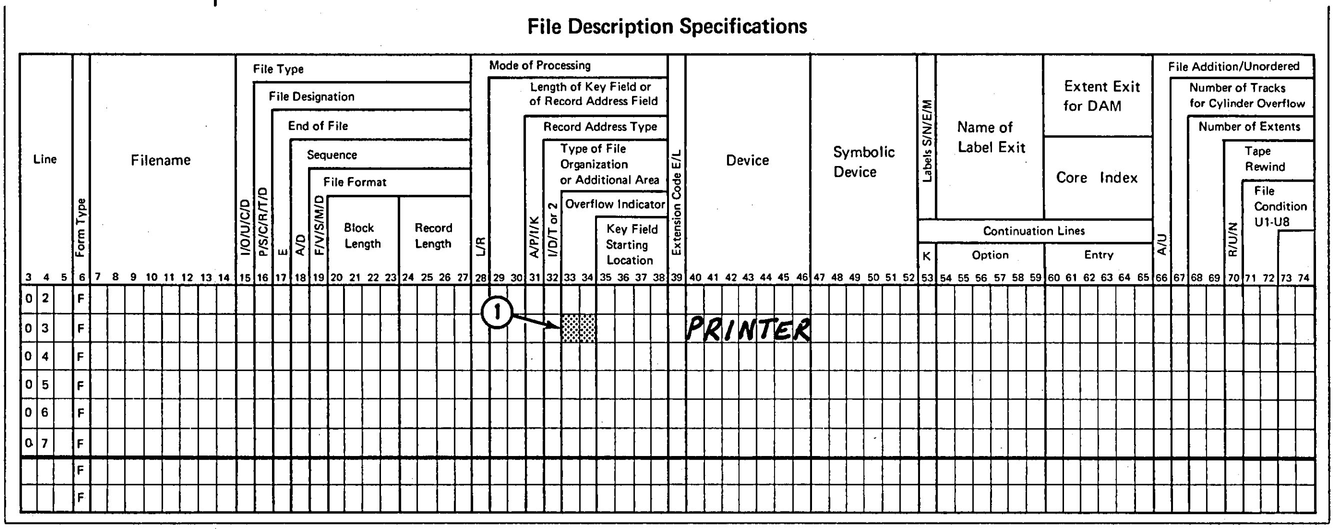
Overflow can be handled automatically by the system or through specifications you write.

Note: This discussion assumes that you are using standard, 66-line forms in the printer.

Printing always begins on line 06 (assuming the operator positions the first page at line 06) and ends with line 60. Overflow occurs after line 60 is printed; that is, forms advance to line 06 of a new page.

When overflow is handled automatically, heading lines can be printed on the first page if 1P is used. No instructions are provided, however, to print headings on overflow pages. Any heading lines print whenever and wherever detail lines print.

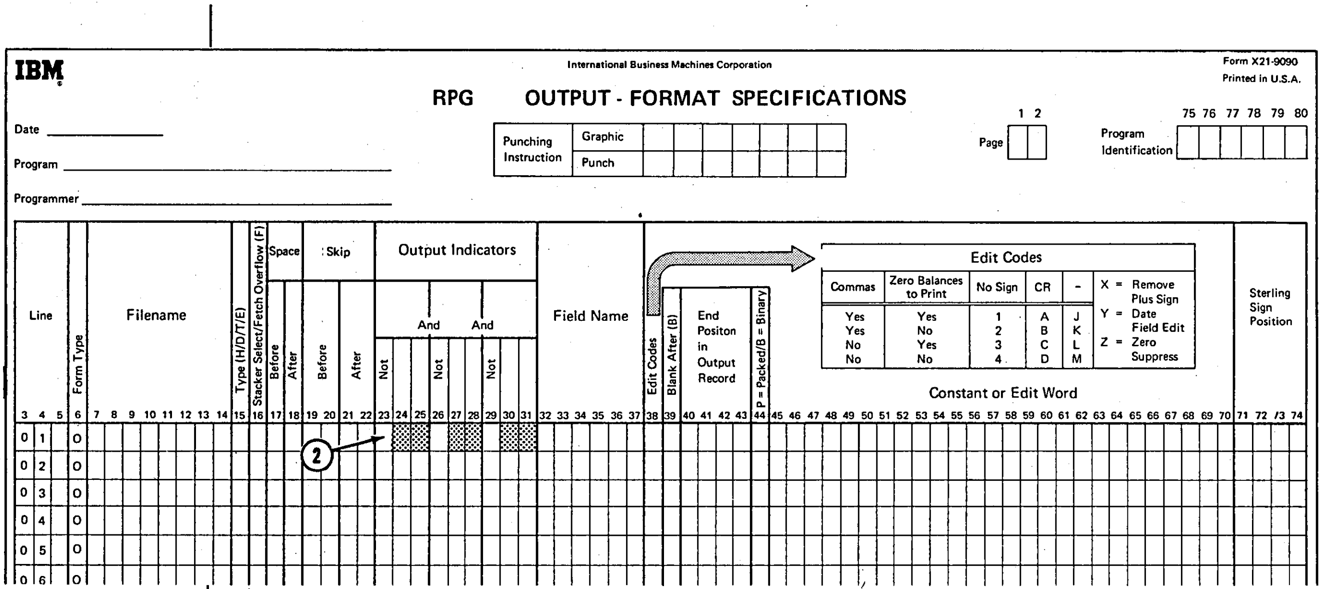
When you don't want overflow handled automatically, you can specify on coding sheets how you want it handled, using overflow indicators:



File Description Specifications

Line (3–5) | Form Type (6) | Filename (7–14) | File Type (15): I/O/U/C/D | File Designation (16): P/S/C/R/T/D | End of File (17): E | Sequence (18): A/D | File Format (19): F/V/S/M/D | Block Length (20–23) | Record Length (24–27) | Mode of Processing (28): L/R | Length of Key Field or of Record Address Field (29–30) | Record Address Type (31): A/P/I/K | Type of File Organization or Additional Area (32): I/D/T or 2 | Overflow Indicator (33–34) | Key Field Starting Location (35–38) | Extension Code E/L (39) | Device (40–46) | Symbolic Device (47–52) | Labels S/N/E/M (53) | Name of Label Exit (54–59) | Extent Exit for DAM | Core Index (60–65) | Continuation Lines: K, Option, Entry | A/U (66) | File Addition/Unordered | Number of Tracks for Cylinder Overflow (67–68) | Number of Extents (69) | Tape Rewind (70): R/U/N | File Condition U1-U8 (71–72) | (73–74)

Line	Form Type	Overflow Indicator	Device
0 2	F		
0 3	F	① → ▨	PRINTER
0 4	F		
0 5	F		
0 6	F		
0 7	F		
	F		
	F		



IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date ____________

Program ____________

Programmer ____________

Punching Instruction: Graphic / Punch

Page (1 2) Program Identification (75 76 77 78 79 80)

Line (3–5) | Form Type (6) | Filename (7–14) | Type (H/D/T/E) (15) | Stacker Select/Fetch Overflow (F) (16) | Space: Before (17), After (18) | Skip: Before (19–20), After (21–22) | Output Indicators: Not (23), (24–25), And Not (26), (27–28), And Not (29), (30–31) | Field Name (32–37) | Edit Codes (38) | Blank After (B) (39) | End Positon in Output Record (40–43) | P = Packed/B = Binary (44) | Constant or Edit Word (45–70) | Sterling Sign Position (71–74)

Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Output Indicators
0 1	O	▨ ▨ ▨
0 2	O	②
0 3	O	
0 4	O	
0 5	O	
0 6	O	

Assign an overflow indicator to the printer at ①, then use it on the Output-Format sheet at ② to show which operations must be done when overflow occurs.

Program Cycle Operations

Figure 10 shows operations in a program cycle in which overflow indicators are used.

The program sets on the overflow indicator you assigned whenever the overflow line is passed. By setting the overflow indicator on, the program remembers that overflow has occurred. As you can see in Figure 10, overflow indicators can be set on at one of two times: at detail time when a detail record prints on the overflow line or at total time when a total record prints on the overflow line. Notice that the only time a check is made to see if the overflow indicator is on is right after total output. If the overflow indicator is on, overflow operations are done in this order:

1. Print any total lines conditioned by the overflow indicator.
2. Skip to new page, provided a skip specification was made on a line conditioned by the overflow indicator.
3. Print all heading and detail lines conditioned by the overflow indicator.

If multiple detail lines are to be printed in a single cycle, printing may occur past the designated overflow line. This is because all detail printing for a single cycle is completed before overflow operations occur.

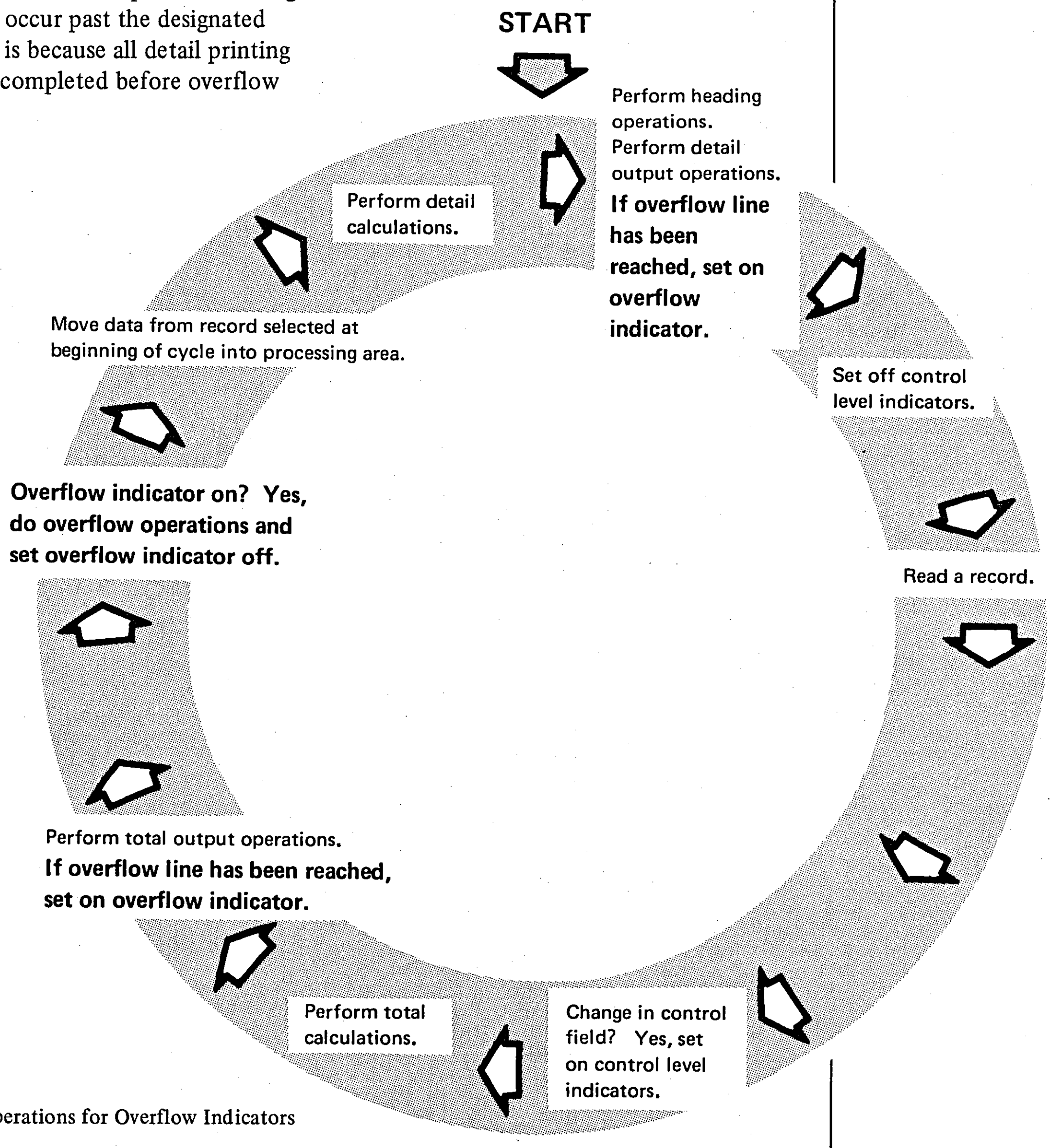

START
Perform heading operations. Perform detail output operations.
If overflow line has been reached, set on overflow indicator.
Set off control level indicators.
Read a record.
Change in control field? Yes, set on control level indicators.
Perform total calculations.
Perform total output operations.
If overflow line has been reached, set on overflow indicator.
Overflow indicator on? Yes, do overflow operations and set overflow indicator off.
Move data from record selected at beginning of cycle into processing area.
Perform detail calculations.

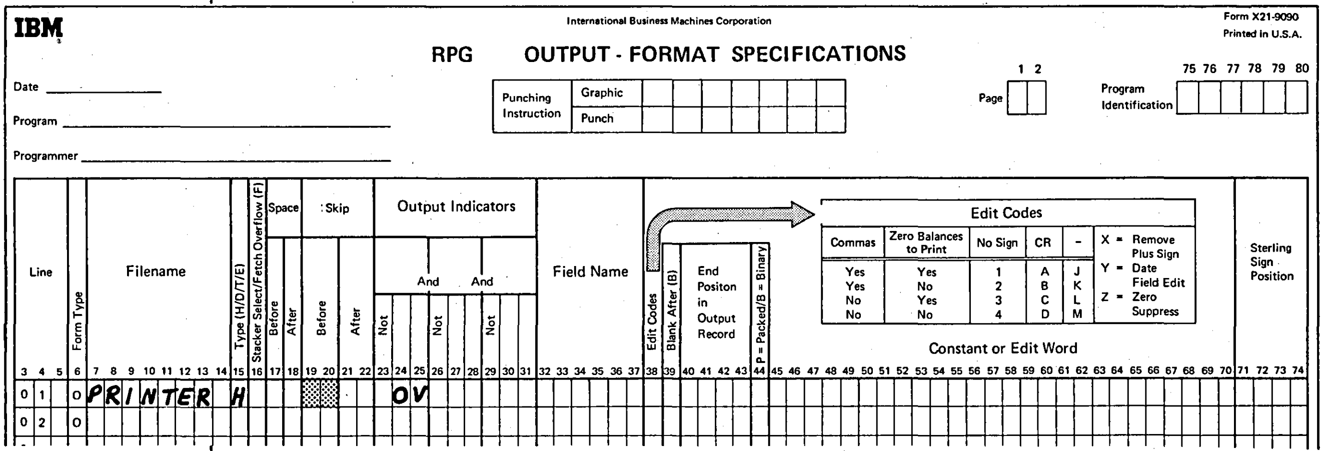
Figure 10. Program Cycle Operations for Overflow Indicators

RPG II Specifications

There are eight overflow indicators: OA through OG and OV. You can enter any one of these indicators in columns 33-34 on the File Description sheet. If you have more than one printer file, however, you must specify a different overflow indicator for each file.

After you have specified an overflow indicator on the File Description sheet, you must specify the *same* indicator on the Output-Format sheet. This specifies what you want done when overflow occurs.

Besides specifying the overflow indicator, you must also specify that forms should advance. You do this by placing a skip specification in columns 19-20 on the Output-Format sheet:



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date

Program

Programmer

Punching Instruction	Graphic							
	Punch							

1 2 Page

75 76 77 78 79 80 Program Identification

Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

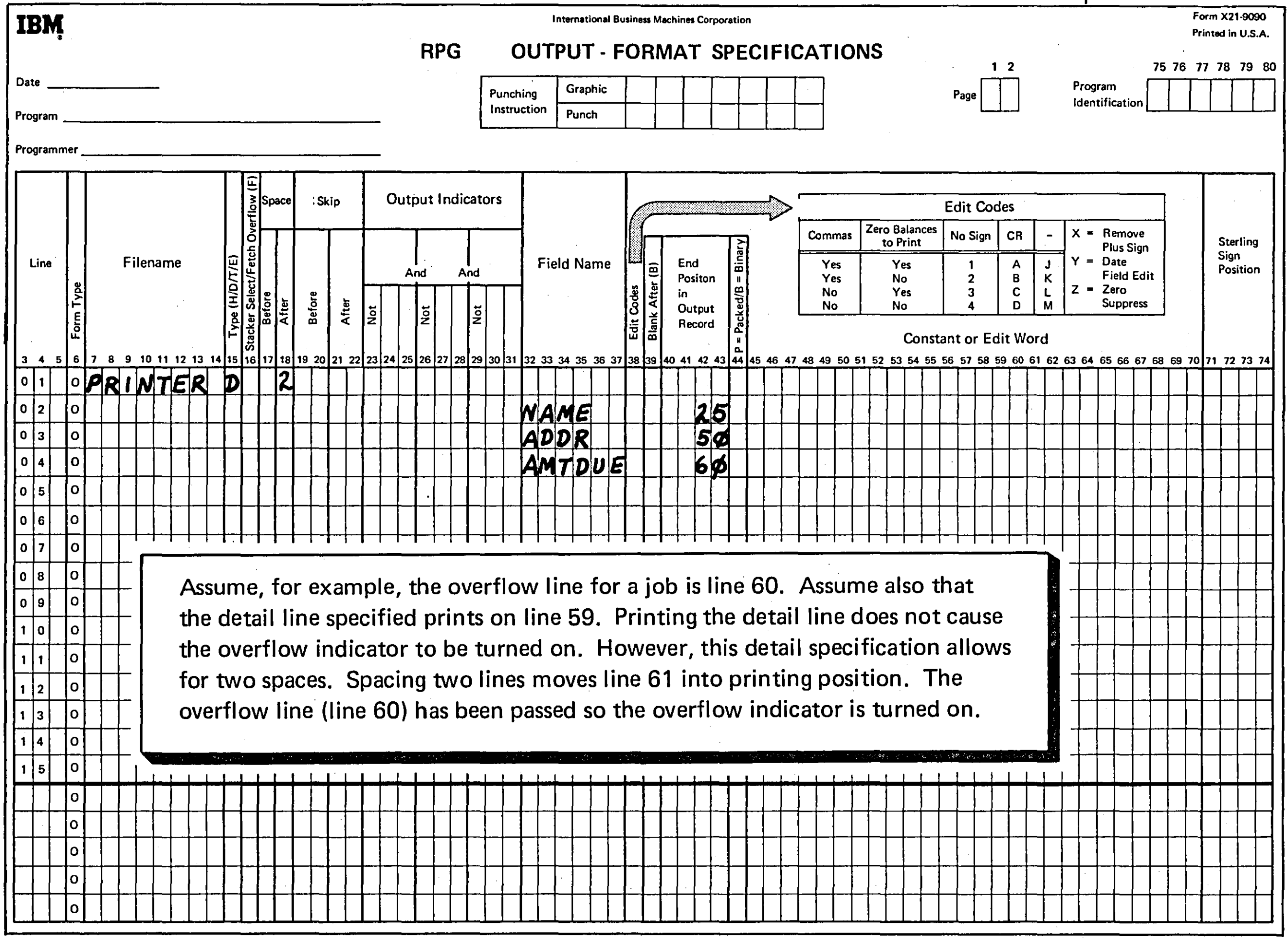
Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators: Not (23)	And (24-25)	Not (26)	And (27-28)	Not (29)	(30-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Positon in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
0 1	O	PRINTER	H				▨			OV											
0 2	O																				

The skip specification should be made on the first line you want printed on the page (usually a heading line). If your printer has a tapeless carriage, the Skip entry is the line number of the beginning line (usually 6). If your printer has a carriage control tape, the Skip entry should be the channel number in the tape that indicates the first printed line on a page.

Always remember to enter a skip specification for advancing forms in a heading line conditioned by the overflow indicator. If you forget, forms will not advance when overflow occurs.

Using Spacing with Overflow

You already know that the overflow indicator is turned on when a record is printed on the overflow line. However, this indicator also turns on whenever the overflow line passes under the printing mechanism. This means that spacing to a line past the overflow line to a line on the same page causes the overflow indicator to turn on. Figure 11 shows an example of spacing after printing.



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date ____________

Program ____________

Programmer ____________

Punching Instruction: Graphic / Punch

Page (1 2)

Program Identification (75 76 77 78 79 80)

Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
01	O	PRINTER	D			2										
02	O									NAME			25			
03	O									ADDR			5Ø			
04	O									AMTDUE			6Ø			
05	O															
06	O															
07	O															
08	O															
09	O															
10	O															
11	O															
12	O															
13	O															
14	O															
15	O															
	O															
	O															
	O															
	O															
	O															

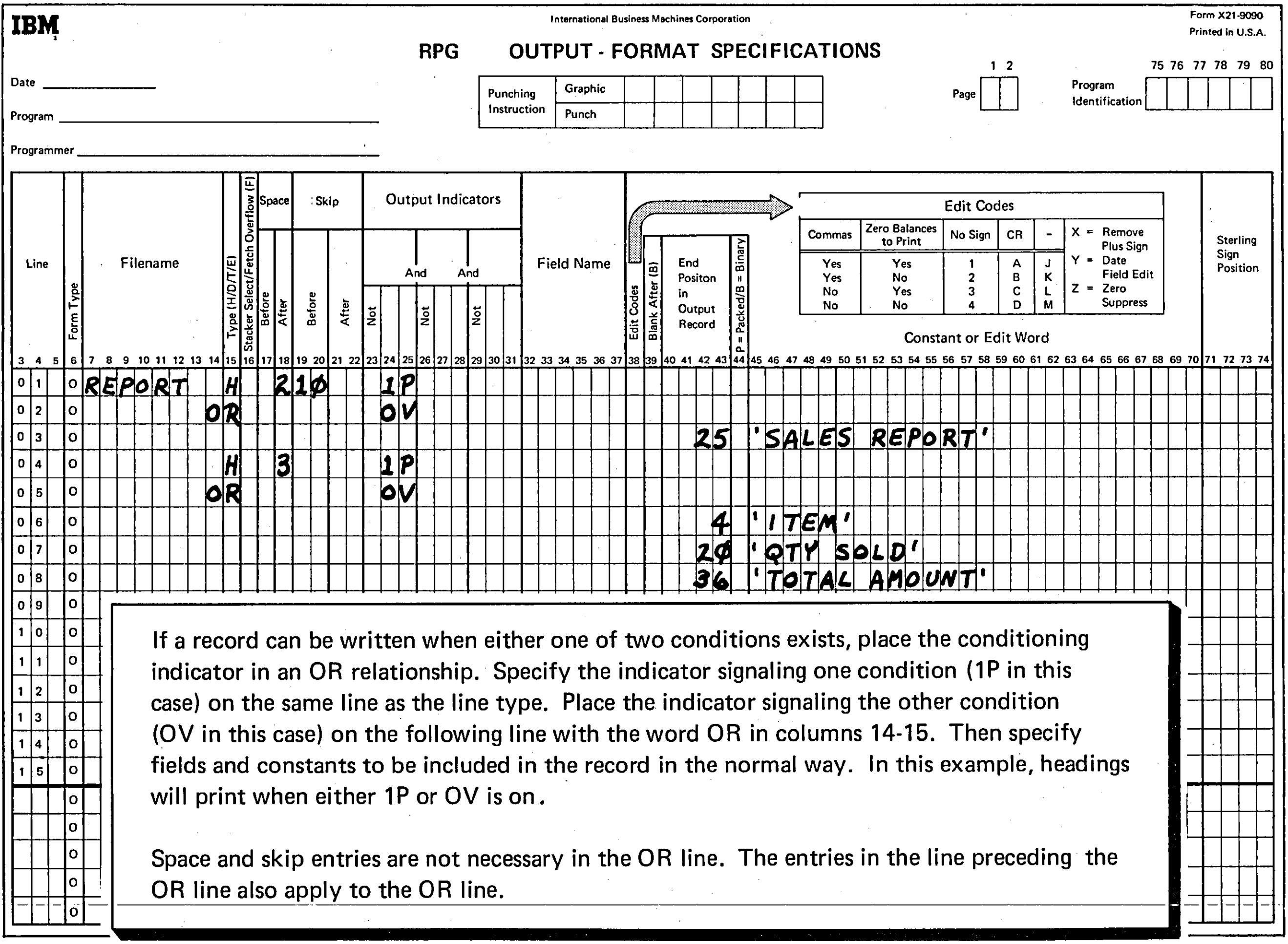
Assume, for example, the overflow line for a job is line 60. Assume also that the detail line specified prints on line 59. Printing the detail line does not cause the overflow indicator to be turned on. However, this detail specification allows for two spaces. Spacing two lines moves line 61 into printing position. The overflow line (line 60) has been passed so the overflow indicator is turned on.

Figure 11. Spacing after Printing

Using Overflow and 1P Indicators Together

The overflow indicator is most often used with other indicators. However, we will discuss the use of overflow indicators only with the 1P indicator. Both 1P and OV cannot be on at the same time.

If you want headings on all pages of a report, you have to use both the 1P indicator and an overflow indicator. 1P causes headings to print on the first page; the overflow indicator causes them to print on all succeeding pages. If a record should be printed when either one or another condition occurs (either 1P or OV is on), you can specify indicators in an *OR relationship:*



IBM — International Business Machines Corporation — Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

```
0 1 O REPORT   H  21Ø  1P
0 2 O       OR         OV
0 3 O                             25 'SALES REPORT'
0 4 O          H  3    1P
0 5 O       OR         OV
0 6 O                              4 'ITEM'
0 7 O                             2Ø 'QTY SOLD'
0 8 O                             36 'TOTAL AMOUNT'
```

If a record can be written when either one of two conditions exists, place the conditioning indicator in an OR relationship. Specify the indicator signaling one condition (1P in this case) on the same line as the line type. Place the indicator signaling the other condition (OV in this case) on the following line with the word OR in columns 14-15. Then specify fields and constants to be included in the record in the normal way. In this example, headings will print when either 1P or OV is on.

Space and skip entries are not necessary in the OR line. The entries in the line preceding the OR line also apply to the OR line.

LAST RECORD (LR) INDICATOR

The last record (LR) indicator is associated with end-of-job procedures. The program uses LR to indicate that the last data record has been read and that end-of-job processing is to take place.

Use of the last record (LR) indicator is optional. When LR is not used in specifications, the compiler automatically supplies end-of-job instructions. If you use LR, you are indicating that certain operations, such as printing a total count of all records read, must be done after all input records are processed.

Program Cycle Operations

Figure 12 shows the operations in the program cycle associated with the last record indicator. RPG II is set up so that it uses an end-of-file record containing some identifying information to indicate end of the data file. For example, card devices use a card with /* (slash, asterisk) in columns 1 and 2 to indicate end of file.

Whenever a record is read, the program checks to see if the record is the end-of-file record. If it is, the program sets on all control level indicators L1-L9. It also sets on the LR indicator to indicate that all records have been processed. All total operations (those conditioned by LR and L1-L9) are performed. After total operations have been done, the program checks to see if LR is on. If it is, processing stops.

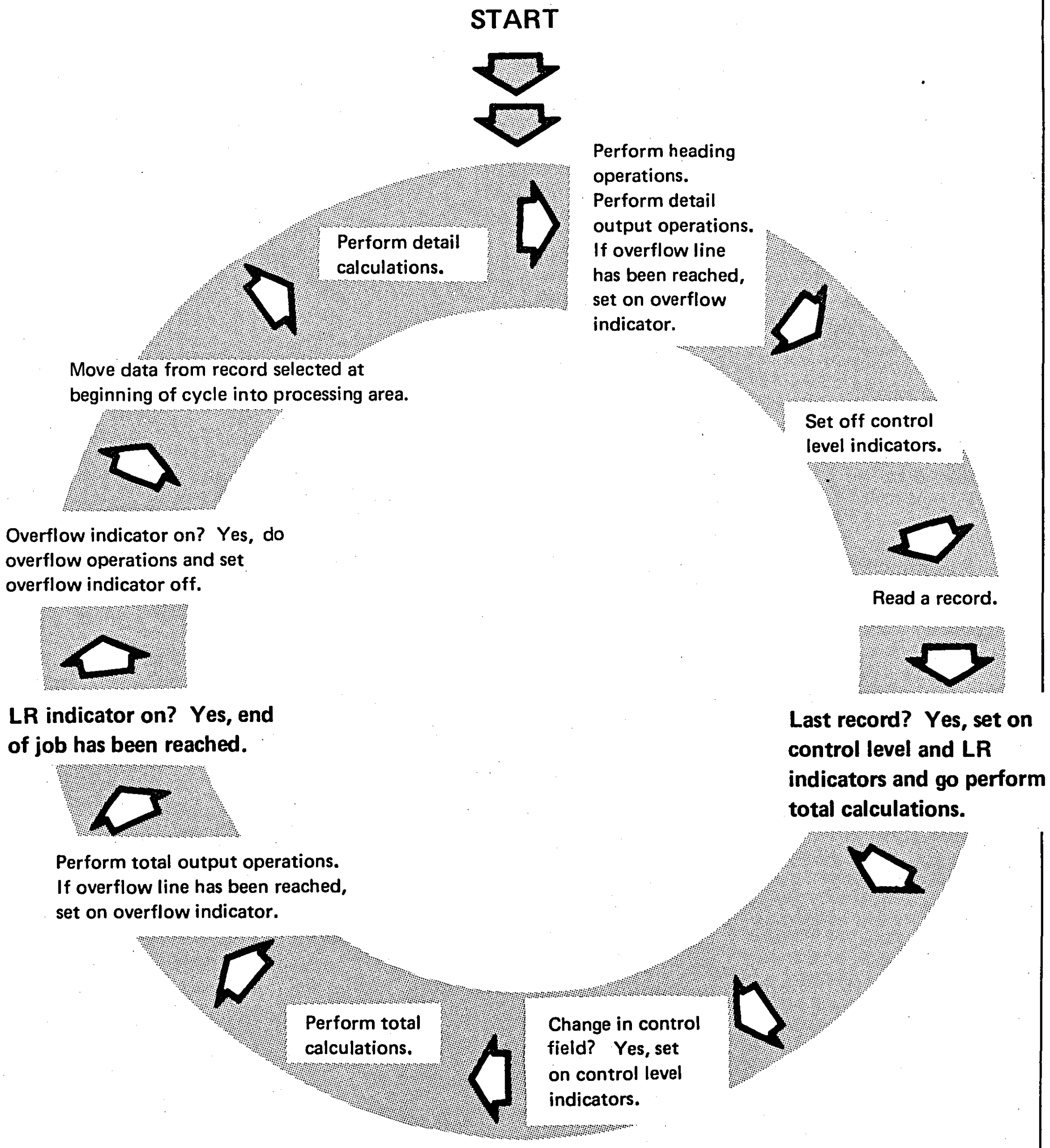
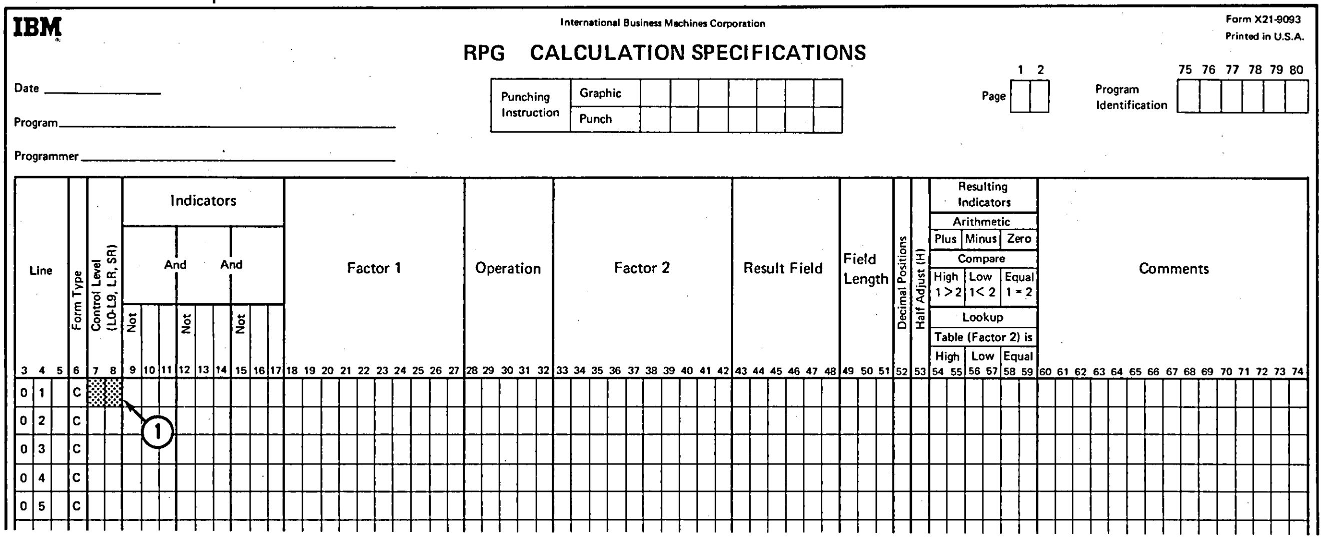

START
Perform heading operations. Perform detail output operations. If overflow line has been reached, set on overflow indicator.
Set off control level indicators.
Read a record.
Last record? Yes, set on control level and LR indicators and go perform total calculations.
Change in control field? Yes, set on control level indicators.
Perform total calculations.
Perform total output operations. If overflow line has been reached, set on overflow indicator.
LR indicator on? Yes, end of job has been reached.
Overflow indicator on? Yes, do overflow operations and set overflow indicator off.
Move data from record selected at beginning of cycle into processing area.
Perform detail calculations.

Figure 12. Program Cycle Operations for Last Record (LR) Indicator

The LR indicator is specified by an LR on the Calculation sheet or Output-Format sheet. This entry specifies which operations are to be done after the last record is processed:



IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date

Program

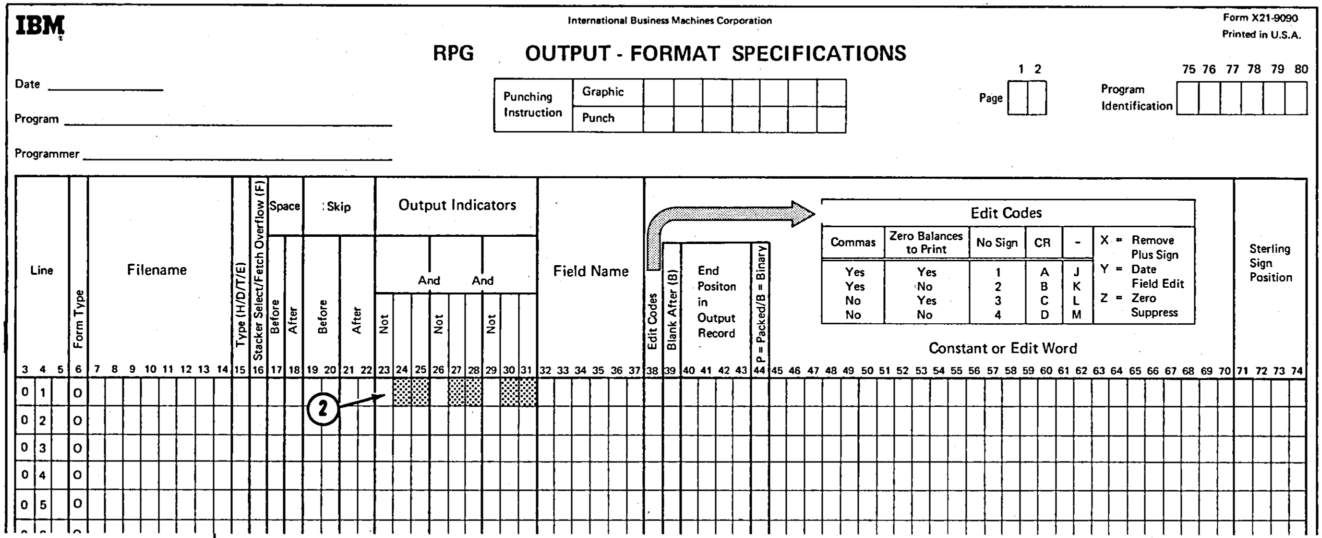
Programmer

Punching Instruction: Graphic, Punch

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3-5)	Form Type (6)	Control Level (L0-L9, LR, SR) (7-8)	Indicators: Not, And Not, And Not (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-48)	Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Arithmetic Plus Minus Zero; Compare High 1>2 Low 1<2 Equal 1=2; Lookup Table (Factor 2) is High Low Equal (54-59)	Comments (60-74)
0 1	C	▨										
0 2	C											
0 3	C											
0 4	C											
0 5	C											



IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date

Program

Programmer

Punching Instruction: Graphic, Punch

Page 1 2

Program Identification 75 76 77 78 79 80

Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators: Not, And Not, And Not (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Positon in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
0 1	O								▨ ▨ ▨							
0 2	O															
0 3	O															
0 4	O															
0 5	O															

The LR indicator is specified at ① to tell the computer which calculations are to be done after the last record is processed. The LR indicator is specified at ② to tell the computer which output operations are to be done after the last record is processed.

Job 4: Using First Page, Overflow, And Last Record Indicator To Print Headings And Totals

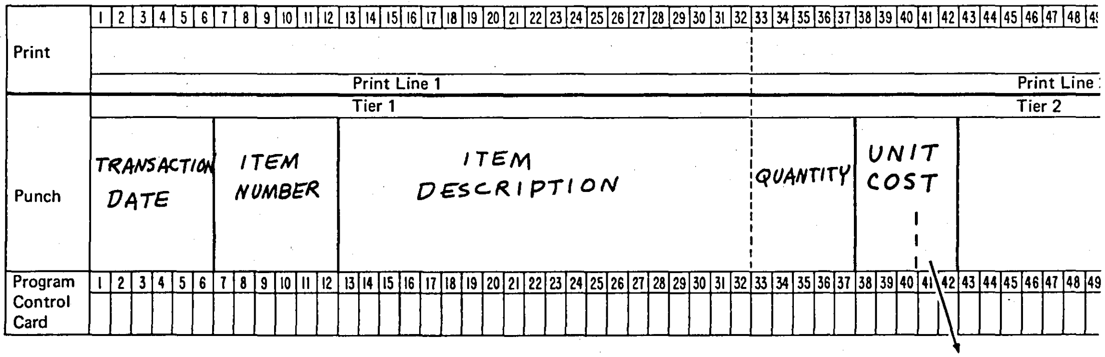
JOB DEFINITION

Print a weekly sales transaction report that lists daily transactions, total sales for the day, and total sales for the week. This report is similar to the one created in Job 3. The only difference is the addition of headings and final total.

The report title and column headings are printed on every page of the report. All items sold each day are listed. Item number, item description, quantity sold, unit cost, and extended cost are included for every item. The date is printed for the first transaction in each group. After all transactions for a day are listed, the daily sales amount is printed. A final total of all daily sales is printed at the end of the report.

JOB REQUIREMENTS

Input: Sales transaction file consisting of 96-column cards. Cards are arranged in ascending order by date. The format of the input records is shown on this Record Layout Form:



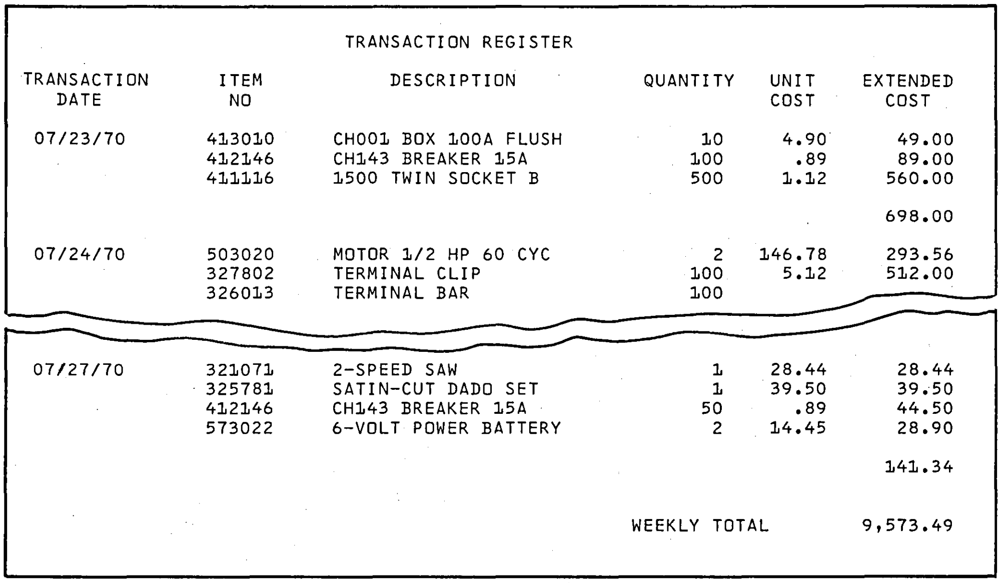
	1 2 3 4 5 6	7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32	33 34 35 36 37	38 39 40 41 42	43 44 45 46 47 48 49
Print			Print Line 1			Print Line
			Tier 1			Tier 2
Punch	TRANSACTION DATE	ITEM NUMBER	ITEM DESCRIPTION	QUANTITY	UNIT COST	
Program Control Card	1 2 3 4 5 6	7 8 9 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

two decimal positions

Processing:

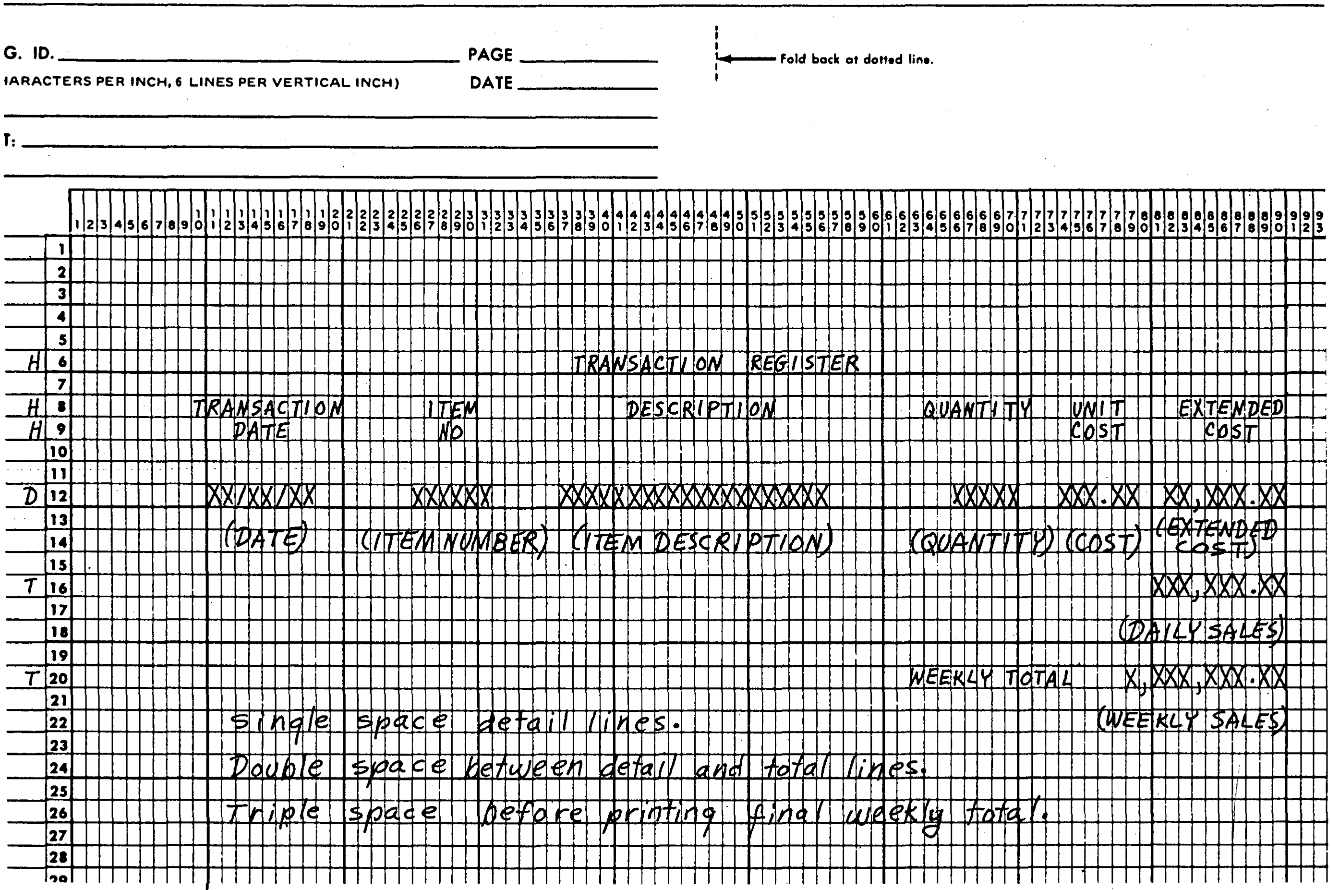
- Multiply quantity times unit cost to find extended cost.
- Accumulate extended cost to find total item sales per day.
- Accumulate total daily sales to find total weekly sales.

Output: A Transaction Register printed on a 96-position printer:



```
                               TRANSACTION REGISTER
TRANSACTION         ITEM              DESCRIPTION               QUANTITY   UNIT     EXTENDED
   DATE              NO                                                    COST       COST

 07/23/70          413010      CH001 BOX 100A FLUSH                   10     4.90       49.00
                   412146      CH143 BREAKER 15A                     100      .89       89.00
                   411116      1500 TWIN SOCKET B                    500     1.12      560.00

                                                                                       698.00

 07/24/70          503020      MOTOR 1/2 HP 60 CYC                     2   146.78      293.56
                   327802      TERMINAL CLIP                         100     5.12      512.00
                   326013      TERMINAL BAR                          100

 07/27/70          321071      2-SPEED SAW                             1    28.44       28.44
                   325781      SATIN-CUT DADO SET                      1    39.50       39.50
                   412146      CH143 BREAKER 15A                      50      .89       44.50
                   573022      6-VOLT POWER BATTERY                    2    14.45       28.90

                                                                                       141.34

                                                     WEEKLY TOTAL            9,573.49
```

This Printer Spacing Chart shows how the report is formatted:



G. ID.________________________ PAGE __________

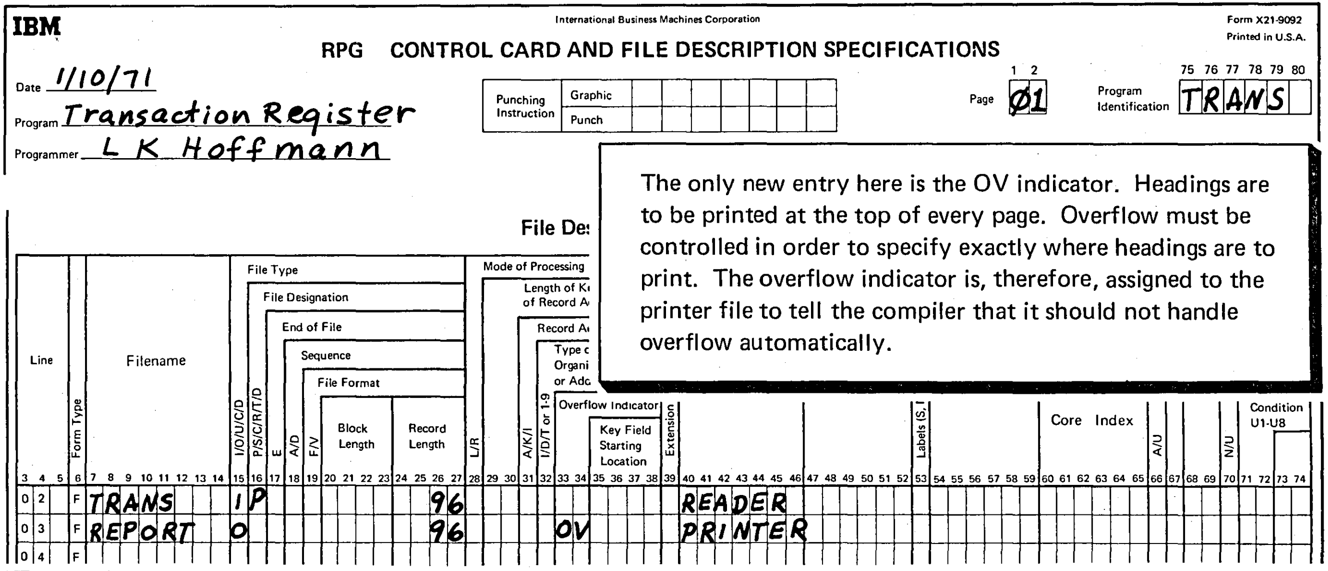
(ARACTERS PER INCH, 6 LINES PER VERTICAL INCH) DATE __________

Fold back at dotted line.

T: ________________________

```
H  6                         TRANSACTION REGISTER
H  8   TRANSACTION       ITEM          DESCRIPTION             QUANTITY   UNIT       EXTENDED
H  9      DATE            NO                                               COST         COST
D 12    XX/XX/XX        XXXXXX    XXXXXXXXXXXXXXXXXXXX          XXXXX    XXX.XX    XX,XXX.XX
  14     (DATE)     (ITEM NUMBER) (ITEM DESCRIPTION)          (QUANTITY) (COST)    (EXTENDED
                                                                                      COST)
T 16                                                                               XXX,XXX.XX
  18                                                                            (DAILY SALES)
T 20                                                      WEEKLY TOTAL       X,XXX,XXX.XX
  22     Single space detail lines.                                           (WEEKLY SALES)
  24     Double space between detail and total lines.
  26     Triple space before printing final weekly total.
```

JOB SPECIFICATIONS



IBM International Business Machines Corporation Form X21-9092 Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71

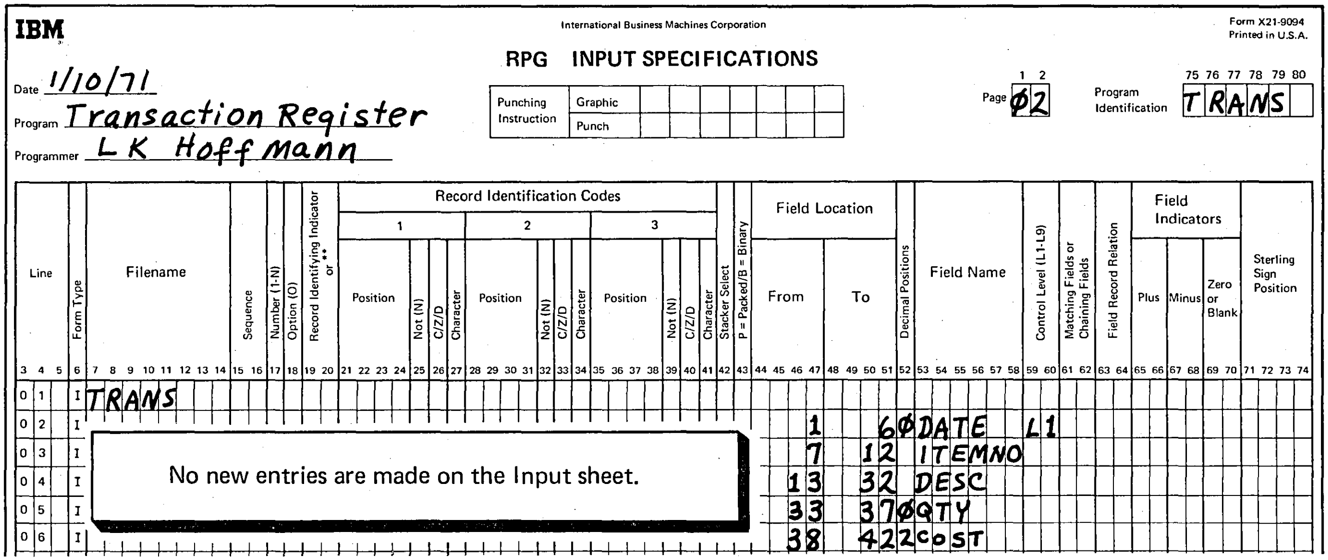
Program Transaction Register

Programmer L K Hoffmann

Page Ø1 Program Identification TRANS

Line	Form Type	Filename	File Type	File Designation	Record Length	Overflow Indicator	Device
02	F	TRANS	I	P	96		READER
03	F	REPORT	O		96	OV	PRINTER
04	F						

The only new entry here is the OV indicator. Headings are to be printed at the top of every page. Overflow must be controlled in order to specify exactly where headings are to print. The overflow indicator is, therefore, assigned to the printer file to tell the compiler that it should not handle overflow automatically.



IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date 1/10/71

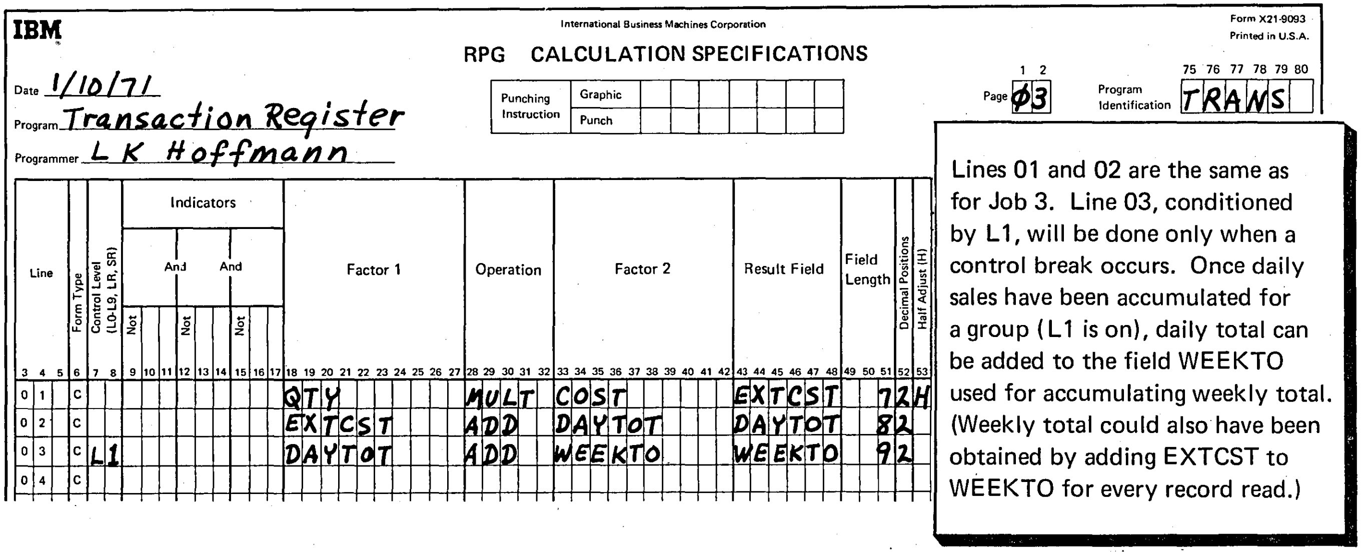
Program Transaction Register

Programmer L K Hoffmann

Page Ø2 Program Identification TRANS

Line	Form Type	Filename	Field Location From	To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	TRANS					
02	I		1	6	Ø	DATE	L1
03	I		7	12		ITEMNO	
04	I		13	32		DESC	
05	I		33	37	Ø	QTY	
06	I		38	42	2	COST	

No new entries are made on the Input sheet.



IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date 1/10/71

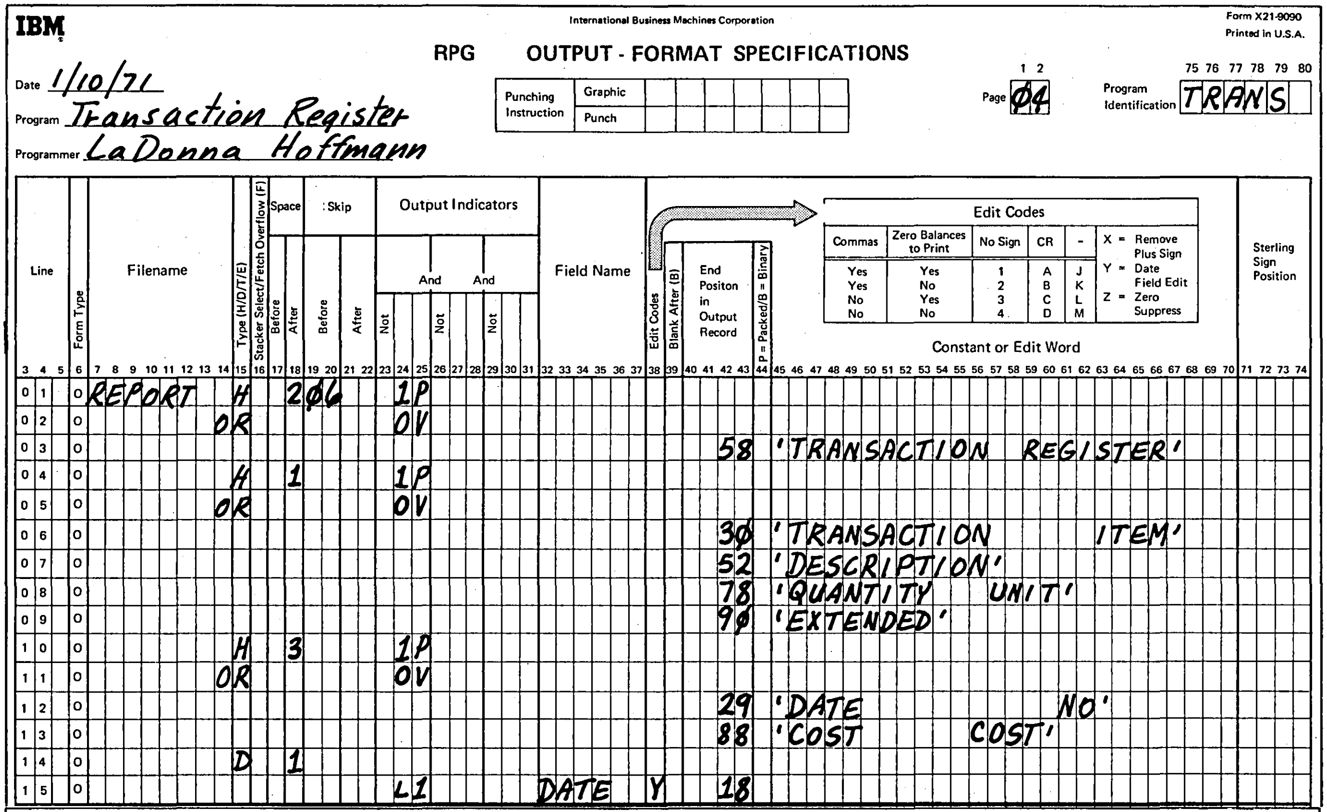
Program Transaction Register

Programmer L K Hoffmann

Page Ø3 Program Identification TRANS

Line	Form Type	Control Level	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)
01	C		QTY	MULT	COST	EXTCST	7	2	H
02	C		EXTCST	ADD	DAYTOT	DAYTOT	8	2	
03	C	L1	DAYTOT	ADD	WEEKTO	WEEKTO	9	2	
04	C								

Lines 01 and 02 are the same as for Job 3. Line 03, conditioned by L1, will be done only when a control break occurs. Once daily sales have been accumulated for a group (L1 is on), daily total can be added to the field WEEKTO used for accumulating weekly total. (Weekly total could also have been obtained by adding EXTCST to WEEKTO for every record read.)



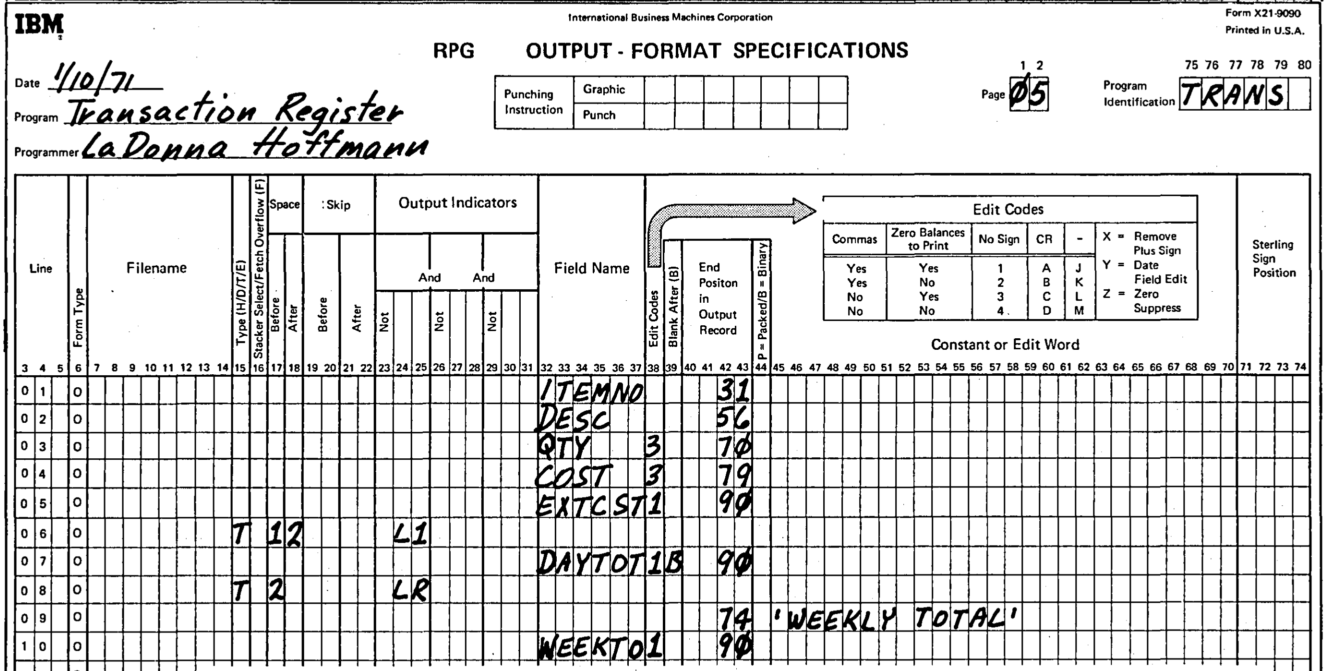
IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71
Program Transaction Register
Programmer LaDonna Hoffmann

Page Ø4 Program Identification TRANS

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After (B)	End Position in Output Record	Constant or Edit Word
01	O	REPORT	H		2	Ø6		1P					
02	O		OR					OV					
03	O											58	'TRANSACTION REGISTER'
04	O		H		1			1P					
05	O		OR					OV					
06	O											3Ø	'TRANSACTION ITEM'
07	O											52	'DESCRIPTION'
08	O											78	'QUANTITY UNIT'
09	O											9Ø	'EXTENDED'
10	O		H		3			1P					
11	O		OR					OV					
12	O											29	'DATE NO'
13	O											88	'COST COST'
14	O		D		1								
15	O							L1	DATE	Y		18	



IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71
Program Transaction Register
Programmer LaDonna Hoffmann

Page Ø5 Program Identification TRANS

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After (B)	End Position in Output Record	Constant or Edit Word
01	O								ITEMNO			31	
02	O								DESC			56	
03	O								QTY	3		7Ø	
04	O								COST	3		79	
05	O								EXTCST	1		9Ø	
06	O		T	1	2			L1					
07	O								DAYTOT	1	B	9Ø	
08	O		T	2				LR					
09	O											74	'WEEKLY TOTAL'
10	O								WEEKTO	1		9Ø	
11	O												
12	O												
13	O												
14	O												
15	O												

Six different lines are to be printed so six are described. The three heading lines should be printed either when 1P is on (first page) or when OV is on (overflow has occurred and forms have advanced). Thus, each heading is conditioned by 1P and OV used in an OR relationship. Notice the first heading line has the skip specification. This must be made before forms will advance.

The weekly total is printed only once on the report, after all records have been processed. It is conditioned by LR. A constant is specified for this total record so the words WEEKLY TOTAL will print as indicated on the Printer Spacing Chart.

RECORD IDENTIFYING INDICATORS

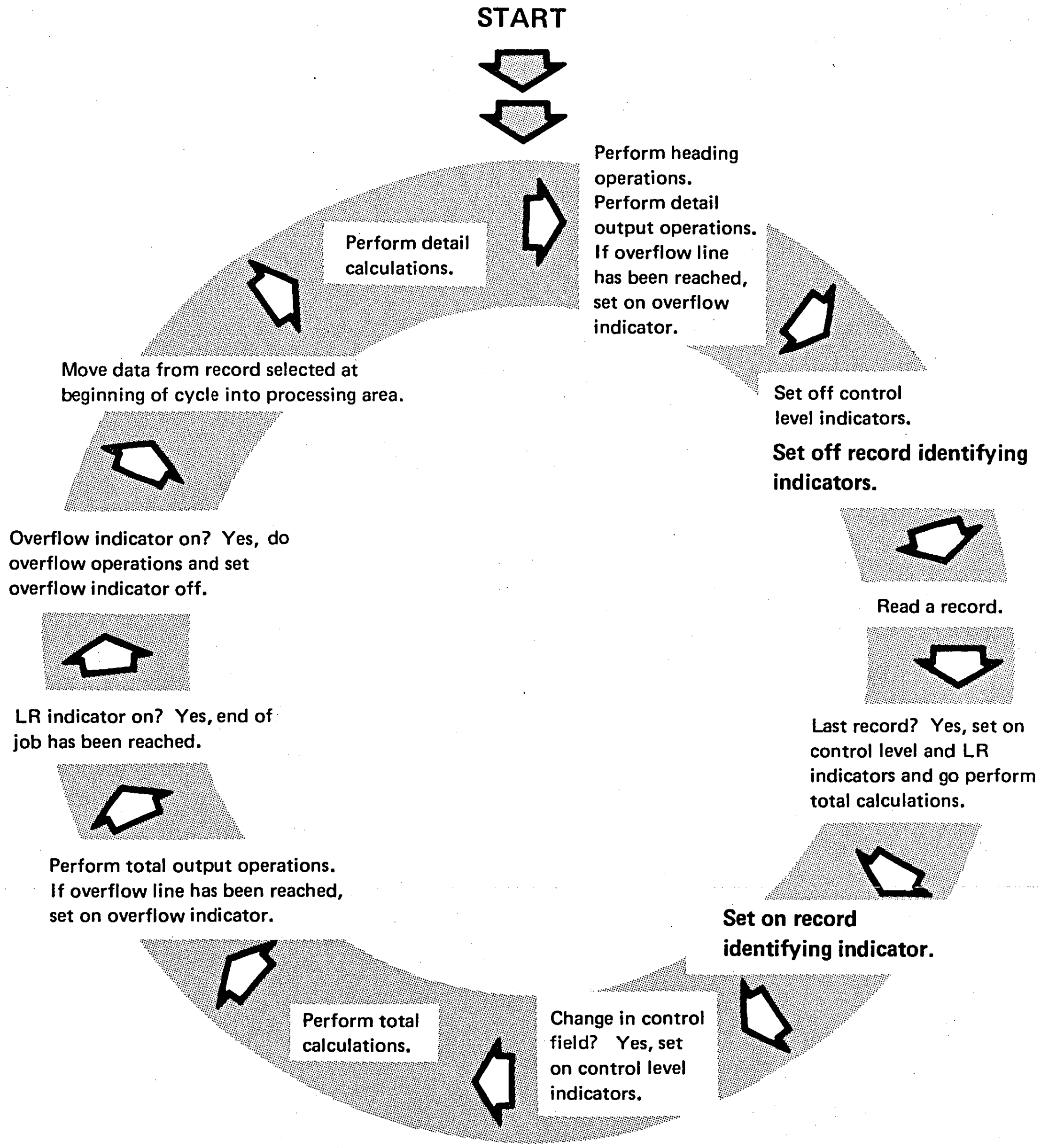
In the jobs discussed so far, we've assumed that all records in the input file were alike. They didn't necessarily contain the same information, but they had the same fields and the same kind of information in each field. They were of the same *type.*

In real jobs, it is unlikely that all data files would contain records of the same type. Files most often contain many types of records with different fields and different information. When using different record types in a job, you must have a way of telling the computer what operations (calculations and output) you want done for each record read. Record identifying indicators are used for this.

Program Cycle Operations

Figure 13 shows program cycle operations associated with record identifying indicators. A record identifying indicator is set on right after a record is read and is set off before the next record is read.

Normally, record identification indicators condition detail calculations and detail output operations because detail operations are done for the record just read (the one associated with the record identifying indicator). On the other hand, total operations are not performed for any one record type; they are done after a certain number of records are processed.

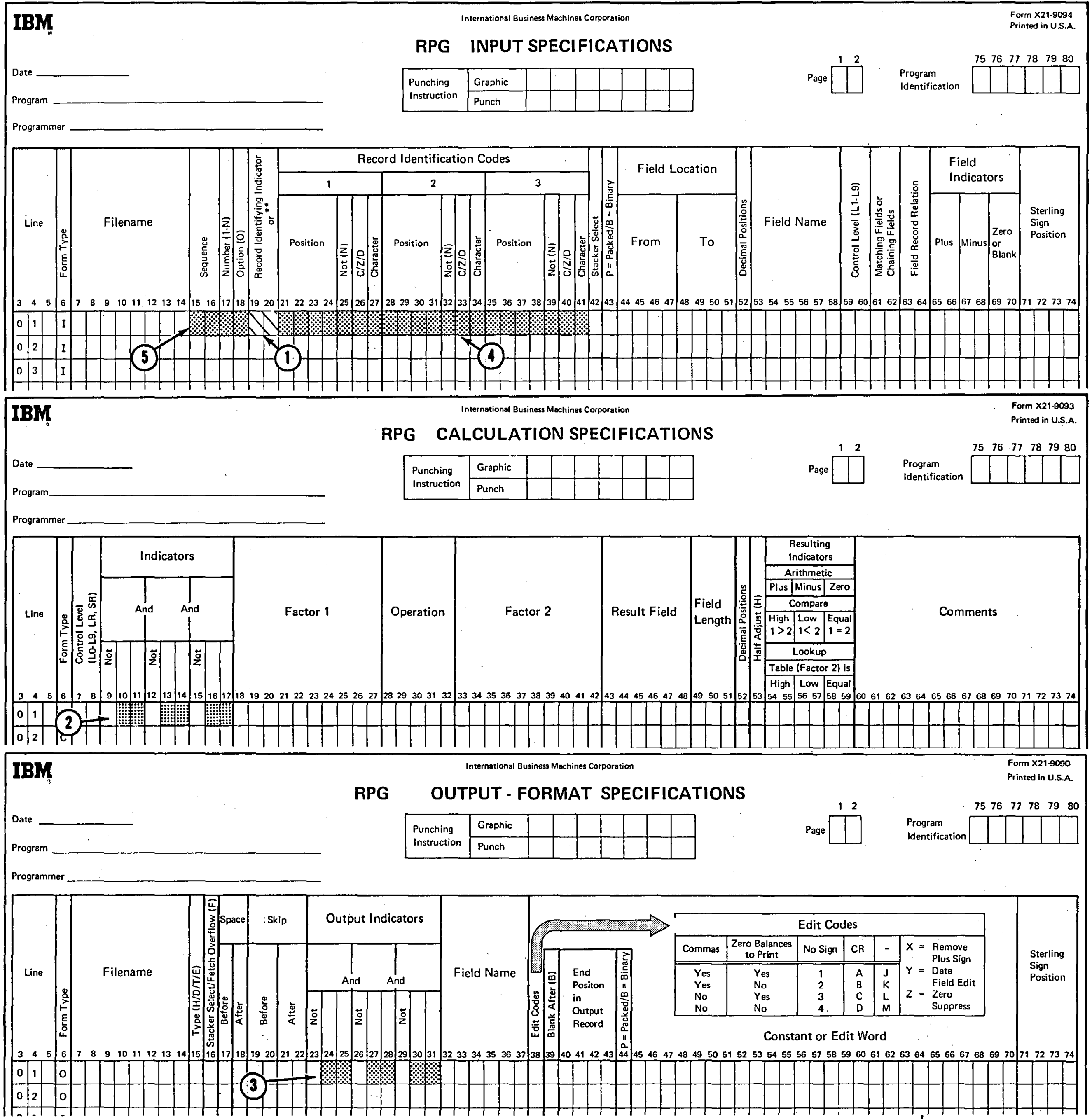


START
Perform heading operations. Perform detail output operations. If overflow line has been reached, set on overflow indicator.
Set off control level indicators.
Set off record identifying indicators.
Read a record.
Last record? Yes, set on control level and LR indicators and go perform total calculations.
Set on record identifying indicator.
Change in control field? Yes, set on control level indicators.
Perform total calculations.
Perform total output operations. If overflow line has been reached, set on overflow indicator.
LR indicator on? Yes, end of job has been reached.
Overflow indicator on? Yes, do overflow operations and set overflow indicator off.
Move data from record selected at beginning of cycle into processing area.
Perform detail calculations.

Figure 13. Program Cycle Operations for Record Identifying Indicators

RPG II Specifications

These are the RPG II specifications you must make when using different record types and record identifying indicators:



IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator or **	Record Identification Codes 1: Position	Not (N)	C/Z/D	Character	2: Position	Not (N)	C/Z/D	Character	3: Position	Not (N)	C/Z/D	Character	Stacker Select	P = Packed/B = Binary	Field Location From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Minus	Zero or Blank	Sterling Sign Position
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
0 1	I		5			1	4																								
0 2	I																														
0 3	I																														

IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators: Not		And: Not		And: Not		Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators: Arithmetic Plus / Compare High 1>2 / Lookup Table (Factor 2) is High	Minus / Low 1< 2 / Low	Zero / Equal 1 = 2 / Equal	Comments
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
0 1	2																		
0 2	C																		

IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space: Before	After	Skip: Before	After	Output Indicators: Not		And: Not		And: Not		Field Name	Edit Codes	Blank After (B)	End Positon in Output Record	P = Packed/B = Binary	Constant or Edit Word	Sterling Sign Position
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
0 1	O						3														
0 2	O																				

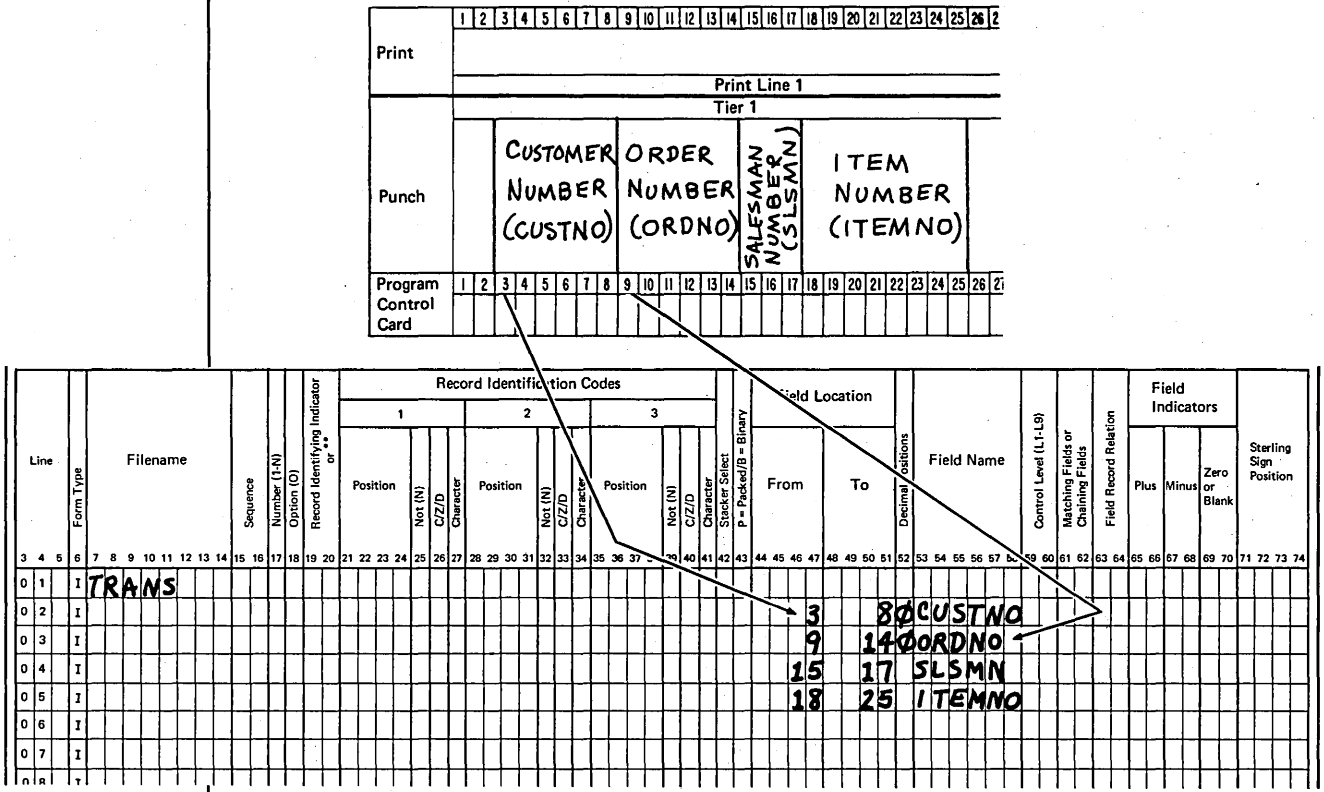
Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Assign a record identifying indicator (a 2-digit number from 01-99) to each record at ①, then use that same indicator to condition calculations at ② and output operations at ③; which must be done for that record only. Identify each record type by describing a code which identifies it at ④. If records must be in a certain order, indicate that sequence at ⑤.

As we said earlier, you must describe each record in your input file. This description includes the names of all fields used from the record, the location of the fields, and the type of data in the fields:



Line	Form Type	Filename	From	To	Decimal Positions	Field Name
01	I	TRANS				
02	I		3	8	Ø	CUSTNO
03	I		9	14	Ø	ORDNO
04	I		15	17		SLSMN
05	I		18	25		ITEMNO
06	I					
07	I					

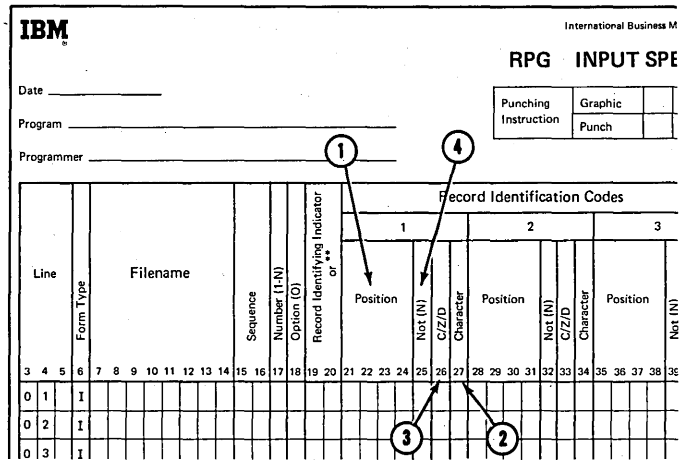
When you describe fields from each record type in the input file, give a unique field name to every *different* field from all record types. If, however, two or more record types contain identical fields, you can assign the same name to the field in each record type. Only one storage area would be assigned for the field from several record types, but it wouldn't matter because the information in the field is the same.

You must do more, however, than merely list the fields on all records, because the computer would not know which fields were on each type of record. Furthermore, it would not know which record it was reading. To give the computer a means of identifying records, you use record identification codes and record identifying indicators.

Specifying Record Identification Codes

When you create records, you should include an identification code on each one. For example, to identify an item transaction record, you might place the code TR somewhere on that record. You can use any combination of letters and numbers for the identification code and you can place the code in any record positions.

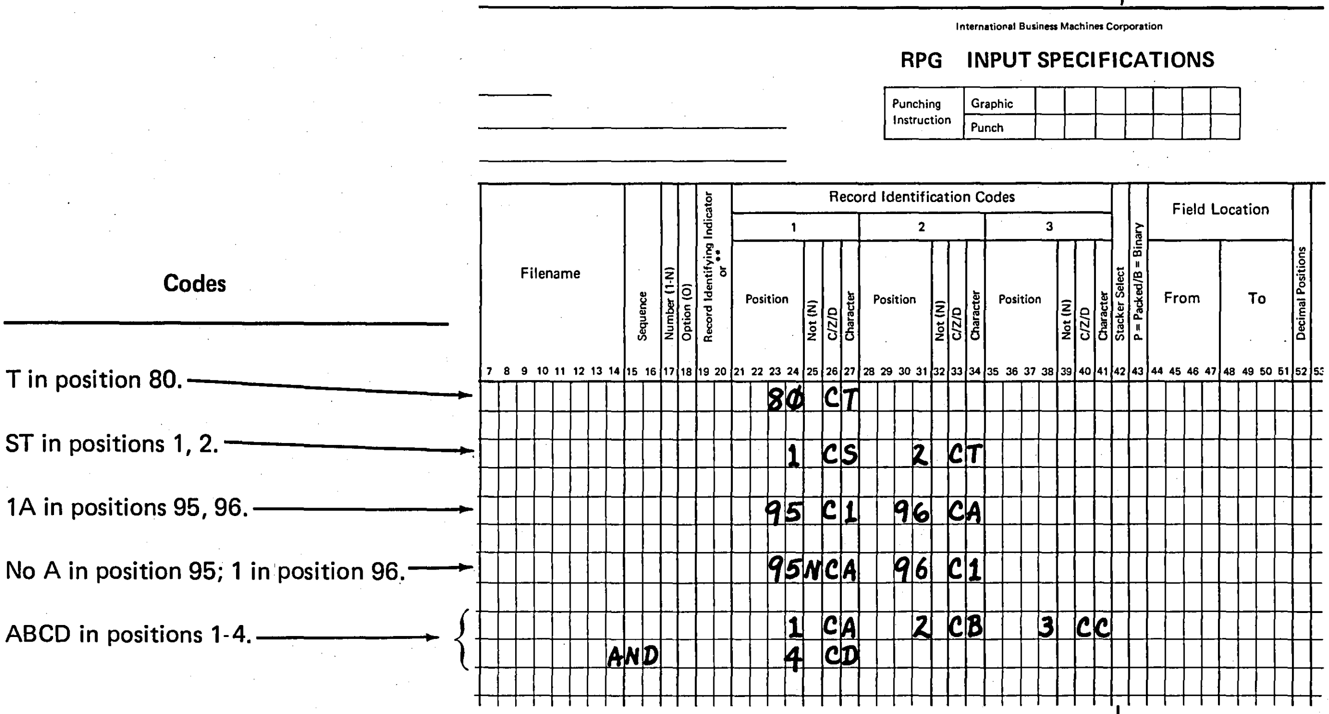
When you describe the record on the Input sheet, you use columns 21-41 to describe the record's identification code and where the code is located on the record:



For each character in the code you must specify:

1. Where in the record the character is found.
2. What the character is.
3. The letter C to indicate character.
4. If the code requires that a character must *not* be present, enter N before the character.

You can specify a code of up to three characters on one line. If your code contains more than three characters, use the next line and the word AND in columns 14-16. Figure 14 shows some examples of how to specify record identification codes.


International Business Machines Corporation
RPG INPUT SPECIFICATIONS
Punching Instruction
Graphic
Punch
Filename
Sequence
Number (1-N)
Option (O)
Record Identifying Indicator or **
Record Identification Codes
Position
Not (N)
C/Z/D
Character
Stacker Select
P = Packed/B = Binary
Field Location
From
To
Decimal Positions
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

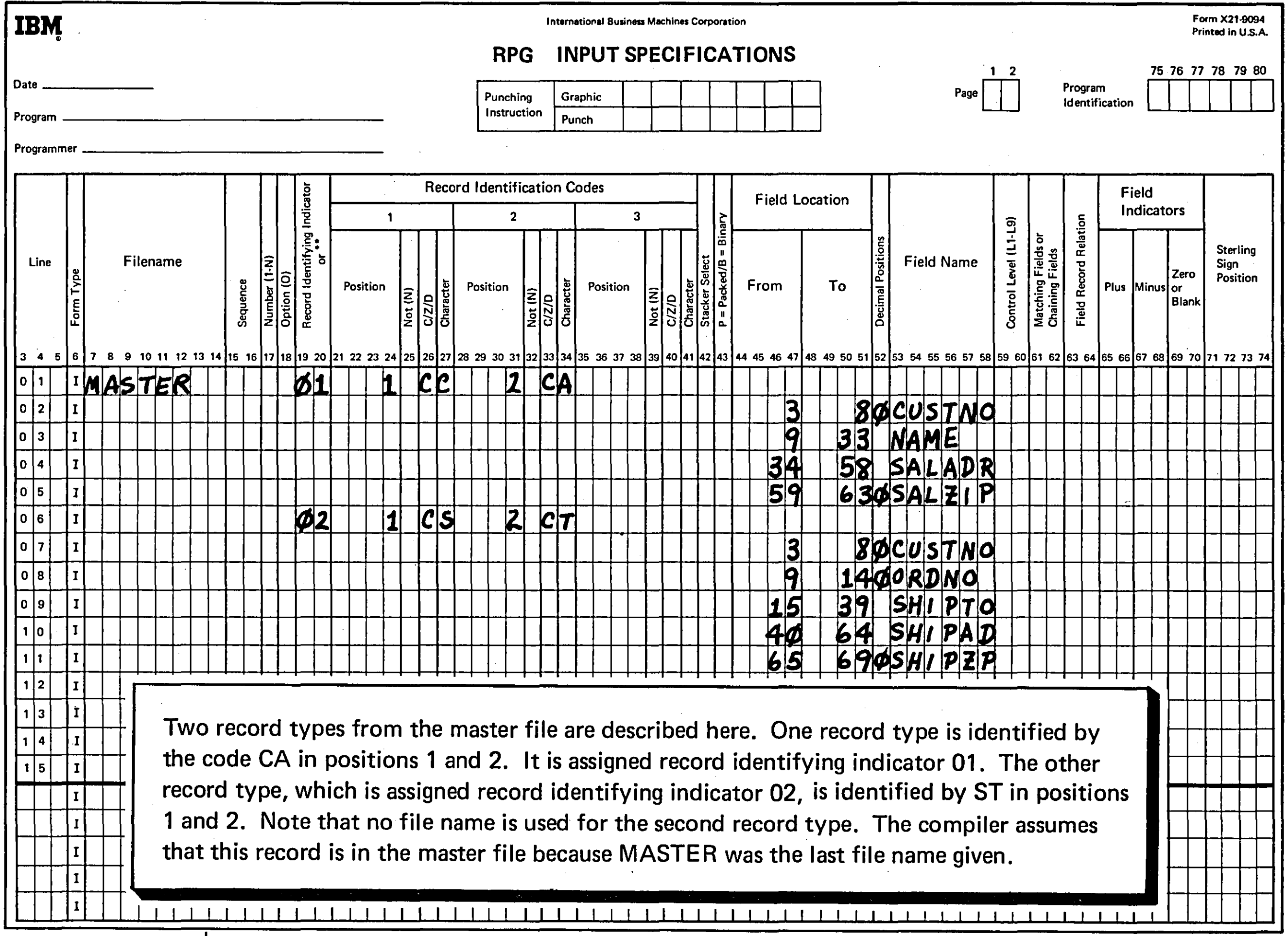
Codes	Filename / Sequence (14-16)	1 Position	1 Not (N)	1 C/Z/D	1 Character	2 Position	2 Not (N)	2 C/Z/D	2 Character	3 Position	3 Not (N)	3 C/Z/D	3 Character
T in position 80.		8Ø		C	T								
ST in positions 1, 2.		1		C	S	2		C	T				
1A in positions 95, 96.		95		C	1	96		C	A				
No A in position 95; 1 in position 96.		95	N	C	A	96		C	1				
ABCD in positions 1-4.		1		C	A	2		C	B	3		C	C
	AND	4		C	D								

Figure 14. Valid Specifications for Record Identification Codes

Specifying Record Identifying Indicators

You specify a record identifying indicator in columns 19-20 for each record type used in the job. Record identifying indicators are numbered 01-99. Use a different number for each record type.

A record identifying indicator is specified on the same line as the identification code. All fields for the record are then listed, starting one line below the identification code and identifying indicator. The file name need be specified only once, on the specification line describing the first record in the file:



IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date

Program

Programmer

Punching Instruction | Graphic | Punch

Page 1 2

Program Identification 75 76 77 78 79 80

							Record Identification Codes														Field Location							Field Indicators			
							1				2				3																
Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator or **	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Stacker Select	P = Packed/B = Binary	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank	Sterling Sign Position
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
0 1	I	MASTER				Ø1	1		C	C	2		C	A																	
0 2	I																				3	8	Ø	CUSTNO							
0 3	I																				9	33		NAME							
0 4	I																				34	58		SALADR							
0 5	I																				59	63	Ø	SALZIP							
0 6	I					Ø2	1		C	S	2		C	T																	
0 7	I																				3	8	Ø	CUSTNO							
0 8	I																				9	14	Ø	ORDNO							
0 9	I																				15	39		SHIPTO							
1 0	I																				4Ø	64		SHIPAD							
1 1	I																				65	69	Ø	SHIPZP							
1 2	I																														
1 3	I																														
1 4	I																														
1 5	I																														
	I																														
	I																														
	I																														
	I																														
	I																														

Two record types from the master file are described here. One record type is identified by the code CA in positions 1 and 2. It is assigned record identifying indicator 01. The other record type, which is assigned record identifying indicator 02, is identified by ST in positions 1 and 2. Note that no file name is used for the second record type. The compiler assumes that this record is in the master file because MASTER was the last file name given.

After reading a record, the program checks the identification codes to determine which record it has read. When it finds a match between the code on the record and the code stored from the Input sheet, it turns on the record identifying indicator associated with that record. This is the program's way of remembering which record it read.

Record identifying indicators can be used to condition calculations, output records, or output fields. In this way, the program performs the proper operations for each different record type.

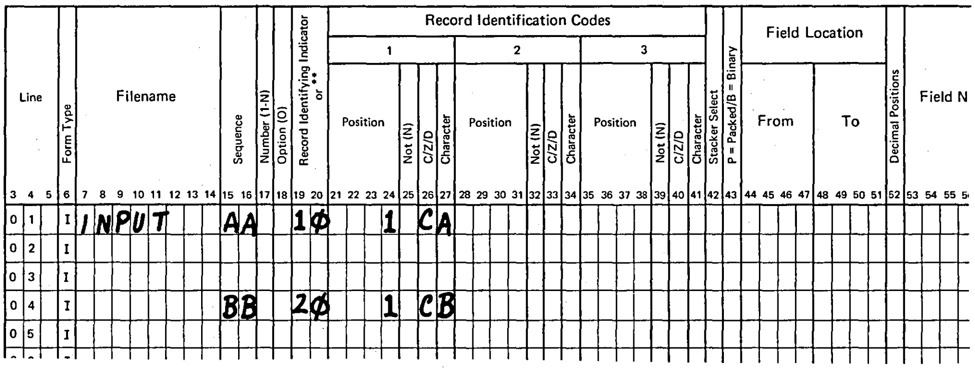
Record identifying indicators must be assigned even when there is only one record type in a file. If you do not use them, the compiler prints a message telling you that record identifying indicators have not been assigned.

When you use record identifying indicators to condition all calculation and output operations, you are assured that these operations are done only for appropriate records. If you do not use these indicators, all operations will be done on all records.

The record identifying indicator should also be used to condition detail output records. This prevents detail lines from being written on the first cycle. If the detail line is not conditioned by a record identifying indicator, any constants you specified on the detail lines would be printed even though the first record had not been read.

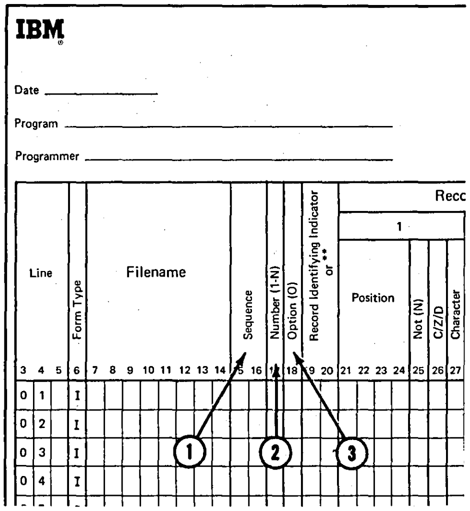
Specifying Record Type Sequence

Sometimes records must be in a particular order within a record-type group, but at other times the order makes no difference. When records need not be in a particular order, enter two alphabetic characters in columns 15-16. You should use different alphabetic characters for each record:



Line (3-5)	Form Type (6)	Filename (7-14)	Sequence (15-16)	Number (1-N) (17)	Option (O) (18)	Record Identifying Indicator or ** (19-20)	Record Identification Codes 1: Position (21-24)	Not (N) (25)	C/Z/D (26)	Character (27)	2: Position (28-31)	Not (N) (32)	C/Z/D (33)	Character (34)	3: Position (35-38)	Not (N) (39)	C/Z/D (40)	Character (41)	Stacker Select (42)	P = Packed/B = Binary (43)	Field Location From (44-47)	To (48-51)	Decimal Positions (52)	Field N (53-5…)
0 1	I	INPUT	AA			1φ	1		C	A														
0 2	I																							
0 3	I																							
0 4	I		BB			2φ	1		C	B														
0 5	I																							

If the records must be in a specific sequence, you must make entries in columns 15-18:



IBM

Date

Program

Programmer

Line (3-5)	Form Type (6)	Filename (7-14)	Sequence (15-16)	Number (1-N) (17)	Option (O) (18)	Record Identifying Indicator or ** (19-20)	Record 1: Position (21-24)	Not (N) (25)	C/Z/D (26)	Character (27)
0 1	I									
0 2	I									
0 3	I									
0 4	I									

(1) Columns 15-16 (Sequence) contain numeric entries (01-99). The sequence entries must be in numeric order.

(2) Column 17 (Number) tells how many records of this type to expect in the record-type group:

1 = One and only one
N = One or more

(3) Column 18 (Option) indicates whether the record type is required in the record-type group:

Blank = Record type is required
O = Record type is optional

Columns 15-16 should have an entry for every record type specified even if there is only one type per file. If you fail to use a sequence entry, the compiler prints a message saying that no entry was made in columns 15-16.

Job 5: Using Record Identifying Indicators To Process Different Record Types

JOB DEFINITION

Print a Stock Status Report. This report is printed whenever the inventory is updated. It gives detailed information on all active merchandise. The first line for each item in the report shows standard descriptive data for the item: item number, item description, quantity on hand, and quantity on order. This information is taken directly from the input record.

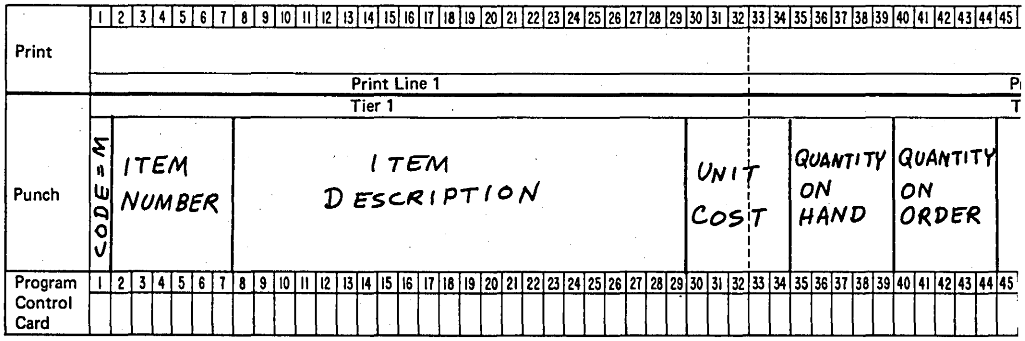
Subsequent lines give the detail on current transactions involving the item: sales to customers and receipts from suppliers. This information is also taken directly from input records.

Quantities remaining on hand and on order are calculated for each item and printed after all transactions for the item are listed.

JOB REQUIREMENTS

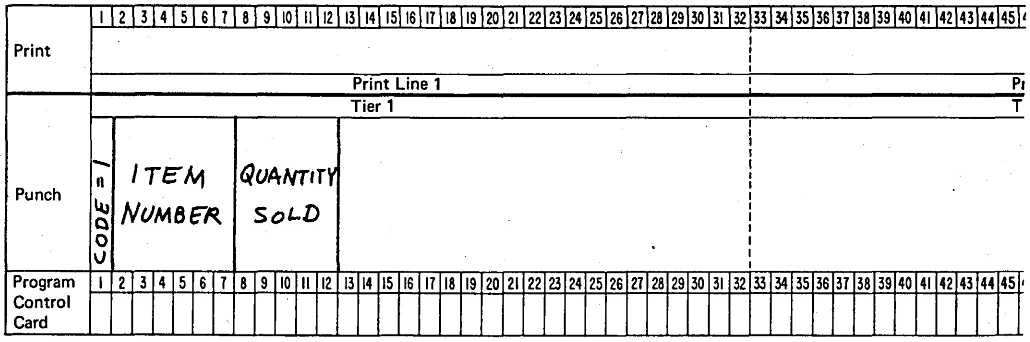
Input: An inventory file consisting of three different types of records. Formats of the three record types are:

Card Name ITEM MASTER RECORD

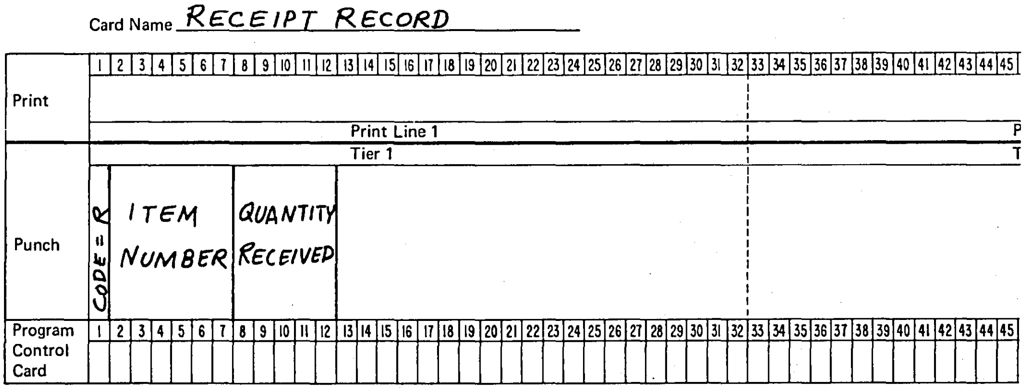


Print	Print Line 1					
Punch	CODE = M	ITEM NUMBER	ITEM DESCRIPTION	UNIT COST	QUANTITY ON HAND	QUANTITY ON ORDER
Program Control Card						

Card Name ISSUE RECORD



Print	Print Line 1		
Punch	CODE = I	ITEM NUMBER	QUANTITY SOLD
Program Control Card			



Card Name RECEIPT RECORD

Print	Print Line 1
	Tier 1
Punch	CODE = R | ITEM NUMBER | QUANTITY RECEIVED
Program Control Card	

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
P
T

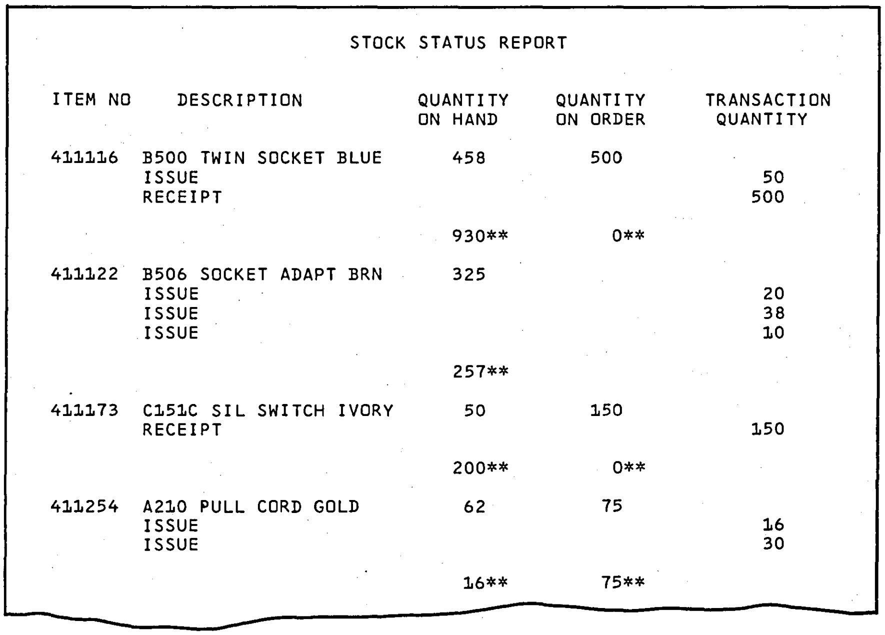
The file is organized in ascending order by item number. For each item one master record is required. Issue and receipt records are optional. When present, however, there may be any number of each. Records for each item are in this order:

1. Item master
2. Issue
3. Receipt

Processing:

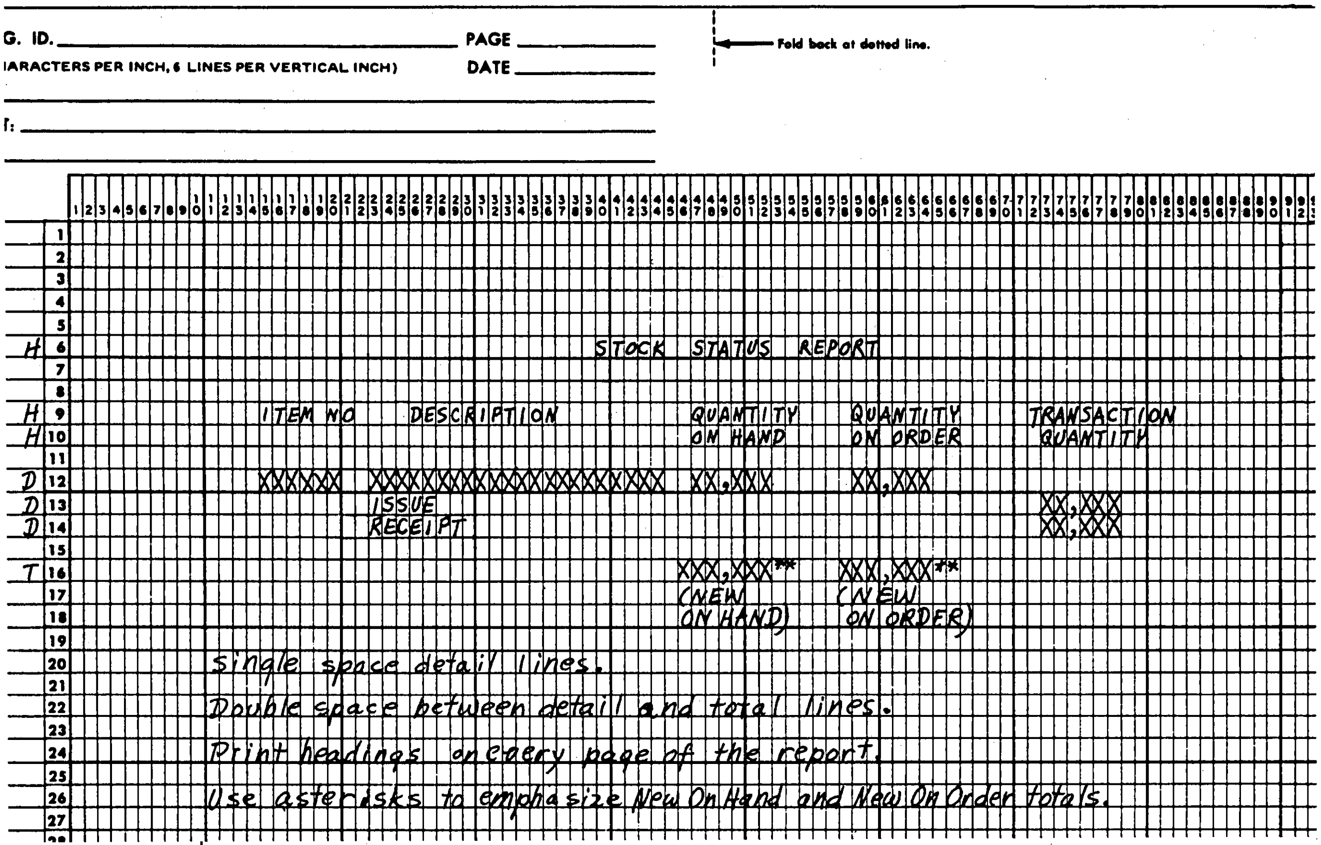
- Find total number of each item sold. To do this, perform the calculation ISSUE + TOTAL ISSUE = TOTAL ISSUE for each issue record.
- Find total number of each item received. Perform the calculation RECEIPT + TOTAL RECEIPT = TOTAL RECEIPT for each receipt record.
- When all transaction records for one item have been read, find new quantity onhand (ONHAND + TOTAL RECEIPT - TOTAL ISSUE = NEW ONHAND) and new quantity on order (ON ORDER - TOTAL RECEIPT = NEW ON ORDER).

Output: A stock status report printed on a 96-position printer:



```
                             STOCK STATUS REPORT

ITEM NO    DESCRIPTION            QUANTITY     QUANTITY       TRANSACTION
                                  ON HAND      ON ORDER        QUANTITY

411116   B500 TWIN SOCKET BLUE      458          500
         ISSUE                                                    50
         RECEIPT                                                 500

                                    930**          0**

411122   B506 SOCKET ADAPT BRN      325
         ISSUE                                                    20
         ISSUE                                                    38
         ISSUE                                                    10

                                    257**

411173   C151C SIL SWITCH IVORY      50          150
         RECEIPT                                                 150

                                    200**          0**

411254   A210 PULL CORD GOLD         62           75
         ISSUE                                                    16
         ISSUE                                                    30

                                     16**         75**
```

This Printer Spacing Chart shows the format of the report:



G. ID. ______________________ PAGE ________ ← Fold back at dotted line.

IARACTERS PER INCH, 6 LINES PER VERTICAL INCH) DATE ________

I: ______________________

```
   1
   2
   3
   4
   5
H  6                                                        STOCK     STATUS     REPORT
   7
   8
H  9                         ITEM NO         DESCRIPTION                  QUANTITY         QUANTITY          TRANSACTION
H 10                                                                      ON HAND          ON ORDER           QUANTITY
  11
D 12                         XXXXXX     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   XX,XXX           XX,XXX
D 13                                    ISSUE                                                                   XX,XXX
D 14                                    RECEIPT                                                                 XX,XXX
  15
T 16                                                                     XXX,XXX**        XXX,XXX**
  17                                                                     (NEW             (NEW
  18                                                                     ON HAND)          ON ORDER)
  19
  20                Single space detail lines.
  21
  22                Double space between detail and total lines.
  23
  24                Print headings on every page of the report.
  25
  26                Use asterisks to emphasize New On Hand and New On Order totals.
  27
```

JOB SPECIFICATIONS

IBM — International Business Machines Corporation — Form X21-9092 Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71

Program Stock Status

Programmer LaDonna Hoffmann

Punching Instruction: Graphic / Punch

Page 1 2: 01

Program Identification 75 76 77 78 79 80: STOKST

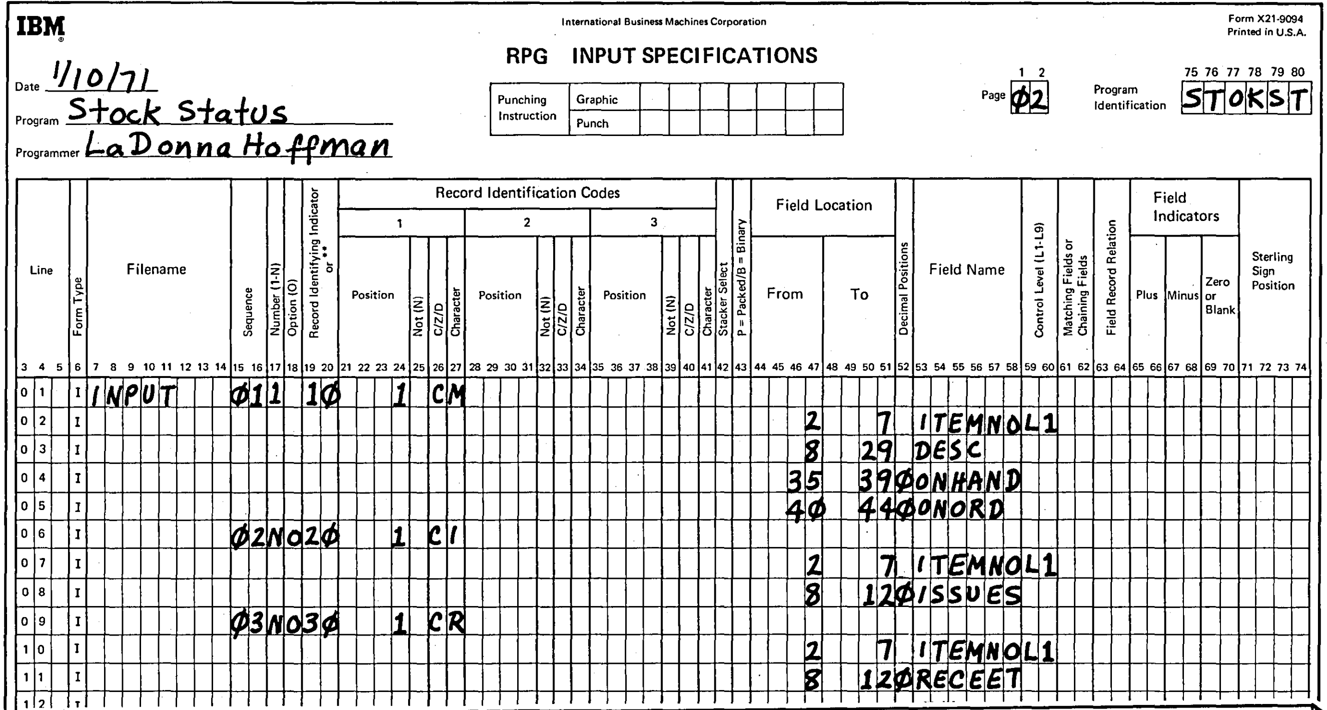
Line | Core Size to Compile | Output | Options | Core Size to Execute | Stacking Sequence | Sterling: Shillings, Pence, I-Shillings, I-Pence | Print | 501 Buffer | Number Of

Two files are used for the job: one input and one output. In the job description given, the output file was associated with the printer, but the input file was not associated with any device. Darkened columns show required entries; the entry depends on the device being used.

File Description Specifications

Line (3-5)	Form Type (6)	Filename (7-14)	File Type I/O/U/C/D (15)	File Designation P/S/C/R/T/D (16)	End of File E (17)	Sequence A/D (18)	File Format F/V/S/M/D (19)	Block Length (20-23)	Record Length (24-27)	L/R (28)	Mode of Processing (29-30)	Length of Key Field or of Record Address Field	Record Address Type A/P/I/K (31)	Type of File Organization or Additional Area I/D/T or 2 (32)	Overflow Indicator (33-34)	Key Field Starting Location (35-38)	Extension Code E/L (39)	Device (40-46)	Symbolic Device (47-52)	Labels S/N/E/M (53)	Name of Label Exit (54-59)	Extent Exit for DAM / Core Index (60-65)	A/U (66)	File Addition/Unordered (67)	Number of Tracks for Cylinder Overflow (68-69)	Tape Rewind R/U/N (70)	File Condition U1-U8 (71-72)	(73-74)
0 2	F	INPUT	I	P					▨	▨								▨										
0 3	F	OUTPUT	O						96						OV			PRINTER										
0 4	F																											

Continuation Lines: K (53), Option (54-59), Entry (60-65)



IBM

International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Form X21-9094
Printed in U.S.A.

Date 1/10/71
Program Stock Status
Programmer LaDonna Hoffman

Punching Instruction	Graphic	
	Punch	

Page Ø2

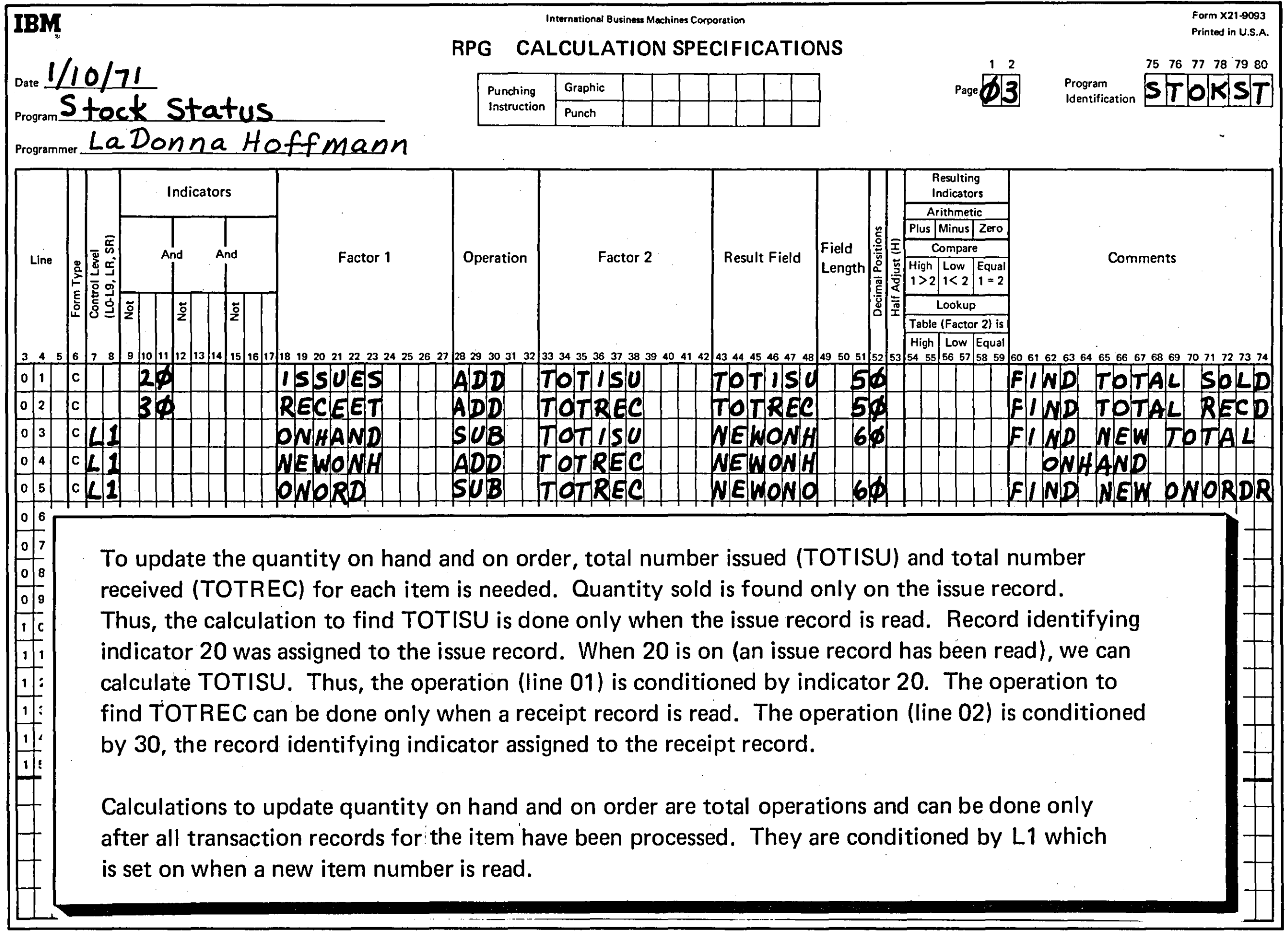
Program Identification STOKST

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator	Position	Not (N)	C/Z/D	Character	From	To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	INPUT	Ø1	1		1Ø	1		C	M					
02	I										2	7		ITEMNO	L1
03	I										8	29		DESC	
04	I										35	39	Ø	ONHAND	
05	I										4Ø	44	Ø	ONORD	
06	I		Ø2	N	O	2Ø	1		C	I					
07	I										2	7		ITEMNO	L1
08	I										8	12	Ø	ISSUES	
09	I		Ø3	N	O	3Ø	1		C	R					
10	I										2	7		ITEMNO	L1
11	I										8	12	Ø	RECEET	
12	I														

All three types of records in the input file must be described. Since they are to be arranged in a certain sequence in the record-type group, they are described in the order they will be read. The first record in the group is the item master record (identified by M in position 1). The 01 in columns 15-16 indicate that this record is first. The record is required (column 18 is blank) and there must be only one per group (1 in column 17). It is assigned record identifying indicator 10. All fields found on the record are then described. Note that the record code appears as a field on the Record Layout Form. However, on the Input sheet it is not described as a field but as the record identification code.

The second record in the group (02 in columns 15-16) is the issue record, identified by I in position 1. There may be several of these records per group (N in column 17). The record is optional (O in column 18). The record identifying indicator 20 is assigned. Fields on the record are then described. The third record in the group is described as were the previous two.

ITEMNO field on the item master record is assigned as the control field. When it changes, all transaction records for the item number have been processed.



IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date 1/10/71

Program Stock Status

Programmer LaDonna Hoffmann

Punching Instruction: Graphic / Punch

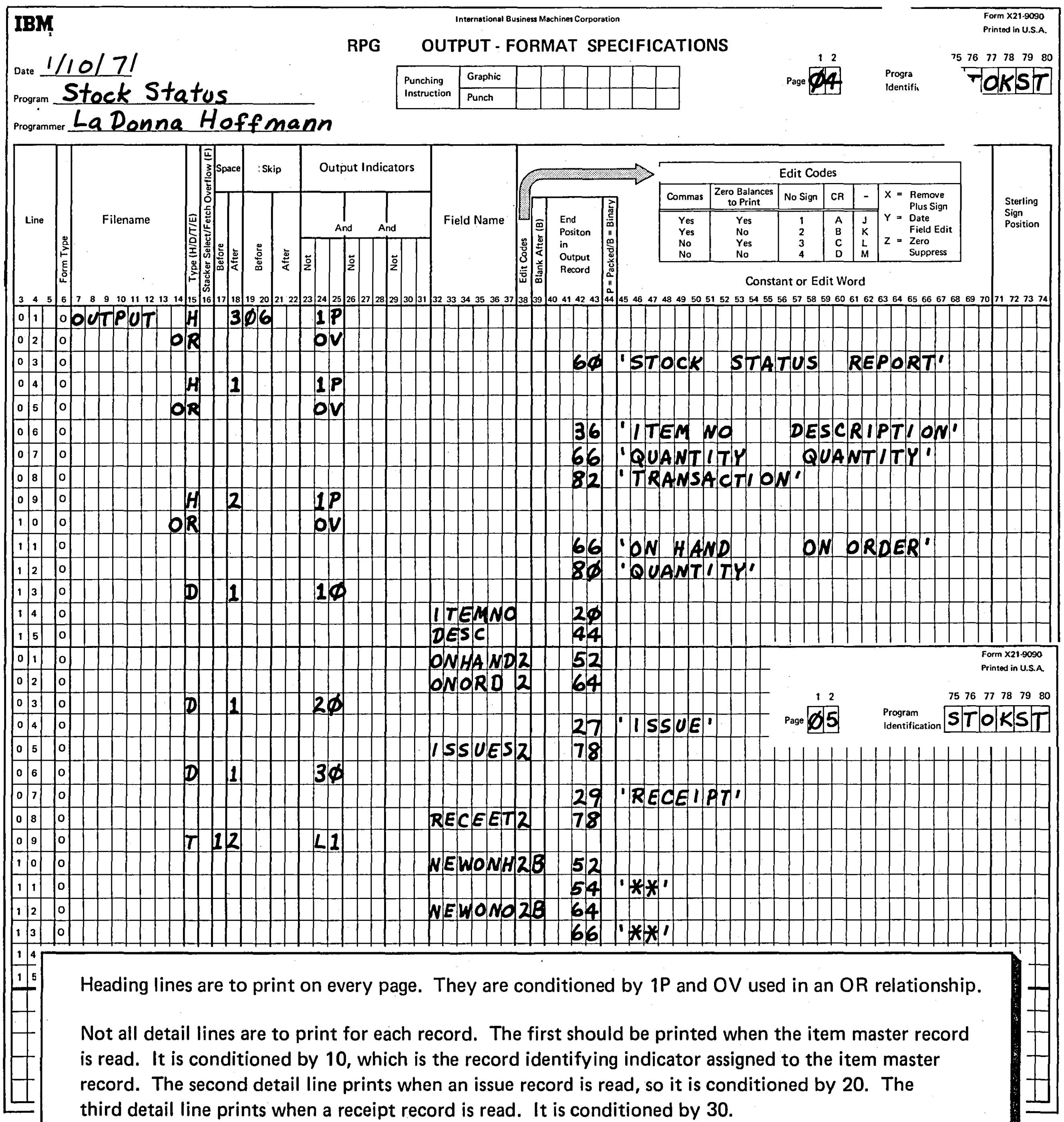
Page Ø3

Program Identification STOKST

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C		2Ø	ISSUES	ADD	TOTISU	TOTISU	5	Ø			FIND TOTAL SOLD
02	C		3Ø	RECEET	ADD	TOTREC	TOTREC	5	Ø			FIND TOTAL RECD
03	C	L1		ONHAND	SUB	TOTISU	NEWONH	6	Ø			FIND NEW TOTAL
04	C	L1		NEWONH	ADD	TOTREC	NEWONH					ONHAND
05	C	L1		ONORD	SUB	TOTREC	NEWONO	6	Ø			FIND NEW ONORDR

To update the quantity on hand and on order, total number issued (TOTISU) and total number received (TOTREC) for each item is needed. Quantity sold is found only on the issue record. Thus, the calculation to find TOTISU is done only when the issue record is read. Record identifying indicator 20 was assigned to the issue record. When 20 is on (an issue record has been read), we can calculate TOTISU. Thus, the operation (line 01) is conditioned by indicator 20. The operation to find TOTREC can be done only when a receipt record is read. The operation (line 02) is conditioned by 30, the record identifying indicator assigned to the receipt record.

Calculations to update quantity on hand and on order are total operations and can be done only after all transaction records for the item have been processed. They are conditioned by L1 which is set on when a new item number is read.



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71

Program Stock Status

Programmer La Donna Hoffmann

Punching Instruction	Graphic	
	Punch	

Page Ø4 Program Identification OKST

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After (B)	End Position in Output Record	Constant or Edit Word
01	O	OUTPUT	H		3	Ø6		1P					
02	O		OR					OV					
03	O											6Ø	'STOCK STATUS REPORT'
04	O		H		1			1P					
05	O		OR					OV					
06	O											36	'ITEM NO DESCRIPTION'
07	O											66	'QUANTITY QUANTITY'
08	O											82	'TRANSACTION'
09	O		H		2			1P					
10	O		OR					OV					
11	O											66	'ON HAND ON ORDER'
12	O											8Ø	'QUANTITY'
13	O		D		1			1Ø					
14	O								ITEMNO			2Ø	
15	O								DESC			44	

Form X21-9090
Printed in U.S.A.

Page Ø5 Program Identification STOKST

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After (B)	End Position in Output Record	Constant or Edit Word
01	O								ONHAND	2		52	
02	O								ONORD	2		64	
03	O		D		1			2Ø					
04	O											27	'ISSUE'
05	O								ISSUES	2		78	
06	O		D		1			3Ø					
07	O											29	'RECEIPT'
08	O								RECEET	2		78	
09	O		T	1	2			L1					
10	O								NEWONH	2	B	52	
11	O											54	'**'
12	O								NEWONO	2	B	64	
13	O											66	'**'
14													
15													

Heading lines are to print on every page. They are conditioned by 1P and OV used in an OR relationship.

Not all detail lines are to print for each record. The first should be printed when the item master record is read. It is conditioned by 10, which is the record identifying indicator assigned to the item master record. The second detail line prints when an issue record is read, so it is conditioned by 20. The third detail line prints when a receipt record is read. It is conditioned by 30.

The Printer Spacing Chart shows that the words ISSUE and RECEIPT are to print in the detail lines. These words are not fields so they are entered as constants in the appropriate detail lines. Note that asterisks indicating totals are also entered as constants in the total line.

RESULTING INDICATORS

Sometimes your decision to do a certain operation is based on the result of a previous operation. Resulting indicators allow you to specify which operations you want done and the conditions under which the operations are to be done. Resulting indicators can be used to determine:

1. Whether a result is larger, smaller, or equal to a predetermined number.

2. Whether a certain result is plus, minus, or zero.

Program Cycle Operations

Figure 15 shows the operations in the program cycle associated with resulting indicators. Resulting indicators are set when the associated calculation operation is performed. This means that resulting indicators can be set either at detail or at total calculation time.

Resulting indicators are not set off automatically. They change their setting only at the time a calculation is performed. For example, if a resulting indicator is set on by a detail calculation, it retains this setting until the next time it is used as a resulting indicator.

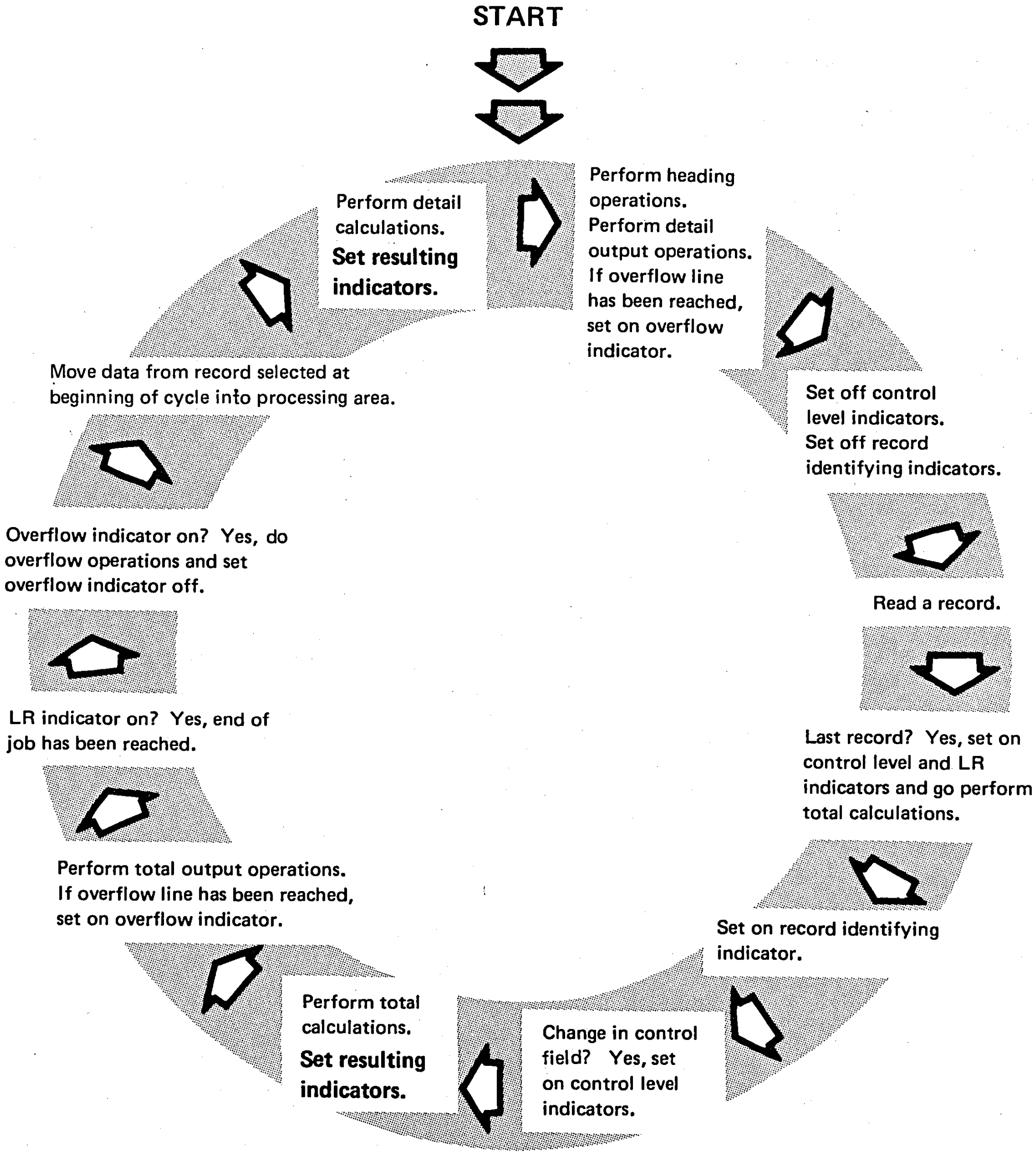
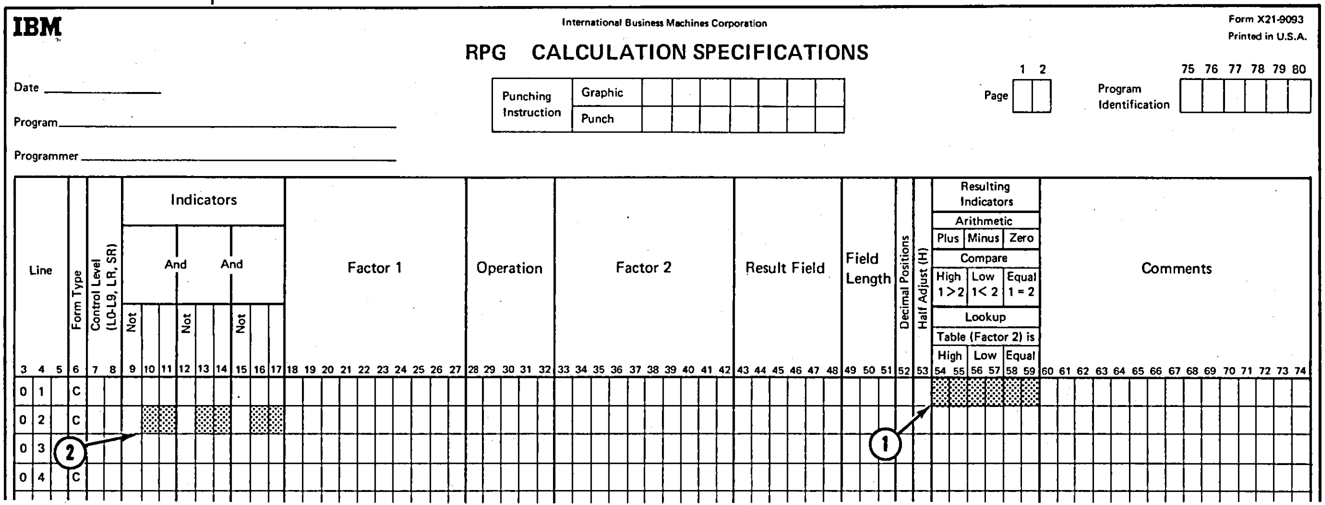

START
Perform heading operations. Perform detail output operations. If overflow line has been reached, set on overflow indicator.
Set off control level indicators. Set off record identifying indicators.
Read a record.
Last record? Yes, set on control level and LR indicators and go perform total calculations.
Set on record identifying indicator.
Change in control field? Yes, set on control level indicators.
Perform total calculations. Set resulting indicators.
Perform total output operations. If overflow line has been reached, set on overflow indicator.
LR indicator on? Yes, end of job has been reached.
Overflow indicator on? Yes, do overflow operations and set overflow indicator off.
Move data from record selected at beginning of cycle into processing area.
Perform detail calculations. Set resulting indicators.

Figure 15. Program Cycle Operations for Resulting Indicators

The type of operation used to check the result field depends on the type of result being checked. If you want to determine whether the result field is larger, smaller, or equal to a certain number, you must use a compare (COMP) operation. If you want to determine if the result field is plus, minus, or zero, use an arithmetic (ADD, SUB, MULT, DIV) operation. You can specify resulting indicators 01-99 on these specifications sheets:



IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

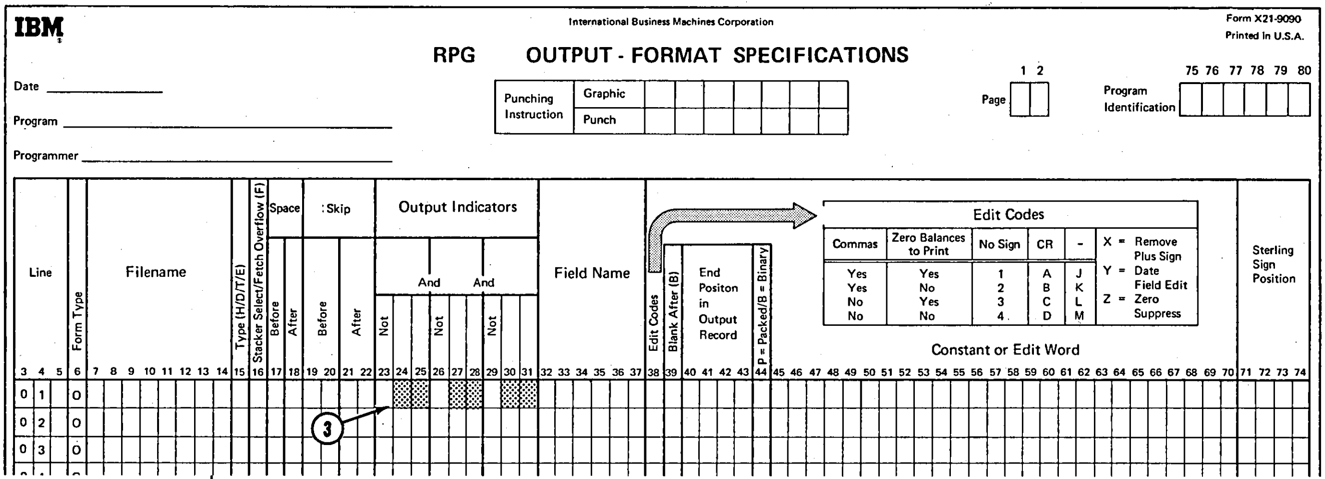
Date ______ Program ______ Programmer ______

Punching Instruction: Graphic / Punch

Page 1 2 Program Identification 75 76 77 78 79 80

Line | Form Type | Control Level (L0-L9, LR, SR) | Indicators (Not, And Not, And Not) | Factor 1 | Operation | Factor 2 | Result Field | Field Length | Decimal Positions | Half Adjust (H) | Resulting Indicators: Arithmetic Plus Minus Zero; Compare High 1>2 Low 1<2 Equal 1=2; Lookup Table (Factor 2) is High Low Equal | Comments

Lines: 0 1 C; 0 2 C; 0 3 C; 0 4 C



IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date ______ Program ______ Programmer ______

Punching Instruction: Graphic / Punch

Page 1 2 Program Identification 75 76 77 78 79 80

Line | Form Type | Filename | Type (H/D/T/E) | Stacker Select/Fetch Overflow (F) | Space Before After | Skip Before After | Output Indicators (Not, And Not, And Not) | Field Name | Edit Codes | Blank After (B) | End Positon in Output Record | P = Packed/B = Binary | Constant or Edit Word | Sterling Sign Position

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

Edit Codes: X = Remove Plus Sign; Y = Date Field Edit; Z = Zero Suppress

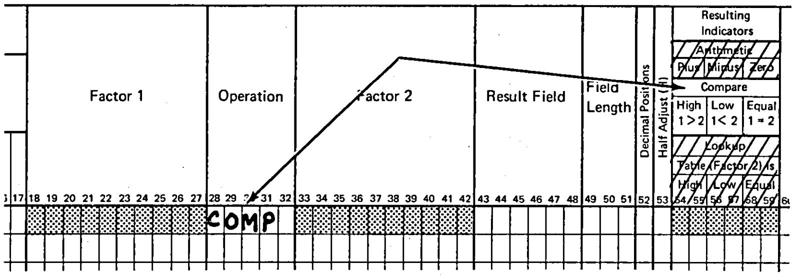
Lines: 0 1 O; 0 2 O; 0 3 O

Resulting indicators are assigned at (1), then used to condition calculation operations at (2) and output operations at (3)

The entries for resulting indicators and record identifying indicators are the same. Be careful not to use the same indicator for both of these purposes in the same program because the computer cannot distinguish between them.

Using the Compare Operation

In many jobs you need to know whether a field is greater than, smaller than, or equal to another field. RPG II language has an operation code, COMP, which allows you to compare fields. The compare operation requires entries in these columns on the Calculation sheet:

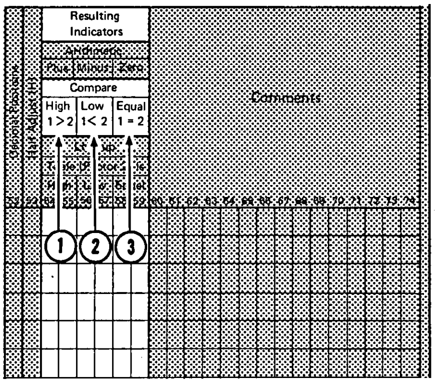


- Factor 1 (either a field name or constant)
- Factor 2 (either a field name or constant)
- Resulting indicators

When compared, Factor 1 and Factor 2 can be in one of three relationships:

- Factor 1 can be greater than Factor 2.
- Factor 1 can be less than Factor 2.
- Factor 1 can be equal to Factor 2.

You indicate that a test should be made to check for one, two, or all three of these relationships by entering indicators in the appropriate columns:



(1) A resulting indicator entered in columns 54-55 tells the computer to determine if Factor 1 is greater than Factor 2.

(2) A resulting indicator entered in columns 56-57 tells the computer to determine if Factor 1 is less than Factor 2.

(3) A resulting indicator entered in columns 58-59 tells the computer to determine if Factor 1 is the same as Factor 2.

The test you specify is made each time the COMP operation is executed. However, the resulting indicator is set on only when the proper relationship exists. If you entered indicator 50 in columns 54-55 to test whether Factor 1 is greater than Factor 2, indicator 50 would be set on only when Factor 1 is greater than Factor 2.

When testing for more than one condition, you can use the same or different indicators in these columns. If you intend to do different operations for each of the three conditions, enter a different resulting indicator to test for each condition on the Calculation sheet:

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators Not	Indicators	And Not	And	And Not	And	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Plus / High 1>2	Minus / Low 1<2	Zero / Equal 1=2	Comments
3 4 5	6	7 8	9	10 11	12	13 14	15	16 17	18-27	28-32	33-42	43-48	49 50 51	52	53	54 55	56 57	58 59	60-74
01	C								MAX	COMP	TOTAL					11	22	33	
02	C			11					TOTAL	MULT	.Ø5	COMM	7	2	H				
03	C			22					TOTAL	MULT	.Ø6	COMM			H				
04	C			33					TOTAL	MULT	.Ø7	COMM			H				
05	C																		
06	C																		
07	C																		
08	C																		
09	C																		
10	C																		
11	C																		
12	C																		
13	C																		
14	C																		

In this operation, the contents of the field MAX is compared to the contents of the field TOTAL. If MAX is greater than TOTAL, indicator 11 is set on and the operation in line 02 is done. If MAX is less than TOTAL, indicator 22 is set on and the operation in line 03 is done. If MAX is equal to TOTAL, indicator 33 is set on and the operation in line 04 is done.

If you want to do the same operations when either one of two conditions exists (Factor 1 is greater than Factor 2; Factor 1 equals Factor 2), you could use the same indicator to test for both conditions on the Calculation sheet:

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators Not	Indicators	And Not	And	And Not	And	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Plus / High 1>2	Minus / Low 1<2	Zero / Equal 1=2	Comments
3 4 5	6	7 8	9	10 11	12	13 14	15	16 17	18-27	28-32	33-42	43-48	49 50 51	52	53	54 55	56 57	58 59	60-74
01	C								TOTSLS	COMP	5ØØØ.ØØ					1Ø	15	1Ø	
02	C			15					TOTSLS	MULT	.Ø2	DISCT	7	2	H				
03	C			1Ø					TOTSLS	MULT	.Ø3	DISCT			H				
04	C																		
05	C																		
06	C																		
07	C																		
08	C																		
09	C																		
10	C																		
11	C																		
12	C																		
13	C																		
14	C																		
15	C																		
	C																		
	C																		

These operations are used for finding the amount of discount to give customers. If the customer purchases goods worth $5000.00 or more, he receives a 3% discount, but if he purchases goods worth less than $5000.00, he receives only a 2% discount. TOTSLS is first compared to $5000.00. If TOTSLS is less than $5000.00, indicator 15 is set on and the operations in line 02 are performed (2% discount is calculated). However, if TOTSLS is either equal to or greater than $5000.00, indicator 10 is set on and the operation in line 03 is performed (3% discount is calculated).

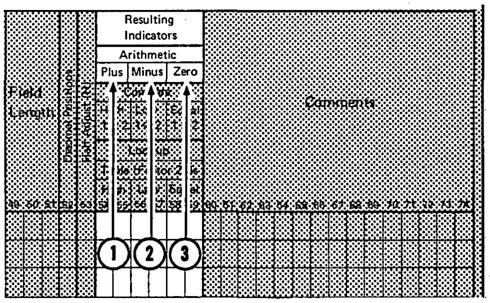
We explained previously that constants can be used in calculation operations, but they must be numeric constants. In a COMP operation, however, constants can be either alphameric or numeric. Rules for using alphameric constants as Factor 1 or Factor 2 are a little different from those for using numeric constants:

Rules for Numeric Constants	Rules for Alphameric Constants
• A numeric constant can be any combination of digits 0-9. Decimal points and signs can also be included.	• An alphameric constant can be any combination of characters. Blanks are also valid.
• The maximum length of a numeric constant is 10 characters, including sign and decimal point.	• The maximum length of an alphameric constant is 8 characters.
• Numeric constants must not be enclosed in single quotes (').	• Alphameric constants must be enclosed in single quotes (').

When you use the COMP operation code, remember to always compare two numeric fields or constants or two alphameric fields or constants. You cannot compare a numeric field or constant to an alphameric field or constant.

Using An Arithmetic Operation

You can test the results of an arithmetic operation (ADD, SUB, MULT, DIV) for plus, minus, or zero by entering resulting indicators in the appropriate columns on the Calculation sheet:



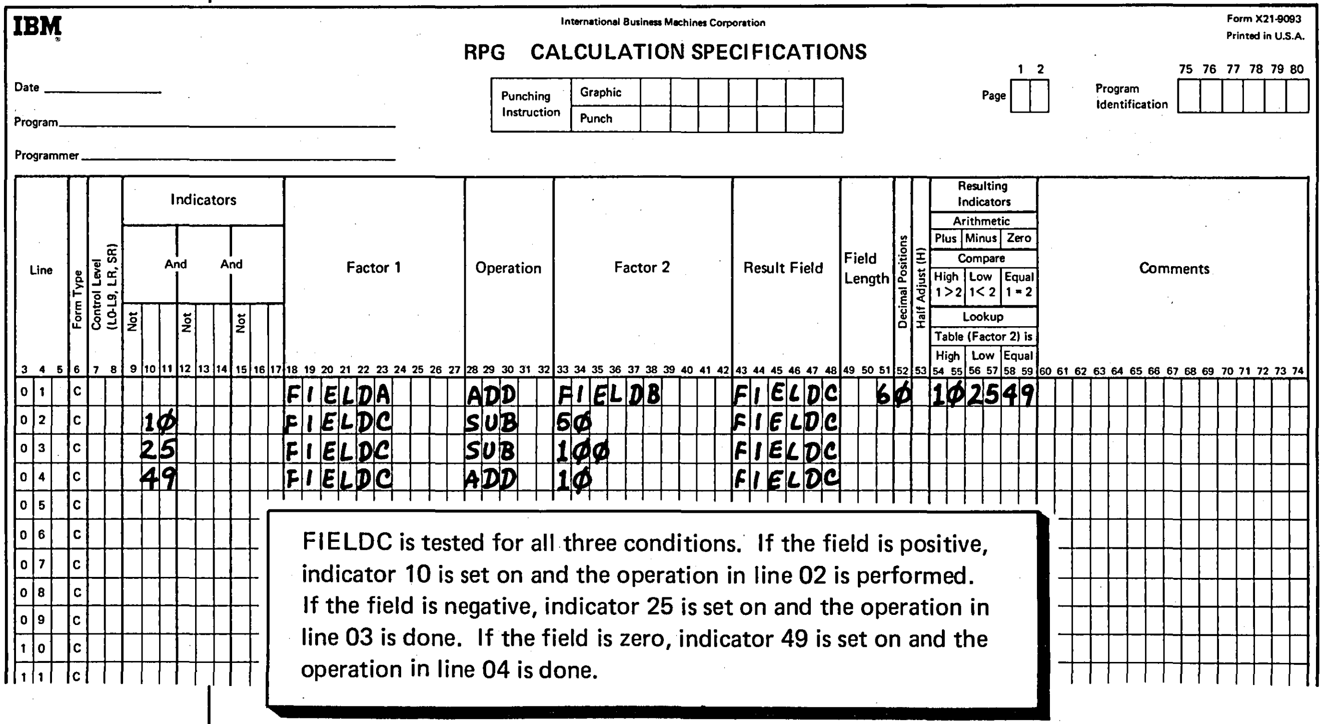
① A resulting indicator entered in columns 54-55 tells the computer to determine if the result field is positive (plus).

② A resulting indicator entered in columns 56-57 tells the computer to determine if the result field is negative (minus).

③ A resulting indicator entered in columns 58-59 tells the computer to determine if the result field is zero.

The tests you indicate are performed each time the operation is executed. However, the assigned indicator is set on only if the field satisfies the condition tested. If you entered indicator 99 in columns 54-55 to test the result field for plus, indicator 99 would be set on only if the result field were plus.

Again, as with the COMP operation, you can test for one, two, or all three conditions at the same time. When testing for more than one condition, you can use the same or different indicators in these columns. If you intend to do different operations for each of the three conditions, enter a different resulting indicator to test for each condition:



IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date ______ Program ______ Programmer ______

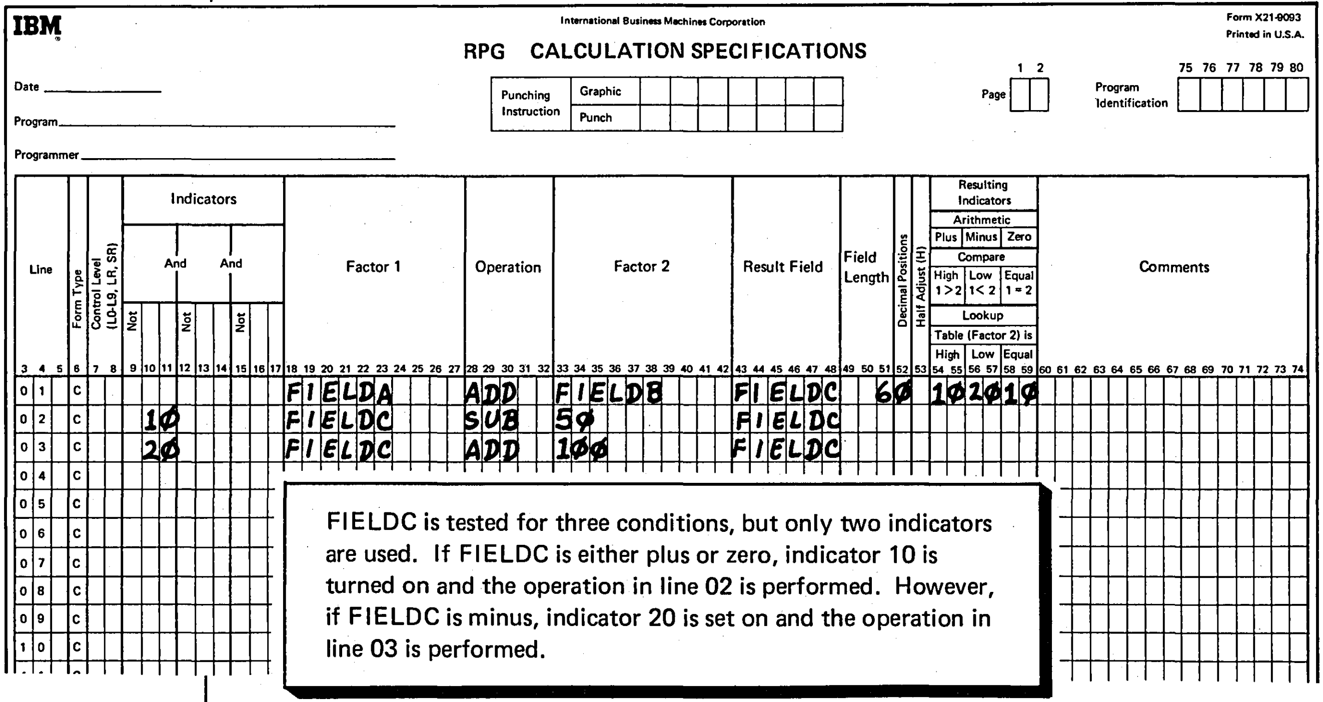
Punching Instruction: Graphic / Punch

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Plus	Minus	Zero	Comments
01	C			FIELDA	ADD	FIELDB	FIELDC	6	Ø		1Ø	25	49	
02	C		1Ø	FIELDC	SUB	5Ø	FIELDC							
03	C		25	FIELDC	SUB	1ØØ	FIELDC							
04	C		49	FIELDC	ADD	1Ø	FIELDC							
05	C													
06	C													
07	C													
08	C													
09	C													
10	C													
11	C													

FIELDC is tested for all three conditions. If the field is positive, indicator 10 is set on and the operation in line 02 is performed. If the field is negative, indicator 25 is set on and the operation in line 03 is done. If the field is zero, indicator 49 is set on and the operation in line 04 is done.

If you want to do the same operations when the result field meets either one of two conditions (plus or zero, minus or zero), you could use the same indicator to test for both:



IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date ______ Program ______ Programmer ______

Punching Instruction: Graphic / Punch

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Plus	Minus	Zero	Comments
01	C			FIELDA	ADD	FIELDB	FIELDC	6	Ø		1Ø	2Ø	1Ø	
02	C		1Ø	FIELDC	SUB	5Ø	FIELDC							
03	C		2Ø	FIELDC	ADD	1ØØ	FIELDC							
04	C													
05	C													
06	C													
07	C													
08	C													
09	C													
10	C													

FIELDC is tested for three conditions, but only two indicators are used. If FIELDC is either plus or zero, indicator 10 is turned on and the operation in line 02 is performed. However, if FIELDC is minus, indicator 20 is set on and the operation in line 03 is performed.

Job 6: Using Resulting Indicators To Test Contents Of Result Fields

JOB DEFINITION

Print a stock status report similar to the one in Job 5. The only difference is the addition of maximum and minimum balances. Item master records usually include the maximum and minimum on-hand quantity for all items. These figures are kept so that checks can be made, whenever the inventory is updated, to determine if quantity on hand is within the limits set.

The first line for each item in the report shows standard descriptive data for the item: item number, item description, quantity on hand, quantity on order, maximum and minimum balances. Subsequent lines give the detail on current transactions involving the item. Quantities remaining on hand and on order are calculated for each item and printed after all transactions for the item are listed. Whenever shipments reduce stock on hand below the predetermined minimum balance or whenever receipts push the quantity on hand above the predetermined maximum, an exception condition is noted on the report.

JOB REQUIREMENTS

Input: An inventory file consisting of three different record types. Formats of the three record types are:

Card Name ITEM MASTER RECORD

Print	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
	Print Line 1				Print Line 2					
Punch	Tier 1				Tier 2					
	CODE = M	ITEM NUMBER	DESCRIPTION	UNI COS	T T	QUANTITY ON HAND	QUANTITY ON ORDER	MAX. BAL.	MIN. BAL.	
Program Control Card	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

Card Name ISSUE RECORD

Print	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
	Print Line 1			Print Line 2
Punch	Tier 1			Tier 2
	CODE = I	ITEM NUMBER	QUANTITY SOLD	
Program Control Card	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

Card Name RECEIPT RECORD

Print	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
	Print Line 1			Print Line 2
Punch	Tier 1			Tier 2
	CODE = R	ITEM NUMBER	QUANTITY RECEIVED	
Program Control Card	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

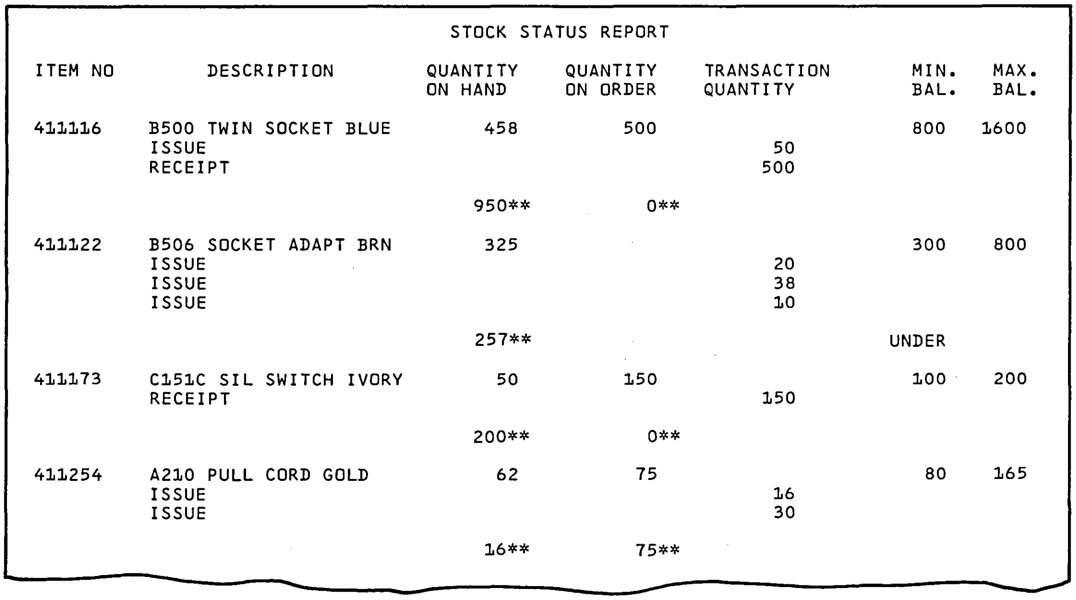
The file is organized in ascending order by item number. For each item, one master record is required. Issue and receipt records are optional. When present, however, there may be any number of each. Records for each item are in this order:

1. Item master
2. Issue
3. Receipt

Processing:

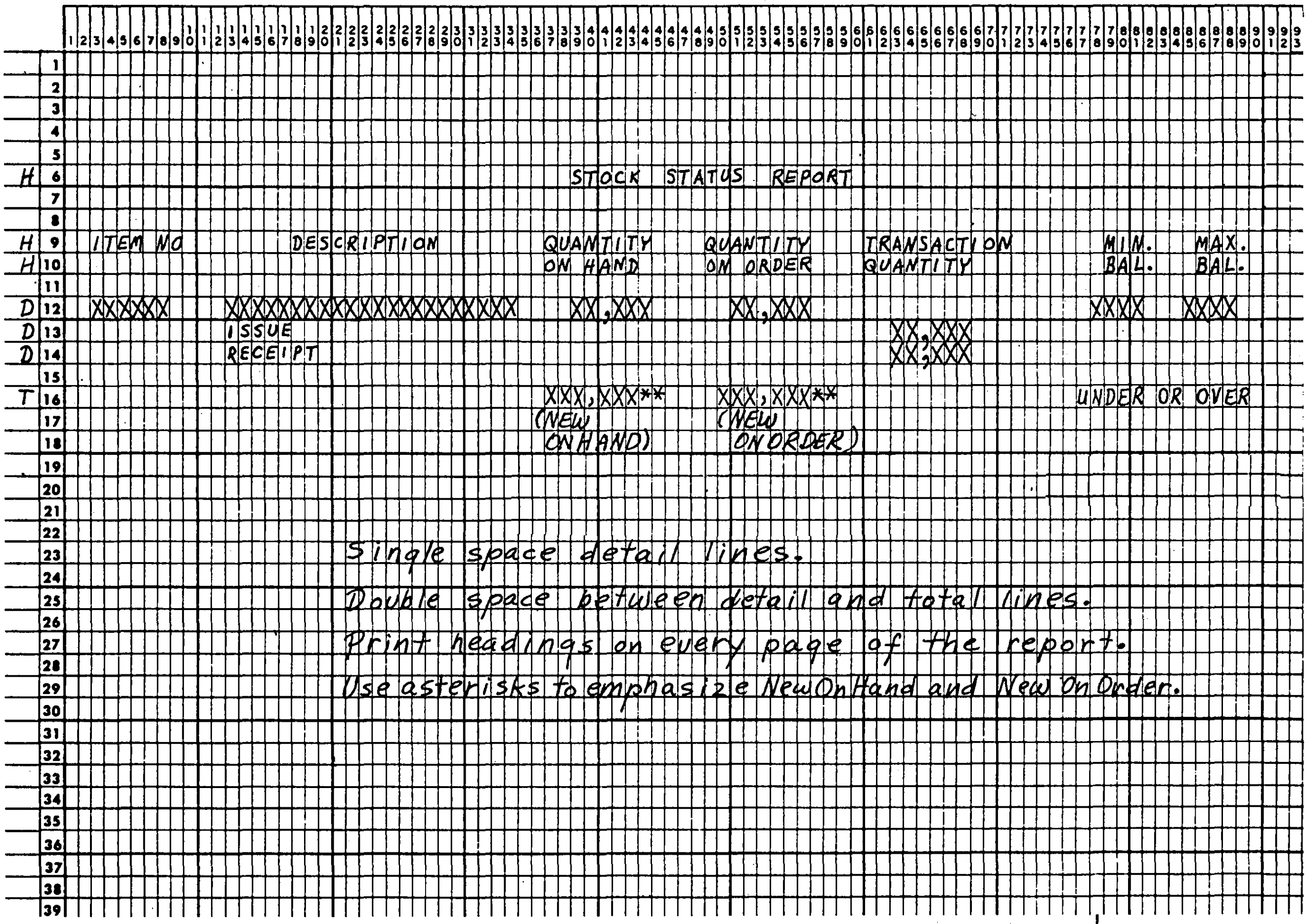
- Find total number of each item sold. To do this, perform the calculation ISSUE + TOTAL ISSUE = TOTAL ISSUE for each issue record.
- Find total number of each item received. To do this, perform the calculation RECEIPT + TOTAL RECEIPT = TOTAL RECEIPT for each receipt record.
- When all transaction records for one item have been read, find new quantity on hand (ON HAND + TOTAL RECEIPT - TOTAL ISSUE = NEW ON HAND) and new quantity on order (ON ORDER - TOTAL RECEIPT = NEW ON ORDER).
- Compare the new quantity on hand to maximum and minimum balances to determine if an exception condition should be noted on the report.

Output: A stock status report printed on a 96-positon printer:



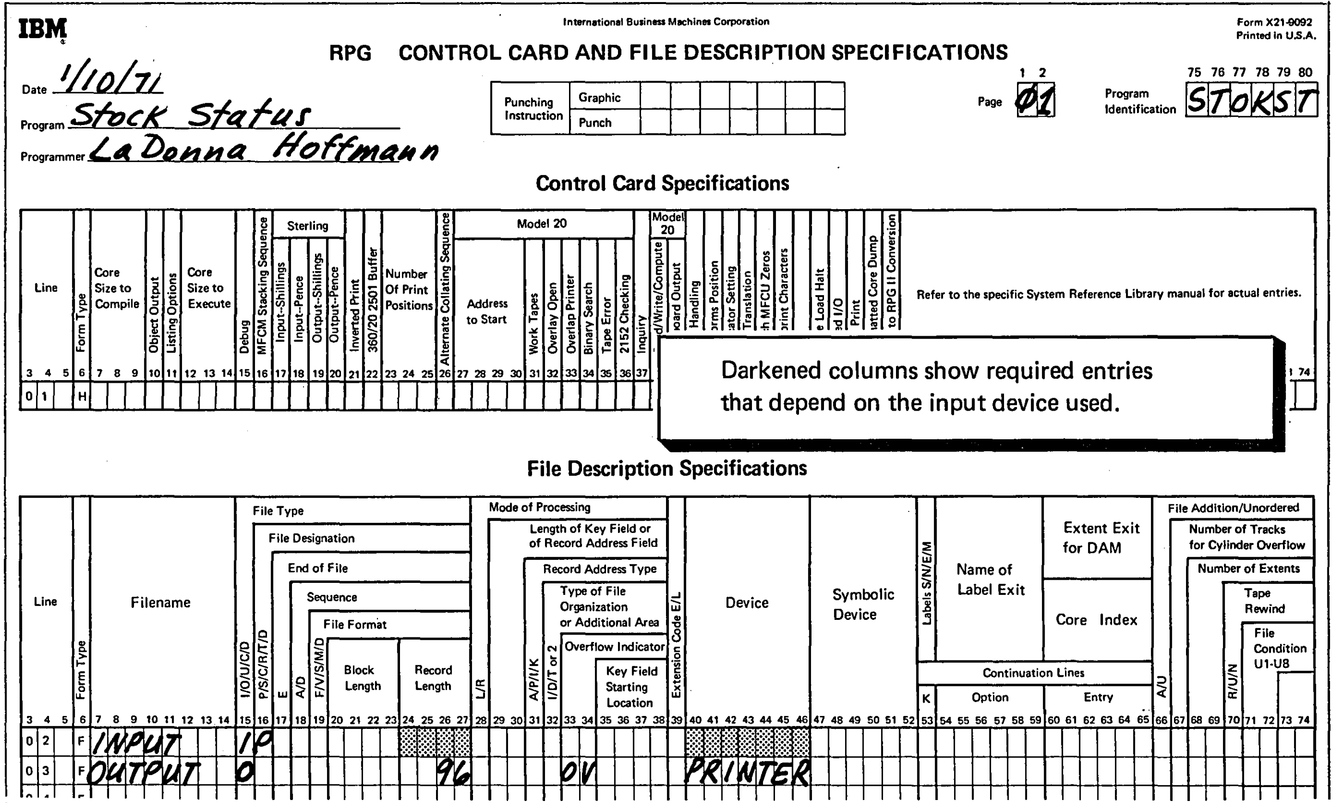
```
                                    STOCK STATUS REPORT

 ITEM NO          DESCRIPTION         QUANTITY    QUANTITY    TRANSACTION        MIN.    MAX.
                                      ON HAND     ON ORDER    QUANTITY           BAL.    BAL.

 411116       B500 TWIN SOCKET BLUE        458         500                        800    1600
              ISSUE                                                   50
              RECEIPT                                                500

                                           950**         0**

 411122       B506 SOCKET ADAPT BRN        325                                    300     800
              ISSUE                                                   20
              ISSUE                                                   38
              ISSUE                                                   10

                                           257**                               UNDER

 411173       C151C SIL SWITCH IVORY        50         150                        100     200
              RECEIPT                                                150

                                           200**         0**

 411254       A210 PULL CORD GOLD           62          75                         80     165
              ISSUE                                                   16
              ISSUE                                                   30

                                            16**        75**
```

This Printer Spacing Chart shows the format of the report:



```
H  6                                       STOCK  STATUS  REPORT

H  9    ITEM NO          DESCRIPTION         QUANTITY     QUANTITY     TRANSACTION       MIN.    MAX.
H 10                                         ON HAND      ON ORDER     QUANTITY          BAL.    BAL.

D 12    XXXXX     XXXXXXXXXXXXXXXXXXXXXXX     XX,XXX       XX,XXX                        XXXX    XXXX
D 13              ISSUE                                                  XX,XXX
D 14              RECEIPT                                                XX,XXX

T 16                                        XXX,XXX**    XXX,XXX**                     UNDER OR OVER
  17                                       (NEW         (NEW
  18                                        ON HAND)     ON ORDER)

  23                        Single space detail lines.
  25                        Double space between detail and total lines.
  27                        Print headings on every page of the report.
  29                        Use asterisks to emphasize New On Hand and New On Order.
```

JOB SPECIFICATIONS



IBM

International Business Machines Corporation

Form X21-9092
Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71

Program Stock Status

Programmer LaDonna Hoffmann

Punching Instruction: Graphic / Punch

Page Ø1

Program Identification STOKST

Control Card Specifications

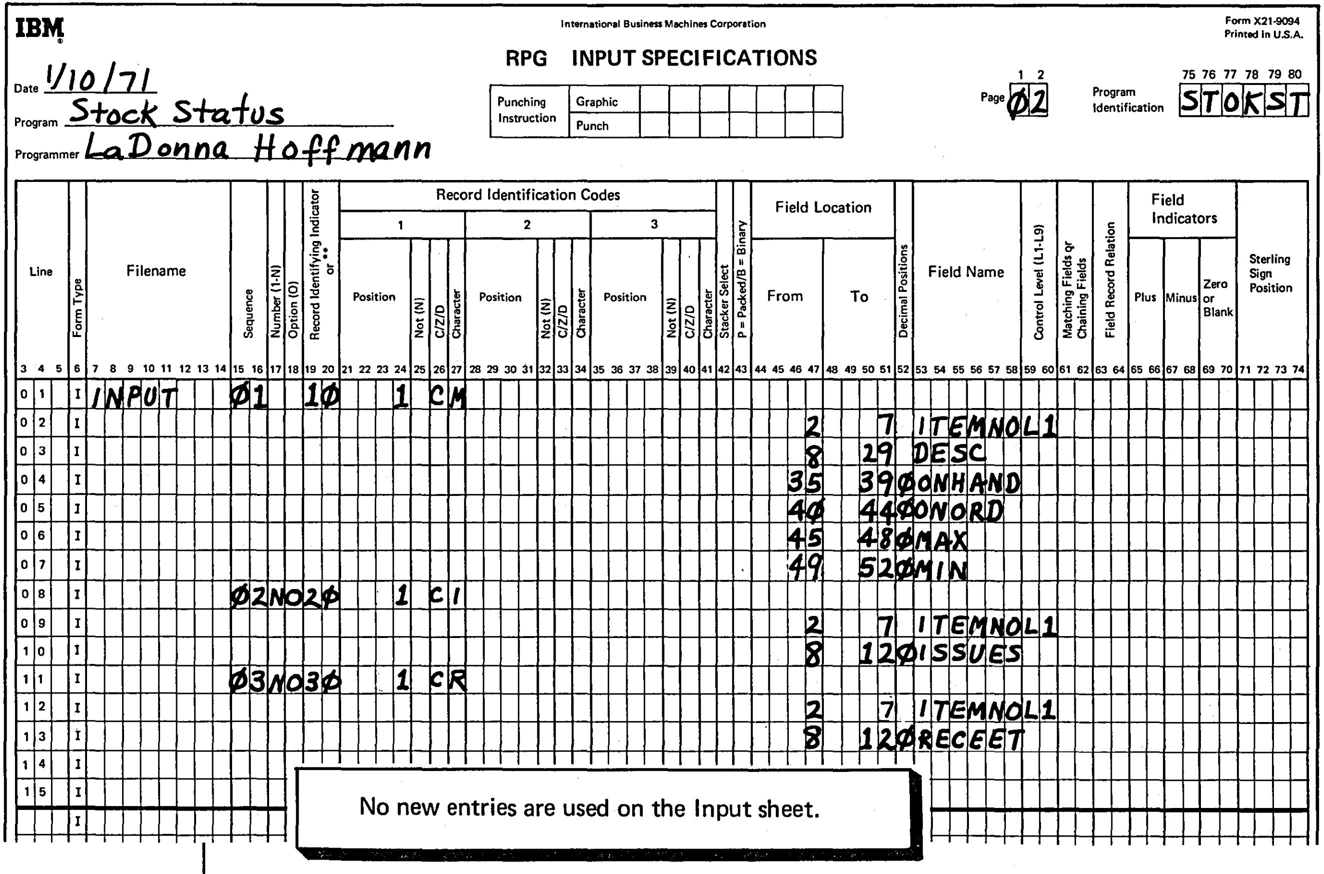
Line	Form Type
01	H

Refer to the specific System Reference Library manual for actual entries.

Darkened columns show required entries that depend on the input device used.

File Description Specifications

Line	Form Type	Filename	File Type / File Designation	Record Length	Overflow Indicator	Device
02	F	INPUT	IP			
03	F	OUTPUT	O	96	OV	PRINTER



IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date 1/10/71

Program Stock Status

Programmer LaDonna Hoffmann

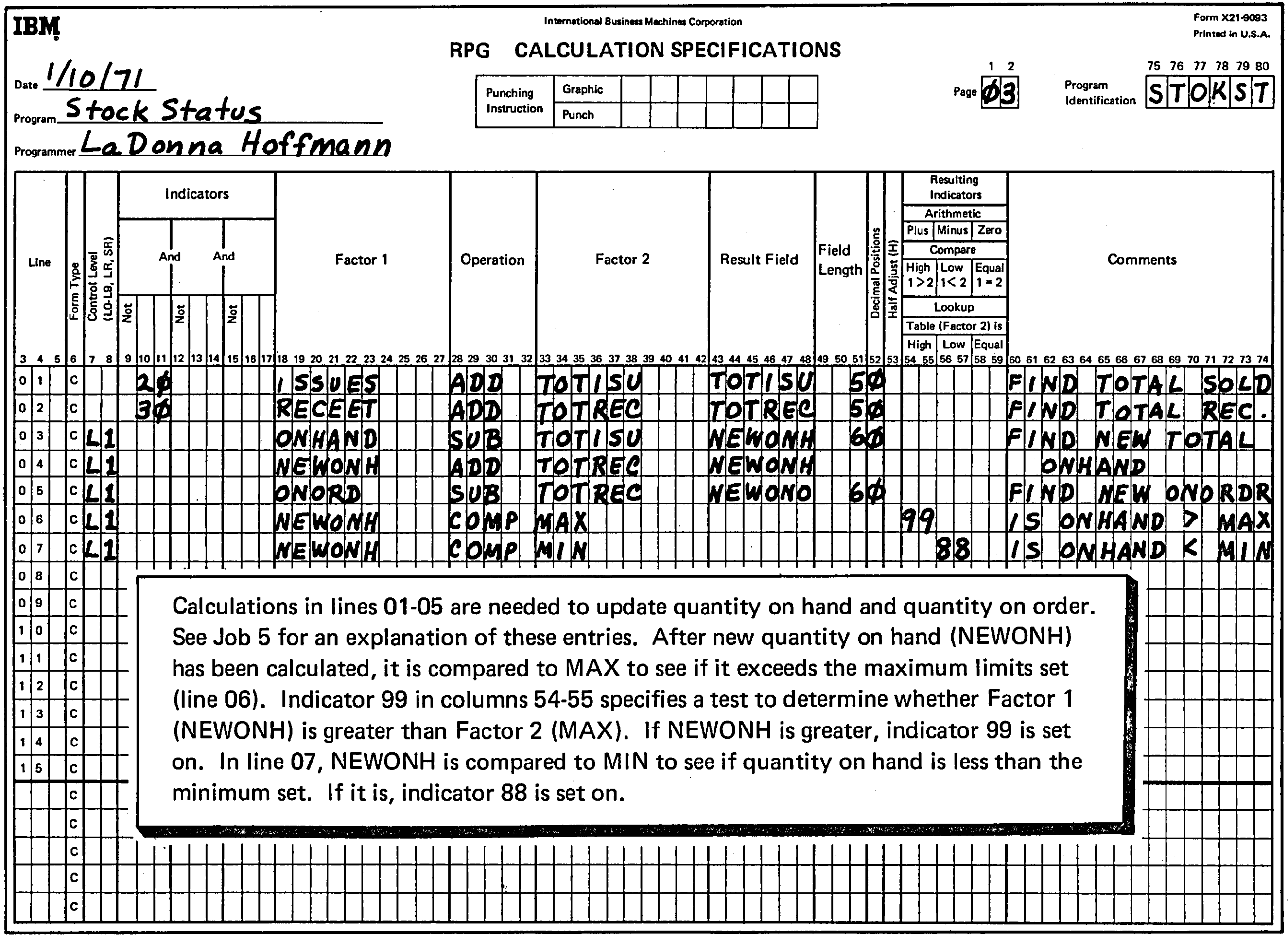
Punching Instruction: Graphic / Punch

Page Ø2

Program Identification STOKST

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator	Position	Not (N)	C/Z/D	Character	From	To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	INPUT	Ø1			1Ø	1		C	M					
02	I										2	7		ITEMNO	L1
03	I										8	29		DESC	
04	I										35	39	Ø	ONHAND	
05	I										4Ø	44	Ø	ONORD	
06	I										45	48	Ø	MAX	
07	I										49	52	Ø	MIN	
08	I		Ø2	N	O	2Ø	1		C	I					
09	I										2	7		ITEMNO	L1
10	I										8	12	Ø	ISSUES	
11	I		Ø3	N	O	3Ø	1		C	R					
12	I										2	7		ITEMNO	L1
13	I										8	12	Ø	RECEET	
14	I														
15	I														
	I														

No new entries are used on the Input sheet.



IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date 1/10/71

Program Stock Status

Programmer LaDonna Hoffmann

Punching Instruction: Graphic / Punch

Page Ø3

Program Identification STOKST

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	High 1>2	Low 1<2	Equal 1=2	Comments
01	C		2Ø	ISSUES	ADD	TOTISU	TOTISU	5	Ø					FIND TOTAL SOLD
02	C		3Ø	RECEET	ADD	TOTREC	TOTREC	5	Ø					FIND TOTAL REC.
03	C	L1		ONHAND	SUB	TOTISU	NEWONH	6	Ø					FIND NEW TOTAL
04	C	L1		NEWONH	ADD	TOTREC	NEWONH							ONHAND
05	C	L1		ONORD	SUB	TOTREC	NEWONO	6	Ø					FIND NEW ONORDR
06	C	L1		NEWONH	COMP	MAX					99			IS ONHAND > MAX
07	C	L1		NEWONH	COMP	MIN						88		IS ONHAND < MIN
08	C													
09	C													
10	C													
11	C													
12	C													
13	C													
14	C													
15	C													

Calculations in lines 01-05 are needed to update quantity on hand and quantity on order. See Job 5 for an explanation of these entries. After new quantity on hand (NEWONH) has been calculated, it is compared to MAX to see if it exceeds the maximum limits set (line 06). Indicator 99 in columns 54-55 specifies a test to determine whether Factor 1 (NEWONH) is greater than Factor 2 (MAX). If NEWONH is greater, indicator 99 is set on. In line 07, NEWONH is compared to MIN to see if quantity on hand is less than the minimum set. If it is, indicator 88 is set on.

IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71
Program Stock Status
Programmer LaDonna Hoffmann

Punching Instruction	Graphic	
	Punch	

Page (1 2) Ø4   Program Identification (75 76 77 78 79 80) STOKST

Edit Codes					
Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
01	O	OUTPUT	H			3	Ø6		1P							
02	O		OR						OV							
03	O												59		'STOCK STATUS REPORT'	
04	O		H			1			1P							
05	O		OR						OV							
06	O												29		'ITEM NO'	
07	O												28		'DESCRIPTION'	
08	O												56		'QUANTITY QUANTITY'	
09	O												71		'TRANSACTION'	
10	O												89		'MIN. MAX.'	
11	O		H			2			1P							
12	O		OR						OV							
13	O												56		'ON HAND ON ORDER'	
14	O												68		'QUANTITY'	
15	O												89		'BAL. BAL.'	

Form X21-9090
Printed in U.S.A.

Page (1 2) Ø5   Program Identification (75 76 77 78 79 80) STOKST

Line	Form Type	Filename	Type	Stacker Select	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After	End Position	P/B	Constant or Edit Word	Sterling Sign Position
01	O		D			1			1Ø							
02	O									ITEMNO			8			
03	O									DESC			34			
04	O									ONHAND	2		44			
05	O									ONORD	2		56			
06	O									MIN	Z		81			
07	O									MAX	Z		88			
08	O		D			1			2Ø							
09	O												17		'ISSUE'	
10	O									ISSUES	2		68			
11	O		D			1			3Ø							
12	O												19		'RECEIPT'	
13	O									RECEET	2		68			
14	O		T		1	2			L1							
15	O									NEWONH	2	B	43			

Form X21-9090
Printed in U.S.A.

Page (1 2) Ø6   Program Identification (75 76 77 78 79 80) STOKST

Line	Form Type	Filename	Type	Stacker Select	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	Blank After	End Position	P/B	Constant or Edit Word	Sterling Sign Position
01	O												45		'**'	
02	O									NEWONO	2	B	56			
03	O												58		'**'	
04	O								88				81		'UNDER'	
05	O								99				89		'OVER'	
06	O															
07	O															
08	O															
09	O															
10	O															
11	O															
12	O															
13	O															
14	O															
15	O															
	O															
	O															
	O															
	O															

Resulting indicators, which indicate whether the new quantity on hand exceeds the maximum or is less than the minimum, are used on the Output-Format sheet to tell when exceptional conditions should be noted. Notice in the total line, the words UNDER and OVER will print only when conditions set by the conditioning indicators are met. UNDER is printed when indicator 88 is on. Remember that 88 is set on when NEWONH is less than the minimum. OVER prints only when 99 is on; that is, when NEWONH is greater than the maximum.

FIELD INDICATORS

Field indicators, like resulting indicators, are used to test the contents of a field and to condition operations based on the results of the test.

Program Cycle Operations

Figure 16 shows the operations in the program cycle associated with field indicators. Note that input fields are tested and field indicators are set to reflect the result of the test at the time data is moved into the processing area. Field indicators are not set off at the end of the program cycle. If a field indicator is set on when data is moved into the processing area in the first cycle, it is not reset until data is moved into the processing area in the second cycle.

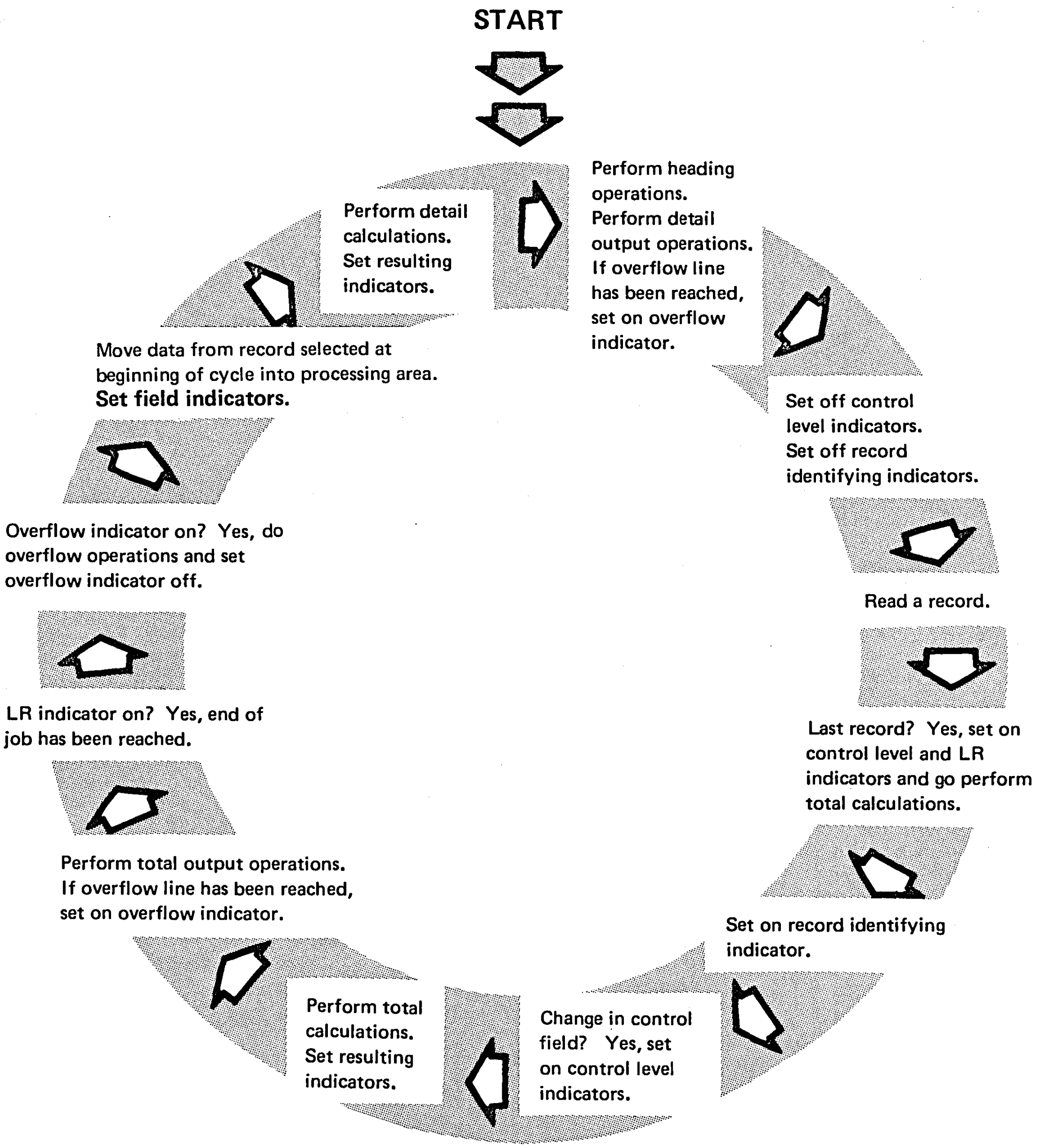
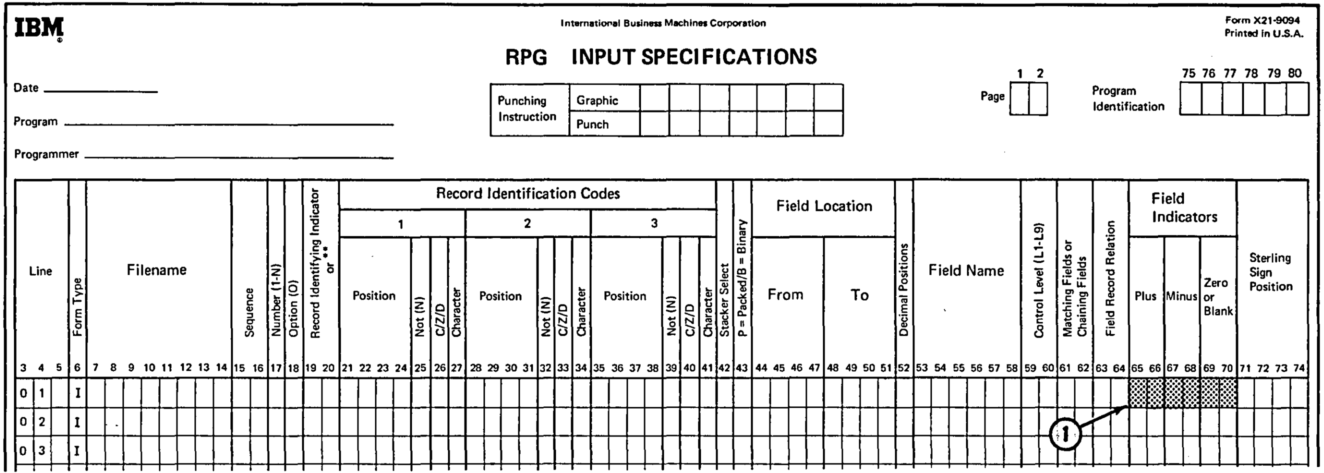

START
Perform heading operations. Perform detail output operations. If overflow line has been reached, set on overflow indicator.
Set off control level indicators. Set off record identifying indicators.
Read a record.
Last record? Yes, set on control level and LR indicators and go perform total calculations.
Set on record identifying indicator.
Change in control field? Yes, set on control level indicators.
Perform total calculations. Set resulting indicators.
Perform total output operations. If overflow line has been reached, set on overflow indicator.
LR indicator on? Yes, end of job has been reached.
Overflow indicator on? Yes, do overflow operations and set overflow indicator off.
Move data from record selected at beginning of cycle into processing area.
Set field indicators.
Perform detail calculations. Set resulting indicators.

Figure 16. Program Cycle Operations for Field Indicators

RPG II Specifications

Make these RPG II specifications when you use field indicators:



IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date
Program
Programmer

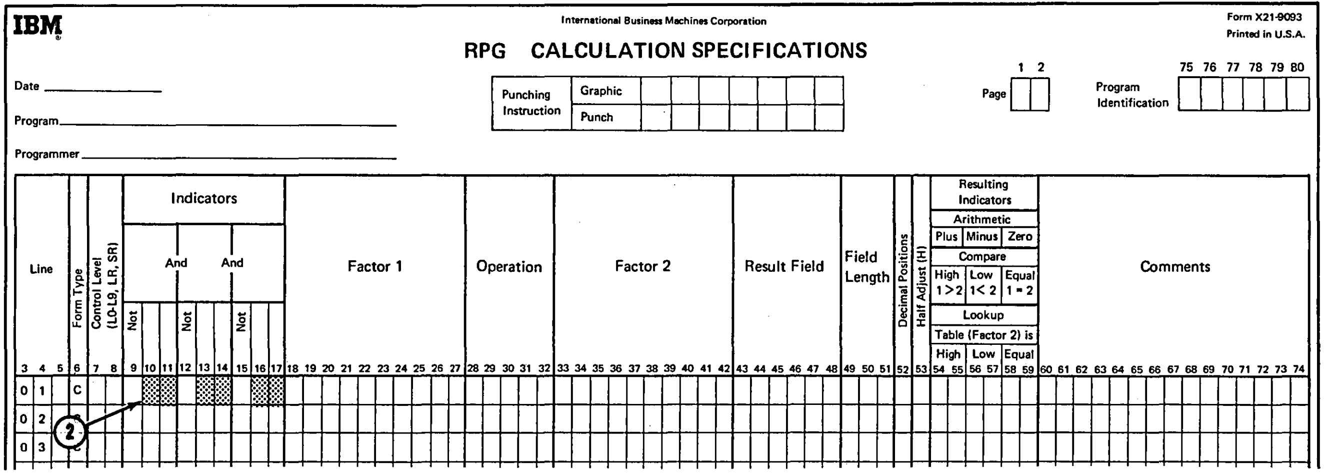
Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator or **	Record Identification Codes												Stacker Select	P = Packed/B = Binary	Field Location		Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
							1				2				3																
							Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character			From	To						Plus	Minus	Zero or Blank	
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
0 1	I																											▨	▨	▨	
0 2	I																														
0 3	I																														

1



IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date
Program
Programmer

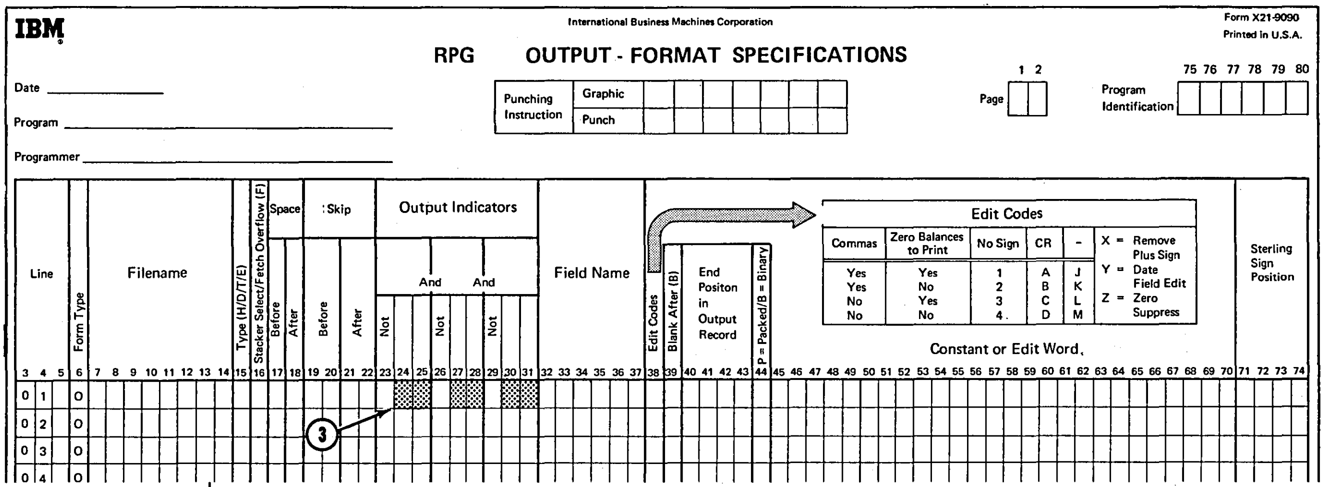
Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
					And		And									Arithmetic Compare Lookup Table (Factor 2) is			
			Not		Not		Not									Plus High 1>2 High	Minus Low 1< 2 Low	Zero Equal 1 = 2 Equal	
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
0 1	C			▨		▨		▨											
0 2	C																		
0 3	C																		

2



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4.	D	M	

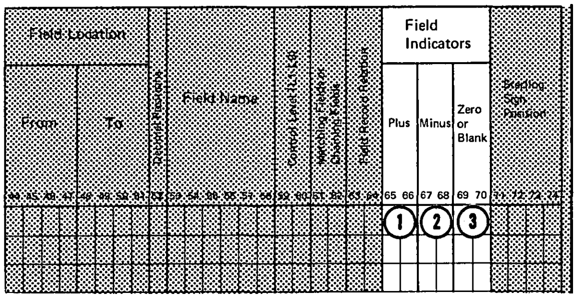
Edit Codes

Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space		Skip		Output Indicators						Field Name	Edit Codes	Blank After (B)	End Positon in Output Record	P = Packed/B = Binary	Constant or Edit Word,	Sterling Sign Position
					Before	After	Before	After			And		And								
									Not		Not		Not								
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
0 1	O									▨		▨		▨							
0 2	O																				
0 3	O																				
0 4	O																				

3

Field indicators are assigned at ①. They may be used to condition calculation operations at ② and output operations at ③ .

You can enter any one of the indicators 01-99 in columns 65-70 of the Input sheet to test an input field. You may assign indicators to test for three possible conditions:



1. A field indicator assigned in columns 65-66 tells the computer to determine if a numeric input field is positive (plus).

2. A field indicator assigned in columns 67-68 tells the computer to determine if a numeric input field is negative (minus).

3. A field indicator assigned in columns 69-70 tells the computer to determine if an alphameric input field is blank or a numeric field is zero.

Job 7: Using Field Indicators To Test Contents Of Input Fields

JOB DEFINITION

Create an Aged-Trial Balance Report that lists:

- Name and customer number of all charge customers who have payments due.
- Amount due.
- Overdue balances.

The customer master file, which contains records for all customers regardless of their balance, is used as the input file. The report is to show only those customers with payments due. Thus, information from customer records that contain a zero or credit balance should not be printed.

JOB REQUIREMENTS

Input: A customer master file consisting of one record type:

Card Name ____________________

97 98 99 100 101 102 103

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 … 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 7

Print Line 1 — Print Line 2

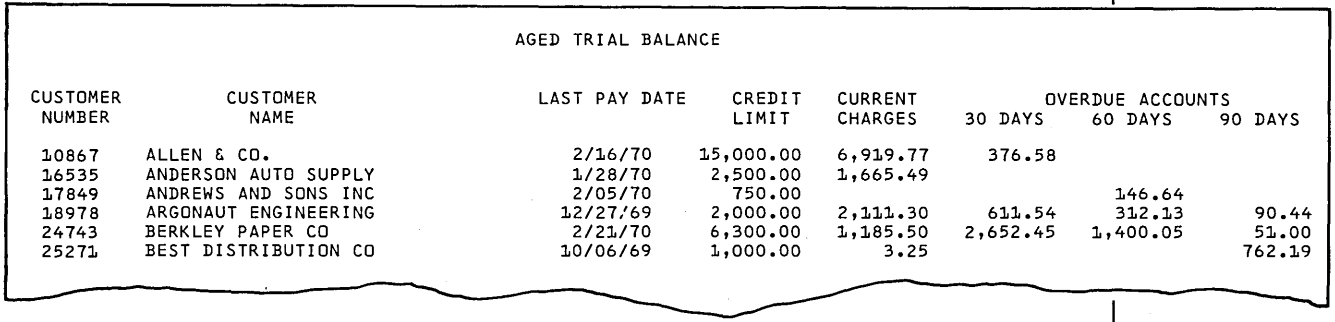
Tier 1 — Tier 2

CODE = M	CUSTOMER NUMBER	CUSTOMER NAME	LAST PAYMENT DATE	CREDIT LIMIT	CURRENT CHARGES	30 DAYS OVERDUE	60 DAYS OVERDUE	90 DAYS OVERDUE

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 2 … 9 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 7

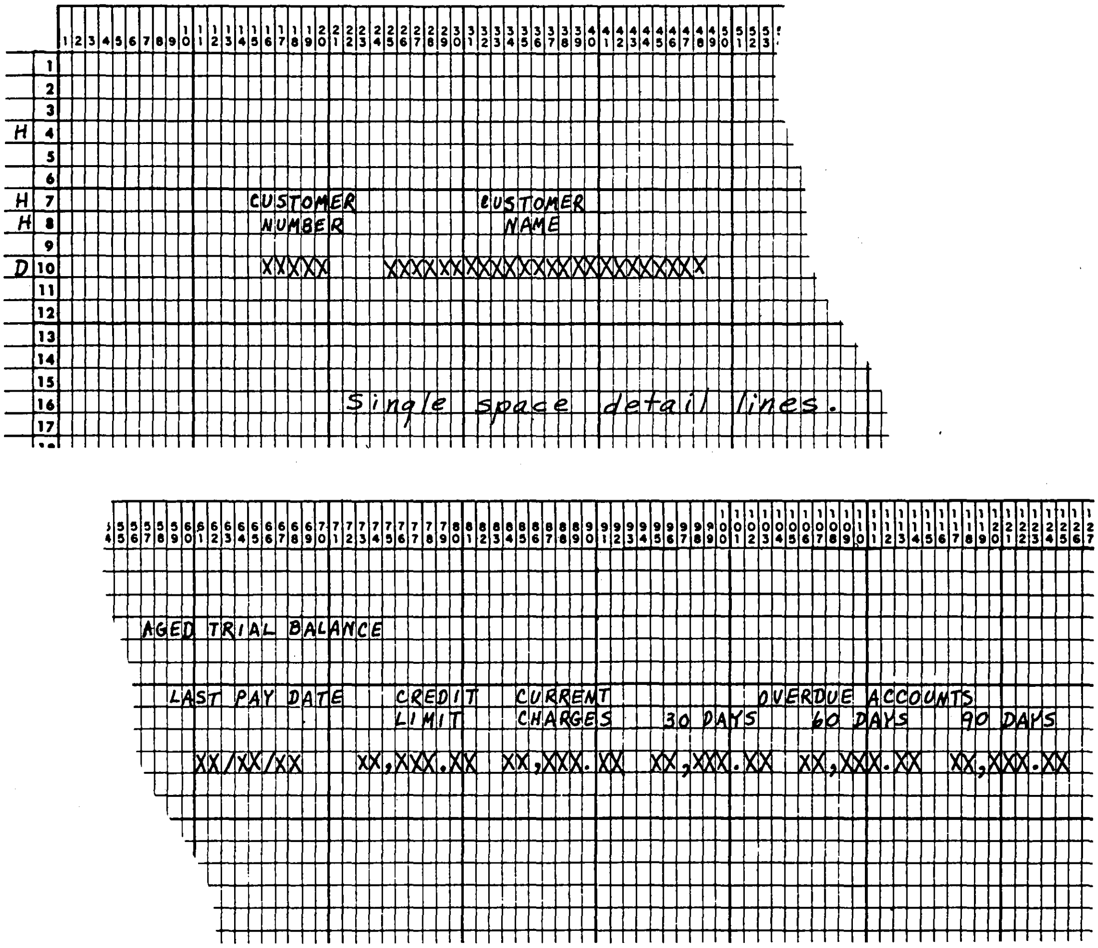
Processing: Check input balance due field for zero or credit balance (use field indicators).

Output: An aged-trial balance report printed on a 132-position printer.

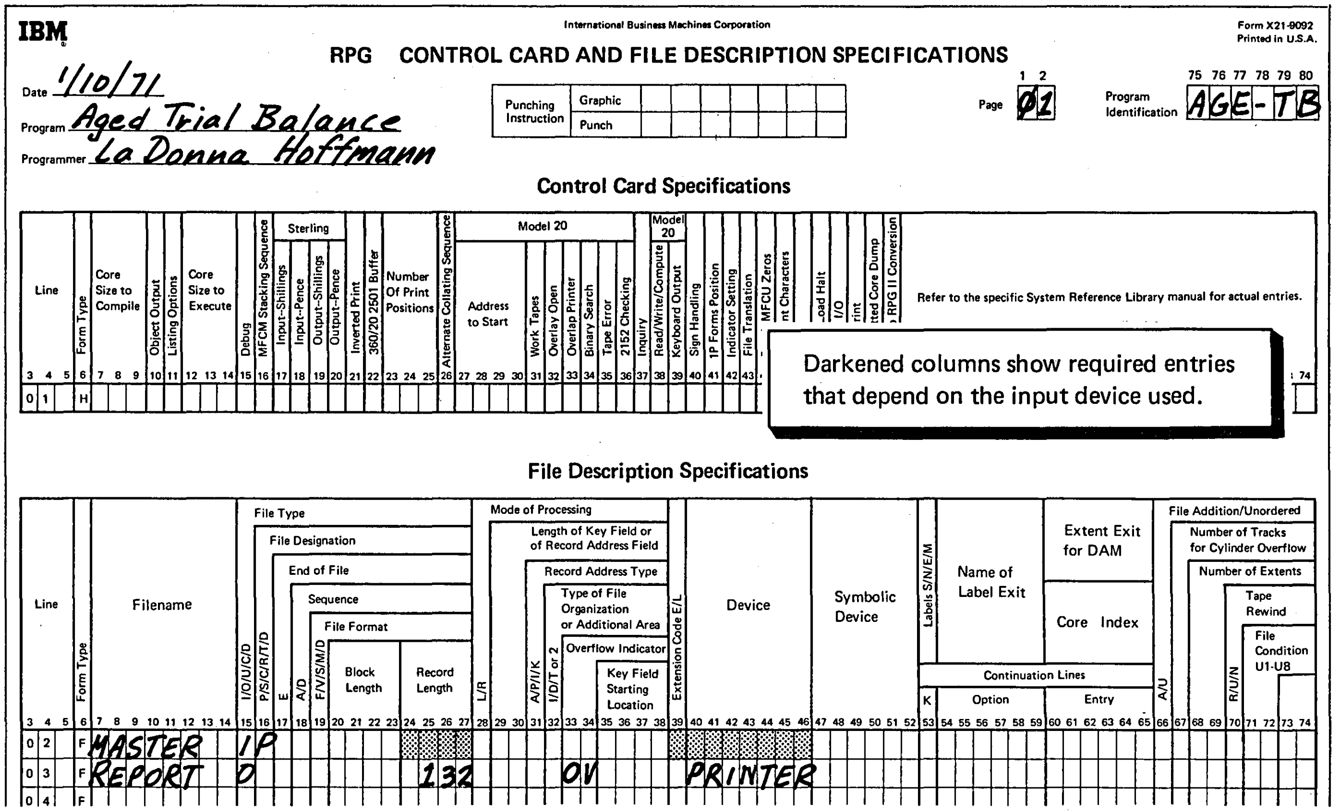


```
                                        AGED TRIAL BALANCE

CUSTOMER          CUSTOMER                         LAST PAY DATE    CREDIT     CURRENT          OVERDUE ACCOUNTS
 NUMBER             NAME                                             LIMIT     CHARGES    30 DAYS    60 DAYS     90 DAYS

 10867    ALLEN & CO.                                    2/16/70   15,000.00   6,919.77    376.58
 16535    ANDERSON AUTO SUPPLY                           1/28/70    2,500.00   1,665.49
 17849    ANDREWS AND SONS INC                           2/05/70      750.00                          146.64
 18978    ARGONAUT ENGINEERING                          12/27/69    2,000.00   2,111.30    611.54     312.13       90.44
 24743    BERKLEY PAPER CO                               2/21/70    6,300.00   1,185.50  2,652.45   1,400.05       51.00
 25271    BEST DISTRIBUTION CO                          10/06/69    1,000.00       3.25                            762.19
```

This Printer Spacing Chart shows how the report is formatted:



JOB SPECIFICATIONS



IBM

International Business Machines Corporation

Form X21-9092
Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71

Program Aged Trial Balance

Programmer La Donna Hoffmann

Punching Instruction: Graphic / Punch

Page (1 2) ϕ1

Program Identification (75–80) AGE-TB

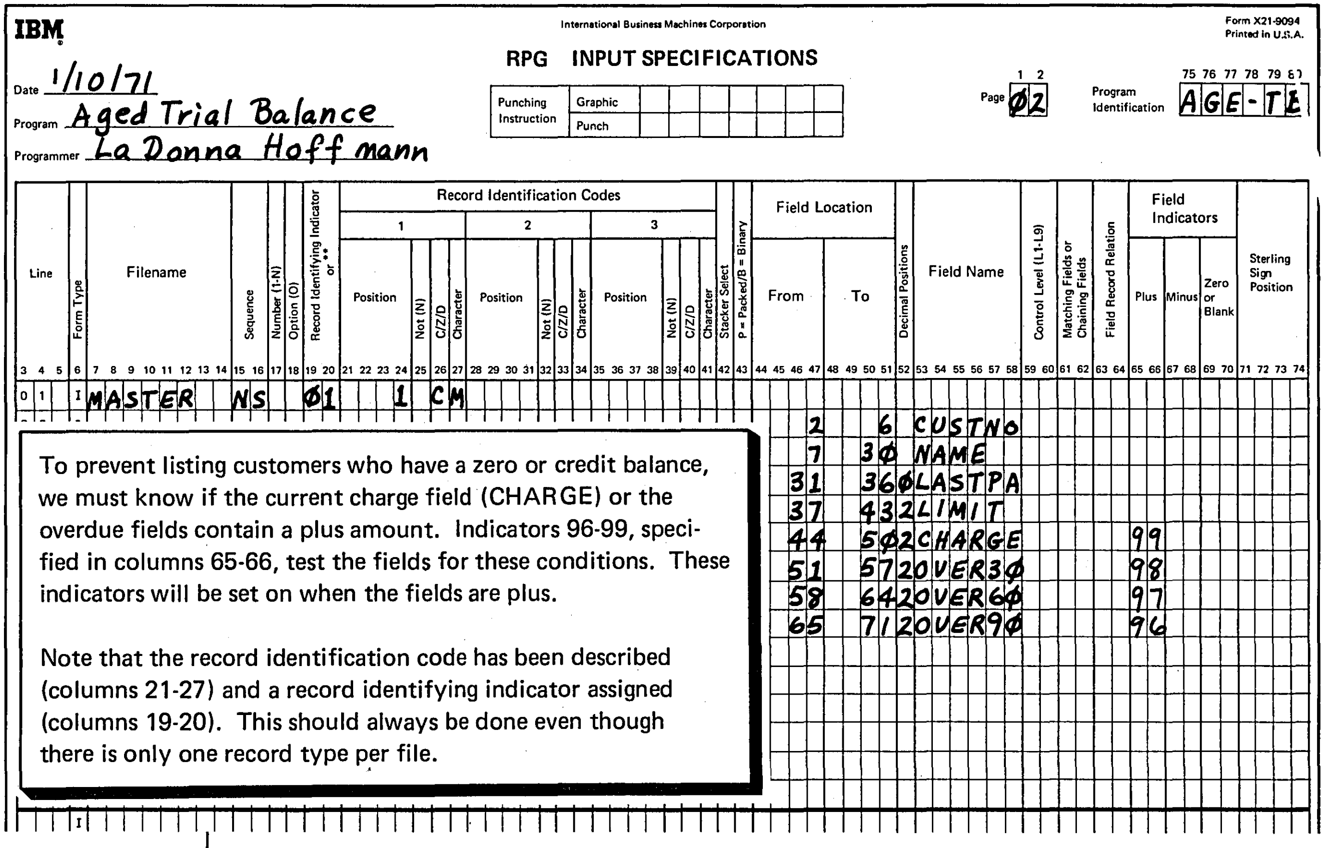
Control Card Specifications

Line (3-5)	Form Type (6)	Core Size to Compile (7-9)	Debug …	Refer to the specific System Reference Library manual for actual entries.
0 1	H			

Darkened columns show required entries that depend on the input device used.

File Description Specifications

Line (3-5)	Form Type (6)	Filename (7-14)	File Type (15)	File Designation (16)	Record Length (24-27)	Overflow Indicator (33-34)	Device (40-46)
0 2	F	MASTER	I	P			
0 3	F	REPORT	O		132	OV	PRINTER
0 4	F						



IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date 1/10/71

Program Aged Trial Balance

Programmer La Donna Hoffmann

Punching Instruction: Graphic / Punch

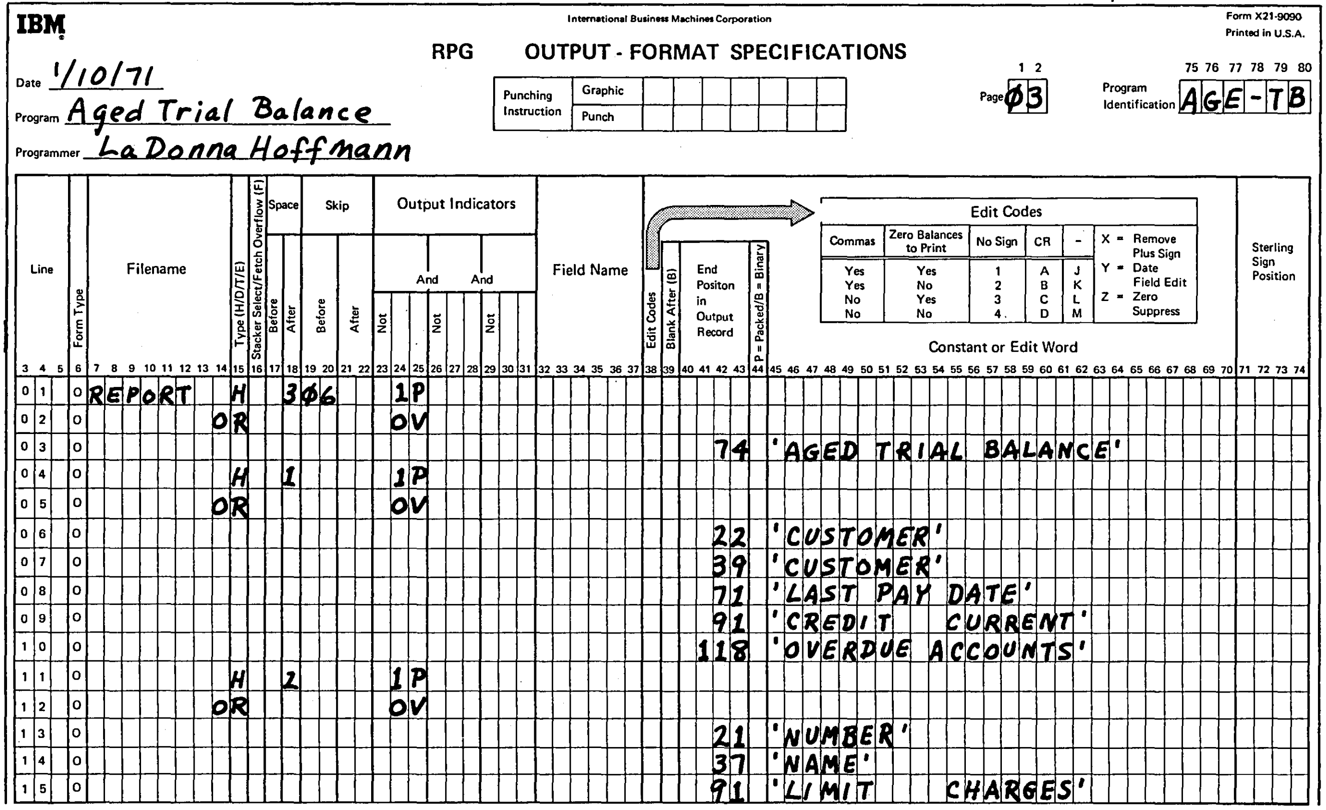
Page (1 2) ϕ2

Program Identification (75–80) AGE-TB

Line (3-5)	Form Type (6)	Filename (7-14)	Sequence (15-16)	Record Identifying Indicator (19-20)	Position (21-24)	C/Z/D (26)	Character (27)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)	Field Indicators Plus (65-66)
0 1	I	MASTER	NS	ϕ1	1	C	M					
								2	6		CUSTNO	
								7	3ϕ		NAME	
								31	36	ϕ	LASTPA	
								37	43	2	LIMIT	
								44	5ϕ	2	CHARGE	99
								51	57	2	OVER3ϕ	98
								58	64	2	OVER6ϕ	97
								65	71	2	OVER9ϕ	96

To prevent listing customers who have a zero or credit balance, we must know if the current charge field (CHARGE) or the overdue fields contain a plus amount. Indicators 96-99, specified in columns 65-66, test the fields for these conditions. These indicators will be set on when the fields are plus.

Note that the record identification code has been described (columns 21-27) and a record identifying indicator assigned (columns 19-20). This should always be done even though there is only one record type per file.



IBM — International Business Machines Corporation — Form X21-9090, Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

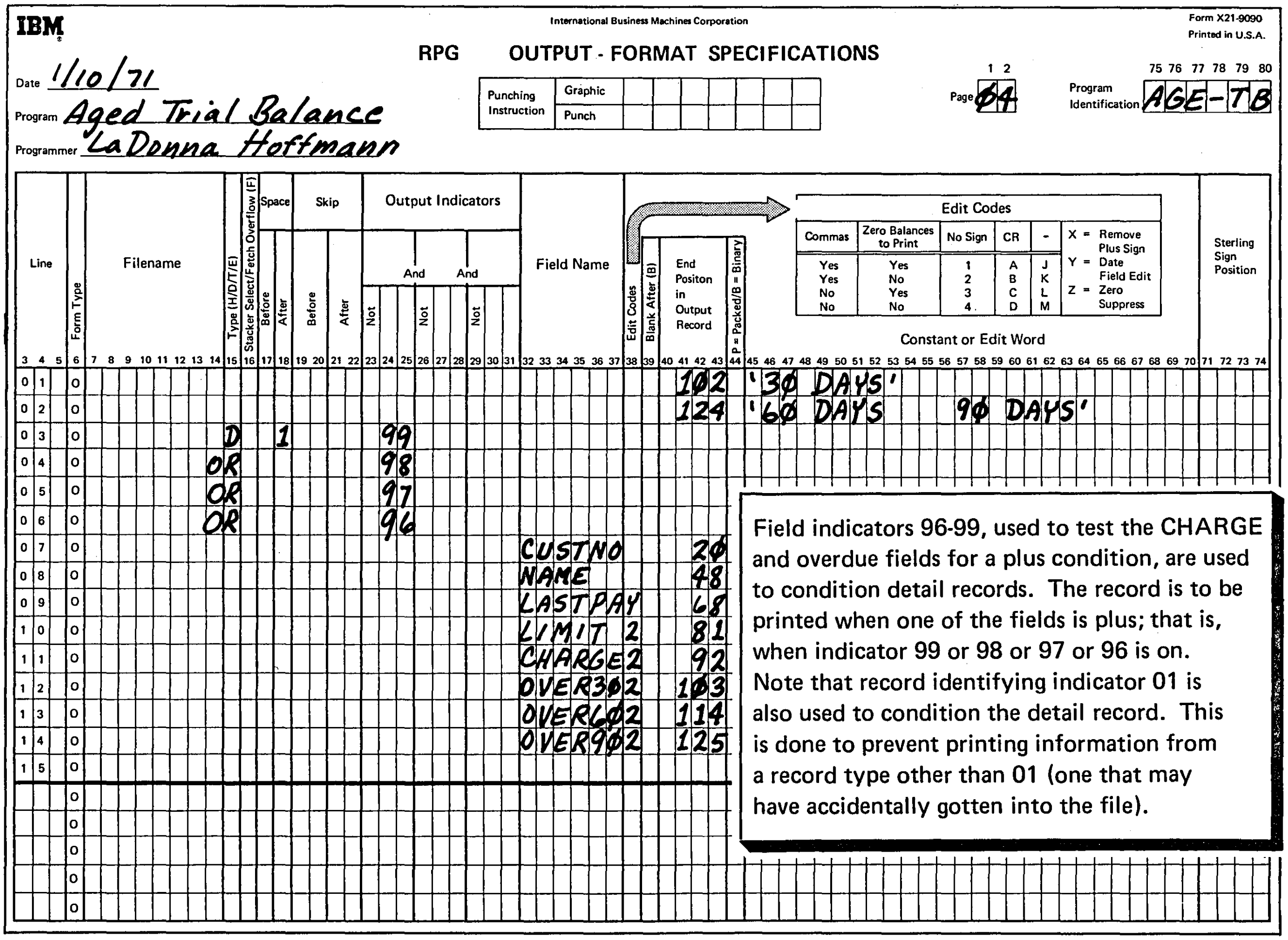
Date 1/10/71
Program Aged Trial Balance
Programmer LaDonna Hoffmann

Page Ø3 — Program Identification AGE-TB

Edit Codes	Commas	Zero Balances to Print	No Sign	CR	-
	Yes	Yes	1	A	J
	Yes	No	2	B	K
	No	Yes	3	C	L
	No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space After	Skip Before	Output Indicators	Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word
01	O	REPORT	H	3	Ø6	1P				
02	O		OR			OV				
03	O								74	'AGED TRIAL BALANCE'
04	O		H	1		1P				
05	O		OR			OV				
06	O								22	'CUSTOMER'
07	O								39	'CUSTOMER'
08	O								71	'LAST PAY DATE'
09	O								91	'CREDIT CURRENT'
10	O								118	'OVERDUE ACCOUNTS'
11	O		H	2		1P				
12	O		OR			OV				
13	O								21	'NUMBER'
14	O								37	'NAME'
15	O								91	'LIMIT CHARGES'



IBM — International Business Machines Corporation — Form X21-9090, Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71
Program Aged Trial Balance
Programmer LaDonna Hoffmann

Page Ø4 — Program Identification AGE-TB

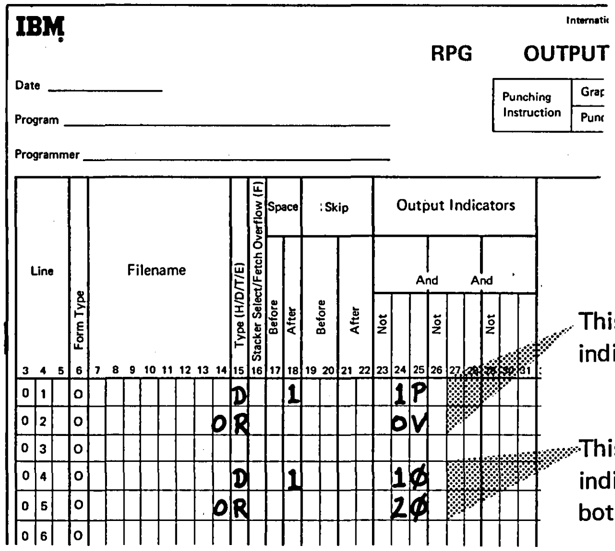
Line	Form Type	Filename	Type (H/D/T/E)	Space After	Skip Before	Output Indicators	Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word
01	O								102	'3Ø DAYS'
02	O								124	'6Ø DAYS 9Ø DAYS'
03	O		D	1		99				
04	O		OR			98				
05	O		OR			97				
06	O		OR			96				
07	O						CUSTNO		2Ø	
08	O						NAME		48	
09	O						LASTPAY		68	
10	O						LIMIT	2	81	
11	O						CHARGE	2	92	
12	O						OVER3Ø	2	1Ø3	
13	O						OVER6Ø	2	114	
14	O						OVER9Ø	2	125	
15	O									

Field indicators 96-99, used to test the CHARGE and overdue fields for a plus condition, are used to condition detail records. The record is to be printed when one of the fields is plus; that is, when indicator 99 or 98 or 97 or 96 is on. Note that record identifying indicator 01 is also used to condition the detail record. This is done to prevent printing information from a record type other than 01 (one that may have accidentally gotten into the file).

CONDITIONING OPERATIONS BY MORE THAN ONE INDICATOR

In this chapter, you have learned about many different kinds of conditioning indicators: control level, first page, overflow, last record, record identifying, resulting, and field indicators. In many jobs you will use two or more conditioning indicators. Indicators used together can be in either an OR or AND relationship.

You have already read about indicators in an OR relationship. You learned that if an operation can be done when either one of two conditions or both conditions exist, you can specify the conditioning indicators like this:



IBM

RPG OUTPUT

Date

Program

Programmer

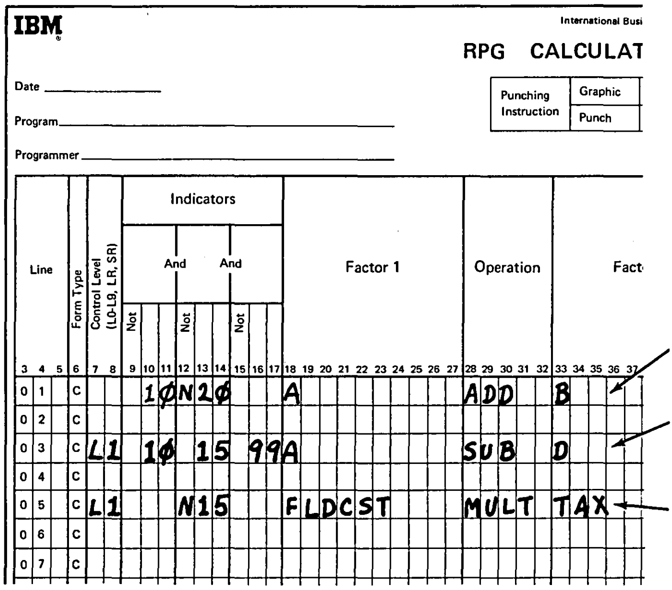
Line	Form Type	Filename	Type (H/D/T/E)	Space After	Output Indicators
01	O		D	1	1P
02	O		OR		OV
03	O				
04	O		D	1	1Ø
05	O		OR		2Ø
06	O				

This line is written when either indicator 1P or OV is on.

This line is written when either indicator 10 or 20 is on, or when both are on.

In some systems, conditioning indicators can be used in the OR relationship on both Calculation and Output-Format sheets. In others, you can specify the conditioning indicator using the word OR only on the Output-Format sheet.

If you specify two or more conditioning indicators on one line, they are in an AND relationship. The AND relationship means that all conditions must be satisfied before the operation will be performed:



IBM

International Busi

RPG CALCULAT

Date

Program

Programmer

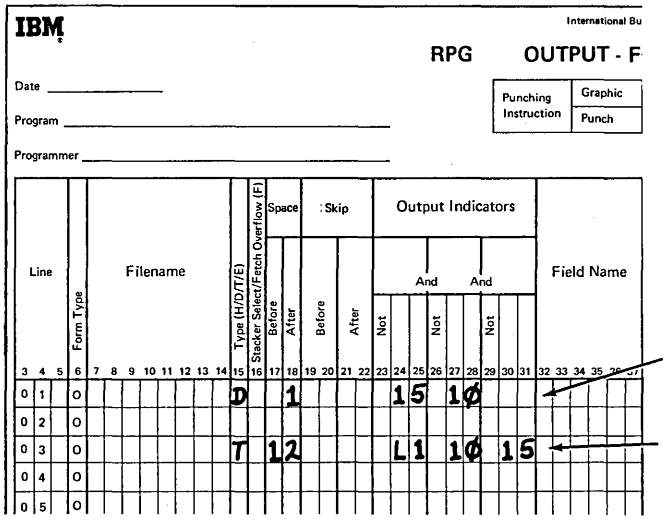
Punching Instruction	Graphic
	Punch

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators Not		Indicators And Not		Indicators And Not		Factor 1	Operation	Fact
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
0 1	C			1Ø	N	2Ø			A	ADD	B
0 2	C										
0 3	C	L1		1Ø		15		99	A	SUB	D
0 4	C										
0 5	C	L1			N	15			FLDCST	MULT	TAX
0 6	C										
0 7	C										

This operation is performed when 10 is on and 20 is not on.

This operation is performed when L1, 10, 15, and 99 are all on.

This operation is performed when L1 is on and 15 is not on.



IBM

International Bu

RPG OUTPUT - F

Date

Program

Programmer

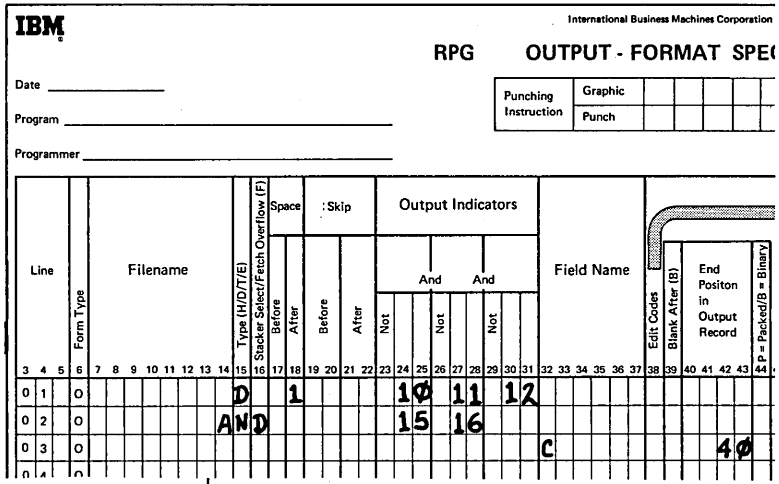
Punching Instruction	Graphic
	Punch

Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators Not		Output Indicators And Not		Output Indicators And Not		Field Name
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
0 1	O		D			1				15		1Ø			
0 2	O														
0 3	O		T		1	2				L1		1Ø		15	
0 4	O														
0 5	O														

This detail line is written when 15 and 10 are both on.

This total line is written when L1, 10, and 15 are all on.

If your calculation or output operation must be conditioned by more than three indicators, additional indicators can be specified on the next line if AND is entered in columns 14, 15, and 16 of the Output-Format sheet or AN is entered in columns 7-8 of the Calculation Sheet:



IBM

International Business Machines Corporation

RPG OUTPUT - FORMAT SPE(

Date

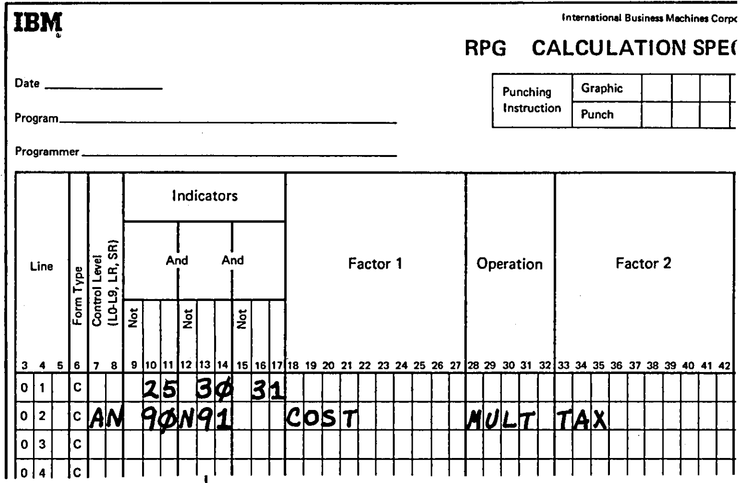
Program

Programmer

Punching Instruction	Graphic				
	Punch				

Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators Not		And Not		And Not		Field Name	Edit Codes	Blank After (B)	End Positon in Output Record	P = Packed/B = Binary
0 1	O		D			1				1Ø		11		12					
0 2	O	A N	D							15		16							
0 3	O														C			4Ø	
0 4	O																		

Five indicators used in an AND relationship condition this detail record. Three indicators are specified on one line; the remaining are specified on the following line with the word AND in columns 14-16. Indicators 10, 11, 12, 15, and 16 must all be on before the detail line will be printed.



IBM

International Business Machines Corp

RPG CALCULATION SPE(

Date

Program

Programmer

Punching Instruction	Graphic			
	Punch			

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators Not		And Not		And Not		Factor 1	Operation	Factor 2
0 1	C			25		3Ø		31			
0 2	C	AN		9Ø	N	91			COST	MULT	TAX
0 3	C										
0 4	C										

This calculation operation is done only if indicators 25, 30, 31, and 90 are on and 91 is not on. Note that the operation is specified on the AN line.

Some systems allow the use of AN on the Calculation sheet; others do not.

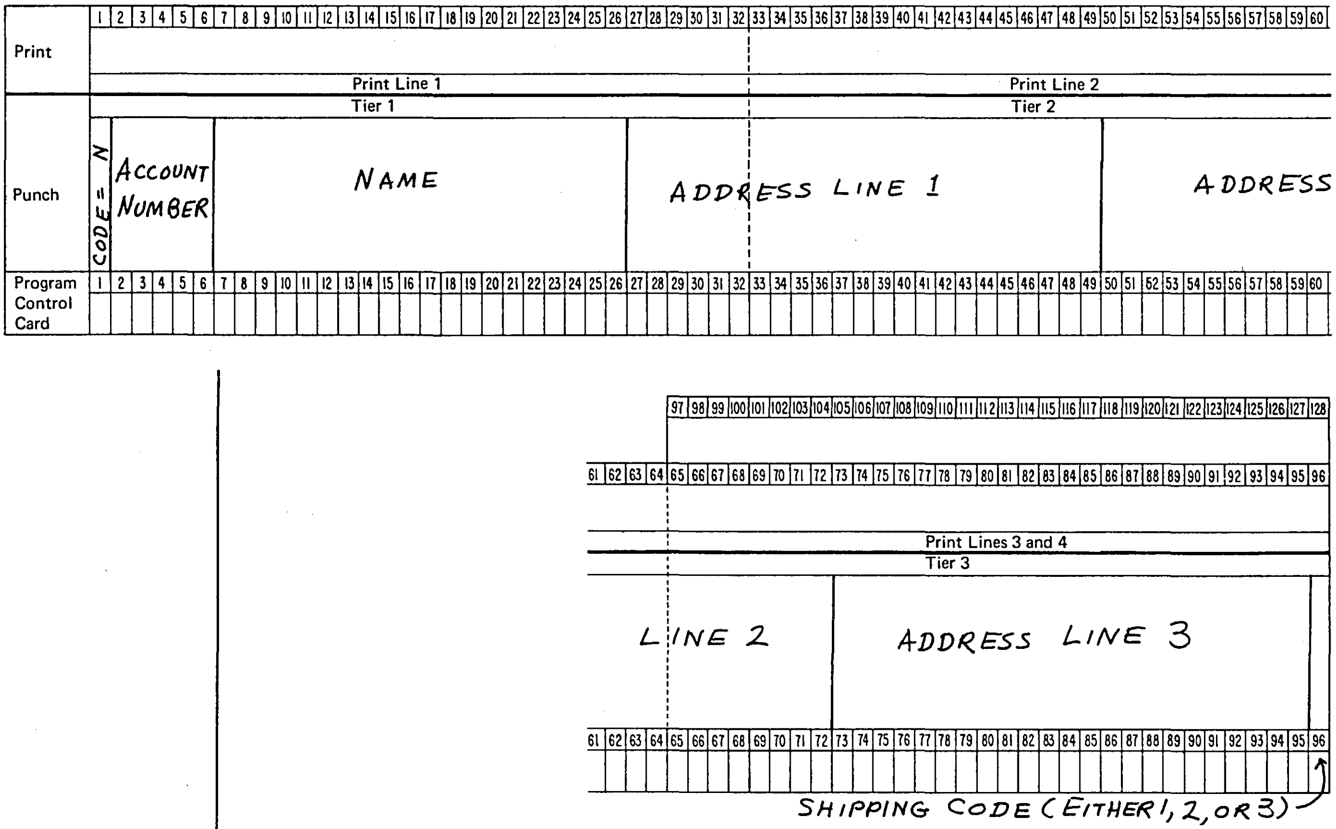
- Analyze the job.
- Determine how the job can be done in RPG II.
- Write the RPG II specifications.
- Prepare for compilation by completing the specifications sheets, desk checking them for accuracy, and having them punched into cards.
- Compile the source program. Be certain that your source program is free of errors by checking the listing.
- Execute the job.

DETERMINE THE JOB REQUIREMENTS

Assume that you are told the following things about a job:

- An invoice is to be prepared like that shown in Figure 17.
- The input file contains two types of records: name/address records for all customers who made purchases on credit during the month and transaction records for each item purchased by the customers during the month. The name/address and transaction records look like this:

Card Name NAME/ADDRESS RECORD



Print

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

Print Line 1 Print Line 2

Tier 1 Tier 2

Punch

CODE = N

ACCOUNT NUMBER

NAME

ADDRESS LINE 1

ADDRESS LINE 2

Program Control Card

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

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

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

Print Lines 3 and 4

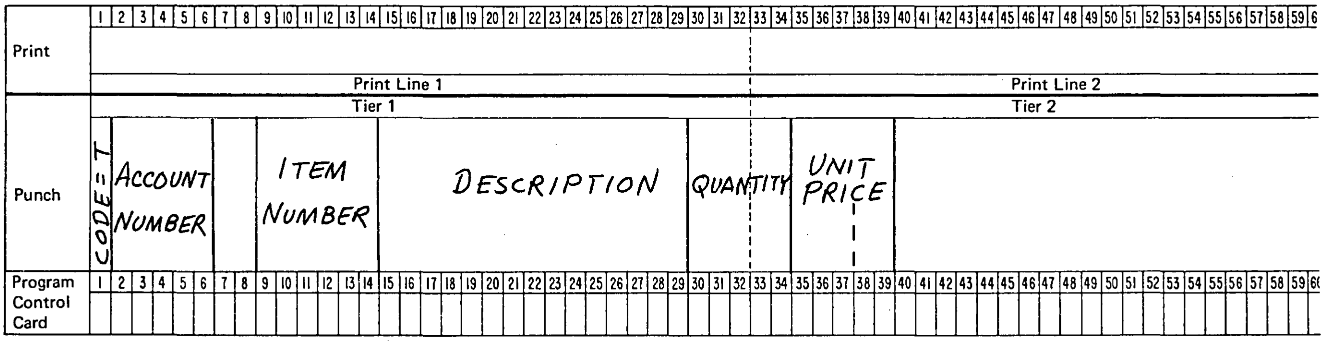
Tier 3

ADDRESS LINE 3

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

SHIPPING CODE (EITHER 1, 2, OR 3)

Card Name TRANSACTION RECORD



Print

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 6

Print Line 1 Print Line 2

Tier 1 Tier 2

Punch

CODE = T

ACCOUNT NUMBER

ITEM NUMBER

DESCRIPTION

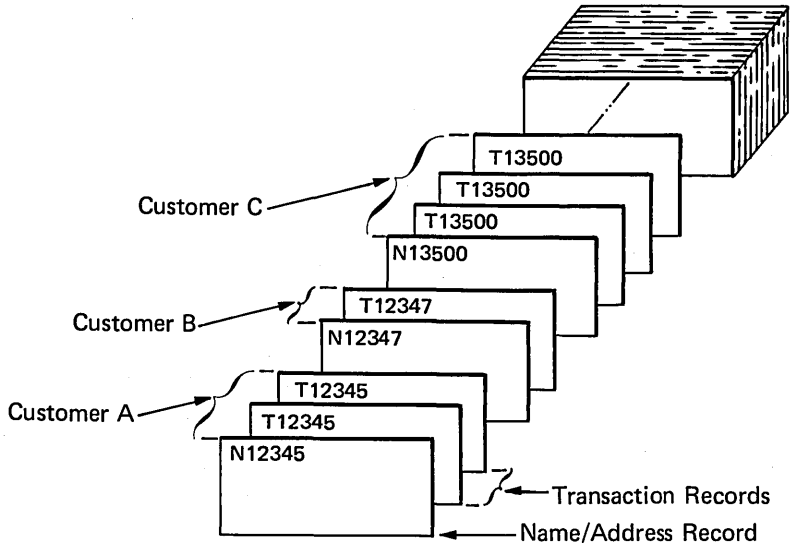
QUANTITY

UNIT PRICE

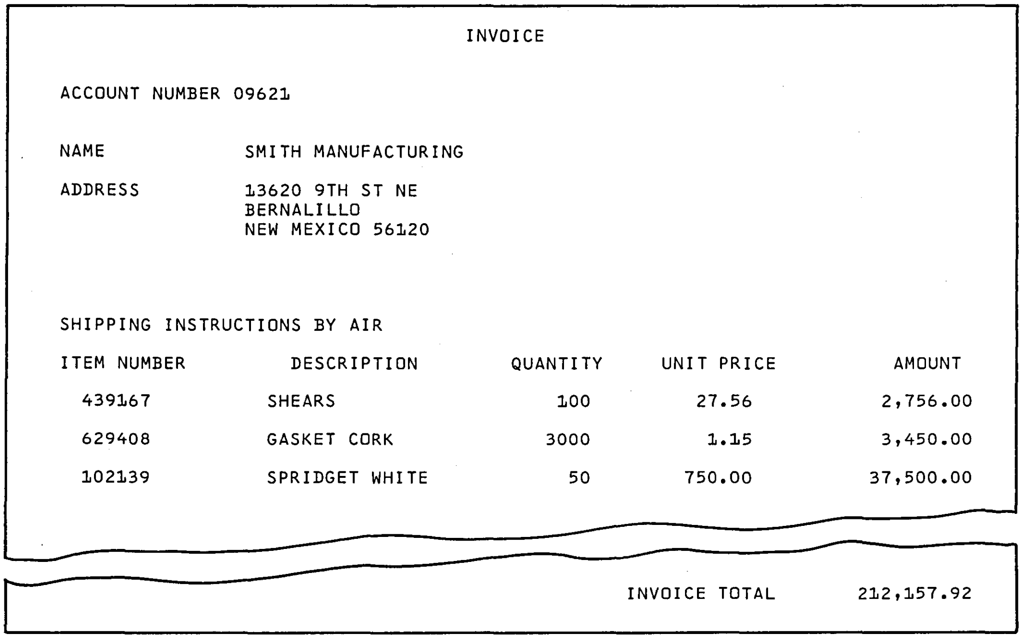
Program Control Card

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

- The input file is organized so that all transaction records for a customer follow the customer's name/address record:



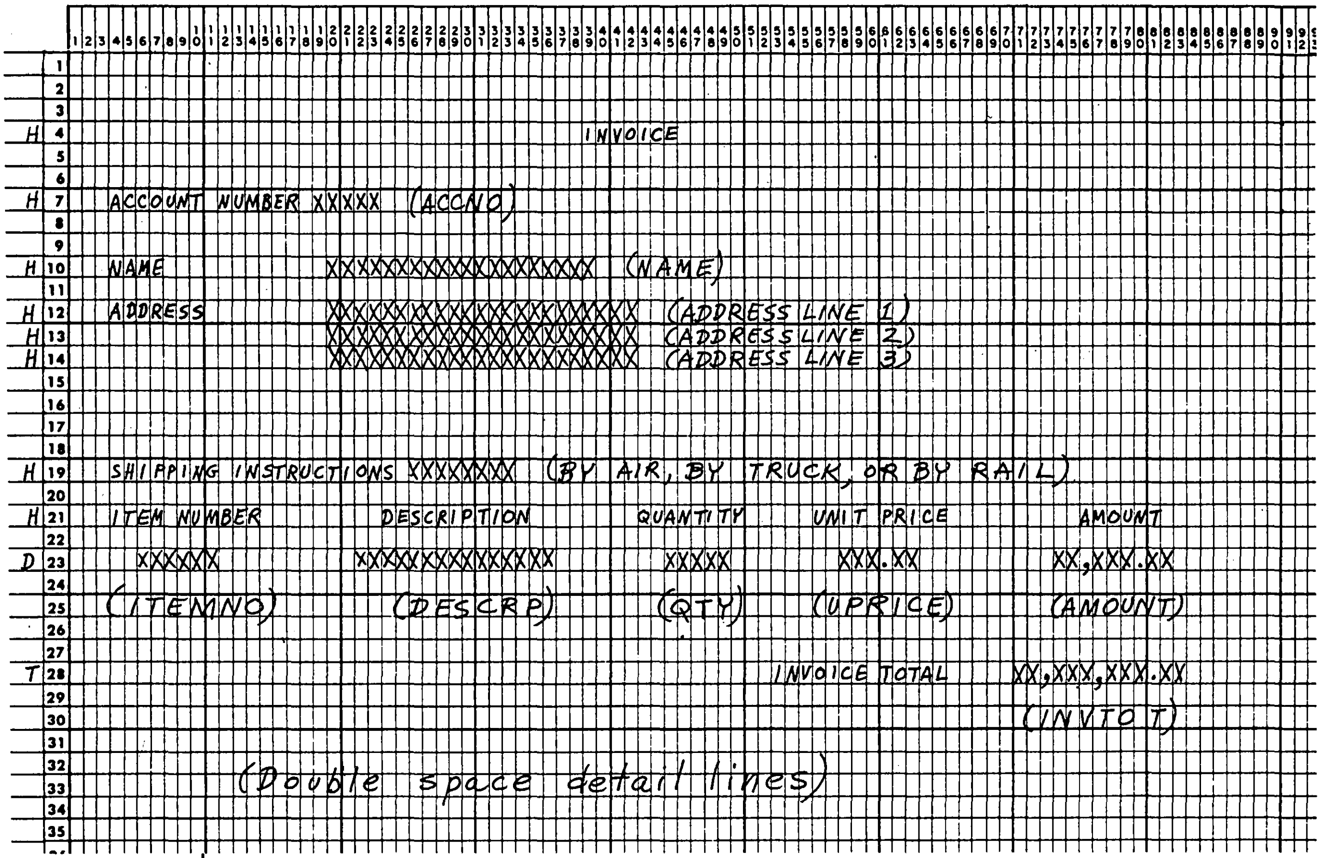
- There will always be one name/address record for each customer, but there may be one or more transaction records per customer.



```
                                   INVOICE

  ACCOUNT NUMBER 09621

  NAME              SMITH MANUFACTURING

  ADDRESS           13620 9TH ST NE
                    BERNALILLO
                    NEW MEXICO 56120


  SHIPPING INSTRUCTIONS BY AIR

  ITEM NUMBER          DESCRIPTION         QUANTITY     UNIT PRICE           AMOUNT

    439167            SHEARS                   100          27.56          2,756.00

    629408            GASKET CORK             3000           1.15          3,450.00

    102139            SPRIDGET WHITE            50         750.00         37,500.00



                                                    INVOICE TOTAL        212,157.92
```

Figure 17. Sample Invoice

- Standard computer paper is to be used. Invoices should look like this:



```
     1
     2
     3
H  4                                       INVOICE
   5
   6
H  7    ACCOUNT NUMBER XXXXX  (ACCNO)
   8
   9
H 10    NAME            XXXXXXXXXXXXXXXXXXXX  (NAME)
  11
H 12    ADDRESS         XXXXXXXXXXXXXXXXXXXXXXX  (ADDRESS LINE 1)
H 13                    XXXXXXXXXXXXXXXXXXXXXXX  (ADDRESSLINE 2)
H 14                    XXXXXXXXXXXXXXXXXXXXXXX  (ADDRESS LINE 3)
  15
  16
  17
  18
H 19    SHIPPING INSTRUCTIONS XXXXXXX   (BY AIR, BY TRUCK, OR BY RAIL)
  20
H 21    ITEM NUMBER         DESCRIPTION        QUANTITY     UNIT PRICE          AMOUNT
  22
D 23      XXXXXX          XXXXXXXXXXXXXXX        XXXXX        XXX.XX         XX,XXX.XX
  24
  25    (ITEMNO)             (DESCRP)           (QTY)       (UPRICE)         (AMOUNT)
  26
  27
T 28                                                    INVOICE TOTAL    XX,XXX,XXX.XX
  29
  30                                                                      (INVTOT)
  31
  32
  33             (Double space detail lines)
  34
  35
```

Your first step is to analyze the problem and decide what processing must be done to get the desired results. Always keep in mind how things are done using RPG II. In your analysis of the job, you would probably think of these points:

- Information for the first part of the invoice is taken from the name/address record; information for the second part (list of transactions) is taken from transaction records.

- In order to print shipping instructions, the shipping code recorded on the record must be determined:

  1 = By truck
  2 = By rail
  3 = By air

- AMOUNT and INVTOT (invoice total) must be calculated because this information is not on input records. These calculations must be done for all transaction records:

  QTY x PRICE = AMOUNT

  AMOUNT + INVTOT = INVTOT

- INVTOT should be printed only after all transaction records for one account have been processed.

- The invoice for each customer must be on a separate page. This means that forms must advance each time a new customer name/address record is found. It is possible that one customer has purchased so many items that they cannot be listed on one page. In this case, forms should advance when the end of a page is reached. When an invoice includes more than one page, headings should be printed on all pages.

DETERMINE RPG II PROGRAM CYCLE OPERATIONS

After you have carefully analyzed the job, determine what RPG II specifications and program cycle operations you'll need. For example, consider the following:

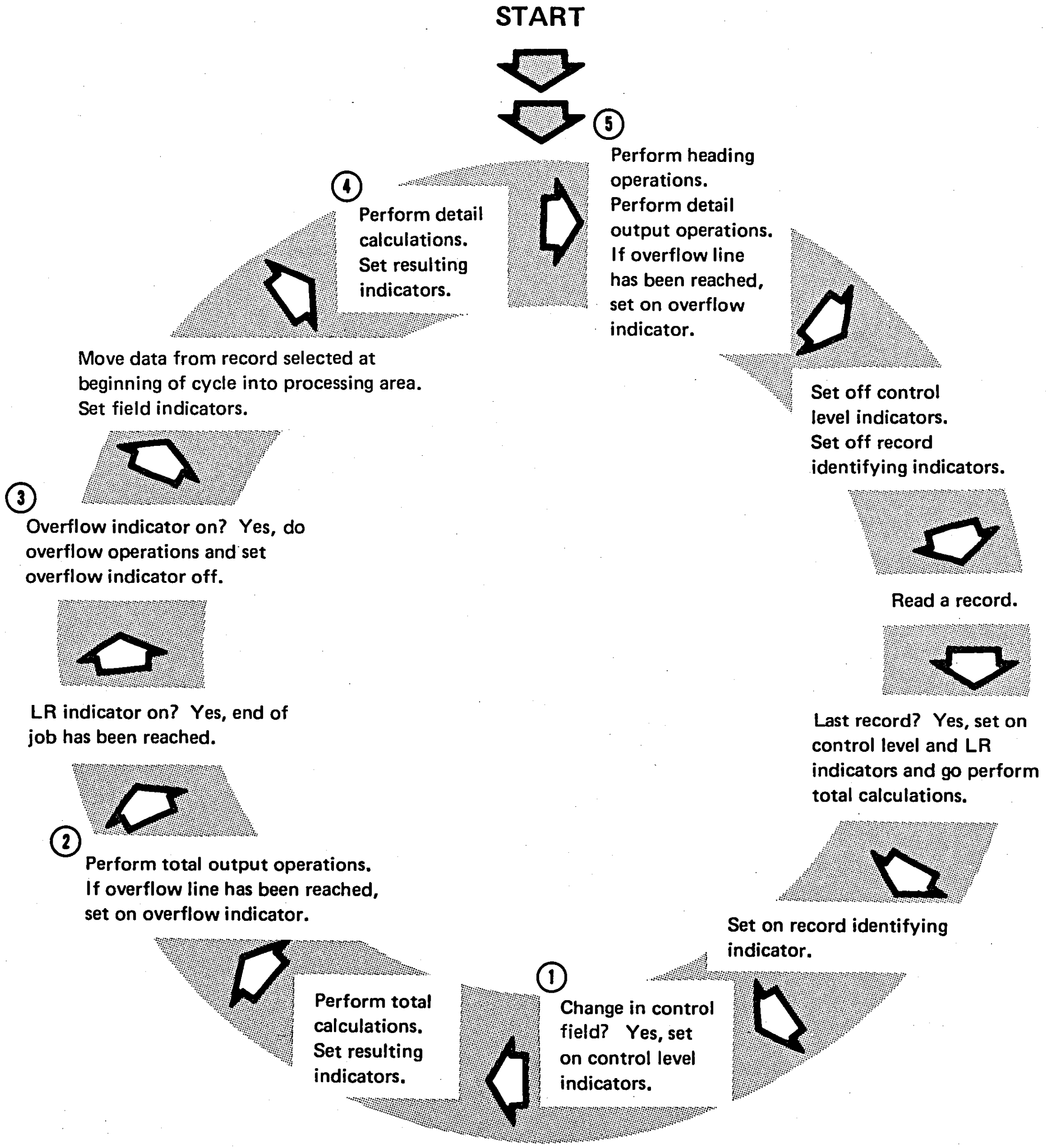
- Different record types are used. This means that record identifying indicators must be specified to tell what to do for each record.

- The transaction code must be determined. One way to do this is to compare transaction code to 2. Through the use of resulting indicators, you can determine if the code is equal to, less than, or greater than 2.

- INVAMT is printed only after all transaction records for one account have been processed. This is a total operation, done only after a group of records has been processed. Therefore, control fields and control level indicators must be used to do a total operation. The account number field can be used as the control field.

- Forms should advance each time a different name/address record is encountered or whenever overflow occurs. Thus heading lines must be conditioned by a record identifying indicator and the OV indicator.

If the indicators and steps just listed are used, the RPG II program cycle would include the steps shown in Figure 18.


START
5
Perform heading operations. Perform detail output operations. If overflow line has been reached, set on overflow indicator.
Set off control level indicators. Set off record identifying indicators.
Read a record.
Last record? Yes, set on control level and LR indicators and go perform total calculations.
Set on record identifying indicator.
1
Change in control field? Yes, set on control level indicators.
Perform total calculations. Set resulting indicators.
2
Perform total output operations. If overflow line has been reached, set on overflow indicator.
LR indicator on? Yes, end of job has been reached.
3
Overflow indicator on? Yes, do overflow operations and set overflow indicator off.
Move data from record selected at beginning of cycle into processing area. Set field indicators.
4
Perform detail calculations. Set resulting indicators.

(1) Did Account Number change?

(2) Print total only after all transaction records for one customer have been processed.

(3) Print all heading lines if overflow occurs.

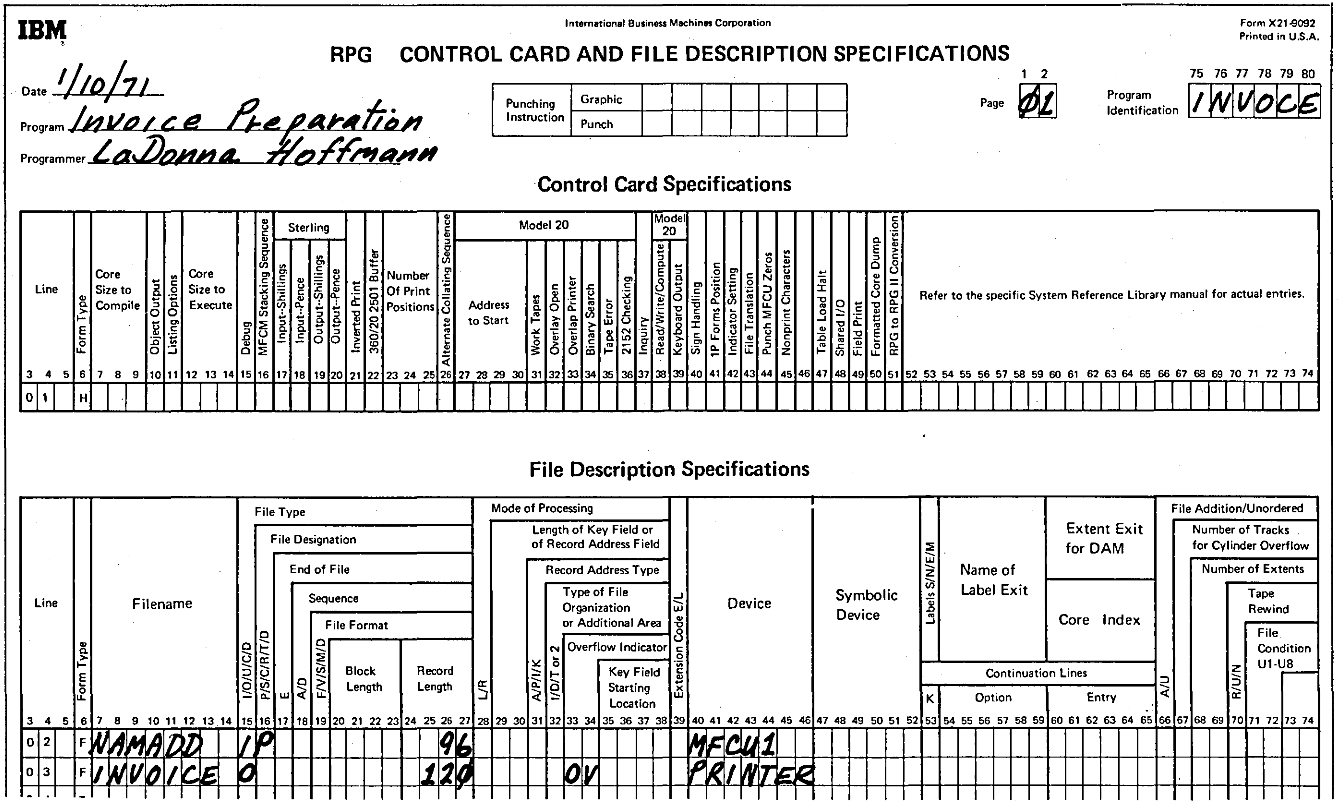
(4) The operation to find shipping instructions can be done only for name/address record. The operations to find AMOUNT and INVTOT can be done only for transaction records.

(5) Headings for the invoice can be printed only when name/address record is read and detail lines only when transaction records are read.

Figure 18. Program Cycle Operations for Sample Job

WRITE THE SPECIFICATIONS

After you have analyzed the problem and determined how to solve it using RPG II, you can write the specifications. Figure 19 shows the specifications for the job.



IBM — International Business Machines Corporation — Form X21-9092, Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date 1/10/71
Program Invoice Preparation
Programmer LaDonna Hoffmann

Punching Instruction: Graphic / Punch

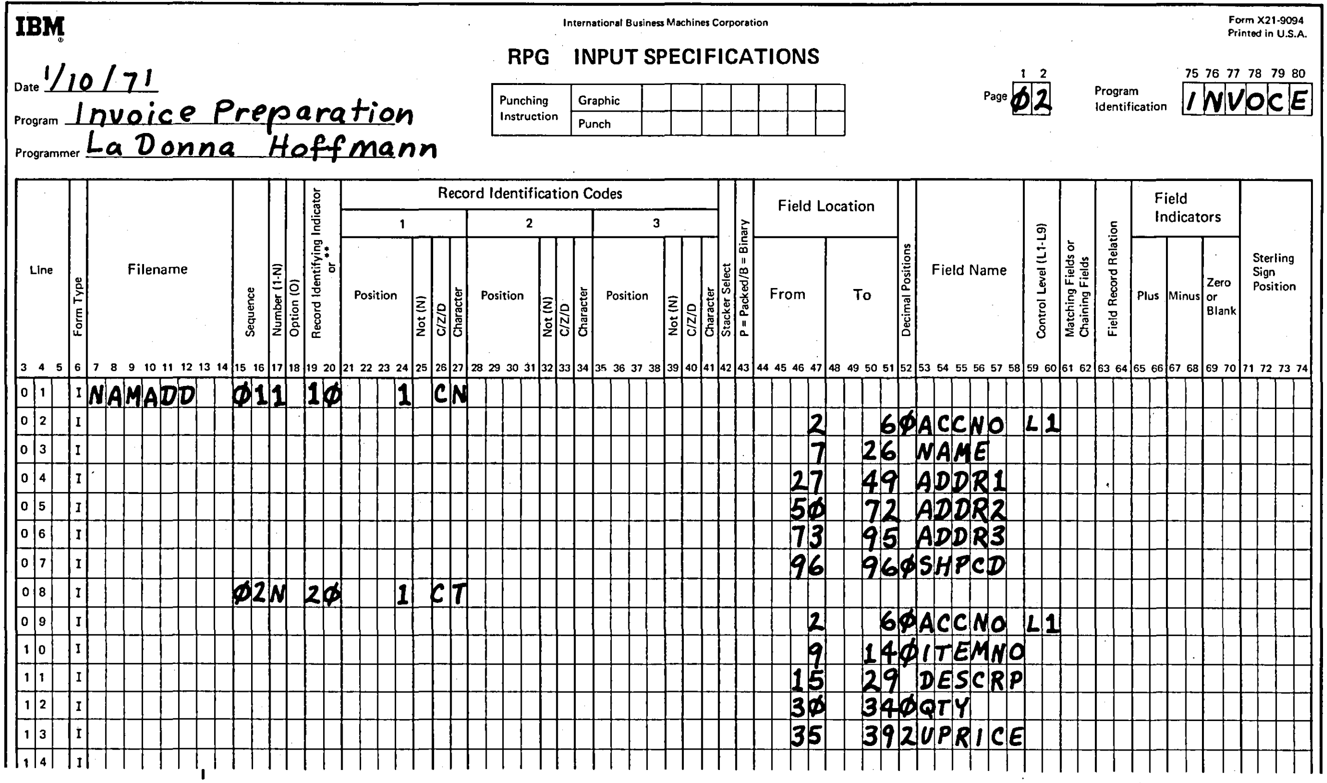
Page Ø1 — Program Identification INVOCE

Control Card Specifications

Line	Form Type	Core Size to Compile	Object Output	Listing Options	Core Size to Execute	Debug	MFCM Stacking Sequence	Input-Shillings	Input-Pence	Output-Shillings	Output-Pence	Inverted Print	360/20 2501 Buffer	Number Of Print Positions	Alternate Collating Sequence	Address to Start	Work Tapes	Overlay Open	Overlay Printer	Binary Search	Tape Error	2152 Checking	Inquiry	Read/Write/Compute	Keyboard Output	Sign Handling	1P Forms Position	Indicator Setting	File Translation	Punch MFCU Zeros	Nonprint Characters	Table Load Halt	Shared I/O	Field Print	Formatted Core Dump	RPG to RPG II Conversion	Refer to the specific System Reference Library manual for actual entries.
01	H																																				

File Description Specifications

Line	Form Type	Filename	File Type (I/O/U/C/D)	File Designation (P/S/C/R/T/D)	End of File (E)	Sequence (A/D)	File Format (F/V/S/M/D)	Block Length	Record Length	L/R	Length of Key Field or of Record Address Field	Record Address Type (A/P/I/K)	Type of File Organization or Additional Area (I/D/T or 2)	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device	Symbolic Device	Labels S/N/E/M	Name of Label Exit	Extent Exit for DAM	File Addition/Unordered (A/U)	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind (R/U/N)	File Condition U1-U8
02	F	NAMADD	I	P					96								MFCU1									
03	F	INVOICE	O						12Ø					OV			PRINTER									



IBM — International Business Machines Corporation — Form X21-9094, Printed in U.S.A.

RPG INPUT SPECIFICATIONS

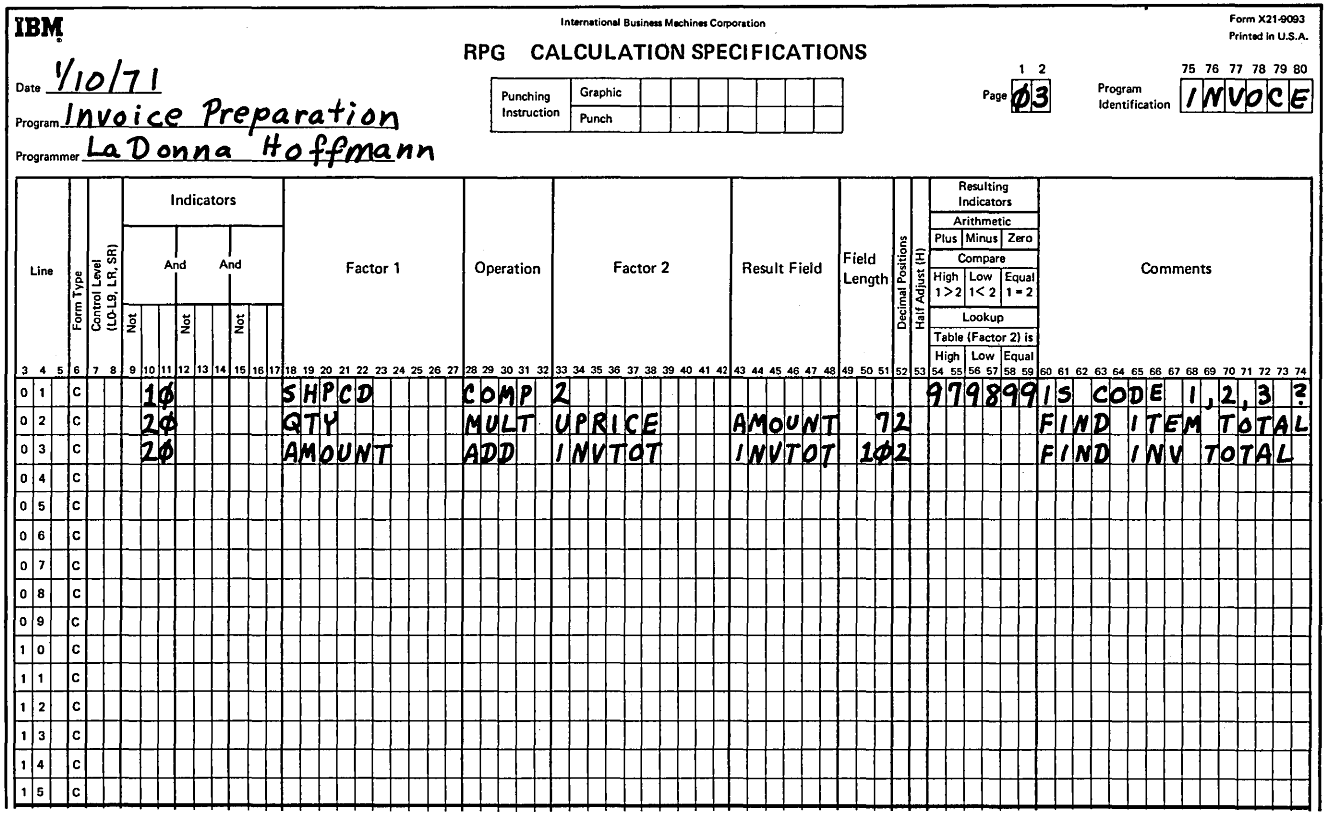
Date 1/10/71
Program Invoice Preparation
Programmer La Donna Hoffmann

Punching Instruction: Graphic / Punch

Page Ø2 — Program Identification INVOCE

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator or **	Position	Not (N)	C/Z/D	Character	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank	Sterling Sign Position
01	I	NAMADD	Ø1	1		1Ø	1		C	N											
02	I										2	6	Ø	ACCNO	L1						
03	I										7	26		NAME							
04	I										27	49		ADDR1							
05	I										5Ø	72		ADDR2							
06	I										73	95		ADDR3							
07	I										96	96	Ø	SHPCD							
08	I		Ø2	N		2Ø	1		C	T											
09	I										2	6	Ø	ACCNO	L1						
10	I										9	14	Ø	ITEMNO							
11	I										15	29		DESCRP							
12	I										3Ø	34	Ø	QTY							
13	I										35	39	2	UPRICE							
14	I																				

Figure 19 (Part 1 of 3). Specifications for Sample Invoice Program



IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date 1/10/71

Program Invoice Preparation

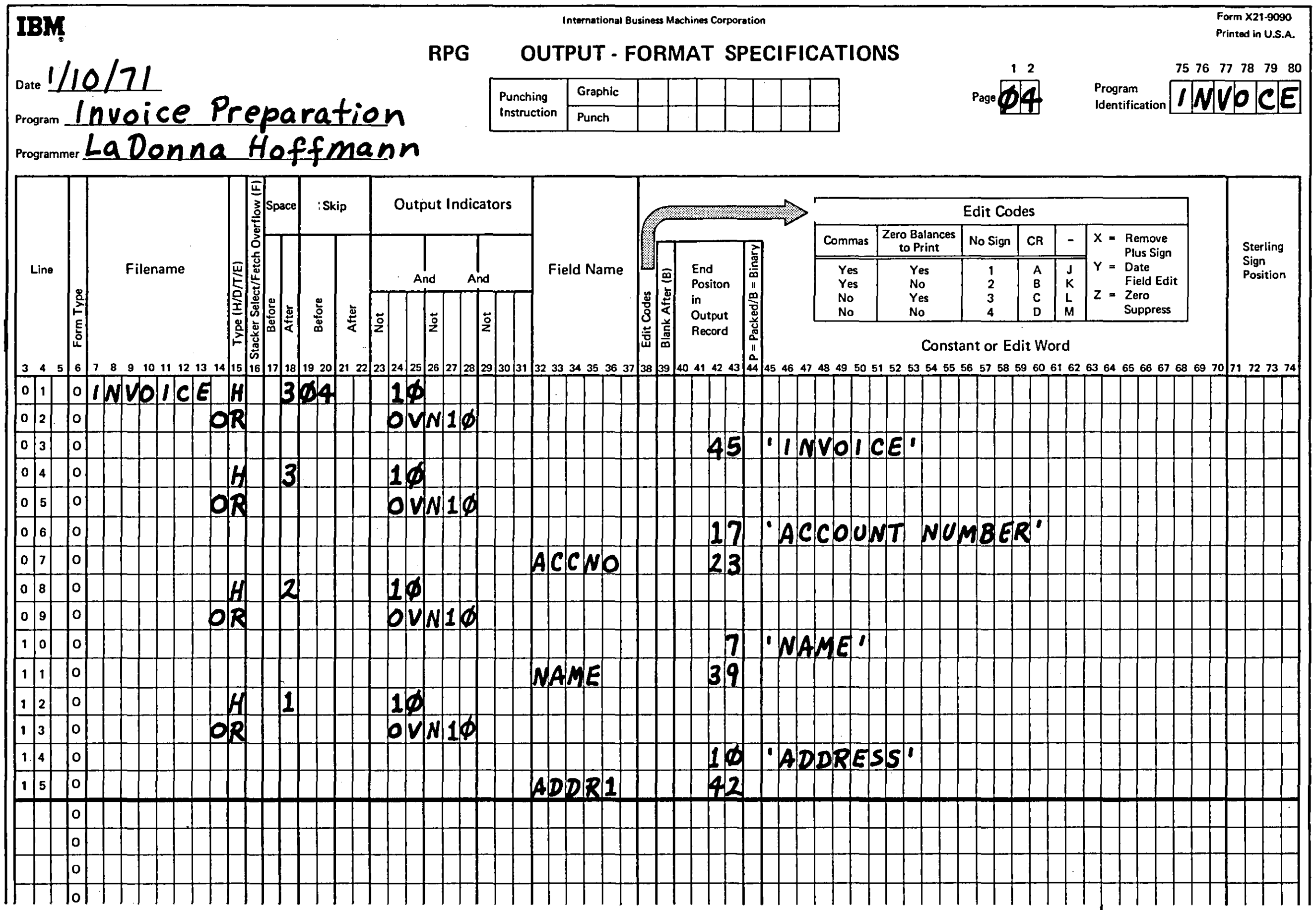
Programmer LaDonna Hoffmann

Punching Instruction	Graphic
	Punch

Page Ø3

Program Identification INVOCE

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators: Not	Indicators	And: Not	And	And: Not	And	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators: Plus / High 1>2	Minus / Low 1<2	Zero / Equal 1=2	Comments
01	C			1Ø					SHPCD	COMP	2					97	98	99	IS CODE 1,2,3 ?
02	C			2Ø					QTY	MULT	UPRICE	AMOUNT	7	2					FIND ITEM TOTAL
03	C			2Ø					AMOUNT	ADD	INVTOT	INVTOT	1Ø	2					FIND INV TOTAL
04	C																		
05	C																		
06	C																		
07	C																		
08	C																		
09	C																		
10	C																		
11	C																		
12	C																		
13	C																		
14	C																		
15	C																		



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71

Program Invoice Preparation

Programmer LaDonna Hoffmann

Punching Instruction	Graphic
	Punch

Page Ø4

Program Identification INVOCE

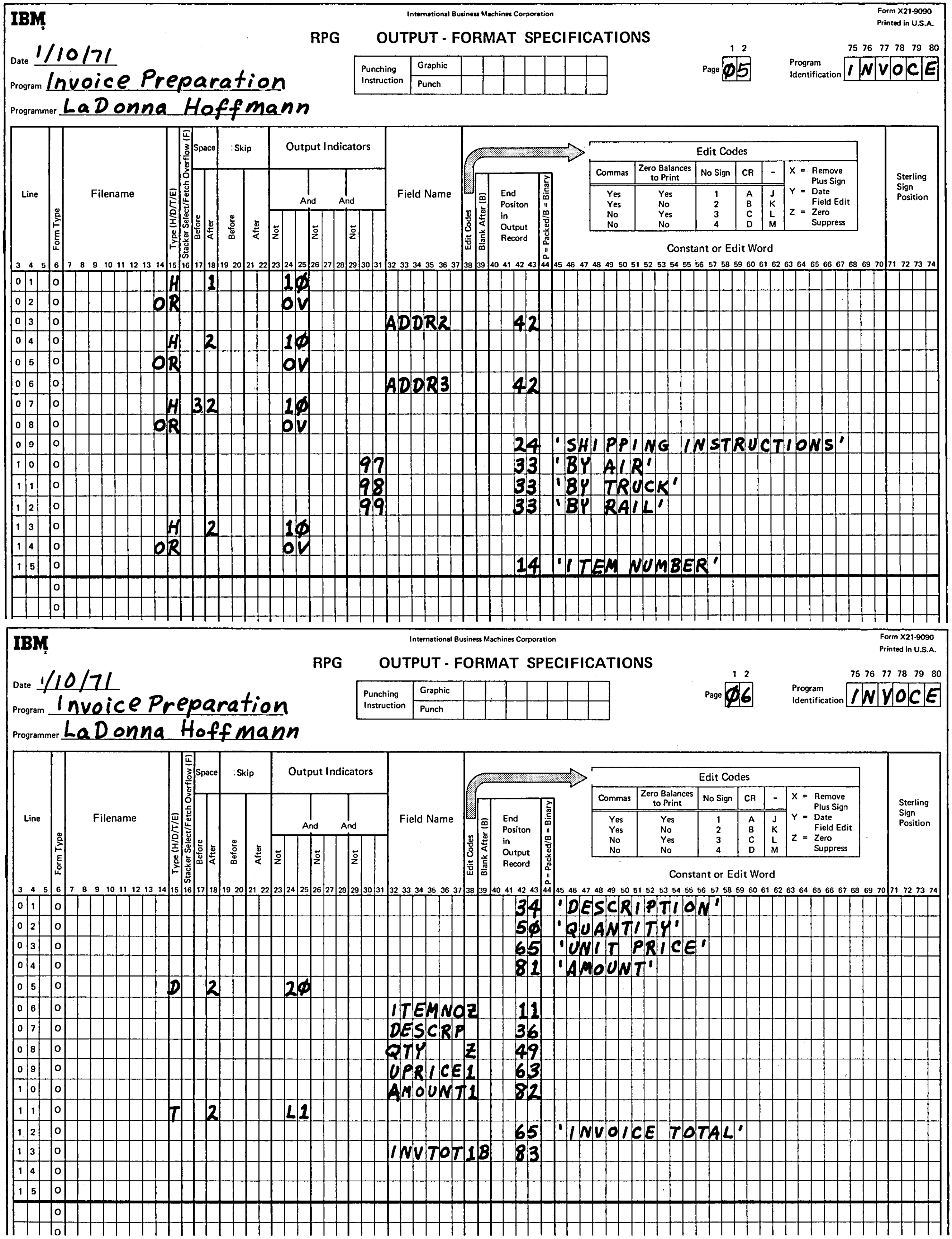
Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fetch Overflow (F)	Space: Before	Space: After	Skip: Before	Skip: After	Output Indicators: Not	Output Indicators	And: Not	And	And: Not	And	Field Name	Edit Codes	Blank After (B)	End Positon in Output Record	P = Packed/B = Binary	Constant or Edit Word	Sterling Sign Position
01	O	INVOICE	H			3	Ø4			1Ø											
02	O	OR								OVN1Ø											
03	O																	45		'INVOICE'	
04	O		H			3				1Ø											
05	O	OR								OVN1Ø											
06	O																	17		'ACCOUNT NUMBER'	
07	O														ACCNO			23			
08	O		H			2				1Ø											
09	O	OR								OVN1Ø											
10	O																	7		'NAME'	
11	O														NAME			39			
12	O		H			1				1Ø											
13	O	OR								OVN1Ø											
14	O																	1Ø		'ADDRESS'	
15	O														ADDR1			42			
	O																				
	O																				
	O																				
	O																				

Figure 19 (Part 2 of 3). Specifications for Sample Invoice Program



IBM | International Business Machines Corporation | Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71

Program Invoice Preparation

Programmer LaDonna Hoffmann

Punching Instruction	Graphic				
	Punch				

Page (1 2) Ø5 | Program Identification (75 76 77 78 79 80) INVOCE

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Not (23)	Output Indicator (24-25)	And Not (26)	Output Indicator (27-28)	And Not (29)	Output Indicator (30-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
01	O		H			1				1Ø											
02	O		OR							OV											
03	O														ADDR2			42			
04	O		H			2				1Ø											
05	O		OR							OV											
06	O														ADDR3			42			
07	O		H		3	2				1Ø											
08	O		OR							OV											
09	O																	24		'SHIPPING INSTRUCTIONS'	
10	O													97				33		'BY AIR'	
11	O													98				33		'BY TRUCK'	
12	O													99				33		'BY RAIL'	
13	O		H			2				1Ø											
14	O		OR							OV											
15	O																	14		'ITEM NUMBER'	
	O																				
	O																				

IBM | International Business Machines Corporation | Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date 1/10/71

Program Invoice Preparation

Programmer LaDonna Hoffmann

Punching Instruction	Graphic				
	Punch				

Page (1 2) Ø6 | Program Identification (75 76 77 78 79 80) INVOCE

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Not (23)	Output Indicator (24-25)	And Not (26)	Output Indicator (27-28)	And Not (29)	Output Indicator (30-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
01	O																	34		'DESCRIPTION'	
02	O																	5Ø		'QUANTITY'	
03	O																	65		'UNIT PRICE'	
04	O																	81		'AMOUNT'	
05	O		D			2				2Ø											
06	O														ITEMNO	Z		11			
07	O														DESCRP			36			
08	O														QTY	Z		49			
09	O														UPRICE	1		63			
10	O														AMOUNT	1		82			
11	O		T			2				L1											
12	O																	65		'INVOICE TOTAL'	
13	O														INVTOT	1	B	83			
14	O																				
15	O																				
	O																				
	O																				

Figure 19 (Part 3 of 3). Specifications for Sample Invoice Program

DOCUMENT THE PROGRAM

An important part of every programmer's job is to explain his program. This documentation provides information for people who will run the program and for programmers who may later need to alter or update it. Documentation is also useful to you. It is not always easy to remember what every program you wrote does. Reading documentation is a much easier way to recall the program than figuring out each instruction.

Documentation consists of:

1. Telling generally what the program does.

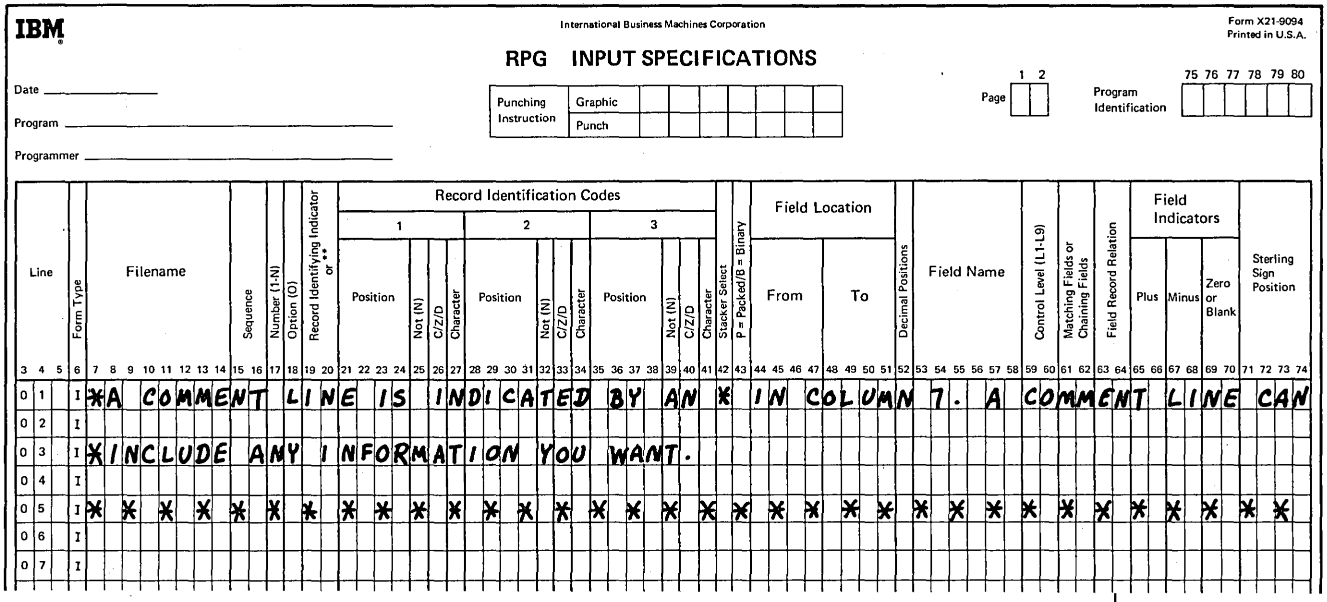
2. Describing input and output. (Record Layout Forms and Printer Spacing Charts are an excellent means of doing this.) File names and field names should be meaningful.

3. Explaining the coding.

4. Telling the operator how to run the program, what to do if the computer stops because of an error, and what to do when the job is completed.

All documentation cannot be done at the time you write specifications. However, when writing your specifications, you can also write an explanation of a line of coding on the specifications forms. You have probably noticed columns labeled comments on the specification sheets. Here is where you write an explanation for your coding.

In addition to using comment columns on the coding sheets, you can use comment lines. A comment line is indicated by * in column 7 of the coding sheet:



IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date
Program
Programmer

Punching Instruction: Graphic / Punch   Page 1 2   Program Identification 75 76 77 78 79 80

Line | Form Type | Filename | Sequence | Number (1-N) | Option (O) | Record Identifying Indicator or ** | Record Identification Codes 1 2 3: Position, Not (N), C/Z/D, Character | Stacker Select | P = Packed/B = Binary | Field Location From To | Decimal Positions | Field Name | Control Level (L1-L9) | Matching Fields or Chaining Fields | Field Record Relation | Field Indicators Plus Minus Zero or Blank | Sterling Sign Position

```
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
0 1   I*A COMMENT LINE IS INDICATED BY AN * IN COLUMN 7. A COMMENT LINE CAN
0 2   I
0 3   I*INCLUDE ANY INFORMATION YOU WANT.
0 4   I
0 5   I* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
0 6   I
0 7   I
```

Comment lines can be used anywhere on any specification sheet. There is no limit to the number you can use. The RPG II Compiler does not regard comments and comment lines as part of the program. This means that it does not translate them into instructions for the computer.

PREPARE FOR COMPILATION

After completing your source program, you must prepare it for compilation.

Specification Sheet Order

Your specification sheets must be in this order:

1. Control Card and File Description sheet.

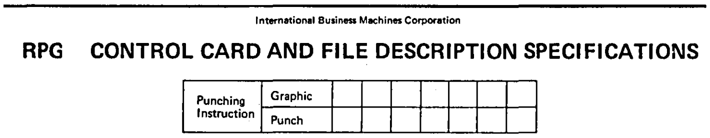
2. Input sheets.

3. Calculation sheets.

4. Output-Format sheets.

Number the sheets in columns 1 and 2. At this time, you might also check to see that the top part of each sheet is completely filled in.

If you are planning to give these specifications to someone to keypunch, it is a good idea to fill in the box labeled punching instructions:



International Business Machines Corporation

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

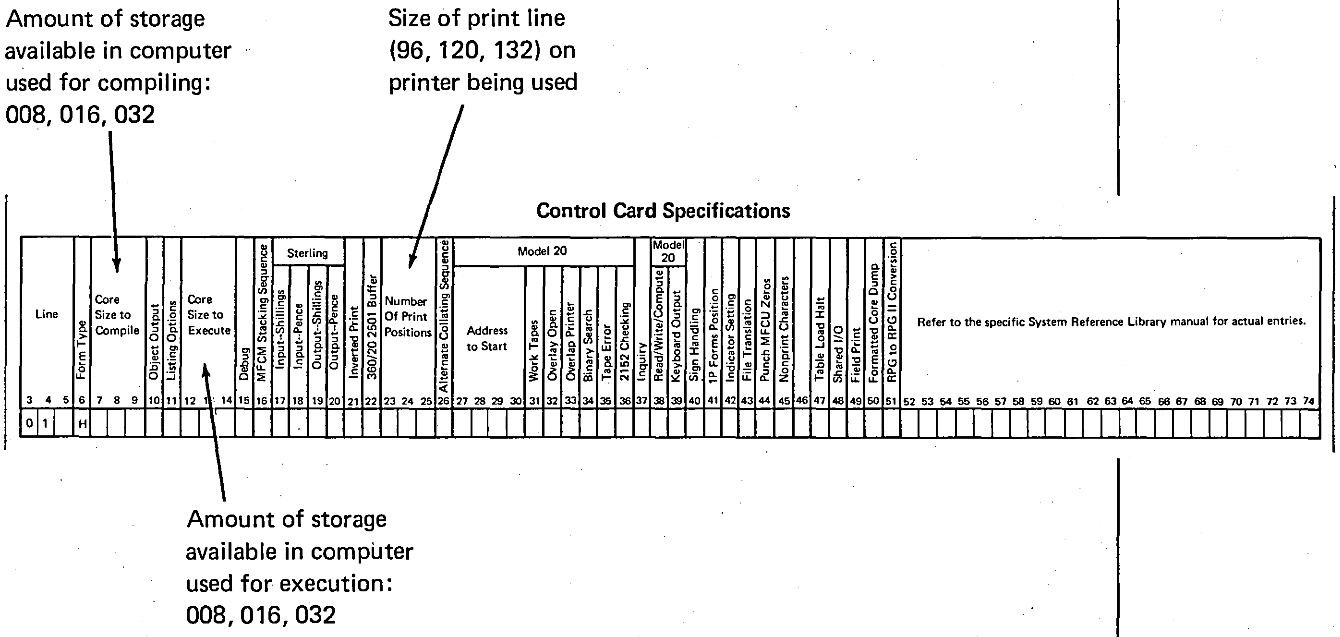
Punching Instruction	Graphic							
	Punch							

You indicate in this box the graphic symbols you are using and their meaning. Some printed letters and numbers are easily confused. For example, it is sometimes difficult to differentiate between the number 0 and the letter O and the number 2 and the letter Z. You may, therefore, devise a graphic symbol that you use for certain letters. Some people use Ø for zero, Ƶ for the letter Z. Explain your symbols so that the keypunch operator will know what to punch when she finds the symbol on the coding sheets.

Control Card Preparation

Some systems require control card specifications. If yours does, you will have to fill out the control card line at the top of the Control Card and File Description sheet.

Control card specifications give the compiler information about the system and tell whether any special RPG II functions are used in the program. The most used entries are:


Amount of storage available in computer used for compiling: 008, 016, 032

Size of print line (96, 120, 132) on printer being used

Amount of storage available in computer used for execution: 008, 016, 032

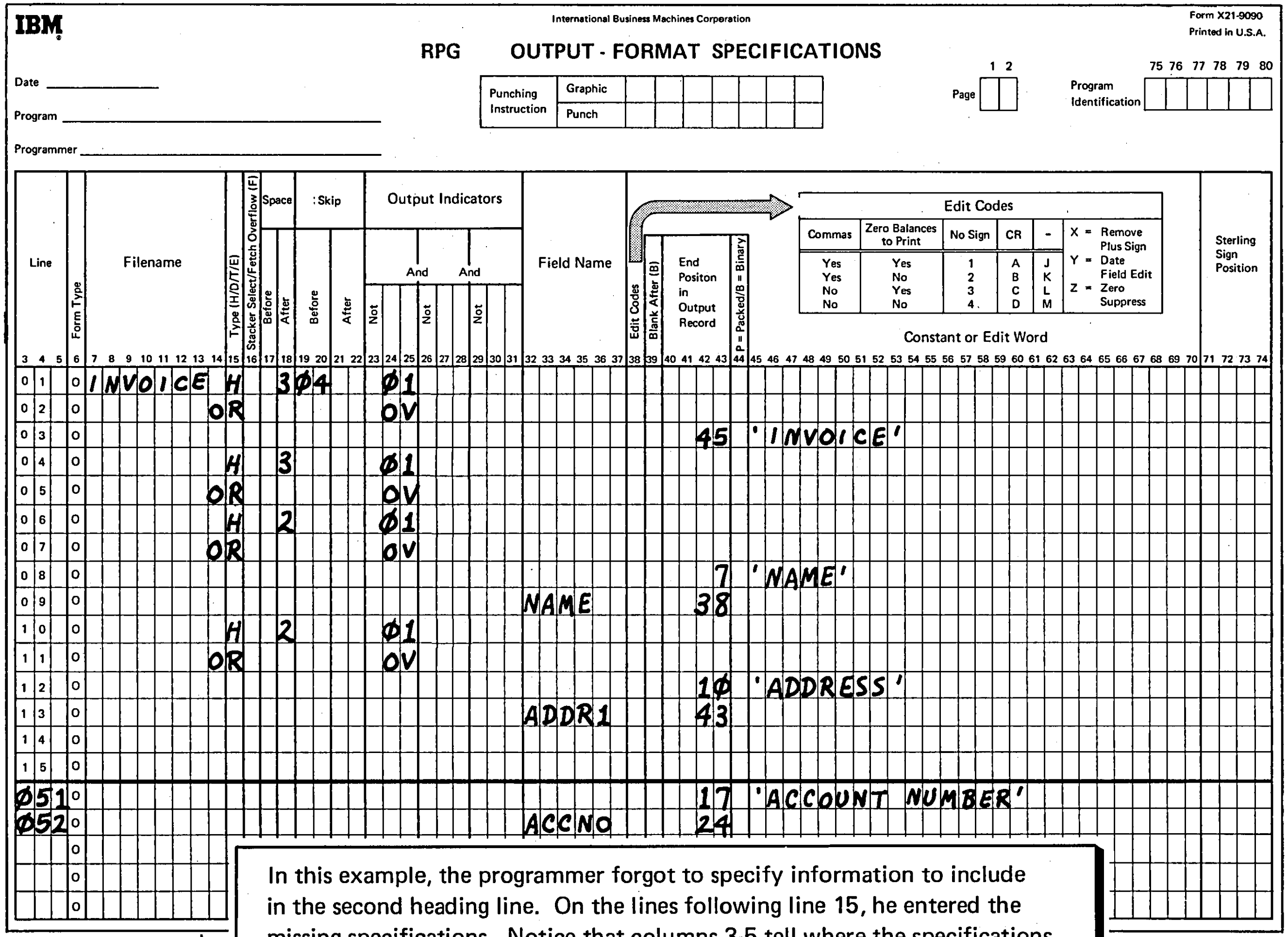
Checking Specifications

Desk checking is a good way to reduce the number of would-be program errors. By desk checking we mean carefully checking through your specifications to see whether you have:

- Placed entries in appropriate columns.
- Used correct entries in columns.
- Spelled field and file names consistently throughout your program.
- Used your indicators correctly.

If you should find that you have omitted a specification (forgot to name an input field or an output field or forgot to enter a calculation), you can enter it on a line following line 15.

Notice that no line numbers have been entered in columns 3-5 of these specification lines. You can place numbers in these columns to tell where the missing specification belongs:



IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date ______________

Program ______________

Programmer ______________

Punching Instruction	Graphic							
	Punch							

Page (1 2) □□

Program Identification (75 76 77 78 79 80) □□□□□□

Edit Codes

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Stacker Select/Fetch Overflow (F) (16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Positon in Output Record (40-43)	P = Packed/B = Binary (44)	Constant or Edit Word (45-70)	Sterling Sign Position (71-74)
01	O	INVOICE	H			3	Ø4		Ø1							
02	O	OR							OV							
03	O												45		'INVOICE'	
04	O		H			3			Ø1							
05	O	OR							OV							
06	O		H			2			Ø1							
07	O	OR							OV							
08	O												7		'NAME'	
09	O									NAME			38			
10	O		H			2			Ø1							
11	O	OR							OV							
12	O												1Ø		'ADDRESS'	
13	O									ADDR1			43			
14	O															
15	O															
Ø51	O												17		'ACCOUNT NUMBER'	
Ø52	O									ACCNO			24			
	O															
	O															
	O															

In this example, the programmer forgot to specify information to include in the second heading line. On the lines following line 15, he entered the missing specifications. Notice that columns 3-5 tell where the specifications belong. The line numbers 051 and 052 indicate that the specifications belong between lines 05 and 06.

If your specification cards are being keypunched, the out-of-order cards must be inserted in the appropriate place. If the source program is being entered directly into the system through a keyboard, the missing specifications will have to be inserted in the appropriate place when the specifications are keyed.

COMPILE THE SOURCE PROGRAM

When you think your source program is free of errors, it can be keypunched (if your system requires a card source deck) or entered directly into the system (if you have a direct entry system). You can then compile your source program. The important part of compilation is, of course, translating the source program in machine language. But in addition to this, the compiler also produces a program listing you will find very helpful.

The most important parts of the program listing are:

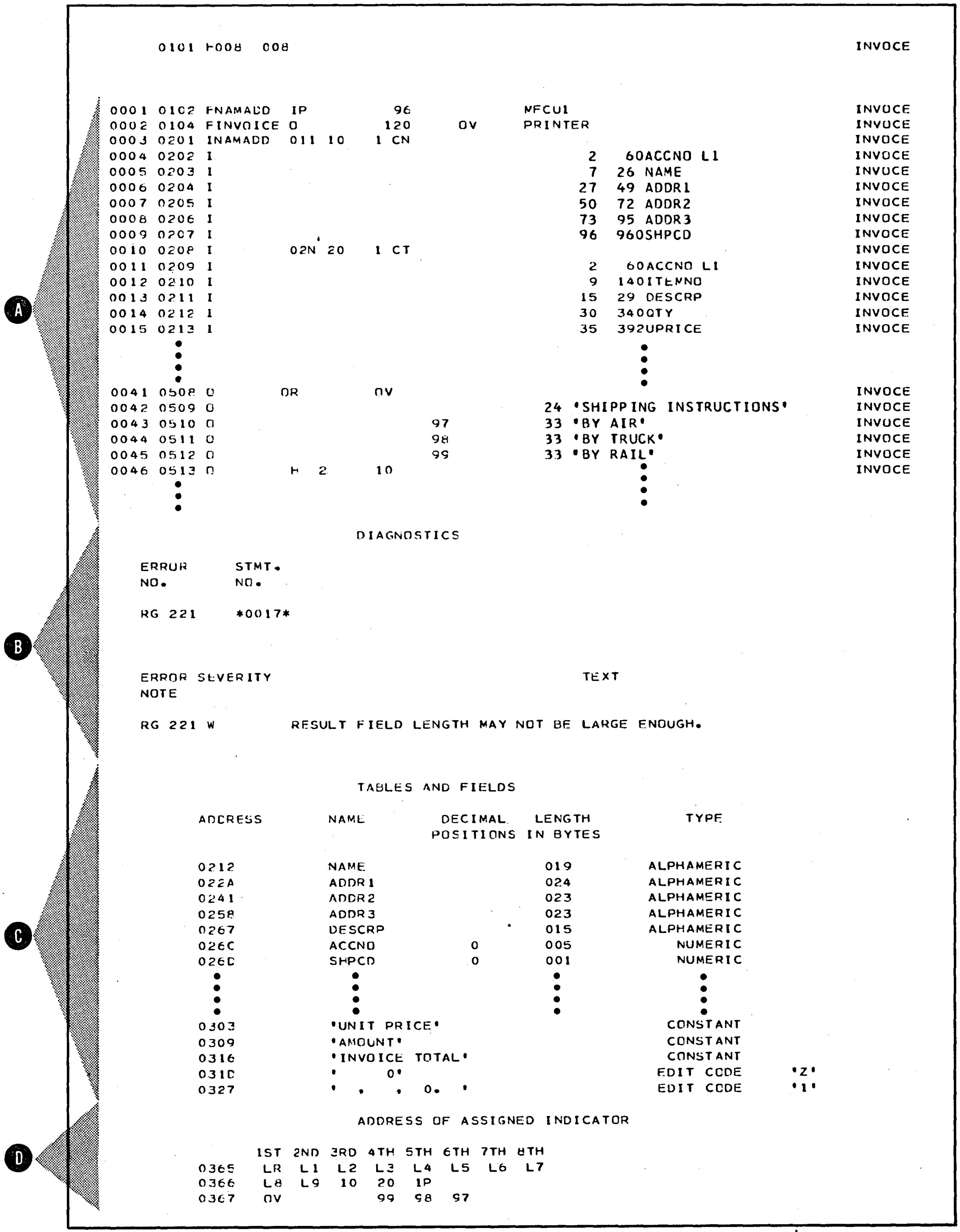
Ⓐ A printout of source specifications including comment lines. Notice the number at the left of each line. This is the sequence number the compiler assigns to the specification.

Ⓑ Diagnostic messages indicating the types of errors made and the statement in which they occur.

Ⓒ A list of all fields used in the program. Included in the list is the storage location assigned to each field and a description of each field as indicated in your specifications. Some compilers also provide this information for constants used in the program.

Ⓓ A list of all indicators used in the program.

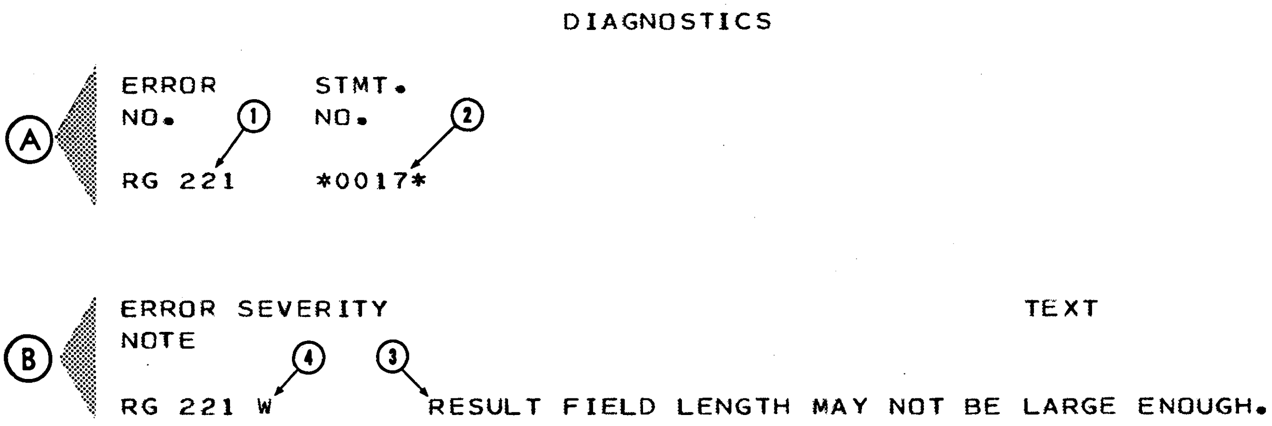
* Note that the sample listing shown is the program listing for the invoice job.



```
          0101 F008   008                                                                   INVOCE


     0001 0102 FNAMADD   IP          96              MFCU1                                  INVOCE
     0002 0104 FINVOICE O           120      OV      PRINTER                                INVOCE
     0003 0201 INAMADD   011 10     1 CN                                                    INVOCE
     0004 0202 I                                                 2   60ACCNO L1             INVOCE
     0005 0203 I                                                 7  26 NAME                 INVOCE
     0006 0204 I                                                27  49 ADDR1                INVOCE
     0007 0205 I                                                50  72 ADDR2                INVOCE
     0008 0206 I                                                73  95 ADDR3                INVOCE
     0009 0207 I                                                96  960SHPCD                INVOCE
     0010 0208 I         02N 20     1 CT                                                    INVOCE
     0011 0209 I                                                 2   60ACCNO L1             INVOCE
     0012 0210 I                                                 9  140ITEMNO               INVOCE
     0013 0211 I                                                15  29 DESCRP               INVOCE
     0014 0212 I                                                30  340QTY                  INVOCE
     0015 0213 I                                                35  392UPRICE               INVOCE
              •                                                      •
              •                                                      •
              •                                                      •
              •                                                      •
     0041 0508 O       OR         OV                                                        INVOCE
     0042 0509 O                                           24 'SHIPPING INSTRUCTIONS'       INVOCE
     0043 0510 O                           97              33 'BY AIR'                      INVOCE
     0044 0511 O                           98              33 'BY TRUCK'                    INVOCE
     0045 0512 O                           99              33 'BY RAIL'                     INVOCE
     0046 0513 O         H  2       10                           •                          INVOCE
              •                                                  •
              •                                                  •
              •                                                  •

                                 DIAGNOSTICS

        ERROR       STMT.
        NO.         NO.

        RG 221      *0017*


        ERROR SEVERITY                               TEXT
        NOTE

        RG 221 W        RESULT FIELD LENGTH MAY NOT BE LARGE ENOUGH.


                                 TABLES AND FIELDS

              ADDRESS        NAME          DECIMAL   LENGTH           TYPE
                                         POSITIONS IN BYTES

              0212           NAME                       019        ALPHAMERIC
              022A           ADDR1                      024        ALPHAMERIC
              0241           ADDR2                      023        ALPHAMERIC
              0258           ADDR3                      023        ALPHAMERIC
              0267           DESCRP                     015        ALPHAMERIC
              026C           ACCNO              0       005           NUMERIC
              026D           SHPCD              0       001           NUMERIC
               •               •                         •               •
               •               •                         •               •
               •               •                         •               •
               •               •                         •               •
              0303           'UNIT PRICE'                            CONSTANT
              0309           'AMOUNT'                                CONSTANT
              0316           'INVOICE TOTAL'                         CONSTANT
              031D           '      0'                             EDIT CODE      'Z'
              0327           '  ,   ,  0.  '                       EDIT CODE      '1'

                                 ADDRESS OF ASSIGNED INDICATOR

                        1ST  2ND  3RD  4TH  5TH  6TH  7TH  8TH
              0365       LR   L1   L2   L3   L4   L5   L6   L7
              0366       L8   L9   10   20   1P
              0367       OV             99   98   97
```

If the compiler finds any errors in your source specifications, it will print diagnostic messages telling you what errors were made. You will find that different types of messages are printed: warning, terminal, or informative. A warning message is an indication that something may be wrong. If you check the questioned specification and find that is is all right for your program, you need not make changes. If you get a terminal message, however, something is wrong with your coding. You must fix the specification and recompile the program before the compiler will actually translate your specifications:

The diagnostic message section of the program listing contains two basic parts: a list of messages (A) and an explanation of each message (B).



```
                              DIAGNOSTICS

  ERROR    STMT.
  NO.      NO.

  RG 221   *0017*


  ERROR SEVERITY                                   TEXT
  NOTE

  RG 221 W          RESULT FIELD LENGTH MAY NOT BE LARGE ENOUGH.
```

Each error message in the list is identified by a 3-digit number (1). Next to the message number is either a statement number identifying the specification in which the error appears or a field name or constant associated with the error (2). Following the list of messages is an explanation of the error (3) and an indication of the severity of the error (W = warning; T = terminal) (4).

The sample shown above shows diagnostic messages printed for the invoice job. Note that the message is a warning. A warning is an indication that something may be wrong. If you check the specification noted and find that it will work for the job, you need not change it.

Checking the message in the listing, you would find that the warning points to the AMOUNT field in statement 0017:

```
0017 0302 C    20        QTY          MULT UPRICE     AMOUNT  92
```

The AMOUNT field is specified as nine characters with two decimal positions. According to the field lengths given to QTY (5 positions) and UPRICE (5 positions), the two fields involved in the multiplication, you could possibly get a 10-digit result.

```
      999.99
     X 99999
     -------
     899991
    899991
   899991
  899991
 899991
 -----------
 99998000.01
```

If you feel that QTY and UPRICE will never contain the maximum amounts, you could leave the specification as is.

TEST THE PROGRAM

It is good practice to test your program before using it for an actual job. To do this, make up test data representing all possible situations that could arise during an actual job. Run your program using that data to see if your program will really handle the situations you think it does. If you get the wrong results when testing, you know your program isn't doing what you thought it would. You can usually find your errors by using actual input data and doing the operations specified yourself, step-by-step, in the order the computer would do them. When doing this, you'll have to follow closely your specifications and the program cycle operations taken by your program. After you've tested your program and the results show it can handle all situations, your job is complete.

Glossary

Address: A number identifying a location in storage.

Alphabetic: In general usage, any combination of the letters A-Z. In RPG II programming, any combination of the letters A-Z and special characters @, #, and $.

Alphameric: Any combination of alphabetic, numeric, and special characters as defined by the RPG II language.

AND relationship: The specifying of conditioning indicators such that the operation conditioned will be performed only when all conditions are met.

Arithmetic/logic unit: An area inside the processing unit where calculations are performed.

Arithmetic operation: An operation such as addition, subtraction, multiplication, and division performed in the processing unit.

Ascending order: The arrangement of data in a specified field from low to high.

Blank after: Changing the contents of a field so that it contains only zeros or blanks after that field has been printed or punched.

Calculation specification sheet: An RPG II coding sheet which specifies the type and order of calculations to be performed on the input data.

Card: In data processing, a card containing combinations of holes representing data to a computer.

Eighty-column card: A punch card with 80 vertical columns representing 80 characters.

Ninety-six column card: A punch card with 96 vertical columns representing 96 characters. The columns are divided horizontally into thirds, such that the columns in the upper third are numbered 1-32, in the middle third, 33-64, and in the lower third, 65-96.

Card file: A group of related punched-card records.

Card layout form: A chart for planning the design and format of cards.

Card punch: A device that records information on a card in the form of combinations of holes representing characters.

Card reader: A device that electronically senses information on punched cards and transfers that information to the processing unit.

Character: Any individual data item that can be represented in printed form; that is, a letter, a digit, or a special character.

Coding: Making entries on RPG II specification sheets.

Comments: Words or statements in a program that serve as documentation rather than as instructions to the compiler.

Compile: Translate a source program (such as RPG II specifications) into an object program (machine language program) using the computer.

Compiler: A program that translates a source program into a machine language program.

Computer: A device or group of devices capable of accepting, processing, and reporting information.

Conditioning: Using indicators to control when calculations or output operations are done.

Constant: A data item that does not change during execution of a program. This item represents itself and is actually used in processing rather than being a field name representing the data. For example, COST is a name representing a field containing data which changes, whereas the constant 100 is actual data used which does not change.

Control break: A change in the contents of a control field.

Control card and file description specification sheet: An RPG II coding sheet which gives, for a particular job, information needed for control of the computer and a description of the files used.

Control field: One or more fields that are compared from record to record to determine when certain operations should be performed.

Control group: A set of records all having the same control field information.

Control level indicator: An indicator used to specify certain fields as control fields and to tell which operations to perform at total time.

Control unit: An area inside the processing unit that determines from instructions what has to be done. It directs other units or devices to perform the required functions.

Data: A collection of facts, numbers, letters, and symbols that can be processed or produced by a computer.

Descending order: The arrangement of data in a specified field from high to low.

Detail record: An output record produced during the detail output operation of the RPG II program cycle.

Detail time: An operation in the RPG II program cycle in which calculation and output operations are performed for each record read.

Diagnostic message: An output message that identifies RPG II specification errors and their severity.

Digit: One of the characters 0-9.

Disk: A thin, round metal plate coated with magnetic material on which information can be recorded (both sides) in the form of magnetized spots.

Disk drive: A device that reads data from or writes data on a disk.

Documentation: A written explanation of a program, its use, its function, and its operations.

Edit: To punctuate a field by suppressing zeros and inserting commas, decimal points, dollar signs, or other constant information.

Edit Code: A number or letter indicating that editing should be done according to a predefined pattern. This includes zero suppression and punctuation.

Eighty-column card: A punch card with 80 vertical columns representing 80 characters.

Error message: (See diagnostic message.)

Execute: To process input data files according to machine language instructions to produce the desired output.

Factor: In RPG II programming, a field name or constant used in a calculation operation.

Field: One or more adjacent record positions which contain related information.

Control field: One or more fields that are compared from record to record to determine when certain operations should be performed.

Result field: The name of a field where the outcome of arithmetic calculations is kept.

Field indicator: An indicator used to determine if a given field on an input record is plus, minus, zero, or blank.

Field length: The number of columns allowed for a given field, determined by the maximum length of information that will be entered in the field.

Field name: In RPG II programming, a combination of six or fewer alphabetic or numeric characters (the first of which must be alphabetic) which identifies a field.

File: An organized collection of related records.

Card file: A group of related records stored on cards.

Disk file: A group of related records stored on disk.

Input file: A set of records a program uses as a source of data.

Output file: A set of records that is written, punched, or printed by the computer.

Primary file: The main file from which a program first reads records. In multifile processing, it is used for determining the order in which records are selected for processing.

Secondary file: Any file other than the primary file used in multifile processing.

File name: In RPG II programming, a combination of eight or fewer alphabetic or numeric characters (the first of which must be alphabetic) which identifies a file.

First page indicator: An indicator used to specify which lines (such as headings) should be printed on the first page only.

Half adjust: A method of rounding off a number by adjusting the last digit to be kept. When the number to the right of the last numeral to be retained is 5 or greater, 1 is added to the last retained digit. For example, 2.475 half adjusted to two decimal places becomes 2.48, but 2.474 becomes 2.47.

Heading: A constant, usually printed at the top of a page, identifying the information or report on that page.

Indicator:

1. A 2-digit or 2-character entry on the specification sheets used to tell when certain operations are to be performed.

2. An internal switch used by the object program to remember when a certain event occurs and what to do when the event does occur.

Control level indicator: An indicator used to specify certain fields as control fields and to tell which operations to perform at total time when data in the control field changes.

Field indicator: An indicator used to determine if a given field on an input record is plus, minus, zero, or blank.

First page indicator: An indicator used to specify which lines (such as headings) should be printed on the first page only.

Last record indicator: An indicator that signifies when the last data record has been processed.

Overflow indicator: An indicator that signifies when the last line to be printed on a page has been passed. It may be used to specify which lines are to be printed on the next page.

Record identifying indicator: An indicator that signifies the type of record to be processed next.

Resulting indicator: An indicator that signifies (1) if the result of a given calculation is plus, minus, or zero, or (2) if a given field is greater than, less than, or equal to another field.

Input: Information to be transferred from cards, disk, or keyboard to storage.

Input specification sheet: A coding sheet used to identify the different types of records in each input file and to describe the fields in each record.

Instruction: A statement that specifies an operation to be performed by the computer and the locations in storage of all data involved in that operation.

Keyboard: A device, similar to a typewriter, used for entering data directly into storage.

Keypunch: A device, similar to a typewriter, used for punching information into cards.

Last record indicator: An indicator that signifies when the last data record has been processed.

Machine language: A language that can be interpreted and used by a computer.

Master record: A record whose information rarely changes (such as a name and address record).

Ninety-six column card: A punch card with 96 vertical columns representing 96 characters. The columns are divided horizontally into thirds, such that the columns in the upper third are numbered 1-32, in the middle third, 33-64, and in the lower third, 65-96.

Numeric: Any combination of the characters 0-9.

Object program: A set of instructions in machine language. The object program is produced by the compiler from the source program.

Operation: A defined action performed on one or more data items (for example, adding, multiplying, comparing, or moving information).

Operation code: A word or abbreviation, specified on the Calculation sheet, that is used to identify an operation (for example, SUB for subtract, ADD for add).

OR relationship: The specifying of conditioning indicators such that the operation conditioned is done when either one or both of the conditions are met.

Output: Data transferred from storage into punched cards, printed form, or disk.

Output-format specification sheet: A coding sheet used to specify the records to be written in each output file and the format of output records.

Overflow: The condition that occurs when the last line to be printed on the page has been passed.

Overflow indicator: An indicator that signifies when the last line on a page has been printed or passed. It may be used to specify which lines are to be printed on the next page.

Overflow line: The line specified as the last line to be printed on a page.

Overflow page: The new page which is advanced when overflow occurs.

Primary file: The file that controls the order in which records are selected for processing.

Printer: A device that records information on paper in the form of printed characters.

Printer spacing chart: A form used to plan the locations of data in the output file.

Processing: To perform operations on data from an input record.

Processing unit: The part of a computer that controls the computer and its attached devices, provides storage area for the programs and data, and performs the operations specified in the program.

Program: A set of instructions that (when stored) tells the computer which operations are to be done and how to do them.

> *Object program:* A set of instructions in machine language. The object program is produced by the compiler from the source program.
>
> *Source program:* A set of instructions that represents a particular job as defined by the programmer. These instructions are written in a programming language, such as RPG II.

Program cycle: A series of operations performed by the computer for each record read.

Program listing: A computer printout which gives information about the source program, such as source statements, diagnostic messages, indicators used, storage addresses of fields, and constants used.

Punch card: (See card.)

Record: A group of related fields or data items treated as a unit; for example, a punched card.

Record identification code: A code placed in a record to identify that record type.

Record identifying indicator: An indicator that signifies the type of record to be processed next.

Record length: The number of characters needed to include all the data for one record.

Record types: Records from one file which have different fields and/or format.

Source program: A set of instructions representing a particular job as defined by the programmer. These instructions are written in a programming language, such as RPG II.

Special character: A character other than a digit or letter (for example, *, +, ¢ , %). In RPG II programming, @, #, and $ are considered alphabetic characters.

Specification sheets: Forms on which an RPG II program is coded and described. The four specification sheets described in this manual are the Control Card and File Description sheet, the Input sheet, the Calculation sheet, and the Output-Format sheet.

Storage unit: An area inside the processing unit where instructions and data are stored.

Total operations: Operations performed only after a group of records has been processed.

Total time: That part of the RPG II program cycle in which operations specified for a group of records are done.

Zero suppression: The elimination of leading zeros in a number. For example, 00057, when zero suppressed, becomes b̸b̸b̸57 (b̸ represents one blank space).

Index

Introduction to RPG II Printed in USA GC21-7514-1

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)