Systems

Introduction to RPG II

IBM

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.

© Copyright International Business Machines Corporation 1971, 1972

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.

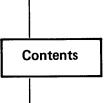
The first chapter describes, in general terms, how a computer operates and the things a programmer must do to make the computer work. The information presented should answer such questions as:

- 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.



Chapter 1.	Basic Computer and Programming Concepts	. 1
	• Parts of a Computer	
	Programs and Programming Languages	
Chapter 2.	RPG II Programming Language	. 7
	RPG II Program Cycle	
	Data Processing Terms and Programming Aids	
	Writing Specifications for Input and Output Operations	
	Program cycle operations Describing the files Describing input records Describing output records Printed reports Sample job	
	Writing Specifications for Calculation Operations	
	Program cycle operations Describing type of operation Describing data to be used Describing the result field Sample job	
	Writing Specifications for Indicators	
	Control level indicators First page indicator Overflow indicators	
	Last record indicator Record identifying indicators Resulting indicators	
	Field indicators Sample jobs for each indicator type	
Chapter 3.	The Programmer's Job	107
	• Determine the Job Requirements	
	Determine RPG II Program Cycle Operations	
	• Write the Specifications	
	• Document the Program	
	Prepare for Compilation	
	Compile the Source Program	
	• Test the Program	
Glossary	• • • • • • • • • • • • • • • • • • • •	127
Indov		122

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	2 75 76 77 78 79 80
Date	Punching Graphic Page	Program
Program	Instruction Punch	Identification
Programmer		
	T	
IBM	International Business Machines Corporation	Form X21-9093 Printed in U.S.A.
	RPG CALCULATION SPECIFICATIONS	1 2 75 76 77 78 79 80
Date	Punching Graphic Page	Program Identification
Program	Instruction Punch	·
Programmer		
Indicators	Resulting Indicator	
	International Business Machines Corporation	Form X21-9094
IBM	RPG INPUT SPECIFICATIONS	Printed In U.S.A.
Date	RFG INFO I SPECIFICATIONS	2 75 76 77 78 79 80
	Punching Graphic Page	Program Identification
Programmer	Punch	
	Record Identification Codes Field Location	Field
IBM	International Business Machines Corporation	Form X21-9092 Printed in U.S.A.
RPG C	ONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS	
Date	Punching Graphic Page	Program
Program	Instruction Punch	
Programmer		
, in the second	Control Card Specifications	
g Sterling	9 Model 20 Model 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	
Core University Section 19 Sectio	isting Secure in the residue in the	·
Line Core Size to Ore Compile Core Fixed Size to County Compile Core Fixed Size to Compile Core Fixed Size Size Size Core Fixed Size Size Size Size Size Size Size Size		ic System Reference Library manual for actual entries.
Piece Committee Country Countr	The Foreign of the Principle of the Prin	
	Alter	
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 2	1 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
	 	
	File Description Considerations	
	File Description Specifications	
File Type	Mode of Processing Length of Key Field or	File Addition/Unordered Extent Exit Number of Tracks
File Designation	of Record Address Field	for DAM for Cylinder Overflow
End of File Sequence	Turn of File	
Line Filename Sequence File Format	Organization or Additional Area	Core Index Rewind
W/D	Overflow Indicator C C C C C C C C C	Condition 2
1/0/U/C/ Block Privile Priv	Record Length E Starting X	nuation Lines S
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 2	1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	1, 1
0 2 F		
0 3 F	┼┼┼╂┼┼╂╂┼╂┼┼┼╂┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼	┤┨╎╏╏╏
0 4 F	┼┼┼╂┼┼╂╂┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼	
06 F	╎╎╏╏╏╏╏╏╏╏╏╏╏╏	
07 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.

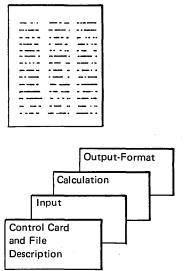
Source Programs to Object Programs

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.

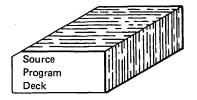
SUMMARY



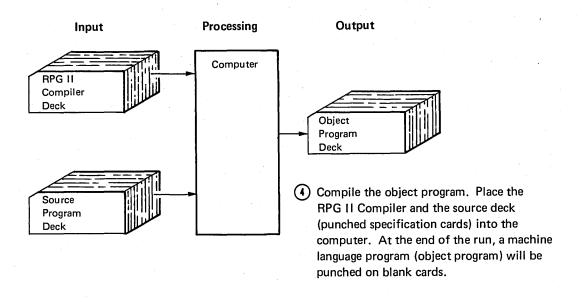
① Determine the requirements of your job.

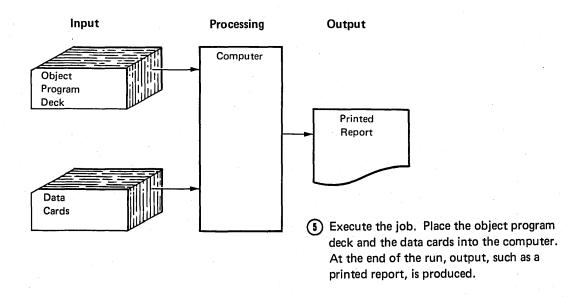
Define the input and required output. Also decide what processing must be done in order to get the proper results.

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



(3) Punch the specifications on cards.





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.



Perform detail calculations. Set resulting indicators.



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.



Move data from record selected at

beginning of cycle into processing area.



Set field indicators.

Overflow indicator on? Yes, do overflow operations and set overflow indicator off.



LR indicator on? Yes, end of job has been reached.



Perform total output operations. If overflow line has been reached, set on overflow indicator.



Perform total calculations. Set resulting indicators.



Change in control field? Yes, set on control level 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.

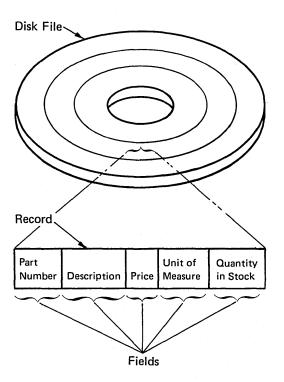
In this chapter, the operations in the RPG II program cycle are explained a few at a time. You will learn:

- 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.

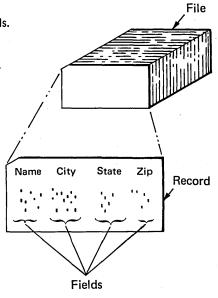
BASIC PROGRAMMING TERMS



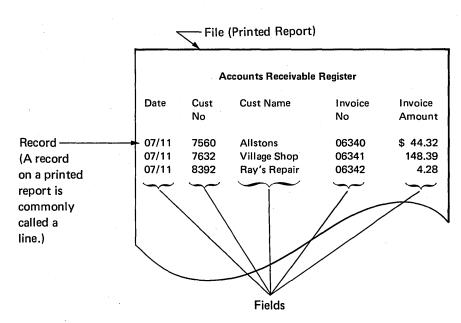
Field: An area on a record reserved and used for a particular item of information.

Record: A group of related fields.

File: A group of related records.



Card



PROGRAMMING AIDS

Record Layout Form

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

IBM,		96 (COLUMN CARD MULTIP	LE LAYOUT FORM	
	Card Name ITEM TRANSACTION RECOR	Ð			
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 3		33 34 35 36 37 38 39 40 41 42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 Print Line 2	57 58 59
	Tier 1			Tier 2	
Punch	CUSTOMER ORDER NUMBER NUMBER NUMBER	ITEM	DESCRIPTION	LIST SELLING PRICE 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	The Record Layout Form show time a file is designed. It show length of each (2). It may als	vs what fields	are in the record 1	and the exact location a	nd
Punch	field (3). There are many different Reco 96-column cards, and tape. The Form.				
Program Control	1 2 3 4 5 6 7 8 9 10 11 12 10 11 10 13 20 21 22 23 27 23 2	ס בון בס בס שי שי שי	דדן נדן זדן ודן טדן פנין טנין טנין טנין ונין טיין ניין טיין טני	סן ספן כפן די פון שטן וטן שטן פרן סדן ודן סדן שד	11 20 29.
Card					
	Card Name				
.	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 2	6 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 5	57 58 59
Print	Print Line 1		· · · · · · · · · · · · · · · · · · ·	Print Line 2	
	Tier 1			Tier 2	
Punch					
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 2	6 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

Printer Spacing Chart

Form GX20-1776

ACTERS PER	R INCH, 6 LINES PER VERTICAL INCH	DATE	i	line.	
					Ø
				, , , , , , , , , , , , , , , , , , , 	
1 2 3 4	456789012345678901234	2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4	4 4 4 5 5 5 5 5 5 5 5 5 5 6 6 6 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2	6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7	7 7 8 8 8 8 8 8 8 8 8 9 9 9
1					
3 1	┨┩╃╫┩╃╃╃╃╃╃╃╃╃╃	┼┼╎╎┧┨┨╎ ┼┼┼┼┼┼┼┼┼┼┼	╎┤╏╏╏╏╏╏ ╏	┡┧┦ ╤ ╽╏╏┩	╒┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋
14					
5 H 6		GASH RECE	IPTS REGISTER	 	3
7					
8 H 9	0,057	OUST NOTE	IWVOICE		
H 10	Ma Ma	WANE	MO	RECEIVED	TAKEN CREDI
11					
D 12	<u> </u>	**************************************		TAN SKINI KAN	XXX X X X X X X X X X X X X X X X X X
14	(CUSTNO)	(MAME) 2	(WVNO)	CCASA	(DISC) CLRE
16		+++++++++++++++++++++++++++++++++++++			╒ ╫╀╫╀╫╫
17					
7 18	TOTA	NUS		XXX XXXX XXX	·XXX·XX** XXX,XX
20	┤┤╎╎╎╏ ┼┼┼┼	╂┼╎╏╣╏╏╏╏╏	┞┪┧╏╏╏╏╏╏	(TOTOASH)	171 10/56) 14-10-10
21					111111111111111111111111111111111111111
22		Single Engl	e detail li	ned	
24		2 1111111111111111111111111111111111111			
25		Triple spac	e between lo	zst detail lin	e and total:
26	╏┪╏╏┋ ╏┼┼┼┼┼┼┼	┨┼╏╏╏ ┇		╒╏╏╏ ╏	╶┤┼╏┤╎╎╎╎╎╎ ┼┼
28			 	 	
30		r Spacing Chart indicates wh			- 5
31		n is included in the report ar			
32		erent lines are needed 🕦 ָ, ۱			
34		n to include in the lines 3			
35		nine where to place informa			
36		orrespond to print positions			
38		orresponding to the desired p			
39		o be printed 🚯 . Field nan			e what field
41	is shown (Any punctuation to be it 	used is also indicated	I on the chart (8).	A İ
42					the state of the s
44	╀┦┦┦╏╏ ┼┼┼┼┼┼┼┼┼┼	┼┼┼┼┼╏╏╏╏╏╏╏	╒╏┋┋┋	┞┦┦┦┦┦┦┨ ╃┤┼┼┼┼	
45					
46		┤┤┤┤┧╏┇┧╏╻╎╏╏╏╏╏	╎╏╏┩ ╃╃╃╃╃╃	 	┵┼╂┼┼┼┼┼┼┼┼┼┼
48	+++++++++++++++++++++++++++++++++++++	 			
49					
50	╌╏┋┋┋┋	╃┼╠╅╏╏╏╏╏╏	╒╫┋┋┋	┟┧╏╏╏╏╏	╃┼╂┼┼┼┼┼┼┼┼╂┼
┸╂┼┼	┪╏╏ ┪	┤┤┼╎╎╏╏╏╏╏╏╏		 	
1 2 3 4	4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4	Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z	7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2	5 5 5 6 7 8 9 0 1 2 3 4 5 6 7	7 7 8 8 8 8 8 8 8 8 8 8 9 9 9 9 8

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.

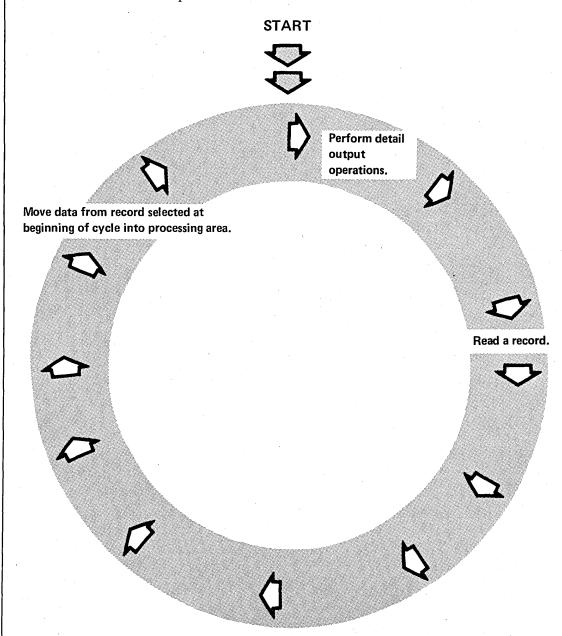


Figure 3. Three Basic Operations in the RPG II Program Cycle

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.

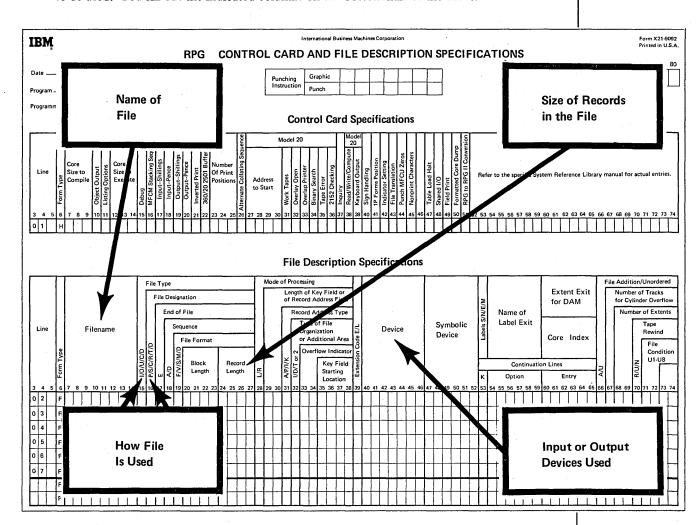
When the program is executed, the program cycle is repeated over and over. All three operations are used for every record in the input file. The term *detail output*, in the cycle operation, means that the specified output operations are performed for every input record.

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:



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.

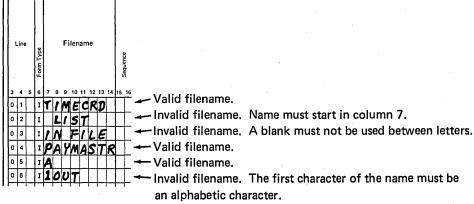
File Names

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?



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.

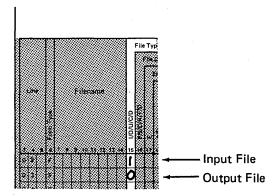
Note: In the examples in this manual, you will see device names or abbreviations of device names rather than actual RPG II code names for devices. Code names differ from system to system and from device to device.

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:

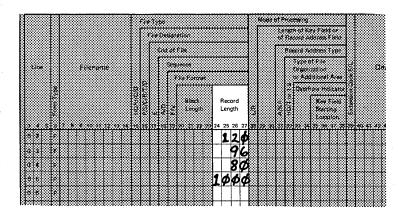


File Designation

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:



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

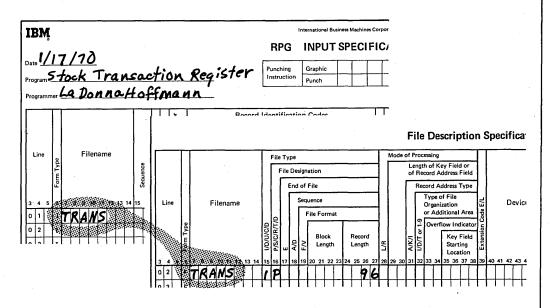
Besides describing files, you must describe the records in each file. The compiler needs this information to create an object program that will read records properly. Input records are described on the Input sheet. Information needed to describe the record in a file includes the name of the file containing 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:

			Form X21-9094
IBM	International Business Machines C		Printed in U.S.A.
	RPG INPUT SPECIF	ICATIONS	75 76 77 78 79 80
Date	Punching Graphic	Page	Program
Program	Instruction Punch		Identification
Programmer	l,		
Record	I Identification Codes	Field Location	Field
	2 3	1 1~1	Indicators
Form Type Record Mumber (1:N) Option (0) Record Identifying Indicator Oracle CIZIO CIZIO CHARACTER CATALOR OF A PARTICIPATION	1111 111	Positions Positions Line Positions P	A Characteristic Control of Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristic Characteristics Characteri
From Type Sequence Number (1-14) Option (O) Option (O) CIZIO	Position Position	From To S	I 등 등 Plus Minus or Position
Form Type Sequence Number (1: Option (0) Not (N) CZZID CZZID CHaracter Character	Not (N) Not (N) C/Z/D Character C/Z/D C/Z/D C/Z/D C/Z/D C/Z/D C/Z/D Character	- Packer - P	Matching I Field Recc Blank Blank Blank
	205		N
3 4 5 6 7 8 9 10 11 2 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 44 45 46 44 49 50 51 2 53 54 55 56 57 58 59 6	50 8 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1 I			
O CONTROL OF THE CONT			
Name of Input File Containing Records You Are Describing			
Name of Input File			
© Containing Records			Name of Fields
			on Record
o a loa Are Describing			· 2 1
0			
0 9 1			***
101			
1 2 I l catio	on of Each	T 4 D 4	
1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-		Type of Data	-{
Field O		in Each Field	┤┤┦┤╏╎╏╎╏ ┼┼┼┪
Record	' !		╏┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋┋
		Mr. goz.	╶┩╸╏╶╏╶╏╸╏╸╏

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:



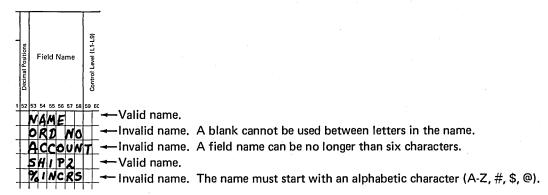
Field Names

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?



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:

Print	1 2	3 4	5	6	7 8	9	0 11	12	13 [14]	15 16	17 [18 19	[20]	21 2	2 23	24 2	5 26	27 2	8 29	9 30	31	32 3	3 34	35	36 3	37 3	8 39	40	41 4	2 43	44 4	5 46	47	48	49 5	0 51	52 5	3 5	
<u> </u>										nt Lin	ne 1											1										Pri		ine	2				
	-	_				г			Tier		- 1					_	_					÷			_						_	Tie			Т				
				- 14	ه-،	1	OT	F	e	41	8	1	· —	ΕN	4																- [L	/ 5	7	1	5	L	-//	
Punch		Cu	15 T	OM	<i>5</i> \			٠.	`	W	BE	N				o		1	7	ΕN	И	į	D	ES	sc	R	l F	7	10	N	-	PK	?/	CE	=	56 PR	2/4	:E	
		N.	υM _	181	ER	N.	UM	BE	:K	SALESMAN	NOM	ĮV.	O I	VI E	761	`						1]] .]		,	/:		
Program	1 2	3 4	5	6	1	J	0 11	12	3 14	15 16	17	8 19	20	21 2	2 23	24 2	5 26	27 2	8 2	9 30	31	32 3	3 34	35	36 3	7 3	8 3	40	41 4	2 43	44 4	5 46	47	48	49 5	0 5/	52 5	3 5	
Control Card							$\overline{}$				П		П																							\mathbb{I}		Τ	
Cara		J. J.	ш.	ــــــــــــــــــــــــــــــــــــــ				_>	~		ш		L		_ـــــــــــــــــــــــــــــــــــــ	LL		ш			ш			ш				ш	!_		ш		1	ш		-]-'			
		Ŧ	\exists	П	è	Т			Rec	ord le	de N	ficat	ion (Code	s			Ť	Т	_					Π				. [T	Ī	Т	Fi	eld		Т		•
				Ш	ndical	<u> </u>		1		Т		2	$\overline{}$	Γ		3		٦ ۱	_	H	ield	Loc	atio	ו	Н					6		5				tors			
Filena	me 1 12 13						esition	Not (N	11	1	sition	-11	Character	1 .	esition	- 1	0	Chare/ter Star, er Selec	ſ	Fro		48	To		3 Decimal Positions			Nam		S Control Level (L1-L9)	Matching Fields or Chaining Fields	1	1		/ 	Zer or Blan	o nk	Sterli Sign Posit	i
TRAN		1	T	1		17	T	-		T	П		7	1	Ţ.		1	1	7	T	П	Ť	Ž	<u>. T</u>	H	Ť	T		T		1			/			+	ÍÌ	
1 1 1 1 1 1 1	1	+	$\forall t$	+	\vdash	╁	+	+	H	++	+	+	╁	H	+	+	H	+	+	╁	H	+	H	\uparrow		c //	<	TΛ	\downarrow	+	H	H	1	H	H	+	╁	H	
+++	╁	+	╁	+	╁┼	H	$\dashv \dashv$	+	\vdash	+	+	+	+	$\dagger \dagger$	+	+	H	$\dashv \dashv$	+	+	+	+	H	+	H	26	D	NC	#	+	††	$\dagger \dagger$	-/	Н	\vdash	+	+	╁	
 	+	11	+	+		††	\top	+	H	††	\forall	+	\dagger	$\dagger \dagger$	+	+	Н	+	+	+	H	+	$\dagger\dagger$	†				ΜΛ		+	††	H	-/-		\vdash	11	+	Н	
 	++	+	$\dagger \dagger$	$\dagger \dagger$	-	$\dagger\dagger$	+	+	\vdash	+	++	$\dagger \dagger$	\dagger	H	+	+	H	+	+	+	H	t	$\dagger \dagger$	\dagger	H	1 7	E	MA	10	+	tt	Ц	/†	\dagger	\dagger	+	╁	H	
	+	$\dagger \dagger$	$\dagger \dagger$	+	\vdash	$\dagger\dagger$	11	十	\vdash	T	Ħ	\top	\dagger	$\dagger \dagger$	+	1	T	-	\top	+	H	\dagger	tt		H	DE	S	c	Ħ	\top	11,	* [T	\top	H	T	\dagger	П	
	††	11	11	\top	\sqcap	T	11	1	\sqcap	\sqcap	11	11	\dagger	Ħ	1	1	П	\top	\top	+	П	+	$\dagger \dagger$	T	H	1	S	T P	Q	1	\checkmark	Ħ	+			11	\top	П	
	++	1-1	$\dagger \dagger$	\top	\vdash	+	++	\top	\vdash	+	++	+	+	${\mathsf H}$	+	H	+	\top	+	+	+	t	$\dagger \dagger$	+	H	<u> </u>	: 7	1 2	a	~/-	††	${\mathsf H}$	+	+	\vdash	\top	╅	H	٠

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:

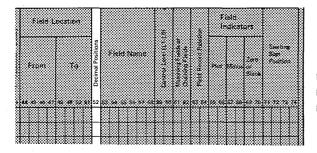
																П	2 3	4	5 6	17	8	9	10 1	1 12	13	14 1	5 16	17	18 1	9 2	0 2	1 22	23	24	25 26	T,							
													Pr	int								1-1					-1			- 1			-11		1	-11-							
																								_	Pı	int	Lin	e 1			_				-	<u> </u>							
																_	_					_				er 1		_								_							
													Pu	nch			000	us. Iui	101 48	ME E	R R	N	RD UN	E1 1B	R ER	,	8 MAN BFR	SMN)	٨	1 T	TE	TN 1E	1 3 <i>E</i> .	R				. •					
																	0	Cu	ST	N	9)	(ORA	D /	10		シチーのション	75)	(1	<i>'T</i>	E	М	Ν	0,									
		•											Pr	ogra	m	ı	2 3	4	5 E	7	8	9	10 11	12	13 1	4 15	5 16	17	18 1	9 2	0 2	1 22	2 [23]	24	25 26	6 2							
													Ca	ntro rd	ol		1	N												T	T					T							
			ļ															/-	_			_/																					_ [
	Γ				T	П	ator				R	ecor	d Ide	ntific	atio	n Co	des		_/		Т	П	/E	ield	Loca	tion	n	П					Γ					ield					11
							Indic			1		T		2				3		Z]	Binary				11101							6		İ	<u>.</u> 6	'	ndic	ator	rs			\parallel
Line	m Type	Filename		Sequence	nber (1-N)	Option (O)	Record Identifying Indicator or	Por	sition	Not (N)	C/Z/D	racter	Positi	on	Not (N)	racter	Posi	tion	(N)	Q	ker Sel	P = Pked/B = Bin	Fre	om.		Ťo)	Decimal Positions	Fie	eld	Nar	ne	Control Level (L1-L9)	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	Chaining Fields	Field Record Relation	Plus	Min	us	Zero or Blank	Ster Sigr Posi		
	For			Seg	ž	opt	Reco			No.	CZZ	S.			ž S	ਤੌ			Not	CZZ	Sta	4			$ \rangle$			Dec					8		Sag	Ë							П
4 5	6	7 8 9 10 11 12 1	3 14	15 1	6 17	18	19 20	21 2:	2 23	24 2	26	27 2	28 29 3	0 31	32 33	34	35 36	37 38	3 39	40 4	1 42	43	44 🛝	46 4	7 48	4.\5	50 51	52 5	3 54	55	56	57 58	59	60 6	1 62	63 64	65 6	67	68 6	9 70	71 72	73 7	4
1	1	TRANS	Τ	П	Т	П	T	П	П	T	Τ	П	\prod	П	T	П		П	П		Τ	П	Т	V		\Box	7	П				T	П	ı	П		П	П	T	\sqcap			1
2	I			П	T	\sqcap		П	\sqcap	1	Τ	П			Ť	П	ŀ		П	T	T	П			3		8	(20	S	T	NC		T	\sqcap		П	П	T	\sqcap	1	П	71
3	I				I	П					Ι				Ι						Ι		\Box	2		1	14	(2 8	D	N	٥									I		
4	I			П	T	П			\Box			П		П	T	П	Г		П	T		П	T	1	5		17		5 <u>L</u>	.5	M	N	П	T			П	П	Τ	П	T		11
5	1			П	1	П	T	П	\prod	T	Τ	П	П	П	1	П	T		П	\exists	Τ	П		1	3	Ž			1 7	ε	M	NO			\sqcap				Τ	77	T		\prod
1.1	1.	 			-1-	\vdash	\neg	\vdash	77		1	11	77.	1-1	1	11	1		\top	\neg	7	\Box	$\neg \vdash$	T 1	\top		\neg		7		\Box		1	\neg	\neg	\neg	\Box	1	\neg	$\neg \neg$	Π.	1	7 I

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:



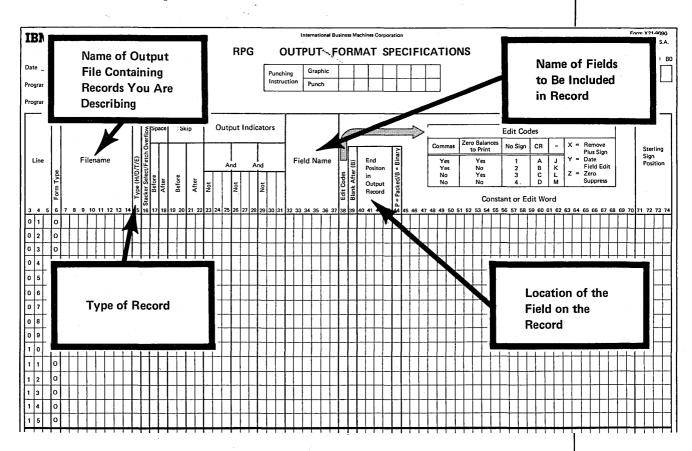
Leave column 52 blank for alphameric fields. Enter 0-9 to indicate number of decimal positions in numeric field.

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:

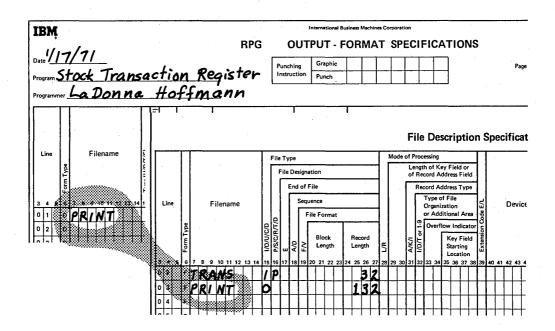


If the output is a printed report, you make additional entries describing the format of the report; that is, entries indicating the spacing and punctuation you want. These entries are discussed later under *Printed Reports*.

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:



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:

	•	ACCOUNTS RECEIVAB	LE TRANSACT	ION REGIS	TER .		
Heading Records		0	7/11/71				PAGE 01
		ST O CUSTOMER NAME	JOURNAL NO	INVOICE NO	CASH AMOUNT	INVOICE AMOUNT	JOURNAL AMOUNT
	07/11/ 7598	20 SOUND OF THE SEVENTI	E ,	063420		\$ 46.23	
	07/11/ 6338	70 OLDE VILLAGE SHOPPE		063421		89.70	
	07/11/ 6429	90 PARAGON TV SALES		063422		20.30	
	07/11/ 1226	20 CANNIZONI STUDIOS		063422		129.76	
Detail Records	07/11/ 6820	30 RAYMONDS RAPID REPAI	R		\$ 63.80		
necorus	07/11/ 7429	50 SARATOGA VARIETY			29.72		
	07/11/ 0142	80 BAKER BRADLEY & CO			43.50		
	07/11/ 8720	60 UNIVERSITY ELECTRIC			97.75		
	07/11/ 8832	90 VILLAGE MUSIC & TV	07-036				\$18.23CR
Total	07/11/ 0062	80 ALLSTONS	07-037				10.70CR
Record		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.

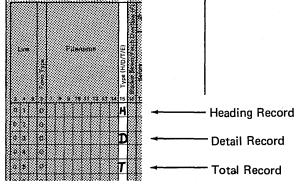
Detail records contain information about an individual item. Information in a detail record is often taken directly from an input record.

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:

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

These entries specify the record type:



Record	Type																							7		4				_	Tot	tal	Re)CO	rd		
1123	3 4 5 6 7 1	1 1 1	1 1 1 2 3 4	1 1 5 6	1 1 1	2 2 2	2 2 2 3	2 2 6 7	2 2 3	3 3	3 3 4	3 3 5 6	3 7 8	3 4	4 4	4 4	4 4 5 6	4 4 7 8	4 5	5 5	5 5 4	5 5	7 8	위이	1 2	6 6 3 4	6 6	6 6 7 8	6 7 9 0	7 7	7 7 3 4	7 7 7	7 7 7	8 8	8 8 3	8845	8
H	$\Pi\Pi\Pi$	Ш	Ш	Ш	Ш	AC	cou	M7	5 6	EC	ΕI	VA	BL	ε	718	AΝ	SA	CT	10	M	RE	G/	57	ΕR	Т	П	П	Π	П	П	П	П	П	П	Π	П	Π
2		Π	Ш	Ш	Ш	TT	П	Ш	TT	П	Π	П		Т	П	Π	l l	\Box	П	TT		\sqcap	П			П	П	П	П	П		П	П	П	Π	П	Π
<i>H</i> 3		Ш				T	П	$\Pi\Pi$	TT	П			XX	ZΧ	X/	XX			\Box	T	\Box					П		П	П		\Box	1	Ac	Æ	XX	\prod	П
4		$\Pi\Pi$			Ш	П	ПТ	Ш	Π	П	П		П	T	П	İ	П		П	П	П	П	П	П	Т	П	П	П	П		П	П	П	П	Π	П	Π
<i>H</i> 5			CUS	7	Ш	TT	ПТ	IΠ	TT	П	П			Т	Jo	UR	NΑ	4	7	MV	01	CE	П	\mathcal{C}	AS	И	П	1	Wν	0/	CE	\prod	JOL	ARIA	AL	П	П
H 6 I	DATE	Ш	MO		CU	570	MER	M S	AME							WO	П		\Box	W	0	П		4M	oυ	M7		A	MO	UN	71	\square	4MC	W	团	П	Π
7	ШП				Ш	П	ПТ	Ш	П	П	П		П		П	П		П	П	П	П		П			П		П	П	П	Шi	П	П	П	Π	П	Π
	/XX//	- XXX	XXX	X	XX	XXX	XXX	XX	XXX	XX	XX	XΧ	XX	T	XX	-x	XX		X	XX	XX	X		ŠΧ	XΧ	. X	X	\$	XX	χŀ	XX	9	XX	(,)	XC	R	Π
9	ШП				Ш	Π		Ш	\prod	П	П			Ţ					\Box	Π		Ш		Ш	I		П				Ш	\prod	\coprod	\prod	П	\prod	П
7 10	$\Pi\Pi\Pi$	Ш			70	TAL	S	Π	\prod	П										\Box	Ш		\coprod	ŝΧ	XX	·Χ	Χ×	\$	XX	χ·	XX		BX)	\odot	XC	RX	
11	$\perp \perp \perp \perp \perp$	Ш	Ш		Ш	Ш	Ш	Ш	П	П	Ш		Ш			Ш	Ш			П		Ш	11	Ш		Ш	Ш	Ш	Ш		Ш	Ш	1	Ш	Ш	Ш	Ш
12		Ш		Ш	Ш		Ш	Ш	\prod	SI	MG	3	S	PA	CE	A	LL	D	ϵ 7	91	4	411	VΕ	SL			Ш	Ш	Ш	Ш	Ш	Ш	Ш	Ш	Ш	Ш	Ш
13	$\Pi\Pi\Pi$	$\Pi\Pi$			Ш	П			TT	П						П				П	П		П		T			\prod	Ш		Ш	\coprod	Ш	\mathbf{H}	\prod		Π
14	ШШ			\coprod	\prod	Π	Ш	Ш	\coprod	Do	$u_{\mathcal{B}}$	LE	S	PA	ϵ	8	ΕŢ	$w\varepsilon$	EM	D	en	AI	4]	41	NE	5	A٨	0	70	74	4.	П	Ш	Ш	Ш	Ш	Ц
15	$\Pi\Pi$	$\Pi \Gamma$				m II				П	\prod									\prod		Ш	П		\perp				Ш		\coprod	\coprod	\coprod	\coprod	П	П	Ц
-1.11	11177	111	ПТ	1-1-1	т	1.1	117	П	177	T	1				1	П			77	T	7	17	17			1	ı F	1	1 F	' ₁□	1.1	1 1	1 1	1 T	1.1	1 [ı r

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:

IB	M																								_				-	Inte	erna	tion	al B	Busin	ess	Mad	hin	es C	orp	orat	tior					_							_
																						R	P	G			(οι	ΙT	P	U	T.	. F	=O	R	M	ΙΑ	T	S	SΡ	Ε	C	IF	10	ÇΑ	ıΤ	IC	N	S				
Date	_					_																			-	Pı	unc	ning		T	Gra	phi	c	Τ		Τ	1	_	Т		Τ		Γ	٦		7							
Progr	am																		_									ıctic		1	Pur	nch		Ť		1	7	_	†	_	t		T	1		1							ľ
Progr																																				_		_							_	_							
_		_	_			_		_	_	Т	_	ΞĪ		_		_	_	7	_		_	_	_					Т		_				Т		-		_							_	-					_		_
1										ı	ľ	NO.	pa	ce		Sk	ip	-		0	ıtp	ut	In	di	cat	or	s	l						L									٨	>	_	_						_	E
										١		Overflow	1	-		_		-						-	T			ł							\lceil	7				1	ry	1	~		C	om	mas		Zero		alan rint		١
Lir	ne				Fil	ena	me			10/1		Stacker Select/Fetch									Ar	nd		Α	 Ind				Fi	eld	I N	am	ie		إ	<u> </u>		nd osit	on	١	- Binary					Y		T		Y			Γ
		Type								T (U.V.) (T./C)		Select	١	Į.	Ortogo	200	١.	1			П		Γ	Γ	1.	Τ	T	1						18		Blank After	ir				ad/B					N	0			Y	es		l
ı		Form									3	cker		After	9	0	After	Į	Š			Not			Š	1		l						Edit Codes		lak		ecc		١	Packed/B						_				_		_
3 4	4 5	١	1	8	9	10 1	1 1	2 1	3 1	1				18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	1 3	5 3	6 3	1	1	1	0 4	11 4	2 4	13	₽ P =	45	46	47	48	49	50	51	52			nsta 55	
0 1	\neg			E					T	1		T	1					7	٦	-			T		T	T	T	Ť	Ī	Ť	T	T	T	Ť	T	1	T	T	T	7			Г	Γ	Τ	Τ	Γ	Γ	Γ	Π	ľ		_
0 2	2	0			1	1	Ť	T	Ţ	T	1		1						٦				Γ		T	T		c	U	5	7	1	VC	1	T	T	T	T	1	1				Г	Т	Γ			Γ		Γ		Τ
0 3	3	0	П		1	T	1	Ť	T	Ť	1	1	1	7					٦					T	T	T	Ţ	_	_	_	7	1/	_	-1-	Ť	T	T	1	T	1		_	Г		T	T	T	T	T	1.	ľ		_
0 4	4	0	П		1		T		1	Ţ	1	T	1				П	٦					Γ		Π	Ī	T	D	ε	S	5	3	T	T	T		1	T	T				Γ		1	I					Г		
0 !	5	0		П	T	T	1	T	T	T	1	T	1										Γ		T	T	T					2 6	=	T	T	1	1	1	T	7			Γ		T		Γ		Γ	Γ	Γ	П	
0.6	R	10	П	П	7	7	+	1	Ť	Ť	t	7	1	7			П	┪	┪		_		1	T	T	1	Τ	Ť	ŕ	Ť	T	T	1	T	Ť	†	7	\top	7	7	Т	П	Г	Τ	T	T	1	T	T		Τ	П	_

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:

	6 LINES PER VERTICAL INCH)	PAGEFold back at detted line. DATE
The common Country C		
Date	TXXXXXX	
Edit Codes Commar / Zero Balances No Sign CR -	Date	Punching Graphic Page Page
	Commonwell Com	Field N. ne Field

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 RPG OUTPUT - FORMAT SPECIFICATIO	NIC	Form X21-9090 Printed in U.S.A.
Date	 	Punching Graphic Instruction Punch	Page Program Identification	76 77 78 79 80
Program		runcn		
Programmer				
	Space Skip Out	t Indicators	Edit Codes	
		Commas	Zero Balances No Sign CR - X = Remove	Sterling
Line	Filename (1)	And Field Name End Series Yes	Yes 1 A J Y = Date	Sign Position
ad.		Positon Yes	No 2 B K Field Edit Z = Zero Suppress	
Ę	Type (H/ type (H/ type (H/ Before Before After After	Bedit Codes Backed B Hold Meser Are Best Are Be	Constant or Edit Word	
3 4 5 6 7	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 2	27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50		39 70 71 72 73 74
0106	REPORT D 1			
0 2 0		CUSTNO 21		
0 3 0		ITEMNO 32		
0 4 0		DESC		
0 5 0				
0 6 0				
0 7 0		To single space the detail lines of a repo	rt, a space	
0 8 0		specification of 1 must be specified in c	olumn 18.	
0 9 0				
1 0 0				

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:

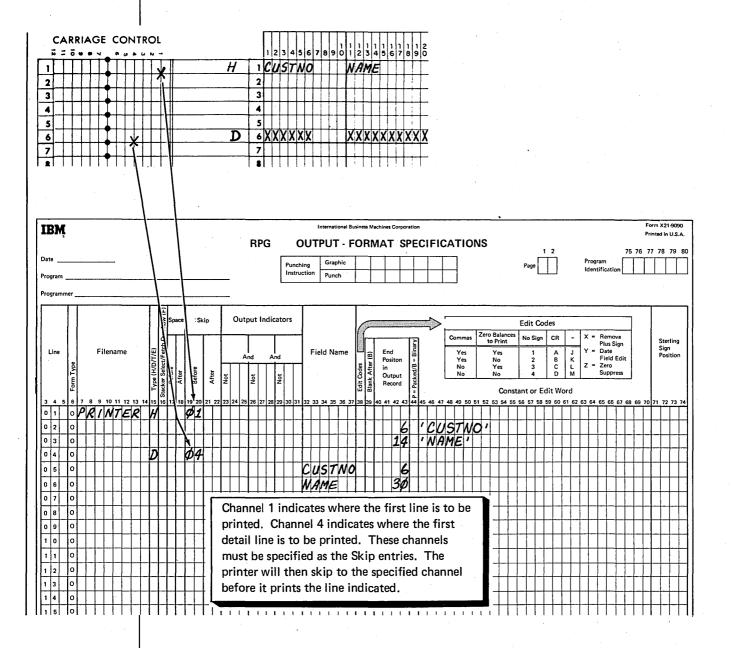
IBM	ļ								•			RI	PG		c				Business FOR					FIC	:A7	ГΙΟ	NS	 S			-	1	2						75.7	Pri	nted in	1-9090 U.S.A. 79 80
Date _														٦	unch	ina	Gra	phic	T		1	T	Ţ	T						Р	age	$\dot{\sqcap}$	Ì			rogr		F	<u></u>	Τ.		<u> </u>
Program										_					nstru		Pur	nch				I								•	90	Щ			10	denti	ificat	ion [1_1	
Programmer																																										
						Overflow (F)	pace	s	kip		Out	put	Indi	cator	rs					1	•			>	Cor	nmas	Ze	ero Bal			it C	Code	es CR	-	T×		Rem		7		Ste	rling
Line	Type		Filena	ime	(H/D/T/E)	Select/Fetch	ie.	Before	ler l		7	and	A	nd L	Τ	Fi	eld N	lame	sopo;		End Positon in Output	8,8			,	Yes Yes No		Ye: No Ye:	s		1 2 3 4		A B C	J K L	1	′ = ! =	Plus Date Field Zero Supp	e d Edit			Sig	
3 4 !	Form		9 10 1	11 12 13	14 15	Stacker	Afte		After	No.	24 25	V N 26	27 28	No.	10 31	32 33	34 3	5 36 3	Edit		Record	P = Pack	45.4	6 47	48 4	9 50	51 5		Cons	stant					2 63	64	65 66	5 67		9 - 70	71 72	73 74
0 1	0			TER			T	10		\sqcap		П	T	П		Т	П	П	T	T	П	\top	П		П	T	П	T	Τ	П	T	Τ	Ĥ	T	T	П	T	T	П	П	Τ	Ш
0 2	0	П		TT	П	П			П	П		П	T	П					П		6	5	17	T R	AA	VS.	4	CT	10	M	1	RE	ρ	OF	27	1		П	П	П		Ш
0 3	0					П	I		П																			П	T	П										\prod		Ш
0 4	0		Ш	⊥Ի	<u>' '</u>		<u></u> -	<u></u>	1 1	* 1			<u> </u>			<u>' </u>	<u></u>			<u>. </u>	1 1 1	<u> </u>	<u>' '</u>	<u></u>		- '		1. 1		•	à				L				Ш	Ш		
0 5	To print a heading on line 10, a Skip specification of 10 in columns												Ш																													
0 6																																										
0 7	0			\perp l					ne '															ľ		•														П		
0 8	0			\mathbb{I}	0							- 1	• • •					9	•														П	1.		П				\prod		
0 9	0		1		/A .	SE, SE	*			CASE I	,		Ţ	, ,	10.00				1 1					1000	WORKSON.		675°4	7 (4)	3e4 S	*		T		T			T		H		T	\Box

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:



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.

	_	_	Sign For	Negati	ve Balance	_			
Edit Code	Commas	Decimal Point	No Sign	CR	– (Minus)	Zero Suppress	Print Out On Zero Balance		
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		
А	Yes	Yes		CR		Yes	.00 or 0		
В	Yes	Yes		CR		Yes	Blanks		
С		Yes ·		CR		Yes	.00 or 0		
D		Yes		CR		Yes	Blanks		
J	Yes	Yes				Yes	.00 or 0		
К	Yes	Yes			-	Yes	Blanks		
L		Yes			_	Yes	.00 or 0		
М		Yes			_	Yes	Blanks		
X*				•					
Y**						Yes			
Z***						Yes			

^{*} The X code removes the plus sign of the field.

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

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.

^{**} 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:

^{***} The Z code removes signs and suppresses zeros.

		ield Length nd Digits	1769532	02	00	000	041345			
	C	ield Character- stics	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			
	Ĭ	2	17,695.32	.02			41.345			
		3	17695.32	.02	.00	0	41.345			
		4	17695.32	.02			41.345			
		Α	17,695.32	.02CR	.00	0	41.345			
		В	17,695.32	.02CR			41.345			
l į		С	17695.32	.02CR	.00	0	41.345			
Edit (Codes	D	17695.32	.02CR			41.345			
		J	17,695.32	.02-	.00	0	41.345			
		к	17,695.32	.02-			41.345			
		L	17695.32	.02-	.00	0	41.345			
		М	17695.32	.02–			41.345			
		×	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:

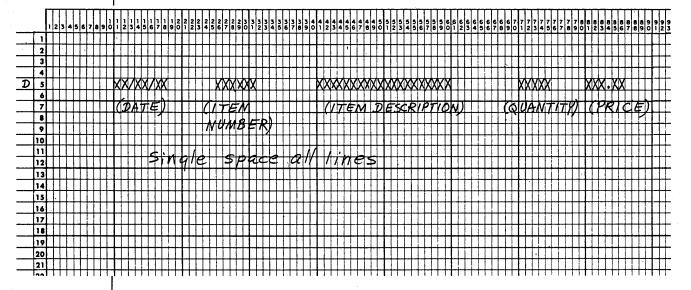
Print	1 2 3 4 5 6	7 8 9 10 11 12	13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 3	2 33 34 35 36 37	38 39 40 41 42 43 44 45 46 47 48
			Print Line 1	<u>'</u>	Print Lir
			Tier 1	1	Tier 2
Punch	TRANSACTION DATE	(ITEM NUMBER	ITEM DESCRIPTION	QUANTITY	PRICE
Program Control Card	1 2 3 4 5 6	7 8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

two decimal positions

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

07/23/70	413010	CHOOL BOX LOOA 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	CHOO5 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:



IBM	International Business Machines Corporation	Form X21-9092 Printed in U.S.A.
Programmer L. K. Hoffmann	CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS Punching Graphic Punch Punch Punch Punch Punch	75 76 77 78 79 80 Program Identification
Core Size to Compile Core Size to Compile Core Compile Core Compile Core Compile Core Compile Core Compile Core C	File Description specifications describe the files used: one input a The input file, consisting of 96-column transaction records, is give TRANS. I in column 15 indicates input file; P in column 16 indicates; 96 in columns 24-27 tells how long the records are. The devict the file is a card reader (the exact code name you enter in columns upon your system). The output file is a printer. Filename is TRA print line (record length) is 96 positions long.	n the name ates primary ce used to read s 40-46 depends
Line Filename File Type File Type File Designation End of File Sequence File I	Notice that entries have been made in columns 1-2 and 75-80 in the hand corner of each sheet. Page number (columns 1-2) along with (columns 3-5) help you keep your specifications in order. These nearly to arrange them in proper sequence by the numbers in column umns 75-80 identify your program.	line numbers numbers are order, it is
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 0 2 FTRAMS / P	Location Location	2 63 64 65 66 67 68 69 70 71 72 73 74
IBM	International Business Machines Corporation	Form X21-9094 Printed in U.S.A.
Date 1/10/71 Program Transaction List Programmer L.K. Hoffmann	Punching Instruction Punch Punch Page 2	Program
Line Filename Filename Post Number (1-N) Post Nu	Record Identification Codes 1	11111
Input specifications describe the named first. The name entered to the input file on the File Descrecords are then described. Mos fields is taken from the Record five fields, all of which are needed are described, starting one line be entries include field location, field indicates type of data (alphamer in arithmetic operations or edited	e input records. The input file is must be the same as the name given cription sheet. Fields on the input to of the information for describing Layout Form. The record contains ed for output. Therefore, all five fields selow the file name. Field description eld name, and decimal position, which cic or numeric). Any field to be used at must be numeric. The output shows is, three fields have an entry in column	

IBM	[Inte	natio	nal B	usines	s Ma	chine	s Cor	pora	tion																					X21-9	
	<i>,</i> .	_	/ -	,											-	RP	G		(Οl	JT	Pl	JT	- F	O	RN	1A	T	SP	EC	CIF	:IC	A.	ΓΙΟ	N	S					1 2							75	76	77	78 7	79 80
Date _						,						,							Punc			¢	iraph	ic		Τ	T			Т									Pag	ge 🗸	03				ogra	ım ficati	[L	1 5	5	r	\prod
Program														_				L	Instr	ucti	ion	P	unch				\perp				\perp											1		iu	:11(1)	iicati	011		- -	- 1.		للل
Program	mer_	L	. <u>k</u>	<u>(. </u>	H	0	£	£	M	a	n	n		-																																						
								ow (F)	Space	ce	SI	cip		0:	utpi	ut li	ndic	ato	rs	T						_						<u>'</u>							Edi	t Co	des				_	_	_	7		Γ		
								Overf	H	+		_	+		1			_		4					1	_					~	_	Co	nmas	Z	ero E	Balan Print	ices	_	Sign	1	R	-	x		Remo		-			Sterli	lina
Line			Fi	lena	me		ē													İ	Fie	eld	Nar	ne	Ш	<u></u>	Er	nd	ı	Sinary			-	Yes	十		es e			1	+-	Α .	J	Y	= [Plus S Date	-			1	Sign Posit	1
1	ype						De (H/D/T/E)	cker Select/Fetch	9		ø	١.	ŀ	Т	An	а Т	Ť	nd T	Т	┨					s e	fter (Po in	sito	1	/B = E				Yes No		١	lo 'es			2	(B	K L	z	- 2	Field Zero		١		l	OSIL	
	rm T						YDe (F	ker Se	Befor	ATE	Before	After	Ş	100		ž		ž							Edit Co	Blank A		utpu		acked		I		No		!	10			4.	يــــــ	<u> </u>	М	L		Suppi	ress	_		l		
	١	, ,					ءً ا	Sta				21 2		2 24	,,,			_					ar .		38	- 1	40 41		42				40	9 50			Co				dit					S5 66					72 7	
0 1	0	TI							Ϋ́	1	720	Ť	1	3 2 7	23	20 2.	20	25	30 3	1 3.	<u> </u>	1	ΪÏ	T	38	39 1		T		7	3 40	Ť		3 50	Ï	52 5	3 54	1	1	7 58		1	1 62	I	Ť	7	ΪÏ	Ĭ	T	Ť	ή	
0 2	0	Ť	ľ	7	Ť	Ħ	Ī	П	Π	7	T	Ħ	1	П	П	7	T	П	1	1	A	1	E	1	Y	1	+	1	8	7	Ī	П		T		T	T	П	\top	T	П	1	T	П	\top	1	Ħ	\exists	1	T	П	П
0 3	0										T		1		П								M	NC	1			3		1				T.													П		Ť	T	П	\Box
0 4	0										I			$\overline{}$			\Box)E							6	Ø					I									L		I	\perp			\perp			
0 5	0		\perp			Ц		Ц	Ц		L	Ц		\perp				Ц			7				Z			7				Ш			Ш			Ш		\perp	Ш		L	Ц	\perp	\perp	Ц		\perp		Ш	
0 6	0						\perp	L	Ц									Ш		F	R	1	C	=	3			8	6																	\perp	Ш			L	Ш	
0 7	0	Ц		_		Ц	1	Ц	Ц	1	1	Ш	1			_	L	Ц	1	1		L	Ц	1	Ш	1	_	1		_		Ш		1		_	L	Ц	1	\perp	Ц	_			4	_	Ц		4	L	Ц	Ш
0 8	0	Ш	\perp	4	4	\perp	1	Ц	Ц	1	1		1	\perp	4	4	L	Н	4	1	4		Ш	1	\perp	4	4	4		4		Ц	4	1	Ц	4	_	Ц	4	1	Ц	4	╀	Ц	4	4	\sqcup	-	4	L	Ц	4
0 9	P	Н	\perp	4	+	\sqcup	+	Н	H	4	\downarrow	1	+	\perp	4	4	╀	Н	+	+	╀	L	Н	1	Н	4	4	╀		4	+	Н	4	+	Н	4	+	Н	+	+	-	+	4	Н	4	\downarrow	\dashv	4	+	╀	Н	4
1 0	0	Н	+	+	+	\dashv	+	Н	H	+	+	H	+	+	H		+	H	+	4	+-	<u> </u>	\vdash	+	\mathbb{H}	\perp	+	+		+	+-	H	+	+	$\left \cdot \right $	+	+	\vdash	+	+		+	+	H	+	+	H	\dashv	+	╀	H	
111	0	+	+	+	+	\dashv	+	Н	H	+	+	H	+	+	Н	+	+	Н	+	+	+	H	Н	+	Н	4	-	╀	H	+	+	\vdash	-	+-	H	+	+	\vdash	+	+	H	+	+	H	+	+	\dashv	\vdash	+	╀	H	+
1 3	0	H	╁	+	+	Н	+	tı								-	_					_											_	_		<u> </u>		-	<u> </u>	<u> </u>					_	_	_			ì	'+	\mathbb{H}
1 4	0	+	╁	+	+	\vdash	+	ŧΙ		() Du	tρι	ıt.	-Fc	orr	ma	t s	рe	ci	fic	cat	tic	ons	d	esc	ri	be	h	οv	v a	n	οι	ıtç	ut	re	ecc	orc	l۷	/ill	lc	ok	ζ.	TI	he	O	utr	pu'	t		ı	+	+
1 5	0	H	t	\dagger	╁	H	╁	tΙ		f	ile	is	n	am	ed	I.	D	is	en	ite	ere	d	in	C	olu	m	n	15	to	o i	nd	lic	ate	e a	d	eta	il	lin	е	(oı	ne	pr	in	tec	1 t	for	•				+	Ħ
H	0	H	\dagger		†	Ħ	十	tΙ		e	ve	ry	Cá	ard	re	eac	i).	P	۱ ۲	Ιi	n	CC	lu	m	n 1	8	sp	ec	ifi	es	si	ng	le	sp	ас	ing	J.	Αl	ll f	iel	ds	to	o b	е	pr	int	tec	ŀ			+	\forall
\prod	0	T	T	\top	T	П	Ť	tΙ		а	ire	nc	w	/ lis	ste	ed,	O	ne	р	er	liı	ne	, s	ta	rtir	ng	10	ne	lit	ne	be	elo	w	en	tr	ies	d	esc	ril	bir	ng	th	e e	ent	tir	e l	ine	e.			7	\top
	0	П	Т				Т	Ħ				d P																																						ı		
	0							II				ord																		_					_										_							
Ш	0				\prod		Ι					he		_								-			-																							t		1	\Box	
						ı		-				zer							-																					•								-		*	_	
												sei									•																											n				
						- 1						d v																							ıu.	30	• •			, .	,G1	ı ıy	a	ادی	ρŪ	,311	.10			1		
										'	161	u v	VI	LI I	۲V۱	, 0	ut	-01	1116	ais	١,	۸,	١٨.	^>	ν,	H	; C (12	IIU	, 0	UII	1111	ıd:	٠.																		

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).

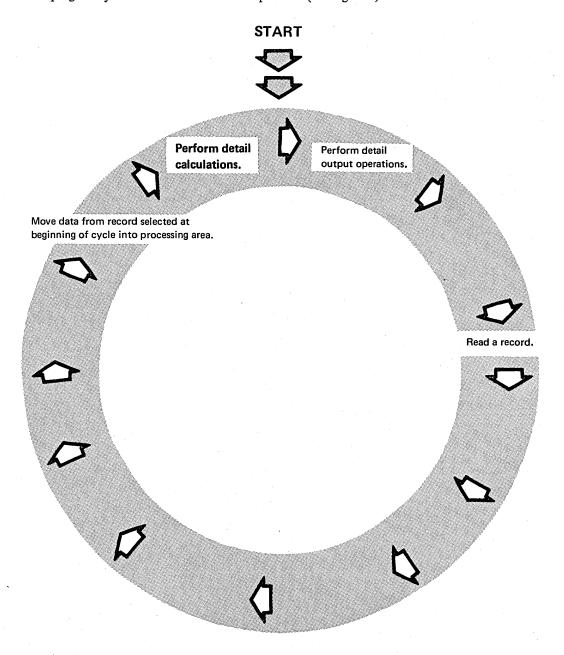
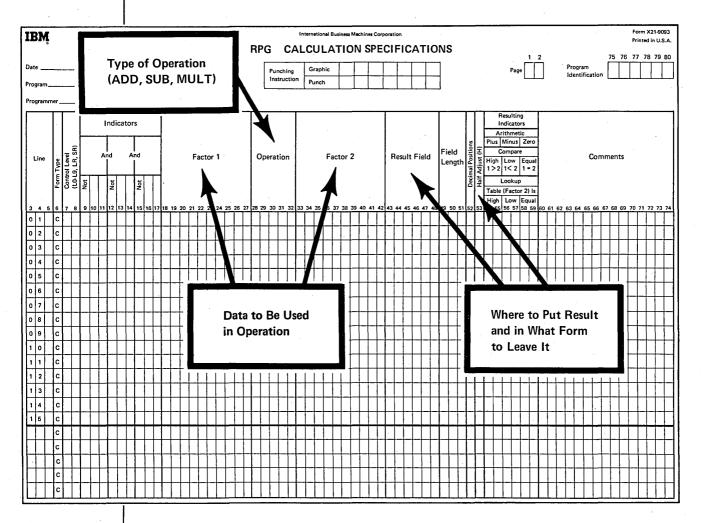


Figure 6. This is a basic program cycle showing the addition of detail operations.

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:



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:

6.17	18	10	20	F	ac	eto	r,1	4 3	5.5		,,	0	pe	rat	ion	1	33 3	14	35	F	act	or i	2	40	41	42	R	esu	ilt i	Fie	eld	- 4	Field Lengt	h	 		Resulting Indicators Arithmetic Plus Minus J Zero Compare High Low Equal 1 > 2 1 < 2 1 = 2 Add a constant 1 to CARDS at detail calculation
	1	13	Ď	<u> </u>	ĺ	Ī	Ī	Ï	Ĭ	Ť		4	D,	D	Ϊ	-		Ā	R	D	S			Ĭ	<u> </u>	**	Ĉ/	Ą	R	D.	5	֓֟֟ <u>֚</u>	_			ti	time, thus providing a count of the records processe
	D	ī	A	M								M	ע	L'	T	,	3	•	1	4	1	5	9				c .	11	R	c	1	1	سد		_		Calculate the circumference of a circle by multi- olying diameter by the constant 3.14159.
	ર	T	4									D	1	V			L	2									Do	၂ ၁ <u>i</u>	Z	E	N	1	*	_	-		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	SUBTRACT
Factor 2 is added to Factor 1 and	Factor 2 is subtracted from Factor 1 and
the sum placed in the Result Field.	the difference placed in the Result Field.
Factor 1 Operation Factor 2 Result Field	Factor 1 Operation Factor 2 Result Field
7 10 10 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 49 4	7 18 19 20 21 22 23 24 25 26 27 23 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 47 47 47 47 47 47 47 47 47 47 47 47 47
AMTZ ADD AMTI TOTAMT	DEDUCT SUB TOTAL DIFF
Either line adds the two amount	The order of subtract operations is
fields. The order of the fields makes	important. The bottom line would
no difference in addition.	not produce the desired result.
MULTIPLY	DIVIDE
MULTIPLY Factor 1 is multiplied by Factor 2 and the product placed in the Result Field.	DIVIDE Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero.
Factor 1 is multiplied by Factor 2 and	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field.
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field.	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero.
Factor 1 is multiplied by Factor 2 and	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field.
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field.	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero.
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field. Factor 1 Operation Factor 2 Result Field F	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero. Factor 1 Operation Factor 2 Result Field F
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field. Factor 1 Operation Factor 2 Result Field F	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero. Factor 1 Operation Factor 2 Result Field F
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field. Factor 1 Operation Factor 2 Result Field F	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero. Factor 1 Operation Factor 2 Result Field F
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field. Factor 1 Operation Factor 2 Result Field F	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero. Factor 1 Operation Factor 2 Result Field F
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field. Factor 1 Operation Factor 2 Result Field Result Field Factor 2 Result Field Factor 2 Result Field Factor 1 Operation Factor 2 Result Field Factor 2 Result Field Factor 2 Result Field Factor 3 Factor 2 Result Field Factor 3 Factor 2 Result Field Factor 3 Factor 2 Result Field Factor 3 Factor 3 Factor 3 Factor 3 Factor 3 Factor 3 Factor 3 Factor 4 Factor 3 Factor 3 Factor 4 Factor 3 Factor 4 Factor 3 Factor 4 Factor 5 Factor 5 Factor 7 Factor 6 Factor 7 Fact	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero. Factor 1 Operation Factor 2 Result Field F The order of the fields in divide operations is important. The bottom
Factor 1 is multiplied by Factor 2 and the product placed in the Result Field. Factor 1 Operation Factor 2 Result Field 7 18 19 20 21 22 23 24 25 28 27 28 29 30 31 32 33 34 35 38 37 38 39 40 41 42 43 44 45 48 47 48 41 HOURS MULT RATE GRSPAY RATE MULT HOURS GRSPAY Either line multiplies the hours and rate	Factor 1 is divided by Factor 2 and the quotient is placed in the Result Field. Factor 2 cannot be zero. Factor 1 Operation Factor 2 Result Field F T 18 19 20 21 22 23 24 28 28 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 48 47 48 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 42 43 44 45 48 47 48 41 41 41 41 41 41 41 41 41 41 41 41 41

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	International Business Machines Corporation
RPG INPUT SPECIFICATIONS	RPG INPUT SPECIFICATIONS
Date _ Punching Graphic Program _ Instruction Punch Programmer _	Date Punching Graphic Page Program Punch Program Punch
Field Location Line File File Field Name Field	*fication Codes 3 Field Location
Position 2 15 25 35 35 35 35 35 35 35 35 35 35 35 35 35	Line File To
0 2 1	0 2 1
International Business Machines Corporation	International Business Machines Corporation
RPG CALCULATION SPECIFICATIONS	RPG CALCULATION SPECIFICATIONS
Punching Graphic	Punching Instruction Punch Punch
Factor 1 Operation Factor 2 Result Field Length	Factor 1 Operation Factor 2 Result Field Field Length
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 47 43 44 45 46 47 48 49 50 51 52 5	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 5 QTY
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.	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:

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							Resulti Indicat		
T				i i		<u>ا</u>	Arithme Plus Minus		
And And	Factor 1	Operation	Factor 2		Field :	ust (H)	Compa High Low	Equal	
					Length	alf Adj	1 > 2 1 < 2		
Not]=	Table (Fact		
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 5	2 53	High Low 54 55 56 53		
	PRICE	MULT	.06	DISCNT	5	ZH		Ш	

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.

4.5088 Slashed digits are dropped since only two decimal positions are required.

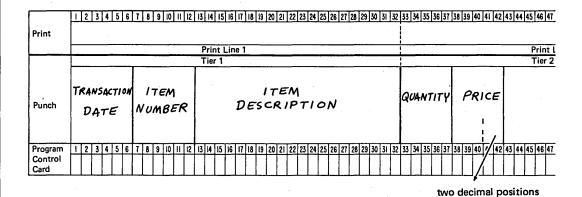
Job 2: Doing Simple Calculations

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:



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

07/23/70	413010	CHOOL 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:

	,	2	3 4	5	6	7 8	9	3	1 2	3	1	3	;	3	9	Ž	2 2	23	2	2 2	2 2	28	29	3	3	3 3	3 4	3 5	3	3 7	3 3	1	;	4 2	4 3	4 4	4	4	4 4	5	5	5	5 2	5 5	5	5 5	5 5	60	5 6	5 6 3	6	6 6	6	6 8	6 7	7	7 7 2 3	7.4	7 7	7,	7 8	7 (9	5 5	8 2	8 3	9	8 6	8,7	8 5	3 9	,	9 :
1	1	\prod	Ι	I	П	Ι		\exists	Ι	Ι		I	Ţ			Ī	I	I			Ι	Ι		П	Ī	I	T	Τ			I	Ι	Ι		П	T	I		I	Ι	П		I	I		T	Τ	П		Ĺ	П	T				П	Ι				П	1	T				Τ	П	T	T		Π
2	Ł	Ш			П		П	1	Ι	L			Ι	L						I		L			Ι	Ι	Γ	L			I	Ι	Ι			Ι	Ι		I	I		П	T	I	П		Γ	П		T.	П	T	П		Τ		T	П	T	Τ		Т	T			I	I	П	Τ	Т		Π
3	L	Ц	l				П	1		L			Ι			Ι	Ι	L		Ι	L				\Box	I	L	L			Ι	Ι	L			Ι	Γ		I	L	\Box		I	I	П		Ι		П	Ι	П		\prod		Τ	П	Ι		T	Γ		I	Ι			Ι	Τ	\prod	Ι	Γ	П	\prod
4	L	Ц	1	Ц	Ц		Ц	_	1	L	П	_	I			_	\perp		\Box		L	L				\perp	I	L			Ι	Ι	L		Li	I							I	Ţ				П	П	I		1	П	\square	I		Ι			I		Ι	Ι			Ι	Ι		m I	Γ		П
5	L	Ц	1	Ш	Ц	⊥	Ц	1	1	L		1	1		Ш	1		L		1	L	L	L		1	Ι	Ι	L			1	l	L			L	l		1		П		Ι				L	П		L	П		П				Ι	П	l	L		Ι	Ι			I	1	\prod		Ι	Ц	Ш
D 6	Ł	Ц	┸	Ц	Ц	1	Ц	_1	<u> </u>	V	М	Х.	42	ŲΧ	Ц	_	\perp	L		_	Ø	<u> </u>	X	M	X	Ι	L	L					X	Χ	X	Φ	Œ	X	XI)	Œ	X	Ø	Ø	XX	Х	Ω	(X	W	Ц	L	Ц	ΧD	J	X	XΧ		1	X	XI)	∢.	X	X	I	X	XI	. >	Ø	X		坙	L	Ц
	1	Ц	1	\perp	Ц	L	Ц	4	4	L	Ц	1	1	L	Ц	1	1	L	Į	\perp	1		Ц	Ц	1				L		1	1	Ĺ		Ц		L		Ι		Ц		\perp		П		L	П	\Box	L	П					Ц		П		L		I	I			Ί	Γ	П	\perp	I	Ц	Ц
•	4	Ц	1	Ц	Ц	1	Ц	4	K	2	И	I	E	1	Ц	4	1	\perp	4	4	4/	U	Д	/4	ΙΔ	11	3/	1	U	Ц	1	1	L	(Z	Œ	11	1	24	1	D	R	1	P	7	Φ	W	Ш	Ц	10	Ø	Щ	М	/ 7	7	II	y	Ц	ŀ	R	L		٩.	L	Q	57	Y 7	ΕÌ	VD	E		Ц
9	-	Н	4	Ц	Ц	1	Ц	4	4	L	Ц	4	Ţ.	1	Ц	1	1	L	Ц	4	1	L	Ц	Ц	4	1	1	1	Ĺ	Ц	1	\perp	L	L	Ц	1	L	Ц	1	L	Ц	Ц	╧	\perp	Ц	4	ľ	Ц	Ц	L	Ľ	\perp	Ц	Ц	1	Ц	1	Ц	1	┸	Ц	4	1	Ц	Ц	4	25	57)_	Ц	H
10	-	Н	4	Н	Н	4	Н	4	4	Ļ	Н	4	4	L	Ц	4	4	ŀ.	Ц	4	1	1	Ц	Ц	4	1	1	↓	L	Ц	1	4	Į.	L	Ц	1	1	Ц	1	1	Ц	Ц	4	4-	Н	4	4	Ц	4	1	11	4	Ц	Ц	1	Ц	1	Н	4	1	Ц	4	1	Ц	Ц	7	1	П	1	1	Ц	Н
111	-	Н	4	Ц	Ц	1	Н	4	4	L	Ц	4	1	L	Ц	4	\$	1	4	ф	1	\$	Ц	Ц	S,	p	k	10	2	2	4	41	1/	L	1	14	4	2	با5	Ļ	Ц	Ц	4	1	Ц	Ц	1	Ц	Ц	4	Ш	4	Ц	Ц	4	Ц	4	Ц	4	╀	Ц	4	4	L	Ц	4	1	Ц	4	┸	Ш	Ц
12	-	Ц	4	4	Ц	1	Ц	4	4	L	Ц	4	1	L	Ц	4	4	L	Ц	1	1	┸	Ц	Ц	Ľ	1	l	┸	Ц	Ц	1	1	L	L	Ц	1	1	Ц	4	┸	Ц	Ц	┙	1	Ц	Ц	┸	Ц	Ц	┸	Ц	1	Ц	Ц	1	Ц	1	Ц	4	┸	Ц	4	1	L	Ц	4	⊥	Ц	┙	L	Ц	Ц
13	-	Ц	4	1	Ц	4	Ц	4	4	Ļ	Ц	4	4	1	Ц	4	4	L	Ц	4	1	Ļ	Ц	Ц	4	4	1	Ļ	L	Ц		1	L	L	Ц	4	1	Ц	4	╀	Ц	Ц	4	1	Ц	Ц	ļ	Ц	Ц	4	Ц	4	Ц	Ц	4	Ц	4	Ц	1	Ļ.	Ц	-	1	Ц	Ц	4	1	Ц	1	Ц	L	Ц
14	-	Н	4	╀	Ц	4	Ц	4	4	Ļ	Ц	4	4	L	Ц	4	4	L	Ц	4	1	Ļ	Ц	Ц	4	4	1	Ļ	L	Ц	1	1	L	L	Ц	\perp	Ļ	Ц	\perp	L	Ц	Ц	1	┸	Ц	Ц	1	Ц	Ц	1	Ц	4	Ц	Ц	1	Ц	4	Ш	4	Ļ	Ц	1	4	Ц	Ц	4	1	Ц	1	\perp	Ц	Ц
15	_	Н	4	1	Ц	4	Ц	4	4	L	Ц	4	1	L	Ц	4	4.	L	Ц	4	╀	Ļ	Ц	Ц	4	1	1	Ļ	L	Ц	4	4	Ļ	L	Ц	1	1	Ц	4	L	Ц	Ц	4	1	Ц	4	1	Ц	Ц	1	Ľ	4	Ц	Ц	1	Ц	4	Ц	4	╀	Ц	4	1	L	Ц	4	1	Ц	4	Ц	Ц	Ц
16		Ц	4	4	Ц	4	Н	Ц	4	Ļ	Ц	4	1	╄	Ц	4	Ļ	L	Ц	4	1	1	Ц	Ц	4	1	ļ	1	L	Ц	1	L	L	L	Ц	1	L	Ц	4	Ļ	Ц	Ц	4	┸	Ц	Ц	4	Ц	Ц	1	Ц	\perp	Ц	Ц	1	Ц	4	Ц	4	ļ	Ц	4	1	L	Ц	4	1	Ц	4	┶	Ц	Ц
17	•	Ц	4	1	Ц	4	Ц	4	4	L	Ц	4	1	L	Ц	4	1	L	Ц	4	1	Ļ	Ц	Ц	4	1	1	1	L	Ц	1	1	L	L	Ц	4	Ļ	Li	4	L	Ц	Ц	4	1	Ц	Ц	\perp	Ц	Ц	L	Ц	4	Ц	Ц	\downarrow	Ц	1	Ц	4	L	Ц	_	1	Ц	Ц	4	1	Ц	4	╀	Ц	Ц
18	-	Ц	4	1	Ц	4	Ц	Ц	\downarrow	L	Ц	4	1	Ļ	Ц	4	1	Ļ	Ц	1	1	╀	Ц	Ц	4	1	l	1	L	Ц	4	4	Ļ	L	Ц	4	L	Ц	4	╀	Ц	Ц	4	\perp	Ц	Ц	╀	Ц	Ц	1	Ц	1	Ц	Ц	4	Ц	4	Ц	4	╙	Ц	1	1	Ц	Ц	4	┸	Ц	1	1	Ц	Ц
19	-	Ц	4	1	Ц	1	Ц	Ц	\perp	L	Ц	1	1	L	Ц	4	┵	L	Ц	1	\perp	\downarrow	Ц	Ц	1	1	L	L	L	Ц	1	1	L	L	Ц	_	L	Ц	1	L	Ц	Ц	1	Ţ	Ц	Ц	!	Ц	Ц	┸	Ц	1	Ц	Ц	1	Ц	1	Ц	1	┸	Ц	1	1	Ц	Ц	1	1	Ц	1	L	Ц	Ц
20		Ц	1	1	Ц	1	Ц	Ц	1	L	Ц	4	1	\perp	Ц	4	1	L	Ц	1	1	L	Ц	Ц	1	1	1	┸	Ц	Ц	1	1	L	L	Ц	1	L	Ц	1	L	Ц	Ц	4	1	Ц	Ц	1	Ц	Ц	1	Ц	\perp	Ц	Ц	1	Ц	1	Ц	1	Ļ	Ц	4	┸	Ш	Ц	1	1	Ц	4	L	Ц	Ц
21	-	Ц	4	1	Ц	1	Ц	Ц	4	1	Ц	4	1	L	Ц	4	4	L	Ц	4	1	1	Ц	Ц	4	1	1	1	L	Ц	4	1	1	L	Ц	1	1	Ц	1	1	Ц	Ц	4	1	Ц	Ц	1	11	L¦.	4	Ц	4	Ц	Ц	1	Ц	1	Ц	4	1	Ц	4	1	\perp	Ц	4	1	Ц	4	1	Ц	Ц
22	-	Н	4	4	Н	1	Ц	Ц	1	L	Ц	Ц	4	1	Ц	4	4	Ļ	Ц	4	4	Ļ	Ц	Ц	4	1	1	1	L	Ц	4	4	Ļ	L	Ц	4	1	Ц	4	1	Ц	Ц	4	1	Ц	Ц	1	Ц	Ц	1	Ц	4	Ц	Ц	1	Ц	4	Ц	4	Ļ	L	4	1	\perp	Ц	1	4	Н	4	1	Ш	Н
23		Ц	4	1	Ц	4	L	Ц	1	Ļ	Ц	Ц	4	╀	Ц	4	4	L	Ц	4	1	╀	Ц	Ц	4	1	1	1	L	Ц	4	1	L	L	Ц	4	╀	Ц	4	╀	Ц	Ц	4	4	Ц	Ц	1	Ц	Ц	4	Ц	Ц.	Ц	Ц	1	Ц	1	Ц	4	Ļ	L	4	4	L	Ц	1	1	Н	4	Ļ	Ц	Н
24	H	1	1	1		1		١,١	ł	ı			1	1	ı	ı	1		1	ı	1			ı	ł	1	1	l	ı		1	l	l	ŀ	H	١			-	ı	ı	H	١	ł	П		1	11	П	1	1	1			1	l	ı	П	-	Í	l	-	1		l	ı		H	ı İ			H

JOB SPECIFICATIONS

IBM	Form X21-9092 Printed in U.S.A.
RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS 1/10/7/ 12/75 75 76 76 76 76 76 76 76 76 76 76 76 76 76	77 78 79 80
Date 1/10/7/ Programmer La Donna Hoffmann Punching Instruction Punch Punch Punch Programmer La Donna Hoffmann	STNO
Control Card Specifications	
Line Size to Compile Size to Compile Size to Compile Size to Compile Size to Compile Size to Size to Compile Size to Size to Compile Size to Size to Compile Size to Size to Compile Size to Compile Size to S	- T- T- T- T- T- T- T- T- T- T- T- T- T-
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	
File Description Specifications	Control of Control
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	on/Unordered
File Designation of Record Address Field for DAM for Cyl	inder Overflow per of Extents
Line Filename Sequence Type of File Organization Device Symbolic Device Device Symbolic Sequence Constitution Device Devi	Tape Rewind
File Format or Additional Area 8 9 0 0 0 0 0 0 0 0 0	File Condition
E	N/O/N
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 120 21 22 23 24 25 26 27 28 129 30 31 33 23 32 43 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
02 FTRANS IP 96 READER PRINTER	
0 4 F	
TBW	Form X21-9094 Printed in U.S.A.
Page 1/10/71 Punching Graphic Page 1/10/71 Page 1/10/71 Page 1/10/71 Page 1/10/71 Page 1/10/71 Page 1/10/71 Page 1/10/71	77 78 79 80
Program IFANSACTION LIST Instruction Punch	STNO
Programmer LaDonna Hoffmann	
Becord Identification Codes Field Location Field L	
	Sterling
Form Type Packer	Sian
Form Type A CAZID A	*
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 7	71 72 73 74
0 1 ITRANS 0 2 1 1 6 ØDATE	+++
0 3 I 7 12 17 EMMO	
0 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
Input specifications are the same as for Job 1	++++
because the same input file is used.	
Input specifications are the same as for Job 1 because the same input file is used.	++++
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	++++
	<u> </u>

IBM	International Business Machines Corporation Form X21-9033
	RPG CALCULATION SPECIFICATIONS
Date 1/10/71	1 2 75 76 77 78 79 80 Punching Graphic Program L/STWO
Program TRANSACTION LIST	Punching Instruction Punch Page P3 Instruction Punch
Programmer LADONNA HOFFMAN	/N
riogrammer	Resulting
Indicators	Indicators Arithmetic
	g Plus Minus Zero
Line Line And And Factor 1	Operation Factor 2 Result Field Field Compare Comments
Cont (L.b.)	Table (Factor 2) is
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24	4 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 QTY	MULT PRICE EXTEST 72
0 2 C	1
This specification tell	Is 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, howev	ver, that a new field, EXTCST, is created to hold the result. We needed
a new result field bec	cause we could not use one of the fields described on the Input sheet.
o 8 c If we had used one of	f them, we would lose information needed for printing the detail line.
0 9 C	
1 0 c Any new field must b	be defined by describing field name, field length, and decimal positions.
	EXTCST. Any valid name could be used. According to the Printer
	XTCST field is 7 positions long with two decimal positions. We, there-
	es when defining field length and decimal positions.
114 C	
115 (0)	
15 C	
IBM	International Business Machines Corporation Form X21-0090 Printed in U.S.A.
	The Hartonal Dusiness Machines Corporation
Date 1/10/71	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 79 80 Punching Graphic Page 04 Program Identification LISTMO
Date 1/10/71 Program TRANSACTION LIST	RPG OUTPUT - FORMAT SPECIFICATIONS Punching Graphic Punch Pun
Date 1/10/71	RPG OUTPUT - FORMAT SPECIFICATIONS Punching Graphic Punch Pun
Program TRANSACTION LIST Programmer LA DONNA HOFFMA	RPG OUTPUT - FORMAT SPECIFICATIONS Punching Graphic Punch Punch Punch Punch Program Identification LISTMO
Date 1/10/71 Program TRANSACTION LIST Programmer LA DONNA HOFFMA	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Punching Instruction Punch Punch Punch Page 044 Program Identification L I S T NO Output Indicators Edit Codes ANN Printed in U.S.A. Program Identification L I S T NO
Program TRANSACTION LIST Programmer LA DONNA HOFFMA	Printed in U.S.A. RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Punching Graphic Instruction Punch Punch
Program TRANSACTION LIST Programmer LA DONNA HOFFMA Line Filename (1) Space : Skip C	Printed in U.S.A. RPG OUTPUT - FORMAT SPECIFICATIONS Punching Graphic Instruction Punch Punch Punch Punch Punch Punch Punch Program Identification L I S T NO Commas Zero Balances to Printer to
Date 1/10/71 Program TRANSACTION LIST Programmer LA DONNA HOFFMA Space :Skip C	Printed in U.S.A. RPG OUTPUT - FORMAT SPECIFICATIONS Punching Instruction Punch Punching Instruction Punch Punch Punch Punch Punch Punch Punch Punch Punch P
Program TRANSACTION LIST Programmer LA DONNA HOFFMA Line Filename (1) Space : Skip C	Printed in U.S.A. RPG OUTPUT - FORMAT SPECIFICATIONS Punching Instruction Punch Punching Instruction Punch Punching Instruction Punch Punch Punch Punch Dutput Indicators Field Name Commas Zero Balances to Print X = Remove Plus Sign Position X = Remove Plus Sign X = Remove
Program TRANSACTION LIST	Printed in U.S.A. Punching Instruction Punch Punching Instruction Punch Punching Instruction Punch Program Identification L I S T NO Output Indicators Edit Codes Commas Zero Balances to Printe No Sign CR - X = Remove to Printe No Sign CR - X = Dus Sign Y = Date Field Edit Yes No 2 B K Field Edit No Yes No 2 B K Field Edit No Yes No 2 B K Field Edit No Yes No 2 B K Field Edit No No Yes 3 C L Z = Zero No No No A Duspress Constant or Edit Word 4 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
Program TRANSACTION LIST	Printed in U.S.A. Punching Instruction Punch Punching Instruction Punch Punching Instruction Punch Program Identification L I S T NO Output Indicators Edit Codes Commas Zero Balances No Sign CR - X = Remove Plus Sign Y = Date Plus Sign Y = Date Position Sign Y = Date Plus Sign Y = Date Plus Sign Position No Yes 3 C L Z = Zero No Yes No Yes 3 C L Z = Zero No No No No No No No No No No No No No
Program TRANSACTION LIST	Printed in U.S.A. Punching Instruction Punch Punching Instruction Punch Punching Instruction Punch Program Identification L I S T NO Output Indicators Edit Codes Commas Zero Balances No Sign CR - X = Remove Plus Sign Y = Date Plus Sign Y = Date Position Sign Y = Date Plus Sign Y = Date Plus Sign Position No Yes 3 C L Z = Zero No Yes No Yes 3 C L Z = Zero No No No No No No No No No No No No No
Program TRANSACTION LIST	Printed in U.S.A. Punching Instruction Punch Punching Instruction Punch Punching Instruction Punch Program Identification L I S T NO Output Indicators Edit Codes Commas Zero Balances No Sign CR - X = Remove Plus Sign Y = Date Plus Sign Y = Date Position Sign Y = Date Plus Sign Y = Date Plus Sign Position No Yes 3 C L Z = Zero No Yes No Yes 3 C L Z = Zero No No No No No No No No No No No No No
Program TRANSACTION LIST	Printed in U.S.A. Punching Instruction Punch Punching Instruction Punch Punching Instruction Punch Program Identification L I S T NO Dutput Indicators Edit Codes Commas Zero Balances to Printe No Sign CR - X = Remove to Printe No Sign CR - X = Phys Sign Position Yes Yes 1 A J Y = Date Field Edit Yes No Yes 3 C L Z = Zero No Yes 3 C L Z = Zero No No No No No No No No No No No No No
Date	Printed in U.S.A. Punching Instruction Punch Punch Punch Punch Punch Program Identification Punch Punch Punch Punch Punch Program Identification Punch Punch Punch Punch Punch Program Identification Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch
Program TRANSACTION LIST	Printed in U.S.A. Punching Instruction Punch Punching Instruction Punch Punching Instruction Punch Program Identification L I S T NO Dutput Indicators Edit Codes Commas Zero Balances to Printe No Sign CR - X = Remove to Printe No Sign CR - X = Phys Sign Position Yes Yes 1 A J Y = Date Field Edit Yes No Yes 3 C L Z = Zero No Yes 3 C L Z = Zero No No No No No No No No No No No No No
Program TRANSACTION LIST	Punching Instruction Graphic Punch
Program TRANSACTION LIST	Page OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 79 80 Page O4 Program Instruction Punch
Program TRANSACTION LIST Programmer LA DONNA HOFPMA Space : Skip O O O O O O O O O	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Instruction Punch
Program TRANSACTION LIST Programmer LA DONNA HOFPMA Space Skip O Skip O O O O O O O O O	Page OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 79 80 Page O4 Program Instruction Punch
Program TRANSACTION LIST Programmer LA DONNA HOFPMA Space Skip O Skip O Skip O Skip O Skip O O Skip O O O O O O O O O	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Instruction Punch
Program TRANSACTION LIST Programmer LA DONNA HOFFMA Space :Skip O O O O O O O O O	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Instruction Punch
Program TRANSACTION LIST Programmer A DONNA HOFFMA	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Instruction Punch
Program TRANSACTION LIST Programmer A DONNA HOFFMA	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Instruction Punch
Program TRANSACTION LIST Programmer A DONNA HOFFMA	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Instruction Punch
Program TRANSACTION LIST Programmer A DONNA HOFFMA Space Skip O O O O O O O O O	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Instruction Punch

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 DATE	I TEM NO	DESCRIPTION	QUANTITY	UNIT	EXTENDED COST
07/23/70	413010 412146 411116	CH001 BOX 100A FLUSH CH143 BREAKER 15A 1500 TWIN SOCKET B	10 100 500	4.90 .89 1.12	49.00 89.00 560.00
					698.00
07/24/70	503029 317802 326917 411121 412997 413088 411174 413090 718326	MOTOR 1/2 HP 60 CYC TERMINAL CLIP TERMINAL BAR 1506 SOCKET ADAPT BRN CH173 BREAKER 30A CH176 BREAKER 60A C151 SIL SWITCH BRN CH005 BR BOX 150A FC803 FUSE 15A	2 100 100 400 60 40 200 10	146.78 5.12 4.12 .19 1.15 1.16 4.98 .32	293.56 512.00 412.00 76.00 68.00 46.00 232.00 49.80 64.00

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.

IBM	International Business Machines Corporation Form X21-906
	Printed in U.S. RPG INPUT SPECIFICATIONS
Date	1 2 75 76 77 78 79 8
Program	Instruction Punch Identification
Programmer	-
Indicator	Record Identification Codes 1 2 3 Field Location Field Location Field Location Field Location
Filename Filename Filename Position Option (O) White Head (1:A) Position Option (O) Position Option (O) Position Filename Filename Option (O) Position Option (O) Posi	Not (N Not (N C/Z/D Charge
0 1 1 0 0 2 1 0 0 3 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
IBM	International Business Machines Corporation Form X21-90 Printed in U.S
	RPG CALCULATION SPECIFICATIONS
Date	Punching Graphic Page Program Identification
Program	Instruction Punch
Programmer	_
Indicators Line ad A 1 9 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	Result Field
0 1 0	
	┦╏╏┩╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒
	International Business Machines Corporation Form X21-909 Printed in U.S.
IBW	Printed in U.S. RPG OUTPUT - FORMAT SPECIFICATIONS
	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 Program Punching Graphic Page Identification Identification
IBW	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 79
IBM,	Printed in U.S. Page Program
IBM, Date Program	Printed in U.S. RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 79 Punching Graphic Punch Punch Poster Punch Pun
Date Program Programmer Date	Printed in U.S. RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 79 Punching Graphic Punch Punch Poster Punch Pun

An input field specified on the Input sheet determines when totals should be calculated

tions are total operations.

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.

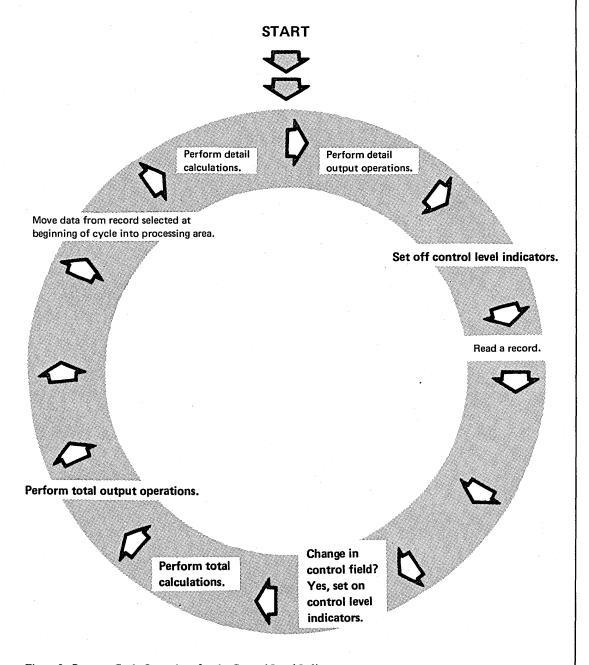


Figure 8. Program Cycle Operations for the Control Level Indicators

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:

1										·	orat A		О	N:	s												•								70	Pri	ntec	X21-	J.S.
on .	Н	_	aph	_	1						_													P	age	İ	2			rogra		tion	- [/5	Ĺ	Ĺ			Ī
tic	on	С	od	es		3					Binary		Fi	eld	L	oca	tic	n									6			io			ielo		ors				
C/2/D	CITIO	Character	P	'osi'	tion	1	Not (N)	C/Z/D	Character	Stacker Select	P = Packed/B = Bin	ı	Fro	m			т	0		Decimal Positions	F	=ie	ld l	Na	me		Control Levei (L1-L9)	Matching Fields or	Criallining Fields	Field Record Relation	PI	lus	Mi	nus	Ze or Bla		s	iterli iign 'ositi	_
3:	3	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57 5	8 5	9 60	61 6	2 6	3 64	65	66	67	68	69	70	71	72	73
ļ.	4		_		L	_	H		L	-	_	_	L		4		L		_	<u>ہ</u>	D	_	_	E	+	١,	1	\dashv	+	+	ł	L	_	L	L	L	Н	4	_

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	1 2 75 76 77 78 79 80
Program	Punching Graphic Instruction Punch	Page Program Identification
Programmer	, ruidi	
Record Record	Identification Codes Field Location	Field
Form Type Sequence Notificator Record Identifying Indicator Outcord Identifying Indicator Outcord	2 3	Indicators
Line Filename (27)	bosition () By Position () By Prosition Field Name Sterling Sign Zero Position	
Form Type Sequence Option (0) Provided to the condition of the condition o	NOT (N) NOT (N	Marching Marching Or Blank
	29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52	
O 1 I I NPUT		
0 3 1	33 370	DATE L1
0 4 1	وبنا نجر کاری بندر کے باترین سازے کا کہ اور کا انتہا کا کا اسازی کے انتہا کا	COST
IBM	International Business Machines Corporation	Form X21-9093 Printed in U.S.A.
Ri	G CALCULATION SPECIFICATIONS	1 2 75 76 77 78 79 80
Date	Punching Graphic Instruction Punch	Page Program Identification
Programmer		
Indicators		Resulting Indicators
	80	Arithmetic Plus Minus Zero
11 619 51	peration Factor 2 Result Field Field Length	Compare High Low Equal 1) 1>2 1< 2 1 = 2
Control I Volume I Vo	Decima	Lookup
	29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52	Table (Factor 2) is High Low Equal 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
O1 C QTY M	ULT COST EXTEST 72	и при при при при при при при при при пр
	DD DACST DACST 92 DD FINCST FINCST 102	
0 4 C		nt approprian
	dicator in columns 7-8 to show that a tot en a control break occurs. The ADD op	
	ation that will be done when L1 is on; th	
the DATE field chang	Jes.	1946/2016/14/7 (#7) 41/3 1400 PROMINE S (#7)
	111111111111111111111111111111111111111	<u> </u>
IBM,	International Business Machines Corporation OUTPUT - FORMAT SPECIFICATIONS	Form X21-9090 Printed in U.S.A.
Date	Punching Graphic Punching Punching Graphic	1 2 75 76 77 78 79 80
Program	Instruction Punch	Page Identification
Programmer		
Space : Skip Output Indic	ators	Edit Codes
	S Writings to	Balances No Sign CR - X = Remove Plus Sign Sterling Sign
Line Filename (H) And Ar	d Positon Yes	Yes
Form Type (H/I) Stacker Sele Before Before Before After Not Not	No No No No No No No No No No No No No N	No 4 D M Suppress
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 The Tin co	lumn 15 indicates which output
0 1 0 0 U T P U T D 1		total records. Every total record
0 3 0	QTY 3 25 should also	have a control level indicator tell the computer when to do
0 5 0		on. The output operation
060712 41	described in	lines 06 and 07 is done only
0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DACST 1 45 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		_				_		_	•	_			_		_	_	-										_	_	_	In	ter	nati	ona	B	sine	ss f	Mac	chir	nes	Cor	por	atio	on		_						-							-	_	_					-	_	_						21-90		٦
																						R	P	G			(o	U	TF	ગ	JΤ	٠.	F	o	R	М	1,4	١T	٠.	SI	PE	EC	1	F۱	C	A.	ΤI	o	N	s																						n U.		
Date				-		_																			1	_	_			_	G	ran	hic		Г	_	Γ	-	Т	7		1		Т		Т	_	1									г	+	² ¬				P	rog	ran	n		75	5 76	5 7 T	′7 T	78 	79	8	ე
Program											-																ınc		ng tior	,		Jnc			┢		┢	_	H	+		+	_	+		\dagger	_	l							۲	age	Ľ	1					łc	ien	tifi	cati	іоп	L	\perp	Ţ		\perp	L	L]
Programn	ner																								١								_	_			ا		٠	_	_	_			_			,																				•							
	T	_ _	_					_		Т	T	<u>-</u> -T	_	7	_	_	_		_	_			_				_	-	_					_	<u>. </u>			_		_	_											_	_			_	_		_								_			_	_			_	ı
		I								l		* *	Spac	ce	;	Sk	ip			Ó	utį	put	In	di	cat	or	5	١							۱,										<u> </u>	۰			-	-	_				Ec	dit	Co	ode				_			_		_	-	1						I
	l									1	1	2	Т	┥	_	_	Г		H			1		-	Т		_	┨							1		_					1>	7			Ī	co	mn	nas	Z		Ba Pr	land	ces	N	lo S	ign	T	CR	T	-	1	x	-			ove		1			Str	erlin	na	١
Line	l	ł		F	ile	na	me	,		Į			Betore	-			l				Δ	nd			l and			1	F	ie	ld	Na	me	9	μ	la			End			Binar				ħ		Ye		t		Ye	s		T	1		Ť	A		J	7	Υ	=	D	ate	Sigr ! ! Ed		ŀ		1	Sig		•	١
	١	2								5		elect	<u>.</u>	Ţ		2		.	H		Ë	1	Τ	Ť	1	Т	Τ	┨							ğ	1		i	Posi in			4/B	3					Ye: No No				Ye No	:s			3		١	B C D		K L M	- 1	z	=	Z	ero					١				
	1	5									E S	Ser	Berc	Afte	Refore	200	1	Arre	Not			Not		ļ	Š	l		١							ig ig	lank.			Out Rec			Packe				L	_	NO		_		_			1_	-4	_		Ť	_		' '	۰			трр	163		j		l				
3 4 5	ا ا	,	7 8	3 9	. 1	0 1	1 1	2 1	3 1	أ	5 1	6	,,	18	19	20	21	22	23	24	25	26	27	28	29	30) 3	,	32 :	33	34	35	36	37	38	39	3 4	10 4	41	42	43	# d 44	4	5 4	16	47 -	18 4	49	50	51	52								t W			2 6	63	64	65	66	3 67	7 68	8 69	9 70	, ,	7:	2 7:	3 74	١
0 1) (T	1	-	1	-	1	П	٦	Г	Γ	Г	Γ	Ī	Γ			Γ		T	1	1			_	Γ		Ī	Ī	T	T					T	1		٦									П				Т	I					Г		Τ	T	T	T	T	Γ	T	Τ	1
0 2	c	,	T	T	T		T		T	T	T		T	1		7	1	5		L	1			Γ	Ī	Γ	T	į	D	A	T	ε			Ī	Γ	T				6		T	T				٦	Ì								Γ	Γ		T	T				Γ	Ī	T			T	T	Т	T	Γ	1
0 3	c	+			I	I	T	1			I	I	I	⅃		╝	_					L					I		2			_			3					2			Ţ	I						\Box									L	Ι	Ţ				I	I	I	I	Ι	I	I	I	Ι	I]
0 4	c	4.	1	1	1	1	1	1	1	1	1	1	1	_	\perp	لـــا	L	L		L	L	L	L	L	L	L	1		C						3		1	4		3			L	1	1	4	4	4			_				i		L	L	L	1	4	_	_	L	L	L	\downarrow	1	ļ	1	ļ	ļ	1	ļ	1
0 5	c	1	1	1	1	1	1	1	1	1	1	1	1	_		┙	_	Ĺ	L	_	L	L	L	Ļ	L	L	1	1	ε	X	T	C	S	7	1	Ļ	1		4	1	5	1	1	1	1	1	1	1			4	Ц			أ		L	L	Ļ	1	1	1		L	L	L	1	1	ļ	\downarrow	1	\downarrow	1	ļ	1
0 6	c	+	1	1	1	4	4	4	4	1	η.	4	1	2	4	Ц	2)	L	L	1	L	L	L	L	L	1	1	4		_	_	L	L	Ļ	Ļ	1	4	_		Ļ	L	1	4	4	4	4	4	4	4	4		Ц	Ц	_	L	L	L	Ļ	ļ	1	4		L	L	Ļ	1	1	\downarrow	1	ļ	Ļ	1	ļ	1
0 7	ļ	4	1	4	1	1	1	-	4	4	4	4	4	4	4	\vdash	ļ.,	L	L	L	L	L	ļ	Ļ	Ļ	Ļ	1	ļ	D.	A	C	S	7	1	1	╀	1	4	-	4	5	L	ļ	4	4	4	4	4	4	4	4	4		_	_		L	L	Ļ	1	+	4	_	_	L	Ļ	ļ	1	4	4	1	\downarrow	+	╀	1
0 8	C	4	+	+	+	+	+	+	+	+	+	٦			_	-				:	_	_		_	_		_	_	_						_		_		_				_				_			_	-	_		_					_	٦	١.	+	_	L	╀	1	╀	+	+	\downarrow	╀	+	+	╀	$\frac{1}{2}$
0 9	10	-1-	+	+	+	+	+	+	+	+	╀	۱.			4 (ı	ŀ	+	_	H	╀	╀	╁	+	+	+	╀	+	+	╀	1
1 1	6	+	+	+	+	+	+	+	+	+	+	- [•	(1	_						es																																							1	+	-	-	-	╁	+	+	+	+	t	+	+	╁	ł
1 2	6	+	+	+	$^{+}$	$^{+}$	$^{+}$	+	+	+	+	۱-			ev																																								_	_	n			ı	ŀ	+	-	H	H	H	t	t	+	+	+	+	+	+	1
1 3	c	+	+	+	†	†	1	†	+	t	t	1		t	he	Э :	sa	m	ıe	S	рє	eci	fi	c	at	ic	n	1	ir	ıe	8	IS	tl	16) I	ir	ıe	1	ty	p	е	e	n	tr	·y	ii	ו ו	CC	οlι	ır	nı	1	15	5	(2	!)				1	ľ	†			r	†	t	†	†	t	t	t	†	t	1
1 4	t	,†	t	†	†	t	†	†	†	t	†	1		i	nc	it	Ca	it	es	t	ha	t	tł	ne	e	n	ti	re	9	lir	16	9	h	O	ul	d	b	Э	١,	VI	ri:	tt	eı	n	W	ιh	er	1	th	е	C	or	١t	ro	ı					ł	ľ	†	_	-	t	t	t	t	†	†	t	†	t	t	1
1 5	þ	7	Ť	†	1	Ť	1	1	1	1	Ť	1		10	ev	/e	li	'n	di	Ca	at	or	į	S	10	١.																																		ı	<i>(</i> -	7		Г	Г	T	T	T	1	T	T	1	T	T	1
\Box	þ	,	T	Ť	Ť	1	T	T	1	T	T	Ļ	_					_	_							_					-																	÷					_	_						-		Ţ				T	T	T	T	T	Τ	T	T	T	1
	c	1		1	I	I	I	I		I	I	I	1		Ĺ	Ĺ	L	Ĺ	Ĺ	Ĺ	Ĺ	Ĺ	Ĺ	Ĺ	L	Ĺ	Ī	Ī		_			L	L	Ĺ	L	1	_			<u> </u>	Ĺ	1	1	_	Ī	_	_	_		_	Ц	Ц	Ц	L	Ц	L	L	L	Ţ	1	I				I	I	I	I	I	I	I	I	I]
	c	1		1	1		1	1	1	1	1	1	1	╛	Ц	Ľ	L	L	L	L		L	L	L		L	1	1	_	_					L		1	1			L	L	1	1	_		1					Ц	Ц		L				\perp	1	1	1			L	1	1	1	\perp	1	1	1	1	1	1
H	c	1	1	1	4	1	1	1	1	1	1	1	4	_	┦	L	L	L	L	L	L	L	L		L	1	ļ	1	4	4				L	L	L	1	1			L	Ļ	1	1	_	1		\downarrow	_	_		Ц		Ц	L	L	L	1	L	1	1	1	_	_	L	Ļ	1	1	\downarrow	1	ļ	\downarrow	\downarrow	1	1
$\coprod \bot$	c	<u>1</u>	\perp	1				\perp		1			⅃		لــا	L	L	L	L	L	L	L	L	L	L								L	L	L		\perp		Ш			L	1	1				┙					Ц		L	L	L	L	L	\perp	1				L	L	\perp		\perp	Ţ	\perp	T	\perp	\perp	J

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:

			_				_	_	_			_		_	Ţ	ie	1				_	_			_		_						_	_		÷					_				_				T	ier :	2	-
F	Punch	T	RA DA	NS	E	-T	101	^	<i>Ι</i> /υ			N BE								D	E		/7 C	R	€ <u>/</u> / /	И 0 ⁻	Τ.	ıc	^	/						۱	२०	ΑΙ	V7	17	Y	C	'N 'C	'' 'S	T	-						
П	Program	T	2	3	4	5	6	7	8	Tg	ī	0 1	ı	2	13	14	15	16	ili	7 1	8	19	20	21	22	23	2	1 2	26	27	28	2	9 3	0 3	1 3	2 3	3 3	4	35	36	37	38	39	40	ŢΛ	42	43	44	45	46 4	7 4	3
	Control Card																																											/	/- L						1	_

two decimal positions

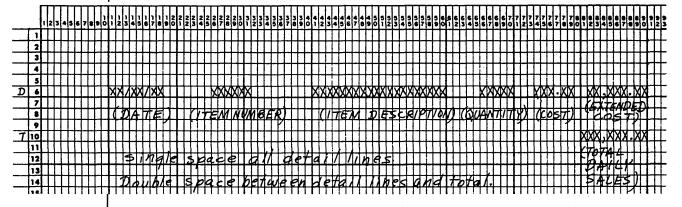
Processing:

- 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	CHOO1 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	CHOO5 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:



JOB SPECIFICATIONS

IBM International Business Mechines Corporation RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS	
. , RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS	Form X21-909 Printed in U.S.
1/10/7/	75 76 77 78 79 8
Program LK Hoffmann Page Punching Instruction Punch Program LK Hoffmann	Program TRANS
Programmer LK Hoffmann	
Control Card Specifications	
8 Sterling 8 Model 20 Model 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
Core Core Core Core Core Core Core Core	
Line Size to Size to Compile Of Execute See See See See See See See See See S	rence Library manual for actual entries.
Mechania (North 1990) Perum	
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 No new entries are made on t	this sheet.
File Description Specifications	
	File Addition/Unordered
1 11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	nt Exit Number of Tracks
End of File Record Address Type Name of	Number of Extents
Line Filename Sequence Type of File Organization Device Symbolic Symbolic Device Symbolic Devi	
Q Verflow Indicator S 5	File Condition U1-U8
Block Record Starting Sta	
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
OZ FTRAMS IP 96 READER OZ FREPORT O 94 PRINTER	
0 4 F	
IBM International Business Machines Corporation	Form X21-9094 Printed in U.S.A.
RPG INPUT SPECIFICATIONS	
1/2 - In a late of the late of	75 76 77 78 79 80
Date 1/10/71 Transcation Registers Punching Graphic Page 52 Punching Graphic Page 52	75 76 77 78 79 80 Program TRAMS
Date 1/10/71 Program Transaction Register Punching Graphic Instruction Punch Punch Punch Punch	rogram TDAAC
Date 1/10/71 Program Transaction Register Programmer L K Hoffmann Punching Graphic Instruction Punch Punch Punch	rogram TDAAC
Date 1/10/71 Program Transaction Register Programmer L K Hoffmann Punching Graphic Instruction Punch Punch Punch	regram TRAMS
Date 1/10/71 Program Transaction Register Programmer L K Hoffmann Record Identification Codes Field Location Field Location Graphic Instruction Punch Field Location Graphic Instruction Codes Field Location Graphic Instruction Punch Field Location Punch Fi	regram TRAMS
Date 1/10/71 Program Transaction Register Programmer L K Hoffmann Record Identification Codes Field Location Field Location Graphic Instruction Punch Field Location Graphic Instruction Codes Field Location Graphic Instruction Punch Field Location Punch Fi	regram TRAMS
Date 1/10/71 Program Transaction Register Programmer L K Hoffmann Record Identification Codes Field Location	regram TRAMS Field Indicators Sterling Sign Sportion
Date 1/10/71 Program Transaction Register Programmer LK Hoff Mann Record Identification Codes Field Location Record Identification Codes Field Location Field Name Field	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date 1/10/71 Program Transaction Register Programmer LK Hoff Mann Record Identification Codes Field Location Record Identification Codes Field Name Filename Fil	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Programmer L K Hoff mann Record Identification Codes Filename Record Identification Codes Filed Location Field Name Fiel	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Date I/Io/11 Program Transaction Register Programmer L K Hoff Mann Record Identification Codes Field Location Field Location Field Name Field Nam	TRAMS TRAMS Field Indicators Plus Minus or Position Blank
Programmer L K Hoff Mann Record Identification Codes Filename Punching Graphic Punch	TRAMS TRAMS Field Indicators Plus Minus or Position Blank

IBM	International Business Machines Corporation Form X21-0093 Printed in U.S.A.
	RPG CALCULATION SPECIFICATIONS
Date 1/10/71	Program T D A A S
Program Transactio	in Register Instruction Punch
Programmer LK Hoff	<u>man</u>
Indicators	Resulting
Indicators	Indicators Arithmetic
G And And	Field Ξ Ξ Compare
	Factor 1 Operation Factor 2 Result Field Length Field Length Field Length Field Length Field
Form Ty Control I (LO-L9, U Not	
	Table (Factor 2) is High Low Equal
3 4 5 6 7 8 9 10 11 12 13 14 1	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 50 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 2 C	ATY MULT COST EXTEST 72H EXTEST ADD DAYTOT DAYTOT 82
0 3 C	
	item, QTY must be multiplied by COST to get EXTCST (extended cost). To get
0 5 c the tota	of all sales made during the day, EXTCST is added to DAYTOT (the field used to
o 6 c accumul	ate daily sales total).
0 7 C	
08 C	
0 9 C	┆╎╏╎╎╎╎╎╎╎╎╎╎╎
1 0 C	╿┦┛┩┩┩┩┩┩┩┩┩┩┩┩┩┩┩┩┩
1 2 6	╎╏╏╏┩┩┡┊╏╏┩╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇╇
	╎╏╏╏┪╏╎╎┞╒╒╘╎╎┞╏╏╏╏╒╒╘ ┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼
1 4 6	╎╣╣╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎╎
IBM	International Business Machines Corporation Form X21-9090
	RPG OUTPUT - FORMAT SPECIFICATIONS
Date 1/10/7/	1 2 75 76 77 78 79 80
Program Transactio	Program Punching Instruction Punch Punch Program TRANS
Program Transactio	n Kegister Instruction Punch
Program Transaction Programmer L K Hoff	n Kegister Instruction Punch
Program Transactio	n Kegister Instruction Punch
Program Transactio	Mann Comman Zero Balances No Sign CR X = Remove Comman Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances No Sign CR Zero Balances
Program Transactio	Mann Space Skip Output Indicators Skip Output Indicators Field Name
Program Transactio Programmer L K Hoff	M Q N N Comman C
Program Transactio Programmer L K Hoff	M REGISTER Mann Space Skip Output Indicators Skip Output Indicators Sterling Space Skip Output Indicators Sterling Sign Output Indicators Sterling Sign Output Indicators Sterling Sign Output Indicators Sterling Sign Output Indicators Sterling Sign Output Indicators Output I
Program	M Q N N Comman C
Program	Constant or Edit Word Suppress Sterling Sign Position Sign CR - X = Remove Plus Sign Sign Position Sign CR - X = Remove Plus Sign Sign Position Sign CR - X = Remove Plus Sign Sign Position Sign CR - X = Remove Plus Sign Sign Position Sign CR - X = Remove Plus Sign Sign Position Sign Sign Position Sign Sign Position Sign Sign Position Sign
Program	Comman Field Name Field N
Program Transactio Programmer L K Hoff Line Filename	Commas Series Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Output Indicat
Program	Comman Sepace Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators Skip Output Indicators O
Program	Comman Constant or Edit Word Constant or Edit Wo
Program Transactio Programmer L K Hoff Line Filename 5 5 6 7 8 9 10 11 0 11 0 REPORT 0 0 3 0 4 0 0 5 0 6 0	Constant or Edit Word Punch Punc
Program Transactio Programmer L K Hoff Line Filename S	Comman Constant or Edit Word Constant or Edit Wo
Program Transactio Programmer L K Hoff Line Filename 5 5 6 7 8 9 10 11 12 13 14 1 0 1 0 R FORT 1 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Constant or Edit Word Punch Punc
Program Transactio Programmer L K Hoff Line Filename S	Comman C
Program Transactio Programmer L K Hoff Line Filename SE FILENAME SE	Name Field Nam
Program Transactio Programmer L K Hoff Line Filename SE SE SE SE SE SE SE SE SE SE SE SE SE S	Sterling Sign Position Po
Program Transactio Programmer L K Hoff Line Filename SE SE SE SE SE SE SE SE SE SE SE SE SE S	Requisitor Mann Space Skip Output Indicators Field Name Fi
Program Transactio Programmer L K Hoff Line Filename State	Instruction Punch Comman
Program Transactio Programmer L K Hoff Line Filename SE SE SE SE SE SE SE SE SE SE SE SE SE S	The struction Punch Sale Sale Sale Sale Output Indicators Field Name Sale Sale Sale Sale Sale Sale Output Sale
Line Filename Fi	Instruction Punch Comman
Program Transactio Programmer L K Hoff Line Filename S	The Register Na N No. 1 No. 1 No. 1 No. 1 No. 2
Line Filename Fi	The process of the pr
Line Filename Section Filename Fil	The Register Na N No. 1 No. 1 No. 1 No. 1 No. 2

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	•	_		-					-	_	_	_			_								-		_			_		li	nter	nat	iona	d Bo	usine	ess 1	Mac	hir	nes (Corp	ora	ation	n			_							_	-	_	_	_			-	-	_	_!		_	_	_	_					-909		
	•																					R	P	G				o	U	TI	Pι	JΊ	Γ	F	0	R	M	IΑ	ıΤ	•	SP	Έ	C	:1:	FI	С	A.	TI	IO	N	s																						U.S.		
Date						_																				Г			_	_	_		hic		Т	-	Т	_		Т		_		-		T		1									\Box	1 : T	2 				Pro	gra	ım		Γ,	75 	76	77	, , T	'8 T	79 	-80	1
Program																												chii	ng tion	,		unc		_	\vdash	\dashv	H	-	H	+		+		+		╀		$\{$							F	age	Ľ	L	_							tior	٦Ĺ	_		Ĺ	L	1		_	ĺ
											_								_							Щ								_		_	_		L	_		Д.				1_	_	J																											
Programs	ner	_		_					_	_	_			_		_	_	_	_																_	_								_			_						_		_							_								_					
ŀ		1								l	١	(E)	Space			:Sk	in			o	uti	put	: In	ndi	cat	tor	s	1								_									_	-		_							_		_		_		_		_		_			_							ı
	1									ĺ	1	ê.		1			_		L	_	_	_			_			┙							1	6							-	1		`	<u></u>	_	_	72	Zero	Ва	alan	ces	$\overline{}$		Co	_		Т		Г	x :	- 6		novi		\dashv		1				-	ı
II				_						L	وا	<u>ا</u> ق	l	١								1			1				_							Γ	T	_	nd	_		nary]			ļ	_		nas	1	t	o P	rint		L	_	ign	+	CR	Ŧ	-			F		s Sig		ĺ				Ster Sign	rling 1	j	ĺ
Line	۰			r	liei	nan	ne			17.0		1 te	l	ı					L,	_	Α	nd	_	4	۱nd	_	_	┙	r	-ıe	ıa	INA	m	е	[s	Blank After (B)	٥	P	osi			= Packed/B = Binary				1		Ye	s			Ye No	9			2			A B C		J K	ı		F		ld E	dit	:			F	Posi	itior	1	
	Ž									/H/		ee.	erore	Ē	Refore	erore	A 6400	121	Not			Not	l		Ş	;		1							Edit Codes	ال م	į	C	n Dut			ked/E				L		No				Ye No			L	4			Ď		L M	Ľ	_			pre	ss			-	í				١
	Form T									F	2 3	Tacke	۱۳	1		- 1					l	1		l	1		l	ł											Reco															nst																1	:				I
3 4 5								13	14		٧,	6	17 1	٠٠١.			21	22	23	24	25	26	27	28	3 29	30	3	1	32	33	34	35	36	37	38	39	3 4	0 4	11 4	42	43	44	45	5 4	6 4	17 4	18	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	3 6	4 6	5 6	6 6	37 F	68 (69	70	71	72	73	74	I
0 1	0	т	7	Ш	٨	7	1	1	L	h	4	4	4		Ø	6	_	Ц		1	P	L	L	Ļ	L	1	1	4	4	_	_	L	L	L	L	Ļ	1	4	4	╛	_		Ļ	1	1	1	4	4	_	Ļ	L		L	L	Ļ	L	_			L	_	L	1	1	1		4	4	\dashv	4	۵	L	L	닏	I
0 2	0	╀	ļ	1	Ļ	ļ.,	ļ	L	╀	ļ	4	4	4	4	_	_	_	L			ļ.	L	-	Ļ	Ļ	1	1	4	4	4		L	L	Ļ	L	Ļ	\downarrow	4	_	5	Ø		ľ	1	7	ĸ,	4	M	S	A	C	7	1	0	N	L	R	E	G	1	S	7	16	=/	?	4	4	4	_	4	۲	F	L	Ľ	I
0 3	0	4	+	╀	Ł	H	Ł	╀	╀	╀	+	+	4	4	4	\dashv		Н	Н		L	L	H	H	╀	+	+	+	+	-	_	L	╀	H	Ļ	ł	+	+	+	4	4	Н	L	+	+	4	1	-	Ц	L	ŀ	L	L	-	L	H		Ц		_	L	H	+	+	+	+	4	4	4	4	\dashv	F	H	H	į
$H \rightarrow H$	+-	-	+	╀	H	┞	H	╀	1	╀	+	+	+	+	-		_	Н	Н	4	H	H	H	╀	╀	+	+	+	4	-	-	L	┞	╀	-	Ł	+	+	+	4	-	_	ŀ	ł	+	+	4	4		_	_		H	-	L	-	H		H		-	H	+	+	+	+	4	4	\dashv	4	\dashv	H	ا	H	ł
0 5	0	╀	╀	╀	ŀ	ļ	L	ŀ	╄	╀	+	+	4	4	4	Н		Н	Ц		_	L	L	┝	╀	+	4	4	+	4	_	L	-	H	H	╀	+	+	+	\dashv	4	_	L	+	+	4	4	4	_	-	-	L		L	L	_				_	L	L	+	+	+	4	4	\dashv	4	\dashv	\dashv	H	H	H	I
0 6	0	+-	+	+	H	-	┞	╀	╀	╀	+	+	+	4	4	Н	_	Н	L	_	L	┞	H	╀	╀	+	+	+	+	4	_	L	-	H	┞	╀	\downarrow	+	+	4	-		ŀ	+	+	4	4	-	_	H	L	H	L	_	L	L	L	Н	'	-	_	H	+	+	+	+	4	+	-	-	\dashv	H	L	۲	1
0 7	0	1	-	╀	╀	Ļ	Ł	-	-	ļ	+	+	+	4	4	-	_	Ц	L	_	-	-	-	ŀ	ŀ	+	+	4	4	4		L	-	ŀ	-	╀	+	+	4	-	4	Н	L	ļ	+	\downarrow	-	-		H	L	L	_	_	-	L	H	H	L		L	ŀ	Ŧ	+	+	+	4	1	\dashv	4	\dashv	H		۲	1
0 8	0	+-	+	╀	-	\vdash	┝	╀	╀	╀	+	+	+	+	-	-	_	\vdash	Н	_	L	H	H	╀	╀	╀	+	+	+	-	_	_	┝	H	⊦	ł	+	+	+	+	4	Н	L	+	+	+	+	\dashv	_		-	H	H	_	H	-	H		_		_	L	ł	+	+	+	+	+	\dashv	4	\dashv	H	H	Н	İ
0 9	0	╀	╀	╀	┞	┞	ŀ	╁	╀	╀	+	+	+	+	4	\dashv	_	Н	Н	_	L	┞	H	╀	+	+	+	+	+	+	_	H	H	⊦	├	╀	+	+	+	+	4	-	ŀ	+	+	+	4	\dashv			L	L			H		H	_	_		-	H	+	+	+	+	+	+	\dashv	\dashv	\dashv	H	H	۲	İ
1 0	0	╀	╀	╀	╀	┞	H	╀	╁	╀	╀	+	+	+	\dashv	\dashv	-	Н	Н	4	L	┞	⊦	╁	╀	╀	+	+	+	4	_	L	┝	H	⊦	ł	+	+	+	+	-		H	+	+	+	+	-		_	_	H	_	-	-		H	H	_		H	┝	╀	+	+	+	+	\dashv	\dashv	-	\dashv	H	H	۲	1
	0	╀	+	+	H	-	H	+	╀	╀	+	+	+	+	4	Н	_	Н	H	4	-	┞	\vdash	╀	╀	+	+	+	+	4	_	_	╀	\vdash	├	+	+	+	+	+	-	_	┞	+	+	+	+	-		H		\vdash	\vdash	\vdash	H	-	H	H	Н	_	-	\vdash	+	+	+	+	+	\dashv	\dashv	\dashv	\dashv	H	H	۲	١
1 2	0	╀	+	╁	┝	┝	┝	┝	╁	ł	+	╀	+	+	\dashv	\dashv		Н	Н	_	H	┞	┝	╁	╀	╁	+	+	+	4		_	├	+	├	H	+	+	+	+	1	Н	├	+	+	+	+	-{	Н		-	\vdash	\vdash	-	\vdash	\vdash	Н	Н		-	\vdash	┝	+	+	+	+	+	+	+	+	\dashv	۲	H	Н	ĺ
1 4	0	₽	+	╁	H	-	H	+	+	+	+	+	+	+	+	\dashv		Н	\vdash	-	H	Ͱ	\vdash	+	+	+	+	+	+	+	-	ŀ	+	╁	\vdash	H	+	+	+	+	+	Н	H	+	+	+	+	\dashv	H	H	-		H	H	\vdash	\vdash	H	_	Н	_	H	\vdash	+	+	+	+	+	\dashv	+	+	\dashv	۲	۲	۲	١
1 5	0	+	+	╁	H	╁	+	t	t	t	+	+	+	+	+	\dashv	-	Н	Н	-	H	H	H	╁	+	+	$^{+}$	+	+	+	-	\vdash	+	t	\vdash	+	+	+	+	+	+	H	┝	+	+	+	+	-	Н	H	-	┢	H	H	\vdash	\vdash		H	H	-	\vdash	\vdash	+	$^{+}$	+	+	+	+	+	+	\dashv	H	H	Н	١
	0	+	+	╁	H	H	۲	+	+	t	+	+	+	+	+	Н	_	Н	Н	_	-	۲	H	t	t	+	+	+	+	+	-	۲	۲	H	┝	H	t	+	+	+	۲		۲	+	+	+	+	+		-	۲	H	┝	۲	-	۲	H	_		-	-	-	+	+	+	+	+	+	+	+	\dashv	۲	۲	۲	١
	0	+	+	+	H	H	H	+	+	+	$^{+}$	t	†	+	+	\forall		Н	Н	-	-	┝	┝	+	╁	+	+	+	+	\dashv	_	H	╁	H	┝	H	†	+	+	+	1	H	l	t	+	\dagger	+	4	Н	┝	\vdash	\vdash	\vdash	\vdash	H	\vdash	H	۲	Н	-	H	H	+	\dagger	$^{+}$	+	+	ᅥ	+	+	\dashv	H		H	l
	0	╀	\dagger	╁	t	╁	t	+	t	t	\dagger	+	\dagger	+	+	Н		Н	Н	_	\vdash	┢	t	t	\dagger	t	+	†	+	1	_	H	1	t	t	t	\dagger	†	+	7	1	Н	t	\dagger	\dagger	+	+	7	-	_	H	H	H	H	1	H	Н	Н	H	-	-	r	t	†	†	+	†	+	\dashv	+	+	F	Ħ	Н	١
H	0	╀	\dagger	t	t	╁	t	t	t	t	\dagger	\dagger	+	+	+		_	Н	H	-	H	┢	t	t	t	\dagger	\dagger	+	+	1		-	+	t	t	t	\dagger	+	+	+	+	Н	t	\dagger	\dagger	+	+	+		-	H		H	H	-	H	H	H		_	-	H	\dagger	†	\dagger	+	+	+	+	+	-	r	H	Н	۱
H	0	╀	t	+	t	\vdash	+	t	t	t	\dagger	\dagger	\dagger	+	\dashv	H		H	\vdash		-	1	t	t	t	t	†	+	+	+	_	\vdash	+	t	t	t	\dagger	+	+	+	1	-	H	+	\dagger	\dagger	+	┪	_	H	H	\vdash	H	H	H	-	-	Н	H	H	H	H	†	\dagger	†	+	+	+	+	+	H	H	H	H	١
لللا	ľ	Ŀ	上	上	L	1_	L.	L	L	L,	┸	1	┸	_	_	Ш	_	L_	Ш		_	L	L.	1	_	_	_		_1			<u>_</u>	_	1_	ㅗ	ㅗ	ᆚ		_1	Ц	Ц	ш	L	_	1	_	_	_		_	Ц.	_			Ц.	_	L.	L		_	Щ	Ц.	Т.	_	_L	_	_	_		_	_	_	_	لب	Í

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.

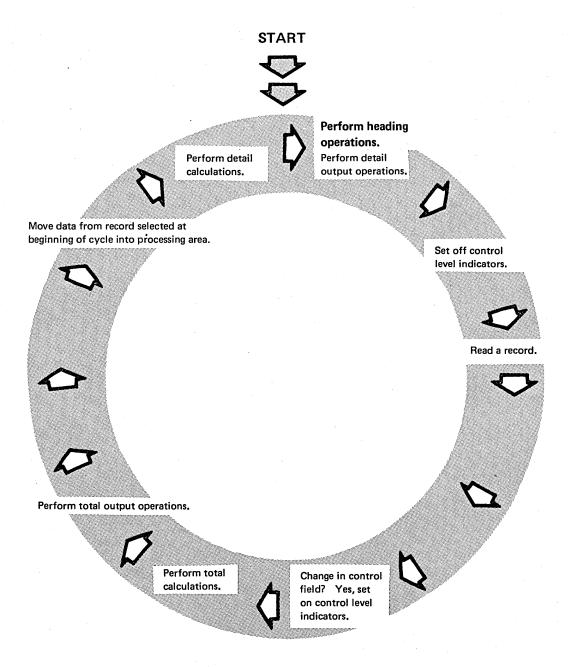


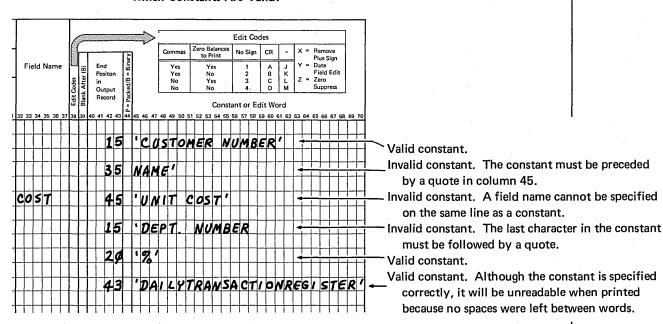
Figure 9. The first operation in the first program cycle concerns output operations conditioned by the first page (1P) indicator.

RPG II Specifications

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:

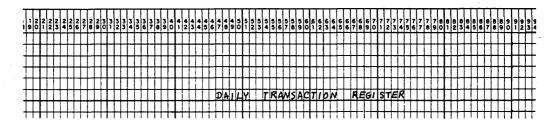
- Constants must be entered in columns 45-70 of the Output-Format sheet.
- Constants can contain either alphameric or numeric characters.
- Constants must be enclosed in single quotes. (The beginning quote is always entered in column 45.)
- No field name can be used on the same line as a constant.
-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

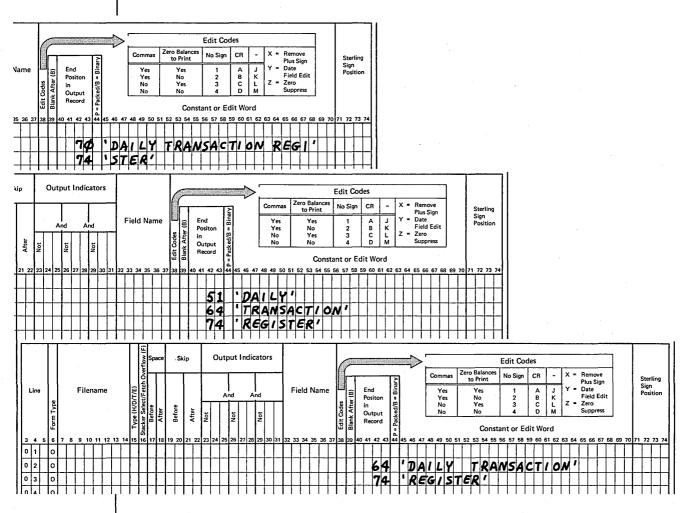
Which Constants Are Valid?



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:





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.

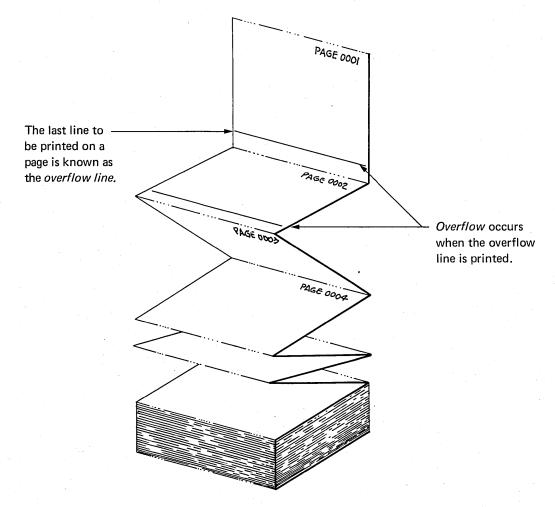
OVERFLOW INDICATORS

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.



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.

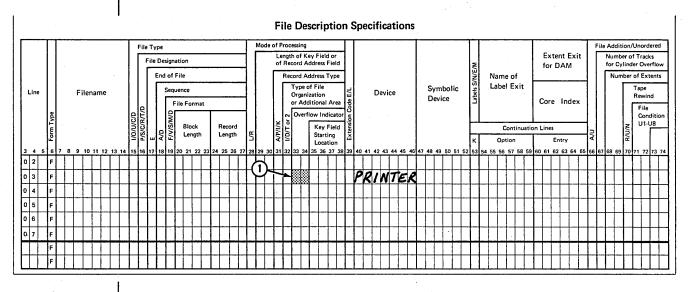
Overflow can be handled automatically by the system or through specifications you write.

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:



									_																																							
IBM																	٠.								es Co																						X21-909 1 in U.S.	
														RF	'G		C)U	TP	UT	ا - ا	FO	R۱	ИΑ	Т	SP	E	CIF	FIC	A	ГΙС	N:	S				1	2						75	76 7	!7 7 [.]	8 79	80
Date				_													Punch	ning	(Grapi	hic			-			Γ									age	П				rogr	ram tificat		П	T	Τ	П	
Program																	Instru	ction	<u>.</u>	Punci	h							\perp									ш	_		,	dent	nicat	ion [
Programi	ner																																															į
[П					Т	Ē		Τ			Γ.	<u></u>	out 1	nd:							Ť																					_	_		Γ		٦
1	П						rf ov	Spac	•	Ski	Р.	ľ	Juis	Jut 1	nan	alu						1	7					٣,	>	_		_				dit (Cod	es		_				_		1		
							P Ove	Т	T	Т								1					Г			Т	آڃَ			Co	mmas	Z	ero B to P	alanc rint	es N	lo Si	gn	CR		_		Rem Plus					Sterling	
Line			File	name		<u>1,6</u>	/Fetch			.			A	nd	Α	nd		F	ield	Na	me	1	e		nd osito	,	= Binar				Yes Yes		Y			1 2		A B	J	.1		Date Field	Edi	it			Sign Position	,
1	Type					Q/H/D	Selec	ore		Delore	ē		Τ	Ţ	Τ		Τ	1				Sego	After	ir		.	Packed/B				No No		Y	es		3		C	L	Z		Zero		.				
	Form					700	acker	Before	3		After	Not		Not		Not		ł				Edit	Blank	F	lecore		Pack							Cor	ıstan	t or	Edi	it W	ord					_				
3 4 5	6	7 8	9 10	11 1	2 13 1	- 1	(O)			20 2	21 22	23 2	4 25	26 2	7 28	29	30 31	32 :	33 34	35	36 3	7 38	39	40 4	11 42	43	44	15 46	47	48 4	9 50	51								2 63	64	65 6°	67	68 f	39 70	7,	72 /3	74
0 1	0	Τ	П	П	П	Т	П	T		J	\bot	- 8				П		П	1.	Π	Т	T	П	П	Τ		1	T		П	T	П	Т		T	П	T	Τ	П	Τ	\Box	П	Τ	П	Т	П	П	٦
0 2	0		П	TT	П	ľ	П		(2	7	T	T	T	П	T	П	T	П	T	П		T	П	П	T	П		T		П		П				П		-	П	Γ	П	П		П	1	П	П	7
0 3	0			П	П	T	П			П				П	Τ	П	T		T			T	П	П						П	T	П	T	П			Ţ	T	П	T		П	T	П		П		7
0 4	0		П	П	П	T	П			Т	T	П		П	T	П		П		П		T	П	П	T		T			П	T	П		П	Т					Τ	П	П		П	T	П	П	
0 5	0					T	П				Τ	П			T	П		П				T	П	П	1									П		П		1				П		П	T	П		П
0 6	0	\top		\sqcap	\Box		П		П	\sqcap		\sqcap			1	\Box	Т	П	\top		7	T	П		\top	\Box	7	1		\Box	1,	П	1	П	T		7		П	Τ			Τ	\sqcap	\top	П	\Box	1

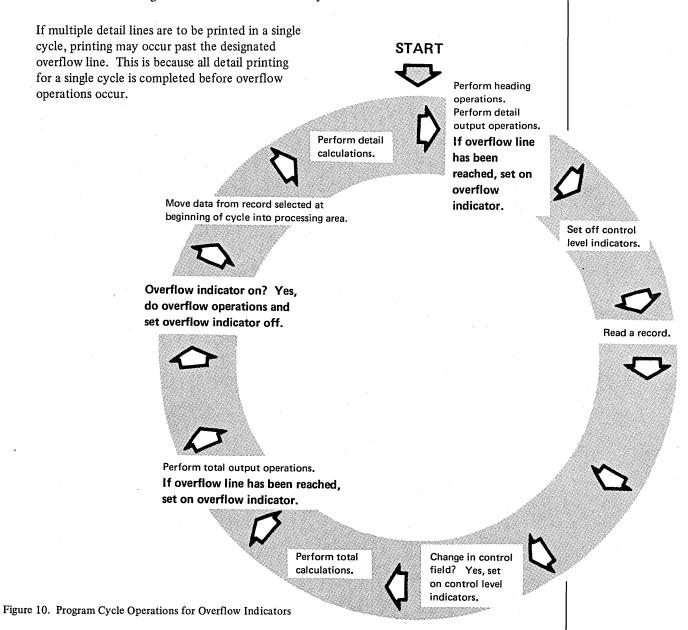
Assign an overflow indicator to the printer at \bigcirc , then use it on the Output-Format sheet at \bigcirc to show which operations must be done when overflow occurs.

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.



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:

IB	V.																						Ir	tern	ation	nal B	usine	ss M	lach	ines	Corp	orati	on															4							X21-9	
	•																R	PG	i		C	υ	TF	PU	T	- F	0	RI	VI	ΑT	. 8	P	C	IF	IC	A.	Tje	O۱	ıs					1	2							75 :	76 7	7 7	8 7	9 80
Date _					_																unch			Gr	aphi	ic	Ι			1				Γ	T								Page	Γ				Pro		m icati	ا م	T	T	T		\prod
Prograi	m												_		_					Ľ	nstru	ctio	<u> </u>	Pu	nch									L		_								_	ــــــا						∟		_			11
Progra	nme	r_													_																																									
		T						T	13 WC	Sp	ace	::	Skip			Out	put	Ind	lica	tor	s	Γ					T							7	 >					_	_		dit	Coc	ies		_			-	_	_ 7				
								١	h Overf	\vdash	П		Т				Т		T			1						<u></u>	Т				7	-		c	mma	as		o Ba to Pi	lanc rint	es	Vo S	ign	CR		Ξ	x		Remo		٦			Sterli	ng
Line		ě		Fil	ena	me		į	I/O/1/E)	9		بو			-		nd	П	An	T	_		Fie	ld I	Van	ne	Codes	fter (B)		End Posi in	iton	regist a Shorter					Yes Yes No			Ye No Ye	o es		1 2 3		A B C	ı	J K L	z	- Z	Zero.	Edit	- 1			Sign Positi	on
		Form Type							tacker Select/F	Befor	After	Before		After	Not		Not			5							Edit S	I٤		Out	put	- Packer	- Lacker				No	_1		No		star	t o	r Ec	lit V		м d	<u> </u>	-	Supp	ress	۷				
3 4	5	6	7 8	9	10 1	1 12	13	14 1	5 16	17	18	19 2	0 2	1 22	23	4 2	26	27	28 2	9 3	31	32	33	34 :	35 3	6 3	7 38	39	40	41	42 4	3 4	45	46	47	48	49 5	0 5	52	53	54	55 56	57	58	59 60	61	62	63 6	4 6	5 66	67	68 €	9 70	71	72 7	3 74
0 1	T	۰	PR	1	٧ 7	rΕ	R	1	4				ı,			DI	1	П		Г										П	1		Γ																				\perp	Ш		
0 2	\Box	0	Τ	T	T	T	M	T	Τ	Γ			Т	Т	П	T	Τ	П	1	T	Т	П	П	T	7	Т	Т	Γ	Γ	П		T	Т						Τ	T		7	Γ			Γ	П	П			П	П	T	П		П
 	\vdash	+	+-	+		+-	 	-+	+	1	Н	\vdash	+	+	1	+	1	Н	+	1	+	Н	-	\dashv	+	-†-	+~	1-	1	Н	\dashv	+	+	+	1	\vdash		-†-	1	+-	\vdash	+	1-	⇈	+	+	Н	1	+	1	\vdash	\dashv	+	П	\vdash	+

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	<u> </u>	-	_	_		_							_		_								-	_	_	_	_		lı	nter	neti	ona	l Bo	usine	ss M	Aacl	hin	es C	orp	orat	tior	1		_				-	-	_		_		-			_					-	_			_				21-9		
																					F	₹P	G			(o	U	TI	Ρl	ΤL	٠.	F	0	RI	M.	IΑ	т	5	SΡ	Έ	С	1F	10	ĊΑ	т	ıo	N	IS																					in U.		
Date																								Į	_	_		-	7	G	rap	hic		Т	7		7		Т	_	Т		Т	7	_	7										2	1			Pr	ogra	am			75	76	; 7	/7 丁	78	79) (30 7
Program	_					_													_							unc		ng tior	,	_	unc	_		H	+		+		\dagger	_	\dagger	_	t	\forall		1							P;	age	L	Ш				lde	enti	ific	atio	on	L	L	\perp	\perp	_	L	L	لـ
Programn	mer	_																						١	_				'					_					-	_			ـــا		_	_																										
l.—	Т	T		_	_	_				Т	Tá	:		Т	_	_	_	Т	_	_	_	-	-			_	Т			_				Т	_			_	_									_	_														_		_			Т		_		٦
	l	ı									1	Before Sold Sold Sold Sold Sold Sold Sold Sold	pac	е	: 5	Skip	•	۱	(Out	рu	t Ir	ndi	cat	or	S	l																	>	_				_				Ed	it (Coc	les			_		_		_		\neg			l				١
]]		ļ								ļ	٤	3	Ţ	t	_	Т		+			T			Г			┨								_	Τ-			_	т	2	i	•		a	omi	nas	1			lan	ces	No	Si:	gn	CF	3	-		×			mo us S			ĺ		l	Ste	erlir	ng	
Line					File	na	ne			(4/E	الْوَالِ		١	l		l		l		,	l. Ind		A	l nd			1	F	ie	ld	Na	me	9	Ш	(B)			nd osit	on	[Bina					Υε		Ť		Ye				1 2		A		J	٦	Y	=	Da		-		ĺ			Sig Po:	gn ositic	on	1
	125	ž.								VDe (H/D/	i la	a la			Before	١	a	ŀ.	T	Τ	T.	Τ	Τ	L	Τ	T	1							sapo	After	۱	ic			1	ed/B=					No	•	١		Ye	es.			3		C	:	Ľ	- 1	z	=	Ze				ı						1
	E E									2	a a	8		1	Bef	l	After	Š	2	1	Š		l	Š	1		l							Edit	Blank	l		ecc		ŀ	Pack				_						_	net:	ant	or.	Ed	i+ \	Ma	_		_	_							l				
3 4 5											10	"	7 1	8 1	9 2	0 2	1 2	2 2	3 2	2	26	27	28	29	30	31	1 3	2 3	33	34	35	36	37	38	39	40	0 4	1 4	2 4	3	ă 44	45	46	47	48	49	50	51	52										2 6	3 6	34	65	66	67	68	69	70	7,	72	2 7:	3 7	4
0 1	0	1	PI	R	1/	1	E	Ŕ	L	D)	1	7	4	1	1	1	1	1	1	L	L	L	L	L	1	1	1	_		·		L	L		L	1	1	1	1	_		L			L	Ц	L	L	L	Ц	4	1	1	1	1	1	1	1	1	_			L	L	L	L	L	ļ	1	1	
0 2	0	+	1	4	1	1	1	1	L	L	1	1	1	1	1	1	1	1	1	L	1	L	L	L	L	1					E		L	L		L	1		24		4		L	L	Ц	Ц		L	L	L	Ц		4	4	1	1	1	4	4	4	4		L	L	Ļ	L	L	L	ļ	1	1	4
0 3	0	4-	+	+	+	╀	+	+	1	ł	╀	+	╀	╀	+	+	+	1	+	+	╀	H	Ļ	-	ŀ	\perp	-	_		D	_		L		L	L	4		5 4		-	_	_			_	Н	L	-			4	4	4	+	+	+	+	+	+	4	_	_	-	╀	1	1	L	Ł	\downarrow	\downarrow	4
0 4	0	+	+	+	+	+	╀	╀	╀	1	╀	╀	╀	╀	+	+	+	╀	+	╀	╀	+	H	┞	╀	╀	1	¥	4	7	D	U	E	\vdash	H	ŀ	+	-	59	4	4		ŀ	_	-		H	L	-	_	Н	-	-	+	+	+	+	+	+	+	\dashv	_		ŀ	╀	H	Ł	⊦	╀	+	+	4
0 6	100	4-	+	+	+	╁	╀	╁	╁	╁	╀	+	+	+	+	+	+	╀	+	╀	ŀ	+	H	H	╀	╀	+	+	+	4	_	_	-	H		┞	+	+	+	+	+		\vdash	H				_	_			+	\dashv	+	+	+	+	+	+	+	4	-	_	H	╀	╀	┝	┞	╀	+	+	+
0 7	10	4	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	H	ŀ	+	+	+	-	4	-		-	H	H	-	+	+	+	+	4		-	-	-	-		-	-	H	Н	-	-	+	+	+	+	+	+	+	\dashv	_		H	╁	╁	╁	┝	╁	+	+	$\frac{1}{2}$
0 8	10	+	+	+	-	_							_								_			_								_		_								_	_		_													٦	Ì	+	+	_		┝	+	H	H	┝	t	+	+	$\frac{1}{2}$
0 9	6	†	\dagger	†	-																													а																								ı	ž	+	+	_	-	┢	t	t	r	-	t	+	\dagger	1
1 0	6	†	+	†	-												•																	ri				_																				ı	44	t	†	_	_	T	T	T	T	t	Ť	+	t	1
1 1	0	t	Ť	1	- [w																					W	S		ı		1	1			T	T	T	r	r	T	T	T	1
1 2	0	1	T	1	_																													ne																								1		Ť	T				T	I		Γ	T	İ	İ	1
1 3	0	I					(οv	er	٠f۱	0	W	li	in	е	(1	in	е	60)	h	as	b	e	er	۱ ۱	pa	as	se	ed	s	0	t	he	C	ΟV	/e	rt	lc	٥V	٧	ir	nd	ic	at	0	r i	S	tı	ır	ne	ed	0	n.						I							L		Ī	I	I	
1 4	0	4	1	1	_				70								_			2,0			_	_				_					- :					*					7111	. 30				100	16.89				136	W.	100	146	10 g		*	1	1			L	Ļ	L	L	L	ļ	1	1	4
1 5	0	+	1	4	1	Ī	+	+	-	-	1	+	+	1	1	7	Ŧ	+	+	1	Ŧ	+	_	-	_	+	+	4	-	_	_		_	-	H	-	+	+	7	+	_		-	-	Н	_	-	_	-	Н	Н	4	4	4	1	+	+	4	4	1	4			L	Ļ	Ļ	Ļ	L	Ļ	\downarrow	1	4
	0	╀	+	4	+	+	1	1	1	L	╀	╀	1	1	\downarrow	4	1	1	+	1	Ł	1	L	L	L	1	1	4	-	4	_		L	Н		L	1	1	+	4	4		L	Ц					L	Ц		4	4	4	4	+	+	4	4	+	4		Ц	L	1	-	L	L	ļ	\downarrow	\downarrow	1
\mathbb{H}	0	╀	+	+	+	+	+	+	H	╀	╀	╀	╀	╀	+	+	+	╀	+	╀	+	╀	\vdash	-	\vdash	+	+	+	-	-	\dashv	_	H	Н	H	H	+	+	+	+	\dashv	_	\vdash	H	Н	4	\dashv		H	\vdash	-	+	+	+	+	+	+	+	+	+	+	4	_	L	+	+	\vdash	\vdash	÷	+	+	4
++	0	+	+	+	+	+	+	╀	+	╀	╀	╀	╀	+	+	+	+	╀	╀	╀	╀	+	┞	H	⊦	+	+	+	\dashv	\dashv	-	_	L	H	4	┞	+	+	╬	+	\dashv		┞	Н	Н	Н	\dashv	_	-	Н	\dashv	-	-	+	+	+	+	+	+	+	+	-	_	-	╀	\vdash	\vdash	H	╀	+	+	$\frac{1}{2}$
H	6	╀	+	+	+	+	+	+	-	╀	╀	╀	H	╁	+	+	+	╀	+	╁	H	+	H	\vdash	H	+	╁	+	+	\dashv	\dashv	_	H	Н	-	-	+	+	+	+	4	-	\vdash	H	Н	\dashv	+		H	\vdash		+	\dashv	+	+	+	+	+	+	+	+	-	4	\vdash	╀	╁	+	-	H	+	+	$\frac{1}{2}$
	To	1	L	_	_	L	ı	Ι	1_	L	L	I.	L	_		T	1.	L	L	_	L	L	L	L.,	L	L	L	_	_	_	_	_	L	Ш		L	L	L		L	_1	_	<u>L</u> .	L	Ш			_	L	Ш	L.I			_1	_L	\perp	1	丄	1	1.	_	J	_	L	ــــــــــــــــــــــــــــــــــــــ	_	Ц.	L	1	上	┸	Ţ

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	'																			Ir	ntern	ation	nal B	usine	ss M	achi	nes C	orpo	oratio	n			_							-		-						-					-9090	٦
***														1	RP	G		(วบ	ΙΤΙ	PU	Т	- F	0	RI	VI /	١T	S	PE	C	IF	IC.	Αī	П	ΟN	S																	U.S.A.	
Date				_													г				Gr	aphi	ic	Т	Т		1	_	Т		Г	_	\neg							_		2 T			Pr	rogra	m	-	75	76	77	78	79 8	٦
Program																	- 1	Punc Instri				nch		+-	+		-	+	+	_	H	+								Page	Ľ	Ш			ld	lenti	ficat	ion]
Programn												٠.	-				_										١				L																							
rrogramn	ner.											_							_																_	_																		٦
								» S	pace	::	Skip	1	0	utpi	ut Ir	ndic	ato	rs	ı					L							~	> -							E	dit	Cor	des							٦		1.			
	l						Ш	Overf	_	L	_	+		-т					-					1					1.	¬	V	1	Cos	nma	s		Bal o Pri			No S	-	CF	1	-	×		Rem					Ster	lina	
Line	l	,	File	nam	е		ū	etch (.								Fie	ld f	Van	ne	L	ļ_	١,	End		Sinary			ŀ	,	Yes	$^{+}$		Yes		+	1	_	A	+	J	Y	=	Date				1	Sign Posi	١ .	
	ě						ĽQ/	lect/F	,	بو ا		ŀ		And	<u> </u>	Т	nd T	$\overline{}$	-					s	fter (E		Posit in	ton	/8 = 1					Yes No	İ		No Ye:	5		3		B	:	K L	z	=	Zero					. 03		
	Ę						Pe (ker Se	After	Befor		N P			Š		Not							Š	ak A		Outp Reco		acked			L	- 1	No	_		No	-	_	4			Ц.	M	1	_	Supp	press			ı			
II	1,5	١.,					٦	Stac	J.,	ļ., .				25										ŭ	1				- L													dit \								co 3	١.,		73 74	
0 1	+-	_	Po	$\overline{}$		_	Н	16 1	-	19	_	22 2	1	_	26 27	28	29 .	30 3	32	33	34	35 3	16 3	/ 38	39	10	1	T	3 44	45	٦	Π̈́	Ť	9 50	T .	52	53	1	5 54	<u> </u>	1	59 6	T	162	Ï	1	55 6	1	Ĭ	69 /	†	Ϋ́	73 /4	۱
0 2	0	75			1	0	1.1	+	r		1	H	Ó	-	t	T	H	t	T	Ħ	_	\dagger	+	t	t	П	\forall	\dagger	†	T	İ	H	7	\dagger	T	r		\dagger	\dagger	✝	П	1	t	T	П	7	1	†	П	Ť	t	T	ΠĦ	1
0 3	0	H	11		\dagger	Ť	n	1	T	Ħ	1		T	Ť	+	Т	П	\top	T	Ħ	T	1	Ť	T	T	П		2 5	5	•	S	A	L	55	;	R	ε	P	> 1	7	•	T	†	T	П	П	T	Ť	Ħ	T	T	Ħ	П	1
0 4	0		Π			-	H		3	П	T		1	P				Ī	T	П	\sqcap		1	Τ	Г	П			T	Ī		П	T	T																	T]
0 5	0		П			0	R						0	V										Π	Γ				Τ	Γ										T]
0 6	0						Ш																					4	4	1	1	7	EΙ	4	'						Ш	Ì	L							L	L		Ш	
0 7	0			Ш		L	Ц												L					L	L			29	1	1	Q	T	4	S	þ	L	D	1	\perp	L	Ц								Ш	┙	\perp		Ш	1
0 8	0	Ш		Ц	_	\perp	Ц	\perp	L	Ц	\perp		\perp		\perp		Ц	\perp	L	Ц		\perp	\perp	L	L	Ц		36	<u>.</u>	Ľ	T	0	T	44		A	1	0	IJΛ	T	<u>'</u>	\perp	Ŀ	L	Ц		4	\perp	Ц	4	╀		4]
0 9	0	Ц					_																		_											_									-					•	1	Ш	\dashv	┨,
1 0	0	Ц		lf	a r	ес	or	d	са	n b	е١	vri	tte	n	wł	ne	n e	eitl	he	ro	on	e (of	tv	٧C	C	10:	٦d	iti	or	าร	e>	cis	ts	, p	la	ce	t	he	C	วท	di	tic	ni	in	g				-	+		\dashv	┨
1 1	0	Н		ine	oib	at	or	in	a	n ()R	re	lat	ioi	nsł	nip).	Sp	ec	cif	у	th	е	in	di	ca	to	rs	sig	na	ali	ng	0	ne	c	01	٦d	iti	or	n (1F	' ir	n t	hi	S					-	+	1	⊢	-
1 2	0	Н		са	se)	О	n ·	th	e s	am	ie l	ine	e as	s t	he	li	ne	ty	'nθ	е.	P	lad	се	th	ıe.	in	di	са	ito	r	sig	gna	ali	ng	j t	he	9 0	th	er	C	on	di	tic	on					i	-	╀	\vdash	\vdash	+
1 3	0	Н		(O	V	in	tŀ	nis	Cá	se) o	n t	he	fc	ollo	οw	'in	gΙ	in	e١	wi	th	tl	he	w	0	rd	0	R	ir	י רו	ol	uı	mr	าร	14	4-1	15		Th	er	ı s	рe	ci	fy					-	╀	+	+	+
1 5	10	H										s t						_																													ngs	s		-	+	+	H	1
	0	Н										her									,													•								•	•				Ŭ			Ī	十	H	+	┥.
$\parallel \parallel \parallel$	10	Н		•••	٠. ,	φ			•••				•		•	_	•			•																															+	H	\vdash	1
	0	Н		Sr		۵,	an.	۸,	Ŀ	n e	nt	rie	c a	re	nc	١t	ne	ce	ccr	a۲۱	, :	n	th	و م	ΩI	R	lir	10	٦	Γh	ıe	en	ıtr	ie	e i	n	th	<u>م</u> ا	lin	e i	nr	eci	he	in	a ·	th	ıe.		-	•	T	T	П	1
	0	Н										to							336	ai y	y '	• •		, D	<u>ا</u> ر		•11	16,	•			U1		10.	<i>.</i> 1	• •	.,,	· ·			۱,		Ju	!	9					•	t	Τ.	П	1
- - -	0	F		0	١ ١	1111	C 6	115	U	1hh	ηУ	ιυ	ui	-	UF	1	1116	ፘ.								_	_	_						_			_									_			-	-	†	-	FF	1
	0				7																																													-	L	1		

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

١

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.

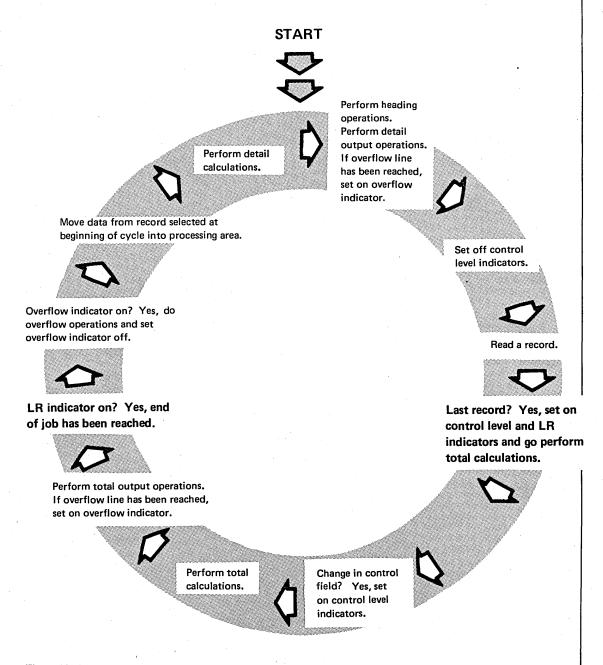


Figure 12. Program Cycle Operations for Last Record (LR) Indicator

RPG II Specifications

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	1	Int	ternational Business Max	chines Corporation		Form X21-9093
		RPG CAL	CULATION	SPECIFICATION	ONS 1 2	Printed in U.S.A. 75 76 77 78 79 80
Date		Punching	Graphic		Page	Program Identification
Program		Instruction	Punch] -	identification [
Programmer					•	
India	ators				Resulting Indicators	
	· · · · · · · · · · · · · · · · · · ·				Arithmetic	
Form Type Control Level (LO.L9, LR, SR) Not	And Factor 1	Operation	Factor 2	Result Field	Field E Compare	Comments
Type					Length & 1 High Low Equal	
Cont (LO-1)	No.				Lookup Table (Factor 2) is	
3 4 5 6 7 8 9 10 11 12 1	3 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	High Low Equal	62 63 64 65 66 67 68 69 70 71 72 73 74
0 1 C						
0 2 C	 					
0 3 C 0 4 C						
0 4 C						
				++++++		
IBM		lr	nternational Business Ma	schines Corporation		Form X21-9090 Printed in U.S.A.
	RPO	G OUTI	PUT - FORN	MAT SPECIFIC	ATIONS	75 76 77 78 79 80
Date		Punching	Graphic		Page	Program Identification
Program		Instruction	Punch			identification
Programmer						
	Space : Skip Output In	dicators				
	Overf			/r	Commas Zero Balances No Sign CR	- X ≈ Remove
Line Filename	1 121 1 1 1 1	Fie	Id Name	End III	Yes Yes 1 A	J Y = Date Sterling Sign
Ape.	And And And	And	Codes k After (E	Positon III	Yes No 2 B No Yes 3 C	L Z = Zero
orm Type	Type (H/D/T/E) acker Select/Fetce acker Select/Fetce After After Not	Not	Edit Codes	End Bind Bind Bind Bind Bind Bind Bind Bi		M Suppress
3 4 5 6 7 8 9 10 11 12 1	3 14 15 16 17 18 19 20 21 22 23 24 25 26 27	28 29 30 31 32 33		ايًا	Constant or Edit Word 48 49 50 51 52 53 54 55 56 57 58 59 60 61	
0 1 0	2					
0 2 0						
0 3 0	+++++++++++++++++++++++++++++++++++++++					 - - - - - - - - - - - - - - - - - - -
0 5 0	 		- 			
	 					

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:

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

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:

DANGAGTTON		7500757500			
RANSACTION DATE	I TEM NO	DESCRIPTION	QUANTITY	COST	EXTENDED COST
07/23/70	413010	CHOOL 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. 1.	28.44 39.50	28.44 39.50
	325781 412146	SATIN-CUT DADO SET CH143 BREAKER 15A	50	.89	44.50
	573022	6-VOLT POWER BATTERY	2	14.45	28.90
					141.34
			WEEKLY TOT	Δ1	9,573.49

This Printer Spacing Chart shows how the report is formatted:

G. ID					_	_					_						_			_ F	PA	G	Ε.	_		_		_							1			- F	old	ь	ock	at	da	tte	d li	ne.																							
IARACTE	R	PE	ER	IN	СН	, 6	LIF	1E5	P	ER	VE	RT	ICA	AL I	IN	СН)			١	DΑ	\TI	E _		_						_				i																																		
T.			_		_	_			_		_					_									-						_																																						
		_				_				_	_																																																										
	П	Ţ	П	T	П	Т	П	T	П	7	Π	T	П	П	П	T	П	Т	П	Т	T	1	7	T	T	_	П	7	T		_ 	Т	П	Т	Т	П	Т	τ	П	1	П	П	Т	П	Т	Т	П	П	Т	П	7	1	П	1	Т		П	7	Т	П	П	7	Т	Т	m	П	Т	Т	Г
	h	2 3	4	5 6	7	8	녱	1 2	3	1 1	6	1 1 7 8	9 6	2 2	2	2 2	3	2 2 6 7	8	2 9 0	3	3 2	3 3	3 3	3	3 7	3	3 6	1	4 2	4	5	4	4 4	5 4 8 9	5	5 2	5 3	3	5 5 5 6	5 7	5 8	5 6	6	6 9	6	5	6 7	6 6 7 8	6	6	7 2	7	4	7	7	7	7 6	î	8	3	4	8 8	3 8 8 7	8	8	9 9	9	9
. 1	П	Τ	П	Т	П	T	П	T	П	T	П	T	П	Т	П	T	П	1	Ħ	1	T	Ħ	1	1	T		П	Ť	T	T	Т	Т	П	Ť	T	П	Ť	1	П	1	T	П	1	Т	Ť	Ť	П	\top	Ť	П	7	Τ	Г	7	T		П	T	T	П	П		Ť	T	T	П	T	T	Г
2	П		П	Τ	П	T	П	T		T	П	T	П	Т	П	T	П	T	П	T	T	П	1	1	Τ	Г	П	T	T			Т	П	1	T	П	1	Т	П	Т	Т	П	1	T	7	T	П		T	П	1	T	Г	7	T		П	1	T	П	П	T	T	T	T	П	T	Т	Γ
3	П	Ι	\prod		П	I	П		П	Τ	П	Ι		Γ	П	T	П			T	Т	П	T	T	T		П	T	Τ		П	Τ		T	Т	П	Т	Τ		T	Τ	П	Т	Т	1	T		П	Т	П	T	Τ		Ţ	T		П	T	Τ	П	П	T	T	T	Γ	Π	Ι	Γ	Γ
4	Ц	\perp	Ц	L		1	Ц	I	П		П	I	П	Ι			П	I	П		L		T	Ţ	I			I	L			Ι					I	Ţ.		Ţ		П	I	Ι	\Box	I			Ι	П	Ι	Ι		Ι	I		П	I	I	П	П		Ι	I	Γ	П	I	Г	
5	Ц	L	Ц	1	Ц	\perp	Ц	\perp	Ц	I	Ц	L							П	İ	L	Ц		I	I	L			L		ì	\perp			Ι		I	I	Ц	1		П		L		I			L		\perp	l			Ĺ	L		1	L	П	Ц		1	l	L	П	⅃.	Ţ	L
_H 6	Ц	1	Ц	1	Ц	4	Ц	1	Ц	1	Ц	\downarrow	Ц	L	Ц	\perp	Ц	1	Ц	4	L	Ц	1	⊥	1	L	1	RV	M	2	4	: 1	1	0/	٧.	Ц	RI	G	1	5 7	IΕ	R	4	L	Ц	⊥	Ц	Ц	⊥	Ц	┙	⊥	Ц	Ц	1	L	Ц	1	L	Ц	Ц	Ц	4	1	L	Ц	┵	┸	L
7	Н	\perp	Ц	1	Н	4	Ц	₹.	Ц	1	Ц	1	Ц	\perp	1	1	Ц	1	Ц	1	Ļ	Ц	4	1	1	L	Ц	4	Ļ	L	4	╀	Ц	4	1	Ц	4	1	Ц	1	Ļ	Ц	4	1	Ц	1	Ц	Ц	4	Ц	4	↓	L	4	1	L	Ц	4	Ļ	Ц	Ц		4	╧	Ļ	Ц	╁	¥	L
_H .	Н	+	H	+	H	4	7	Ψ	М	SA	9	Τ!	0/	4	Н	+	Н			Εþ	1	Н	4	4	4	L	Ц	4	╀	D	Εļ	C	R	1 F	77	Щ	0/	V _	Н	4	Ļ	Н	4	╀	4	G	W	ΑM	V T	1	Ţ	4	L		W		Ţ	4	╀	Н	Ε				/D		4	+	L
_# 9	Н	+	Н	+	Н	+	H	+	K	47	4	+	H	H	Н	+	H	+	M	4	╀	H	4	4	╀	H	Н	+	╀	Н	+	+	Н	+	+	Н	4	+	Н	+	╀	Н	+	₽	4	+	Н	Н	+	Н	+	+	H	_K	:10	S	Ц	+	╀	Н	Н	-	Ξþ	15	Ţ	H	+	╀	H
10	Н	+	Н	+	Н	+	Н	+	Н	+	-	+	H	+	Н	+	Н	+	╢	+	╀	H	+	+	+	+	Н	+	╀	╀	Н	+	Н	+	+	Н	+	+	Н	+	+	₩	+	╀	Н	+	Н	1	+	₩	4	┿	H	+	+	╀	Н	+	╁	Н	Н	+	+	+	╀	Н	+	╀	-
D 12	Н	╁	Н	+	Н	+	Н	ⅹ┖	Н	vb	╁	νV	+	+	Н	+	Н	₩	₩	v	₩	Н	+	+	+	Y	И	V.	₩	₩	u	X	W	4	ďχ	U	ν	λ	W	٧k	+	Н	+	╁	H	+	Н	Y	₩	V	V	+	+	Χþ	٨	+	V	桾	╁	X	М	Н	₩	\star	/ -	₩	4	+	H
13	Н	┿	Н	+	Н	+	Н	Y	Ή	4	Н	₩	+	╁	Н	+	Н	₩	Ή	4	₩	Н	+	+	+	r	A	ᄱ	4	Α	Y	4	Α	4	₩	Ą	4	ŲΧ	H	¥	+	Н	+	╁	Н	+	Н	4	¥	44	4	+	H	4	Ψ	۴	Α	4				4	4	Ψ	ŀ	A	4	╀	۲
14	Н	┿	H	+	Н	+	H	+/	7	À.	t	-)	+	t	7	7/7	E	加	1	n	λz		d	ø	+	╁	И	1	Ł	4	Н	1/4	H	7	R	,	D -	+,	H	W	1	Н	+	t	1	+	W	A	A	+	, 1	t	1)	7	7	4	Н	-)	К	ď	Х	7	4	₩.	2	Đ	ተ	+	-
15	•	+	H	✝	Н	+	Ħ	+	Ť	+	H	1	11	+	1	"	Ħ	"	М	4	T	М	4	4	1	†-	Ħ	1	Ť	4	۲	1	Ħ	7	4	Н	7	4	М	4	1	Н	+	$^{+}$	H	۲	۳	П	4	H	Н	₩,	۲	۲	٣	1	П	1	t	Ħ	Н	9	٦	#	7	H	†	+	r
7 16		+	H	Ť	Ħ	1	Ħ	†	Ħ	+	H	\dagger	\dagger	Ť	Н	+	H	+	H	+	1	Н	+	+	+	t	Н	1	t	t	1	1	Н	+	+	Н	+	+	H	+	Ť	Ħ	7	t	H	†	П	H	+	H	H	†	t	H	+	t	Н	+	ħ	X	V		Χþ	ለ	đ.	X	XT-	+	r
17		+	П	1	Ħ		T	1	Ħ	†	П	1	H	†	П	†	Ħ	1	Ħ	+	1	Ħ	Ħ	+	Ť	t		†	t	T	Н	T	H	1	+	П	1	+	Ħ	1	T	П	T	T	Ħ	Ť	Ħ	1	+	П	T	1	T	1	Ť	T	П	1	Ť	T		7	*	T	۲	Ħ	1	1	r
18	П	T	П	1	П	П	П	1	П	1	Ħ	7	П	T	П	1	Ħ	T	Ħ	7	1	Ħ	Ħ	+	T	1		T	t	T	П	T	Ħ	T	T	П	T	T	Ħ	T	T	Ħ	1	T	7	1	П	П	T	П	Т	1	T	П	T	T	0	10	Ā	14		7	3	Ā	U	73	丌	T	Γ
19	П	Т	П	T	Т	П	П	T	П	1	П	T	П	T	П	\top	П	1	П	T	T	П	T	T	T	T	П	1	T	T	П	T	П	T	T	П	T	T	П	7	T	П	1	T	7	T	П	П	T	П	T	1	Γ	П		Τ		1	Т	П	П		T	Ť	T	П	Т	Т	Γ
T 20	П	T	П	T	П	П	П	T	П	1	П	T	П	T	П	T	П		П	T	T	П	T	7	T	Τ	П	7	T	Т	П	T		T	Ť	П		T	П	T	T	П	1	T	1	VE	E	K	44	1	T	dī	A	1	Т	i	П	XI.	N	X	X		XI)	ХY	(X	ХĪТ	Т	
21	П	Τ	П	T	П		П		П	Ţ	П		П	Т	П	T	П		П		T	П	П	T	T	T	П	1	T	T	П	T		П	Т	П		T	П	1	T	П	Ţ	Т	П		П	П	Τ	П	T	T	Γ		1	1		7	Τ			7	T	Τ	Ι	П	I	Ι	Γ
22		Ι	П	\prod	Π	\Box		T	5	7	h	al	e	Ι	5	p	a	C	e		6	le	? Ŧ	a	li	7		I	1	10	2	5 .		\Box	Τ			Τ		I	Ι	П	Ι	Ι	\square	Ι		П	Τ	П		Ι	Ι		1	U	V.	EE	1	ţ۲,	1		<u>\$</u>	A	L	ES	I	Ι	
23	L		П					Ι			П	1	П	L		1		Ι	\prod	Ι	Ι			I				\Box	Ι	L	П	I			Ι			Τ		I		\prod	\perp	I				П	Ι	П	\square	Ι	L	\Box	l	L			l				I	I	Ι	Ц	1	\perp	
24									${\mathcal D}$	de	1	ıΙδ	2	!	Ы	0	2	9	\$	V	le	17	u	Je	4	ŧ,	7	4	10	11	a	ď		a	n	И	-	to	И	di	Æ		ik	10	Ş	1		Ш		Ш			L			L			L		L		┙			Ш	⅃	\perp	
25	П	\perp	\square	I	Π					I		Ţ	П	Ι	1	Ι	\square	I			Γ			I	I	Ι		Ι	Ι	I		Ι			Ι		Ι	I		\perp	Ι	П	I	Ι		Ţ			I			Ţ,		П	I	L			I		П		I	I	I	П	1	\perp	L
26	П		П	\prod		П	П		Ĺ	\mathbf{d}	M	le	1	2	1	20	4	e	1		1/	10	21	4	2	Ł	e	I	b	r	1	t_{I}	1	þ	1		4	11	19	1		4	de	16	Ŀk	J		Ŀ	I_{ℓ}	27	d	1	Ĺ		1			\perp	1	Ĺ	Ц		1	1	T	П	1	\perp	
27	L	1	Ш	1	\perp	Ц	Ш	1	\perp	\perp			Ц	L	Ľ		\prod				ľ	П		Ī		L	Ц	_[Ī	Ĺ	\prod	\perp	Ц	1	\perp	Ц		L	Ц		\perp	ĹĬ		L	Ц	5	1	Ц	1	П	Ц	\perp	1	Ц	1	L	Ц	Ц	1		Ц	Ц	\perp	1	1	Ш	1	\perp	L
28		1	Ш	I		Ц	\sqcup	I		Ι	П	I	П	L	П	I	П	\perp			Γ	П		I	Ι			I	Ι			Ι			I	П	\Box	Ι		I	I	П	Ι	L			\perp	Ц	\perp	Ц		\perp	L	Ц	Ţ	L	Ц	Ц	I	\perp	Ц	Ц	1	1	L	Ц	1	1	L
20		- 1	П	-1	1		1]	1	1	- [ŧΪ	- [Ιſ	Γ	ΗĪ	Ī	\Box	1	17	-	1	П	T	T	Ι	1	1]	ſ	ľ	1	1	1	1]	ſ	Ī			T	ΙŢ	1	1	ΗĪ	ſ	Ī	lĺ	t	1	ł ſ	1	11	ı	1	!		1	1	П		ı		H	H	ļ	1	1	1 1	4		1

JOB SPECIFICATIONS

IBM	International Business Machines Corporation	Form X21-9092 Printed in U.S.A.
RPG CONT	ROL CARD AND FILE DESCRIPTION SPECIFICATI	IONS
Date 1/10/71	Punching Graphic	Page Program TRANS
Date 1/10/71 Program Transaction Register Programmer L K Hoffmann	Instruction Punch	identification [
Programmer L K Hoffmann		
		ne OV indicator. Headings are
	File De: to be printed at the top of ev	· · ·
File Type	Mode of Processing	exactly where headings are to
File Designation	Length of Ki print. The overflow indicator	r is, therefore, assigned to the
End of File	Record At Overflow automatically	or that it should not handle
Line Filename Sequence	Type c Over How automatically.	<u> </u>
Q File Format	or Adc	Constitution
Block Record Length Length	Key Field Starting St	Core Index
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	Location Location 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 67 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 2 F TRAMS IP		
03 FREPORT 0 9	S PRINTER	┌┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼
110 4 IFI		Form X21-9094
IBW	International Business Machines Corporation RDC INDIT SPECIFICATIONS	Form X21-9094 Printed in U.S.A.
1/10/71	RPG INPUT SPECIFICATIONS	1 2 75 76 77 78 79 80
Transaction Register	Punching Graphic Instruction Punch	Program TRANS
Date 1/10/71 Program Transaction Register Programmer LK Hoff Mann	Punch	
l e	cord Identification Codes	
Form Type Sequence Number (1:N) Position (0) Record Identifying Indicator Record Identifying Indicator Option (0) Not (N) Not (N)	cord Identification Codes Field Location	Field Indicators
l l l l l l l l l l l l l l l l l l l	Big 1	eld Name Chairming Fields C
Line Filename VI (O P) Position	. [] [] [] [] [] [] [] [] [] [eld Name Sign Pus Minus or Sign Position
Florm Type Sequence Sequence Option (N) Mumber (1:N) Position (O) Position (O) Record Identi (N) Not (N) Not (N) (N) (N) (N) (N) (N) (N) (N) (N) (N)	Position 1 1 1 1 1 1 1 1 1	old Name Starting Field Record Relation Pield Record Relation Starting Sign Pred Record Relation Starting Sign Pred Blank
ο		55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
OI ITRAMS		
	1 60 DA	TE L1
No new entries are made of	the Input sheet	EMNO
No new entries are made o		
	33 37ØQT	
		Form X21-9093
IBW	International Business Machines Corporation RPG CALCULATION SPECIFICATIONS	Printed in U.S.A.
Date 1/10/7/		1 2 75 76 77 78 79 80 Program
Program Transaction Register	Punching Instruction Punch Punch	Page \$\vec{\phi_3}\$ Program Identification TRAMS
Programmer L K Hoffmann		
<u> </u>		Lines 01 and 02 are the same as
Indicators	1 1 1 1 1 1	for Job 3. Line 03, conditioned
		by L1, will be done only when a
Line gas Garage And And Factor 1	Operation Factor 2 Result Field Field Length & Early	control break occurs. Once daily
Form Ty Control 1 Not Not	1 15151	sales have been accumulated for a group (L1 is on), daily total can
		be added to the field WEEKTO
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 3	7 28 29 30 31 32 33 34 35 38 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53	used for accumulating weekly total.
0 1 C QTY		(Weekly total could also have been
O 3 C L 1 DAYTOT		obtained by adding EXTCST to
0 4 C		WEEKTO for every record read.)

l	ļ									nternationa																	Form X	
Date /	.//	0/7/					RPG		-	PUT -	FOR	TAN	SPEC	IFIC	AT	ION	S			Ø	2		Proc	ıram				79 80
Program	Z	Fansaci	tion	Re	ais.	ter			ching ruction	Graphic Punch	+1		-	++	-				Page	Ø	4		ider	ram Itificat	tion	TR	AN	S
Programi	mer_	10/71 Fansaci La Doni	na i	40 i	ffm.	ann	,																					
	Τ		Ē		Skip		put Ind	licators	I					~ '													T	
			Overflo			ļ	T	1	4					~> 	Comi	mas Z	ero Bal	ances	Edit No S		es CR	-	x =			-	ξ,	erling
Line	l	Filename	T/E) /Fetch			۸ ا	And	And	Fie	ld Name	, H _e	End Positor	Binar		Ye Ye		Yes		1 2		A B	J K	γ -	Plus Date Field			Si	
	n Type		r Select	After	Berore After	Not	Not	Not	1		Codes k After	in Output			No No	5	Yes No		3		C D	L M	Z =		press			
	2	7 8 9 10 11 12 13	Fig	11	1	111	111	28 29 30 3		24 25 22	Blar	Record 40 41 42				ro re			ant o				ca ca	or o		en en		
0 1		REPORT	HOR	20		119		28 29 30 3	32 33	34 35 36	37 38 39					50 51	52 03 .		30 57									
0 2	0		DR	+	H	OV	1	+++	\mathbb{H}		+		8 '	TO	A A	CA	1 7	10	-		ر ہر د	,	67	,,,,	2/		+	##
0 4	0	 	4	1	Ш	10	1							1	nn	חכ			/ V		, E	7/	3/				+	++
0 5	0	╂═╂═╂═╂═╂═┼═┼═┼	OR	Ш	Ш	OV	1	\Box				0	۱,	TO	24	- 0								-		\parallel	\bot	Ш
0 6	0	 		++	H	+++	Н	╂	+			3	2 1	TR	AN SC	DH	07	10	N N	-	╁	+	//	EM	7	$\frac{1}{1}$	+	++1
0 8	0				Ш		Ш					5 7 9	8 1	DE QU EX	AN	71	7 Y		υ	N/	7	/					廿	111
0 9	0	 		3	Ш	1 /			+	44		9	0 (ΕX	TE	ND	ED				\parallel		H	\dashv	- -	+	$+\!\!\!+$	+++
1 1	0		OR		H	OV		+++	++-				\forall	\vdash	H		+	$\forall \exists$	+	\vdash	$\dagger\dagger$		+	+	+		+	++
1 2	0			\Box	П		Ш		П			2	9 '	DA	TE							40	•	П	П		\prod	
1 3	0	 		1	H	++	++		\mathbb{H}	+	H	8	RI I.	Co	5 7		+	\mathbb{H}	CO	5/		+	+	╁	+	\dashv	$+\!\!+$	+++
1 5	o				Ш	L1			DA	TE	Y	1	8														丄	
IBM	ļ									nternational																	Form X2 Printed i	
Date	///	וד/מ					RPG	i '	וויטט	PUT -	FOR	/IAI	SPEC	IFIC	AII	ION	5			ø:	2		Proc	Fam				79 80
Program	T	0/71 vansact	OH	Reg	rist	<u>e</u> r		Punc	thing ruction	Graphic Punch	+	-	+	\vdash	\dashv				Page	0	5		Iden	tificat	ion	R	AN	5
Programn	mer 🛊	La Donn	a H	07	Y mo	<u>ann</u>	,																					
			(F) WG	pace :	Skip	Out	put Ind	licators		,				~ '					Edit	Code						7	T	
			Overfic	+	$\overline{}$	┼	1	1	-				12		Comr	nas Z	ero Bala	nces	No S		CR	-	x =	Rem		1	St	erling
Line		Filename	/T/E)			A	And	And	Fie	ld Name		End	Ē	ŀ	Ye	-	Yes		1 2		A B	J K	Y =	Date Field	e d Edit		Sig Po	gn sition
	m Type				'						1 1 -	Positor	\ <u>@</u>		Ye		No					L		Zero)	1		ľ
3 4 5	Ē		pe (H//	After	After	Not	Not	Not]		it Codes ink After	in Output	믦			s			2 3 4		C D	M	Z =	Supp	oress	J		
-	6 6	7 8 9 10 11 12 13	F	7 18 19	- 1	2 23 24 25		111	1 32 33	34 35 36	Blan Edit	in Output Record	= Packed/B	46 47	No No	es o	No Yes No	onsta	ant o	. r Edi	t Wo	м rd		Supp		E8 69 '	0 71 7	2 73 74
0 1	0	7 8 9 10 11 12 13	F	11	- 1			111	17	EMN	화 명 37 38 39	Output Record	13 44 B = Backed/B	46 47	No No	es o	No Yes No	onsta	ant o	. r Edi	t Wo	м rd		Supp		68 69	0 71 7	2 73 74
0 2	0		F	11	- 1			111	IT DE	EMN SC	37 38 39 0	in Output Record	13 44 45 E Packed/8	46 47	No No	es o	No Yes No	onsta	ant o	. r Edi	t Wo	м rd		Supp		68 69	70 71 7	2 73 74
0 2 0 3 0 4	0 0		F	11	- 1			111	UT DE QT CO	EMN SC Y ST	37 38 39 0 3	in Output Record	1	46 47	No No	es o	No Yes No	onsta	ant o	. r Edi	t Wo	м rd		Supp		68 69	70 71 7	2 73 74
0 2 0 3 0 4 0 5	0 0 0		14 15 16 1	7 18 19	- 1	2 23 24 25	5 26 27 2	111	UT DE QT CO	EMN SC	37 38 39 0 3	in Output Record	1	46 47	No No	es o	No Yes No	onsta	ant o	. r Edi	t Wo	м rd		Supp		68 69	70 71 7	2 73 74
0 2 0 3 0 4 0 5 0 6	0 0 0		14 15 16 1	11	- 1		5 26 27 2	111	DE QT CO EX	EMN SC Y ST TCS	14/93 83 39 39 39 39 39 39 39 39 39 39 39 39 39	40 41 42 57 7 7 9	13 44 45 14 45 = Dacked/B	46 47	No No	es o	No Yes No	onsta	ant o	. r Edi	t Wo	м rd		Supp		68 69	70 71 7	2 73 74
0 2 0 3 0 4 0 5 0 6 0 7 0 8	0 0 0 0		14 15 16 1	7 18 19	- 1	2 23 24 25	5 26 27 2	111	DE QT CO EX	EMN SC Y ST	14/93 83 39 39 39 39 39 39 39 39 39 39 39 39 39	40 41 42 3 5 7 7 9 9	44 45 E Backed/B		Ye No No	50 51	No Yes No	Consta	4 4 ant o	r Edi 58 59	t Wo	м rd		Supp		68 69	70 71 7	2 73 74
0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9	0 0 0 0 0 0 0		14 15 16 1 7 3	7 18 19	- 1	2 23 24 25	5 26 27 2	111	DE QT CO EX	EMN SC Y ST TCS YTO	1 1 1 B	10 Output Record 40 41 42 3 5 7 7 9 9 9 7	13 14 45 Packed 8		Ye No No	50 51	No Yes No	Consta	4 4 ant o	r Edi 58 59	t Wo	м rd		Supp		68 69	70 71 7	2 73 74
0 2 0 3 0 4 0 5 0 6 0 7 0 8	0 0 0 0		14 15 16 1 7 3	7 18 19	- 1	2 23 24 25	5 26 27 2	111	DE QT CO EX	EMN SC Y ST TCS	1 1 1 B	40 41 42 3 5 7 7 9 9	13 14 45 Packed 8		Ye No No	50 51	No Yes No	Consta	4 4 ant o	r Edi 58 59	t Wo	м rd		Supp		68 69	70 71 7	2 73 74
0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 1 0 0 1 1 1 2 2	0 0 0 0 0 0 0 0 0		7 J	7 18 19 1 2 2	20 21 22 21 22 1 22 1 22 1 22 1 22 1 22	2 23 24 25 L1 LR ent lin	nes a	28 29 30 3	DA DA DA	EMN SC Y ST TCS YTO	37 38 39 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	10 output Record 40 41 42 3 55 7 7 7 9 9 9 1 7 9	9 des	₩ €	Ye No. No.	50 51 LY	No Yes No C	27.	### A L	r Edi 58 59	t Wo	Iin	63 64 63 64 63 64	65 60	ulc		70 71 7	2 73 74
0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 1 0 0 1 1 1			7 1	7 18 19 12 2 1 ix di e pri	ffere	LR ent lin	nes ai	re to hen 1	DA DA be pi	EMN SC ST TCS YTO EKT	37 38 39 39 37 38 39 39 39 39 39 39 39 39 39 39 39 39 39	in Output Record 40 41 42 5 7 7 7 9 9 9 1 1 x area	99 des	we cribe	Ye No No	50 51	No Yes No C	27.	AL he	r Edi 58 58 7 adi w ł	ng	line	es s	65 66	ulc	i	71 770 73 7	2 73 74
0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 1 0 1 1 1 2	0 0 0 0 0 0 0 0 0		7 1 7 5 b	ix die pri	ffere nted	LR ent line e adva	nes arer whanced	re to Inen 1Fd). T	DA DE DA DA be pi be pi hus,	EMN SC Y ST TCS YTO EKT rintecon (fi each	37 38 39 9	in Output Record 1 40 41 42 45 45 45 45 45 45 45 45 45 45 45 45 45	gpayard 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	we cribe	EK ed. DV	The is o	No Yess No C	27.	AL he	r Edi 58 58 adi w ł	ng use	line occed i	es s cur	sho and	ulc I ar	I	71 770 71 7	2 73 74
0 2 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 1 0 0 1 1 1 1 2 1 3 1 4			7 1 5 S b fc re	ix die pri	ffere nted have	ent line e adva	nes are whance otice	re to Inen 1Fd). Te the	DA DA be propies of hus, first	EMN SC Y ST TCS YTO EKT rintecon (fi each	37 38 39 9	in Output Record 1 40 41 42 45 45 45 45 45 45 45 45 45 45 45 45 45	gpayard 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	we cribe	EK ed. DV	The is o	No Yess No C	27.	AL he	r Edi 58 58 adi w ł	ng use	line occed i	es s cur	sho and	ulc I ar	I	71 70 71 7	2 73 74
0 2 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 1 0 0 1 1 1 1 2 1 3 1 4			7 1 5 S b fc re	ix die pri	ffere nted have	ent line e adva	nes are whance otice	re to Inen 1Fd). T	DA DA be propies of hus, first	EMN SC Y ST TCS YTO EKT rintecon (fi each	37 38 39 9	in Output Record 1 40 41 42 45 45 45 45 45 45 45 45 45 45 45 45 45	gpayard 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	we cribe	EK ed.	The is o	No Yess No C	27.	AL he	r Edi 58 58 adi w ł	ng use	line occed i	es s cur	sho and	ulc I ar	I	71 770 71 7	2 73 74
0 2 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 1 0 0 1 1 1 1 2 1 3 1 4			7 1 5 S b fc	ix die pripresselation	ffere nted have a form	ent line either advarip. Norms with the text and the text	nes al acceptance of the control of	re to Inen 1Fd). Te the	DA DA DA be pi chus, first e.	EMN SC Y ST TC S YTO Fintecon (fine each head	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	in Outputh Record 1 40 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 42 41 41 41 41 41 41 41 41 41 41 41 41 41	des r wh	we cribe en C ditio	EK ed. OV one p sp	The is ode by opecial	Po No Yes	ree ver	heeflood Con.	adi w ł	ng nas use s m	line occed inus	es s cur in a	sho red an (ulc I ar OR	I and e	70 71 7	2 73 74

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.

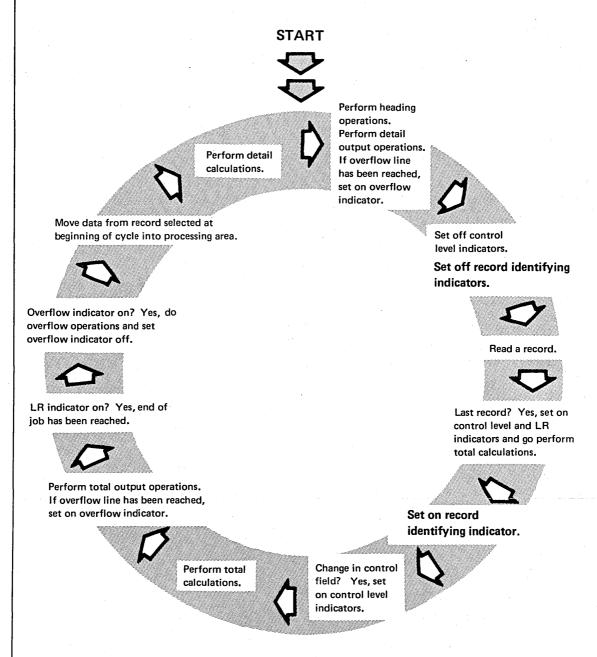


Figure 13. Program Cycle Operations for Record Identifying Indicators

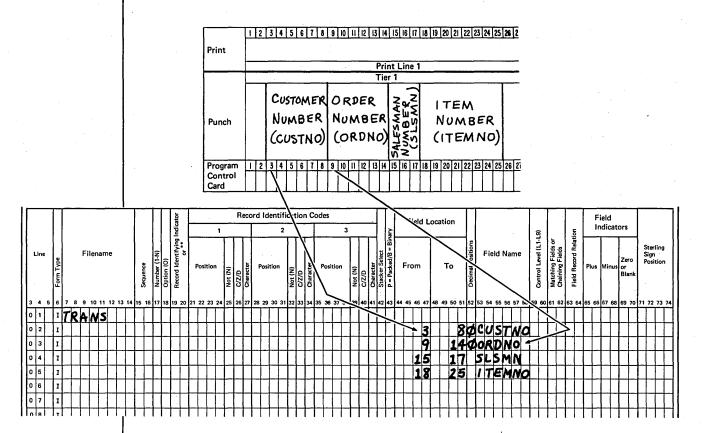
RPG II Specifications

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	1 2 75 76 77 78 79 80
Date	Punching Graphic Instruction	Page Program Identification
Program	Instruction Punch	
Programmer		
Form Type Sequence Number (1-N) Potorior (0) Pecord (dentifying Indicator or ***	Record Identification Codes 2 3 Field Location	Field Name Field F
	Not (N) Not	Natching Gordon Municipal Section Position Plant Matching Gordon Position Plant Minos of Plant M
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 3 1		
TORE	International Business Machines Corporation	Form X21-9093
IBM	RPG CALCULATION SPECIFICATIONS	Printed in U.Ş.A.
Date	[1 2 75 76 77 78 79 80
Program	Punching Graphic Instruction Punch	Page Identification
Programmer		
		Resulting
Indicators Ex And And Factor 1	Operation Factor 2 Result Field Field Length	Indicators Arithmetric Plus Minus Zero E High Low Equal 1 > 2 1 < 2 1 = 2
Line 8 8 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	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	Lookup Table (Factor 2) is High Low Equal
		
		Form X21-9090
IBM	International Business Machines Corporation RPG OUTPUT - FORMAT SPECIFICATIONS	Printed in U.S.A.
Date		1 2 75 76 77 78 79 80
Program	Punching Graphic Instruction Punch	Page Program Identification
Programmer		
	utput Indicators	Edit Codes
l log		Balances o Print No Sign CR - X = Remove Plus Sign Sterling
Form Type Type (H/D/T/E) Blacker Select/Fetch (a Before After After Not	And And Field Name @ End @ Yes Yes	Yes 1 A J Y = Date Sign Position
rm Type (H/D cer Seleci Seleci Before Before Not	Becord B Rec	Yes 3 C L Z = Zero No 4 D M Suppress
		Constant or Edit Word
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 0 3		

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:



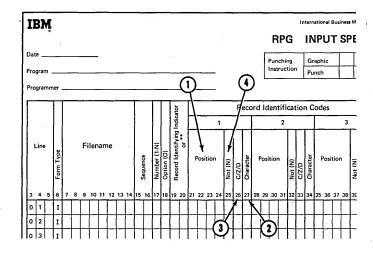
When you describe fields from each record type in the input file, give a unique field name to every *different* field from all record types. If, however, two or more record types contain identical fields, you can assign the same name to the field in each record type. Only one storage area would be assigned 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.
- What the character is.
- 3 The letter C to indicate character.
- 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.

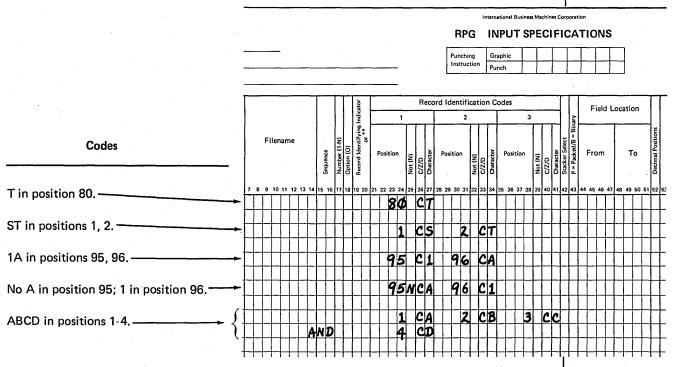


Figure 14. Valid Specifications for Record Identification Codes

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:

Program		_						_1							_	_	_						_			_		_		_													_					_				_	_						
Date	IBN	ſ																					1	Inter	matic	onal	Busi	ness f	Mach	nines	Corp	oorat	tion																										
Program Progra		•																			RI	PG	;	11	NF	'n	T	SP	ΕC	CIT	=10	CA	T	10	NS	3																							
Program Prog	Date					_														Γ.				T.		_			Ť	\neg	_	_	Т	_	_		٦							Par	_	1 2	2		Pr	rogra	ım		7	5 7	76 7	77	8 7	79 8	ាំ
Filed Continue Filed Con	Progran	٠_														_								-			\dashv		╁	+	-		+	┪		-	\dashv								Ľ	.1.	J		ld	ienti	ficat	tion	L	_	_			_	لـ
Record Identification Codes 1 2 3 4 6 6 7 8 9 10 11 12 13 4 15 16 17 16 19 20 12 12 12 14 15 16 17 16 19 10 17 17 18 18 18 17 18 18	Program	nmer																		L	-			1		<u></u>							1_			l																							
Filename Filename			_						_					_		_										_					_										_					_		_			_					_			_
3			l							ı	1	cator		<u>_</u>	_			Rec	orc	d ld	enti	ific	atio	n	Coc	des	_				_			Fi	eld	Loc	cati	on																		ı			
3	ı	ļ							1	-	ł	Indi			_	1	_		1			2	_	_	1	_	_	3		_	4	nary	L			_										ē				tion	L	ın	aica	310	rs —	1			
3	Line				Fit	enai	me			ı	إ	i si					П	i				1	ľ	ı	1			-	Н		٦	3= Bi								itions	F	ield	dМ	Jam	e	1		ospla str	اد	1 Rela		ı		١					1
3		2	3				•			92	=	2 5		Po	sitio	,		١	5	Posi	tion		_	١	<u>.</u>	Pos	itior	,		;	r Sele	/ked/		Fro	m			Го		al Pos						1	Š	ng Fi	-	Secon	PI	us	Mini	us	Zero or				
3		1	5							edne	팀						ĕ	/Z/D	9			1	72/5		ğ			ď	0 t	0/Z/	tacke	- Pa	ļ							ecim						1		latchi		ield F		1		ď	Blani	1			
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.	1.	"									Ţ	1				١	П	ı	1			Ţ	1	Į.	1			ļ		-	Ţ.	1														ľ			1		l	ļ							
Secus Two Secu	-	5 6	-	_	_		_	13 1	15	16	17 1	-	_	21 2		~	-			8 29	_	-	_	_	_	5 36	37	38	39 4	40 4	1 42	43	44	45	46 4	7 48	8 49	50	51	52	53	54 5	55 T	56 5	7 58	59	60	61 6	2 6	3 64	65	66	67 E	58 6	69 70	2 71	72	73	4
1 1 1 1 1 1 1 1 1 1	0 1	-1:	/	IA	5	TΕ	K	+	╀	Н	+	P	1	+	+	1	H	בנ	4	+	H	4	+	- 6	4	+	H	Н	\dashv	+	+	H	H	H	-	╬	+	+	0	_	•		_			H	4	+	+	+	╀	Н	H	+	+	+	-	\vdash	4
SALADR S			+	+	H	+	\dashv	+	╀	\mathbb{H}	+	+-	H	+	+	Н	H	+	╀	+	Н	4	+	+	+	+	H	Н	+	+	+	H	H	Н			╀	-	ğ	Ø	9	7	2	Ц	VC	H	Н	+	+	+	₽	Н	Н	+	+	+	╀	H	4
S I		-1-	+	╁	Н	+	+	+	╀	H	+	╁	Н	$^+$	+	Н	\vdash	+	+	+	Н	+	+	+	+	+	₩	Н	+	+	╀	\vdash	H	-			╁	2	2	H	N	4	7	-		Н	-	+	+	+	₽	Н	Н	+	+	╁	╀	Н	\dashv
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.		-+-	+	+-	\vdash	+	+	+	╀	Н		╀	+	\dashv	+	Н	Н	+	+	+	Н	+	+	+	+	+	╁	Н	+	+	+	╀	 	Н	<u>5</u> ,	<u>†</u>	+-	5	X		2	1	-	A!			Н	+	+	+	₽	Н	H	+	+	╀	╁	\vdash	\dashv
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.		-1-	+	╁	Н	+	+	+	+	H	+	1	12	\vdash	+	4	Н	7	_	+	H	,	+	,	+	+	╁	Н	+	+	╁	+	H	Н	Э,	7	+-	6	3	Y	2/	4/	-	Ž į	F	Н	\vdash	+	+	+	₽	-	H	+	+	╁	┾	╁	\dashv
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.	\square	-	+	╁	H	+	+	+	+	H	+	_¥	14	\dashv	+	1	H	4	1	+	H	-	╀	1	+	+	┼-	Н	\vdash	+	╁	+	┝	Н	١,	-	+	╁	0	4	0	,,	_	Ŧ.		Ӈ	\dashv	+	+	+	₽	Н	H	+	+	╁	╁╴	H	\dashv
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.	\Box		4-	╁	H	+	+	+	╁	Н	+	╁	+	\vdash	+	Н	H	╬	╁	╁	H	+	+	+	╬	╁	\vdash	H	\dashv	+	╁	Н	H	Н			+	4	0	2		0 1	ว ก			1	\dashv	+	+	+	╀	Н	H	+	+	╁	╁	Н	┨
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.	 - - 	-1-	+	╁	Н	+	+-	+	╁	+	+	÷	+	\vdash	+	Н	Н	+	╁	+	Н	+	╁	╁	+	┿	H	Н	\vdash	+	╁		-	Н			+	15	6	Ψ	2		יע	20 7		Н	Н	+	$^{+}$	+	╁	Н	H	+	+	╁	-	₩	\dashv
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.	 			+	H		╁	+	╁	+	+	╁	╁	\vdash	╁	H	Н	+	╁	╁	Н	+	$^{+}$	+	+	+	╁	Н	\vdash	十		╁	H	Н	7	2	╁	2	1	H			;	DA	17	\mathbb{H}		+	$^{+}$	+	+-	Н	Н	+	+	+	╁	H	\dashv
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.	H	+	+	$^{+}$	Н	+	+-	+	$^{+}$	Н	+	╁	t	十	+	H	H	+	╁	╁	H	+	╁	\dagger	+	+	+	H	H	十	╁	┢	H	H	7	4	+			4	S	יוני ע	,	D:	7 7	3	H	+-	+	+	H	Н	H	+	+	╁	╁	H	\forall
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.		٦,	†	+		<u>-</u>	<u> </u>			 	_	±	1-	ᆣ	土	<u> </u>	ᅼ		-	+		+	+	+	\pm	土	<u> </u>	<u>⊢</u>	<u> </u>	<u>+</u>	-}	<u> </u>	<u> </u>		0	1	\pm	_	1			7	4	4	- 1	1	\vdash	<u> </u>	<u>+</u>	<u>+</u>	<u>+</u>	-	<u> </u>	†	+	十	╁	H	┨
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.	1 3	1	1	1	1		_																																														1	1	+	十	t	H	┪
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.	1 4	1	1	+	Ħ																																																ł	lt	+	+	十	H	┪
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.	1 5	1	ı	╁	Ħ	•	the) C	od	e (CA	li /	n þ)09	iti	or	าร	1	an	ıd	2.	١	t	is	as	si	gn	ed	l r	ес	or	d	id	ler	nti	fy	/in	g	in	di	ca	to	or	0	1.	Т	h	9 0	th	ne	r		ı	▋┞	+	t	t	H	┪
that this record is in the master file because MASTER was the last file name given.		١,	†	T	Ħ	ı	rec	or	d t	typ	oe,	, w	/hi	ch	is	as	ssi	gn	ec	l r	ec	or	d	id	ler	nti	ify	/in	ıg	in	di	са	tc	or	02	2,	is	id	er	nť	ifi	ed	ı	οу	S	Τi	in	pc	osi	iti	or	าร	ı	۱t	╈	十		Ħ	┪
that this record is in the master file because MASTER was the last file name given.		1	1	1	Ħ		1 a	nd	2	. 1	No	ote	t t	nat	t n	0	fil	e r	าล	me	e is	sι	ısı	ed	f	or	· tl	ne	se	ec	on	d	re	ecc	orc	j t	ij	ре		T	he	С	0	mţ	ilic	er	as	ssu	ım	nes	;			I	\top	+	T	Ħ	1
		1	:	T	Ħ																																																	It	\top	+	T	H	1
		1	1	T	١L																							•						_				_		-			-	J.			•							It	\top	†	T	\prod	1
		\prod_{i}		I		ı	1		1	1.1		1.	1		Ī							-	I	ı	7	T	ı.	1		T	Ī	<u> </u>	i I		1	1	,	ı		_		1	-		,			-	-	,	-			!	\top	T		П	٦

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:

Γ	_			Γ					_					1	T	T	iţo	1						R	есо	rd	ld	ent	tifi	cat	ior	ı C	od	es			_						Ei	ماد	-		atio	20						-
				1										١		1	Indicator					1							2							3					Binary			CIC	, _	.00	,,,,				ŀ			
	Lin	e	Form Type			F	ile	na	me	•			Sequence			Option (O)	Record Identifying	or	P	ositi	ion	13	Not (N)	C/Z/D	Character	F	Posi	tior		Not (N)	C/Z/D	Character	P	osii	tion	- 1	Not (N)	C/Z/D	Character	ة	4/B		ro	m			т	ō		Decimal Positions	 	Fie	ld N	
3	4	5	6	7	8	9	10	0 1	1 1	2	13	14	15 1	16	17	18	9 :	20	21 :	22 2	3 2	4 2	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39 4	10	41	12	43	44	45	46	47	48	49	50	51	52	53	54	55 5	ji
0	1		ī	1	A	ı	70	1	7				A	A		į	1	ø		_		L		C	A																													
0	2		I		T	I				I			1		I	I																				1																		
6	3	Γ	1	Γ		T	I	T	T				7					1			1	1	I	_												ł	T				1												T	
0	4		1	Γ	Τ	Γ			T				B	В	T	Ţ	2	Ø			1	L	(C	B				Ĭ							Ī	T	T	٦	7							П	П						•
0	5		1	Γ	T		T	T	1	1			1		1	7	1	٦		7	1	1	1							1				П		1	1	1	7	7	1						П	П						•
	٦.		1-	Т		Т		7	_	1	\neg		\neg		7	_		-1	\neg	_	-	1	7		\neg	_			_	\neg	_	-				\neg	-	_	_	7	╛	_	\neg		1	$\overline{}$		\Box	$\overline{}$				\neg	•

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

IJ	31	۸ľ																						
Dat	e _																							
Pro	grai	m.	_			_							_											
Pro	gra	mm	er .	_																				
Γ		_	Γ	Г		_		_				Г		Γ	Г	ţ		Г	_			_	R	ecc
												ĺ				Indica					1			
1	_ine		Form Type			Fi	ler	am	ie			Cocerosco	Sedneure	Number (1-N)	Option (O)	Record Identifying	:	ľ	Posi	tior		Not (N)	C/Z/D	Character
3	4	5	6	7	8	9	10	11	12	13	14	1	16	14	18	9	20	21	22	23	24	25	26	27
0	1		I					Ĺ			/			\prod		\								
0	2		I				Γ			/							1							
0	3		1						[1)			(2)		. 1	(3)					
0	4		I								Ī			Γ										
Γ.	Γ.	г	_	_	Г	г	_	_	_	_	_	_	~	-	_			_	T~		_		_	

- Occiones 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		2	3	4	5	6	1	I		9]	0	11	12	Р	rir	t l	l6			19	2	0 2	1 2	2 2	3 2	4 2	5 2	6 2	27	28	9	30 3	SI];	32 3	3	34 3	5	36 3	37	38	[61	40	41 4	42]	43 4	14	15 P
	L	_						_	_					Т	ier	1						_									_			Ļ							_						T
Punch	CODE=M		T //		-		R						D	E			E			т	، -	0	^	· /							ı	UI Ca		ij			0				ı	0	N		riT FR	1	
Program		2	3	4	5	6	7	8	I	I	0	ij	12	13	14	15	16	17	18	19	2	0 2	1 2	2 2	3 2	4 2	5 2	6	27	28	29	30 3	31 3	32 3	3	34 3	5	36	37	38	39	40	41	42	43	44	15
Control Card											I	Ţ																																			_

Card Name ISSUE 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 4
Print			_									-		_																															
	<u></u>												Р	rin	t L	.in	e 1															- 1	<u>L</u> .												Pı
		_						_					T	ier	1													_										_							T
	1		17	E	-/	И		Q	UF	ł٨	TI	тү																					1 [
Punch	SODE	٨				EP	र		S	ol	-£	TY																																	
Program	-	2	3	4	5	6	7	8	9	Ιο	III	12	13	14	15	16 [17	IR	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
Control Card	Ė	٦	·		Ů	ľ	Ė	Ť	Ť			-	10	1	Ĭ		Ï		10	٤٥	-		_	LT	-	20	-		-	50	91	J.	~	31	٦	"	31	30	00	10	11		13	1	

Card Name RECEIPT RECORD

Print	Ī	2	3	4 :	5	6 1	1	8	9	10	11]	12 1	3 1	4 1	5 16	17	18	19	20	21	22	2 23	2	4 25	26	3 27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
	\vdash						_	_	_		_		Pr	int	Lit	ne '	1	-	_		_		_		_		_	_		_		 	_	_	_		_	_		_		_		
							_						Ti	er 1																		1						_						
Punch	CODE = R	^	T Vui	E,	M 81	ER	2 1	QU RE	VA EC	N El	TI'	ΓΥ D															-									<u> </u>								
Program	1	2	3	1 5	5 6	6 7		B !	9	10	11	12 I	3 14	1 15	16	17	18	19	20	21	22	2 23	2	4 25	26	6 27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Control Card						ĺ					1																																	

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:

ITEM NO	DESCRIPTION	QUANTITY ON HAND	QUANTITY ON ORDER	TRANSACTION QUANTITY
411116	B500 TWIN SOCKET BLUE ISSUE RECEIPT	458	500	50 500
		930**	0**	
411122	B506 SOCKET ADAPT BRN ISSUE ISSUE ISSUE	325		20 38 10
		257**		
411173	C151C SIL SWITCH IVORY RECEIPT	50	150	150
		200**	0**	
411254	A210 PULL CORD GOLD ISSUE ISSUE	62	75	16 30
		16**	75**	

ARACTERS PER INCH, 6 L		GE	Fold 1	back at detted line.		
· ————————————————————————————————————	 					
123456789	1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2	3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 2 3 4 5 6 7 8 9 0 1 2 3 4	4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	555568666666	67-777777777778	8 6 8 8 8 8 8 9 9 9 1
1 2			30,703012300			7234367690128
3 4						
5 H 6		STOCK	STATUS R	EPORT		
7 8						
H 9 H	ITEM WO DESCRI	PTION	QUANTITY OM HAMP	QUANTITY ON ORDER	TRAWSACTI	ON
D 12	XXXXXXX XXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXAXXX	XX ₂ XXX		
D 13 D 14	I I I I I I I I I I I I I I I I I I I					
15 T 16			XXX,XXX**	XXXI,XXXX**		
17			ON HAND	ON ORDER)		
19	Single space detail	/ lines				
21 22	Double space between	en detail a	nd total 1	ines.		
23 24	Print handings and	208511 10000	of the re	part		
25 26	Use asterists to s		() () () () () () () () () ()	nd Macil On Ora	len totals	
27	MAE MAIN MAIN	7,10,14 5,26 1,46	4 17 74 7 A A A A A A A A A A A A A A A A A			
	JOB SPECIFICA	TIONS				
	JOB SPECIFICA	TIONS				
BM	<u>.</u>	International Business M	achines Corporation		·	Form X21-9092
-	RPG CONTROL	. CARD AND FILE	DESCRIPTION		1 2	Printed in U.S.A. 75 76 77 78 79 80
Stock S	status [Punching Graphic Instruction Punch		Pag	Program Identificati	STAUST
rogrammer La Doni	na Hoffmann		<u> </u>		· · · · · · · · · · · · · · · · · · ·	
	Iwo	files are used for , the output file				
	E sterning	i, the output me ciated with any de				
Line Core Size to Size to Size to Size to Size to Size to Size to Size to Size Size Size Size Size Size Size Size		nds on the device			·	
Intromolia Introteu	TO PART I THE LETTER	File Description	Specifications			
TI	File Type M	ode of Processing	Specifications		<u> </u>	File Addition/Unordered
	File Designation	Length of Key Field or of Record Address Field		W/3	Extent Exit for DAM	Number of Tracks for Cylinder Overflow
Line	End of File Sequence	Record Address Type Type of File	Position	Symbolic Kabel		Number of Extents Tape
Line Filename	File Format	Organization or Additional Area	Device	Device g	Core Index	Rewind File
Typ d.	DOJOJOJU Block Record E/J	C Overflow Indicator C S L C Key Field Starting	TO THE STATE OF TH		entinuation Lines	Condition U1-U8
3 4 5 6 7 8 9 10 11 12		9 30 31 32 33 34 35 36 37 38 3	9 40 41 42 43 44 45 46 4	K Opti	on Entry 57 58 59 60 61 62 63 64 65 6	6 67 68 69 70 71 72 73 74
3 FOUTPUT	1/P		PRIMTER			++++++
	- - - - - - - - - - - - - - - - - - -	 	1 1 1 1 1 1 - 1	┤┤┤┤┤┤	 	- - - - -

IBM	International Business Machines Corporation	Form X21-9094 Printed in U.S.A.
Date 1/10/71 Program Stock Status Programmer La Donna Hoffman	RPG INPUT SPECIFICATIONS Punching Graphic	Page Program 75 76 77 78 79 80 STOKST
	H Identification Codes Field Location ≥	Field Indicators
Sequen Sequen Option Record C/Z/D Charac	A Position S Character Select T P Packed B = Binary Positions Openinal Positions	Pield Name Matching Fields or Chaining Fields or C
0 1 1 NPUT 011 1 10 1 1 CM	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
	2 7	ITEMNOL1
0 3 1	8 29	DESC
0 4 1		ONHAND
0 5 1	40 440	ONORD
0 6 1 Ø2MOZØ 1 CI		
0 7 I	2 7 8 12¢	ITEMNOL1
08 1	3 120	ISSUES
0 9 I		
10 1	2 7 8 12¢	ITEMNOL1 RECEET
	1120	RECEET

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.

														_									_													_			_												
IBY	Ţ																			terna																													rm X2 nted in		
		. ,													RF	PG.	1	CA	٩L	.Cl	JL	Α.	TIC	NC	ıs	PE	CI	FI	C	ΑT	10	N	S							1 :	2					75	5 76	77	78	79	80
Date _																Pi	unch	ing		Gra	phi	С	Γ	T	T		Γ	Τ	7	-	Г	7						Pa	ge /	3	2			gran		6	<u>.</u>	Ы	K	S	T
Program	<u>,S</u>	to	<u>.</u> k_	St	a	+(JS					_				In	nstru	ctio	n	Pui	nch			T	7		Г	\top				1							Ā	_			Ide	ntiti	ication	' =	1	$ldsymbol{\succeq}$	الث	_	<u> </u>
Program	nmer_	La	Do	n	na	١,	H_{c}	24	£	M	ai	יוני	1																			_																	~		
Г	Т	I 7		Indi			_	Г			_		_	_	Τ	_			Γ		_	_					Т					Τ		Т	Т	_		sulti		٦		_							_	_	-
				mai	cato	15																					1							l				dicat thme		\exists											
		SR)													ľ																	Ę	ield	ě	E	Plu	_	Minus	_	ro.											
Line	ē,	E E	,	And		Ind 		Ì		Fa	actor	r 1			0)pera	atio	n			1	Fac	tor	2				Resu	ult	Fiel	ld		engt	h E	ust		h I	Low	Eq	ual					Con	nme	nts				
	Ę	Control (L0-L9,	Į.	╁	_	-		ł							-																			i i	alf Ad	1>	ı	1< 2 ooku	Ľ												
	l.	8 =	Ş S	Not		Š		l							l												ļ					l		ľ	Ť	Tat		Fact	_) is											
3 4	5 6	7 8	9 10 1	1 12	13 14	15	16 17	18	19 20	21	22 23	3 24	25 2	6 27	28	29 30	3 31	32	33	34 3	35_3	6 3	38	39	40 4	1 42	43	44	45	46 4	7 48	49	50 !	1 5:	53			Low 56 57			60 6	1 62	63 (64 6	5 66	67 6	8 69	70 7	71 72	2 73	74
0 1	С		29						S				•		A	D1	2		7	0	T	I S	U				Τ	0	T	15	5 0	L		50				I	\mathbb{L}		F	ıN	D	7	ГО	TA	1 L	K	So		D
0 2	С	Ш	39			Ц		R	EC	Ε	E1	Ш			A	DI)		7	0	T 3	RI	C	Ц	\perp	1	7	0	T	RI	EC	L		5 9	<u> </u>	Ц			L		F	1/	D	- 7	ГО	TA	\L		RE	c	D
0 3	С	41	Ш	Ш	1	Ш	Ш	p	NH	Α	NI			╧	S	UE	3	L	T	0	T	/ S	U	Ц		_	N	ε	W	0/	V H	L	Ц	5 9	L	Ц	1	\perp	L	Ц	FI	4	D	1	YE	W	T	07	ΓA	L	L
0 4	С	41		Ш	_	Ц		W	ΕV	10	NE	Ц		4	Α	DD)	L	T	0	rk	ŞΕ	C		4	1	N	Ε	W	01	IH	L	Ш	1,	L	Ц	1	_	L		4	O	N	4/	4N	D	1	Ц	\perp	Ľ	L
0 5	С	L 1	-1-1-	11	4	Ц	4	0	NC	R	D _	\sqcup	4	1	S	UE	3	Ц	T	0	r ji	3/	C		Ц	4.	W	Ε	W	01	10	1		4	1	Ц	4	1	퇶	Ц	F	IN	D	-1	NE	W.	o)	N	OR	D	R
0 6 0 7 0 8 0 9 1 C	Ė			_						_	_		_				_								_				_			_																_	Ť	.	L
0 7		To	upe	dat	e t	he	αι	ıaı	nti1	tv ·	on	ha	ind	d a	ınc	l o	n ,	or	de	er.	to	ota	al i	nu	ım	be	r	iss	ue	ed	(T	-C	T	รเ	J)	ar	١d	to	ta	ıl ı	าน	mi	ber	-						-	L
0 8	1		ceive				•			•																																								-	Ļ
0 9	1		ius,																														-											if۱	/ind	a				i -	├
11			dica																			•																								_				i -	╀
 			Icula						_																																									I -	╁
			icuia id T																																											d				l -	╁
													_																				a l	LIC	,,,	(11	116	= (12	, 13	5 C	Oi	ıuı	uc	ле	u				l -	╁
1 !!		D	' 30,	, tn	e r	ec	orc	1 1	aer	ודו	ГУП	ng	ın	aı	ca	tor	a	SSI	ıgı	ne	ונו	το	τr	ıe	re	се	ıp	τr	ec	COI	ra.	•																		i-	H
		_		_									_																																					-	t
	1		ılcul					•			•		•																	-																			1	ľ	T
		af	ter a	ill t	rar	ารส	cti	or	ı re)CC	rd	s f	or	th	ie i	te	m	h	av	e l	эе	er	р	ro	ce	SS	ed	١.	TI	he	y a	ar	ес	on	di	tic	n	ed	b	У	L1	V	vhi	ch	1				-	1-	T
		is	set o	n o	wh	en	a	ne	w i	te	m I	nu	ml	oe:	r is	; re	ac	d.																															1	1-	T
	_							_		_	_				_									_										_														_	J	1-	T

IBI	٧Į				_											D	PG			_					lusine												c										-					_				U.S.	
Date _ Progra Progra	m _	<u>S</u>	to	c	Ł	S	to	2+	US H	- L	? e			_		ĸı	ru			nchir	ng	G	iraph	nic	=0 	HIV	/IA	\	21						ııc	אוע	13				Pa	ige	ø					rogr lent			7	5 70			_	79 S 7	
Progra	mme	r _•	-4	-	_		14	П	(F)			kip	 			out	Ind	icat	ors															_							Edi	it C	od	es					_	_	-	1	-	Τ			٦
Line	,			File	nan	ne		T/E)	/Fetch Overfi	Γ		T	1		A	nd		And	-		Fi	eld	Nar	me		(B)	_	nd osito		- Binary			[``	nma Yes	s	Zero	Ba o Pr Ye No	int s		_	Sig 1 2		CR A B	\dagger	- J K	×		Date	s Sig	n			1	Sign	rling n ition	- 1
		Form Type						Type (H/D/	Stacker Select Before	After	Before	After		Not		Not		Not							Edit Codes	Blank After	ir O	utpu lecor	ıt d	P = Packed/B				1	No No			Ye No	s Cor		ent	3 4 or		c D it V	/ord	ь м d	z			opre]					
3 4 0 1 0 2			ď					H			19 20 Ø 6		22 2		25 P	26	27 2	18 29	30	31 3	12 33	3 34	35	36 3	7 38	39	40 4	1 42	43	44	45 4	16 4	7 45	8 4	9 50	51	52	53	54	55 !	56 5	57 5	8 59	9 60	61	62	63	64	35 6	6 6	7 68	3 69	70	71	72	73	74
0 3		0						Н		1			1		P		1				-			1	E		1	6	Ø		•	57	rc) (: K	-		S	7	Ą	7	יט	5		R	ε	P	0	R.	7	'	+	1	Ē			
0 5		0		+	+		0	R	\downarrow				\downarrow	0	V		$\frac{1}{4}$	1		1	+	F		+	-		+	3	6			11	Γε	^	4	N	0				1	DE							1		٧,	+	+	-	L	H	-
0 7 0 8 0 9		0		+	+		+	Н		2			#	1	P		1	+	H	+	+	-		+	-		+	8	62		•	8 (T 1	RA	\ /	V S	A	C	T	,	0	N	′	Q U	A	N	_	/	1	Y	+	+	+	+	+			1
1 0	+	0		1			0	R	1						V													6	6		• () ^	1	Н	A	N	D				1	C) A	1	0	R	D	E	R	•	1	-	T	I			
1 2 1 3 1 4		0		+	+		+	D	+	1			+	1	¢		+	-			7	- 5	M	A/	\ \		1				-	9 (96	3/	V 7	/	T	4	,	+	7	+	+			-		-	+	+		+	+	F			-
1 5 0 1		0					+						1	+						1	DE DA	S	C A	N1	2			5	942		+		+							_	1	<u>†</u>		_						1	1	_				-9090 U.S.A	
0 2 0 3 0 4		0		_	-		+	D	+	1		-	+	2	ø		+	+		- (NC	0	R	0	2		+		4	-	•	1 6	= =	.,	JΕ	,		-	+	-	Pa	ge (ø					ogra	am ficat	tion	_					79 5 7	_
0 5		0		†	Ė		+	D	+	1	+		†	3	Ø		#	†		1	S	S	U	ES	2				8		†	<u> </u>				F		1	1	- +	+	+	+	_ 	H	<u> </u>	Н	+	+	+	+	+	+	+	_ -	 	_
0 7	-	0		1										$\frac{1}{1}$						1	RE	c	E	E 1	2			7			1 1	? (EC	26	E 1	P	7	,			1	1							I	1		#	I	F			
0 9 1 0	1	0		+	-		+	7	1	Z			+	L	1		+	+		^	ΙE	W	01	V/	/2	В	+	5	2		۱ .	(+)	4,	+	+	_		+		+	+	+	+	+					+	+	+	+	+	-		H	1
1 2	-	0												ļ				F		٨	ΙE	W	0	Nc	2	В		6					<i>()</i>		1						1	1	+	F					#	#	+	Ŧ	+				
1 5			He	ad	in	g l	ine	es a	re	tc	pr	in	t c	on	ev	/er	У	ра	ge		Th	iey	/ a	re	cc	one	dit	tio	ne	ed	b	У	1P) a	ane	d (٥١	V	us	sec	d i	'n	aı	n (ЭF	3	re	la	tic	on:	sh	ip	 >.		-		-
											are ior																																									d			5. Jan 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
			rec	or	d.	T	he	se	со	nc	d de nts	eta	il	lin	е	pr	in [.]	ts	wł	nei	n a	ın	iss	sue	re	ec	or	d i	s I	ea	ad	, s	0	it	is	С	or	١d												51					200	1	1
			Th Th	e F	Pri e v	nt vo	er rds	Sp:	a∕ci e r	ทg าด	ı Cl t fi	hai elc	rt : ds	sho so	ow th	vs ney	th	at are	th e	ie v	wo	oro ed	ds as	IS	SU	JE sta	aı ınt	nd ts i	R in	E th	CE ie	EII ap	PT op	Γa	are	e t	0	pı																	The second of the second		
			ast	eri	sk	s i	nd	ica	tin	ıg	tot	als	a	re	al	so	er	nte	ere	ed	as	CC	ons	sta	nt	s i	n	th	e 1	to1	tal	Hi	ine	e.																					i	ı	

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

on control level

indicators.

Figure 15. Program Cycle Operations for Resulting Indicators

indicators.

RPG II Specifications

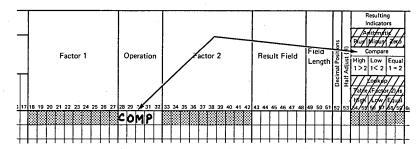
The type of operation used to check the result field depends on the type of result being checked. If you want to determine whether the result field is larger, smaller, or equal to a certain number, you must use a compare (COMP) operation. If you want to determine if the result field is plus, minus, or 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
TDIVE		RPG CA	ALCULATION SE			Printed in U.S.A.
Date			Graphic		1 2	75 76 77 78 79 80 Program
Program		Punching Instruction			Page	Identification
Programmer						
The state of the s			1		Resulting	
Indica	tors				Indicators Arithmetic	
		į			Plus Minus Zero	
Form Type Control Level (L0.L9, LR, SR) Not Page A Not	And Factor 1	Operation	Factor 2	Result Field Length	Tis Compare High Low Equal	Comments
	1				TE D 1>2 1< 2 1 = 2	
Con (LO Not	Not	İ		1	Table (Factor 2) is	
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 5	High Low Equal 1 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 2				<u> </u>	2111111111	
0 4 TC						
TDM			International Business Machines	Corporation		Form X21-9090
IBM		RPG OU		SPECIFICATION	NS	Printed in U.S.A.
Date				TITTE TO	1 2	75 76 77 78 79 80 Program
		Punching Instruction			Page	Identification
		<u> </u>	Toncii			
Programmer						
	Space Skip O	Output Indicators			Edit Codes	
					Zero Balances No Sign CR -	X = Remove
Line Filename	E)		Field Name	Yes	to Print No Sign Ch Yes 1 A J	Plus Sign Sterling Y = Date Sign Position
, be	Type (H/D/T/E) Stacker Select/Fetch C Before After Before After After After Not	And And	티 Po	Yes No No No No No	No 2 B K Yes 3 C L	Z = Zero
Ę	re Sele Before After Before Before After	Not Not	Edit Codes Blank After B O D	put Sold No	No 4 D M	Suppress
	Stack		111	[u]	Constant or Edit Word	63 64 65 66 67 68 69 70 71 72 73 74
0 1 0 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 5	1 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 0	3				 	╎┊╎╎╎╎╎╎╎ ┦
0 3 0	 	 			 - - - - - - - - - - - - - - - - - - -	
1		 	 			

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:

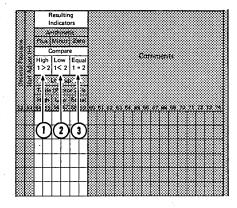


- Factor 1 (either a field name or constant)
- Factor 2 (either a field name or constant)
- Resulting indicators

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			, ,	And	Not		Ar	Not	1	177	18	. 19	20		ac			. 25	. 21	6 2	7 2	O _f	er 3	atio	on 1 3:	2 3	13 ;	34	35	F	ac	to:			0 4	.1 4			esu 14 4			-			ngtl		Decimal Pos	Half Adjust (H)	Plus High 1 >	Arit M Cor Lo Lo	mpa ow 2 oku oku ow 6 5	ors etic Z re L or 2 D or 2	qua = 2		0 61	1 6:	2 €	63 6 4					ent		70	71	72	73	74
0 1	С		Т	П	T	T	T	٦		Γ	Γ	V	A	J	T	Γ	Γ	Γ	Γ	Τ	Т	1	CZ	A	4 6	7	F	ī.	0	7	Δ	L	Ι	Τ	Τ	Т	T	T	T	Т	T	T	T	T	Т	Т	T	Т	1	LÍ	12	12	3	13	3	Т	Т	Τ	Τ	Τ	T	Т	T	Т	П	П	П	П	7	٦
0 2	С	П	1	1	1	1	1	1		T	T	Г	C	_	_	L	İ	t	T	Ť	1	7	nı	,	-	r	Ť	4	•	r_	r	Ť	1	Ť	Ť	†	†	7	:	٥k	•	и	†	1	7	-	1	21	1	Ī	T	T		Ī	1	T	T	Ť	1	Ť	Ť	7	7		T	П	П	ıT	T	1
0 3	С	\vdash	T	2	_	7	7	1	_	T	t	П	·c	-	1	1	т	t	T	t	T	V	1	1		r	Ť	-	•	6	+-	T	T	T	Ť	Ť	Ť	7	Ţ) /	4		1	Ť	Ť	T	ľ	1	44-	†	T	t	t	t	t	T	t	t	\dagger	†	†	†	7			М	П	П	1	1
0 4	С	-	╁	3		7	+	1	_	t	+-	+~	c		+-	+	+	t	t	†	†	1	ij,	1	, ,	7	ť			7		t	t	+	t	†	7	1		2/4	4	7	†	†	7	+	†	Ţ	ان	†	†	†	t	t	†	\dagger	†	+	十	†	十	†	寸	\forall	T	Н	Н	П	7	┪
0 5	c	H	t		1	7	†	1		+	t	ľ	+	+	ľ	-	+	+	t	+	+	f	1	1	7	+	ť	Ŧ	•	•	t	1	†	t	+	+	+	Ť	1	1	7	1	+	+	+	+	+	†	4	+	$^{+}$	+	+	+	1	+	+	+	\pm	†	+	†	7	7	\exists	Н	Н	\dashv	\forall	1
0 6 0 7 0 8 0 9 1 0 1 1 1 1 2 1 3	0 0 0 0 0 0 0												1 T N	on 1 0	is T.	en Se A	ts Et L,	o , i	of n no lu	t a ik	he ne ca	e · d at	i, fie th or T	e e 2	0	T(pe	o er	T/ at	A tie	or o	n n	l ir a	f i l	M in d	A e th	X C	i 2	s is	gr s c	ea do rat	n tic	er e. on	ti i	ha If n	an M li	T A ne	X	T(s i	L le:	, i ss do	nd th	dio na e.	ca n		or														
1 4	c	-	+	H	+	+	+	-	_	+	+	1	1	ł	F	Ť	+	t	-	+	+	+	+	+	Ť	-	1	+	-	H	-	-	+	+-	+	+	+	+	+	+	+	+	Ŧ	+	1	7	1	+	+	+	+	+	+	+	+	+	+	+	$\overline{+}$	<u>-</u>	+	4	4	\dashv	-	\vdash	H	Н	4	4

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							s		l																						1						Ţ					-	_	dica			1										1			١
	Form T		Not 9 10		Not		nd Not	16 1	118	19	20 :	Fa	cto	r 1	25	26	27	O ₁	oera	tior	n 32	33	: 34	35	Fa	act	or 2	2	40	41		Re				id		ielo eng	th	Decimal	ξ	Hig 1>	Co jh 2 2 ble	Minu Low 1< :	E 1 1 Lup	qua = 2 2) is		61	62	63	SA (-	om				. 71	77	73	74
0 1	c	ŤŤ		П	\top	T		Ħ	7	b	T	جآ		:	Ī	Ť		C	M	P	П	5	Ø	d	d		Ø	d	Ī	Ï	T	Ť	Ť	Ť	Ť	Τ	Ť	T	Ī	-	٣	11	-31	1 2	-	1	7	ĬΪ	٦	Ň	T	T	Ť	T	T	Ť	Ť	Ĥ	Ï	Ή
0 2	С		1	5	Ť	t	H	Ħ	7	'n	7	Ś	75		t	Ħ	Н	MI	11	7	Н	-	ø	-	Ŧ	Ť		7	7	\forall	7)/	<	5 (,	-	t	╁	7	2	H	7			1	1	1	H	┢	H	1	1	†	\dagger	\dagger	t	t	t		П
0 3	С		1	d	+	+		П	7	6	7	5	د ر		T	\vdash	Н	MI) L	r	Н	_	Ø	3			7	7	1	1	7	וכ	-1-	:/	7	7	t	T	1	Γ	н	7	1	\dagger	T	T	†	H	-	H	7	7	1	†	Ť	t	t	T	T	П
0 4	c		1	7	\dagger	t	\vdash	-	۲	1	4	7	_	+	+	\vdash	H		+	1		ř		7		H	_	+	-		Ť	7	_	4	4	+	+	+-		H	••	-	- }	+	+	+	+	Н		Н	7	1	+	†	†	+	t	h		\dashv
0 5	c	П	\top	H	†	†																																													7	1	+	\dagger	†	†	t	П		Ⅎ
0 6	c		1	H	\dagger	t	l		T١	nes	е	O	рe	ra	tic	on	S	are	u	se	d	f	or	fi	'n	di	ng	j t	th	e	an	no	u	n	t (of	d	isc	0	ur	٦t	tc) (yiç	е					ı	7	1	†	†	\dagger	t		П	T	Ⅎ
0 7	c		T	Ħ	\dagger	†			cι	ıst	or	ne	rs		lf	th	ie	CL	st	on	ne	er	pı	ur	cl	na	se	s	go	oo	ds	s v	vc	or	th	1 \$	35	00	0	.0	0	or	rr	nc	re	,				ı	7	1	+	╁	\dagger	\dagger	t	T	┢	Н
		\dashv	+	H	+	†			he	e re	ec	ei	/es	a	3	3%	d	isc	:oı	้าทา	t.	b	out	i	f I	hε	e 12	u	ırc	ch	as	es	a	0	oc	ds	w	or	th	ı I	es	s 1	th	ar	1					ı	7	7	+	+	+	+	t		H	H
0 9	c	\Box	\top	H	+	†	ĺ											ive			•						•						_																	ı	+	1	\dagger	†	$^{+}$	t	t		H	
1 0	c	\top	_	П	T	Ť												О			•	•																					•							ı	7	7	7	\dagger	†	\dagger	t	H	<u> </u>	
1 1	c		T	Ħ	T	†	l																																											ı	7	1	\dagger	\dagger	†	†	t	T	r	П
1 2	С			П	\top	Ť												s i																													•			ı	7	1	1	Ť	\dagger	Ť	T		T	П
1 3	С			Ħ	1	t												SL:							-						_												-	ın	d١	-				1	7	7	†	†	Ť	t	t	M	f	T
1 4	С			П	T	†												an			е	O	pe	ra	ıti	OI	n i	n	li	ine) ()3	i	s į	pε	erf	O	rm	e	d	(3	%									7	1	\dagger	Ť	t	\dagger	t	T	T	1
1 5	С			H	1	Ť			di	SC	ou	'n	t is	S C	al	Cι	ıla	ite	d)																																7	7	+	t	Ť	t	t	1	┢	Н
$\top \Box \Box$	С			П	T	T	L			_																																				_					T	7	1	T	Ť	t	T	1	Г	┨
$\Box\Box$	С	П		П	T	Τ			1	Ī	1	i	ī	1		1	1 1	Ī	ı	1				1				Ī	ı	Ī	١	Ī	Ī	Ī	1	1	ı	ı			1	1	Ī	-	1	Ī	l		1		7	7	7	†	T	Ť	t		1	H

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:

A numeric constant can be any combination of digits 0-9. Decimal points and signs can also be included. The maximum length of a numeric constant is 10 characters, including sign and decimal point. An alphameric constant can be any combination of characters. Blanks are also valid. The maximum length of an alphameric constant is 8 characters.

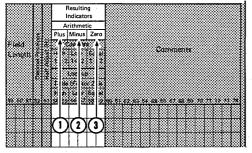
When you use the COMP operation code, remember to always compare two numeric fields or constants or two alphameric fields or constants. You cannot compare a numeric field or constant to an alphameric field or constant.

Using An Arithmetic Operation

Numeric constants must not be

enclosed in single quotes (').

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).

Alphameric constants must be

enclosed in single quotes (').

- 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.
Date	RPG CALCULATION SPECIFICATIONS	1 2 75 76 77 78 79 80
Program	Punching Graphic Page Instruction Punch	Program Identification
Programmer		
		Comments -2 -2 -2 -3 -2 -3 -3 -3
IBM	International Business Machines Corporation RPG CALCULATION SPECIFICATIONS	Form X21-8093 Printed in U.S.A.
Date Program Programmer	Punching Graphic Punch Punch Punch	1 2 75 76 77 78 79 80 Program Identification
Lindicators Control Level Not Not Not Not Not Not Not Not Not Not	Arithmetic Pius Minus 2 Result Field Length Fiel	Zero qual = 2
3 4 5 6 7 8 9 10 11 12 13 14 15 15 17 18 19 20 21 0 1 C FIEL		qual 8 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 2 C 10 FIEL 0 3 C 20 FIEL	DC SUB 59 FIELDC	

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

	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					
	Pri	nt Line 1		Print Line	2
	Tie	r 1	-	Tier 2	
Punch	NUMBER	ESCRIPITON	COST ON HAND		
Program	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
Control Card					

	I	ιŢ	2	3	4	5	Te	1	I	8	9	10	II	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	3	32	3	3 34	4 3	5 3	6 3	7 3	3	4(4	42	43	44	45	46	47	48 4	49 5	0 5	1 5	2 5	3 5	4 5	5 5	6 5	58	59	60
Print	L																																																												
																Pri	nt	Lir	ne '															-	I				_								F	rin	t L	ine	2										
	Ι					_									7	۲ie	r 1																		ī												٦	Tier	2										1		
Punch		CODE=/	i N	T	E	NE	1 8 <i>E</i>	R		QI S	VA O	N	TI D	Τ))						-																																			-						
Program	L	П	2	3	4	5	6	17	٤	Ц	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	3	32	33	3 34	4 3	5 3	6 3	7 3	8 3	9 4) 4	42	43	44	45	46	47	48 4	19 3	50 5	i [2 5	3 5	4 5	5 5	6 5	1 58	59	60
Control Card							1																												Γ																										

Card Name RECEIPT RECORD

Print	Ī	2	3	4	5	6	7	8	9 1	0 1	11 12	2 I	5 14	1 15	16	17	18	9 2	2	22	23	24	25	26	27	28 2	9 3	0 3	32	33	34	35	36	37 3	8 3	9 40) 41	42	43	44 4	5 46	6 47	48	49 5	0 51	52	53	54	55 5	6 5	7 58	8 59	60
	-		_	_	_				_		_		Pri	nt l	Lin	e 1		_			_	_								<u>i</u>					_						Pr	int l	Line	2				_	_				
		_					_						Tie	er 1	_															1											Ti	er 2											_
Punch	CODE = R	~	ד וטי	E 11.	EN BI	1 = F	2	QU RE	AI C	N1 El	IT. VE	4																																									
Program	I	2	3	4	5	6	7	8)	0 1	1 12	1	114	15	16	17	18	9 2	2	22	2 23	24	25	26	27	28 2	29 3	0 3	32	33	34	35	36	37 3	8 3	9 4	0 4	42	43	44 4	5 4	6 47	48	49 5	50 51	52	53	54	55 5	6 5	7 58	8 59	60
Control Card																																																					

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 ST	ATUS REPORT			
ITEM NO	DESCRIPTION	QUANTITY ON HAND	QUANTITY ON ORDER	TRANSACTION QUANTITY	MIN. BAL.	MAX. BAL.
411116	B500 TWIN SOCKET BLUE ISSUE RECEIPT	458	500	50 500	800	1600
		950**	0**			
411122	B506 SDCKET ADAPT BRN ISSUE ISSUE ISSUE	325		20 38 10	300	800
		257**			UNDER	
411173	C151C SIL SWITCH IVORY RECEIPT	50	150	150	100	200
		200**	0**			
411254	A210 PULL CORD GOLD ISSUE ISSUE	62	75	16 30	80	165
		16**	75**			

This Printer Spacing Chart shows the format of the report:

																																																													_												
			П	П	П		Π	Τ,	Ι,	Л	Π,	١,	Ţ	Τ,	Ţ		,	Τ,	Ι,	П	٦.	J.			.].	Τ.	Ι.			Ι.	Τ.	[١.				Ī.		L		Ι.				_				1		ارا	٦,	Ι.	J				Ţ.	Ţ		1.	Ţ			Ι.	Ι.	Г	IJ	Ţ	Ţ	
			2	5 4	6	7 8	9	όĮi	2	3	4 5	6	żΙ	8 9	ΙÓ	f	2 3	4	5	6	ᆌ	5 5	õ	1	2	3 4	5	6	7	3	ijō	Ιĩ	2	3	13	6	7	8	13	ĩ	2	3	4	6	7	8	3	3	2	3	4	6	7	8 9	ó	[6]	2 3	4	5	5 7	8	9	ö	2	3	4	5	6 7	8	9	أأة	1 2	3
	\Box	٦	TΤ	П	П	П	П	Ţ	П	П	Т	П	П	Т	Т	П	Т	Τ	Т	П	Т	Τ	П	П	Т	Τ	Τ		٦	T	Т	Γ	П	Т	Т	Π	П	П	Т	П	П	Т	Ţ	Τ	Т	П	T	Т	П	П	T	Т	П	Т	Τ	П	Τ	П	Т	Т	Γ	П	Т	Т	П	П	Т	T	П	Π	Τ	Τ	Γ
_		2	П	П	1	Т	П	T	Т	П	Т	П	Т	T	T	П	T	Τ	Т	П	7	T	П	٦	T	7	T	П	7	T	Т	Γ	П	1	T	Т	П		Т	П	П	T	T	T	Т	П	T	T	П		1	1	П	T	Т	П	T		T	Τ	Г	П	Т	Т	П	П	Т	T	П	П	T	Τ	Γ
Ξ		3	П	П	П		П	T	П	П	Ι	П		Т	Т	П	Т	Т	Τ	П	T	T	П	٦	T	T	Τ		1	T	Τ	Γ	П	1		П	П	T	Т				1	T	Г			Τ	П		Ī	Т	П	Ţ	Т	П	T	П	1	T	Γ	П	T	T	П	П	Т	T	Г	П	Τ	Ι	
Ξ		4	П	П	П	T	П	T	П	П	T	П	Т	T	П	П	T	Т	Т	П	T	Т	П	٦	T	T	T	П	7	Ţ	Τ	Г	П	į	Т	П	П	٦	Т		П	7	T	T	Т	П	7	T	П	П	1	1	П	7	T	П	T	П	1	Τ	Т	П	Т	Т	П	П	Т	Т	Г	Π	T	Т	Γ
_		5	П	П	П		П	Т	П	П	T	П	П	T	Т	П	T	T	T	П	T	Т			1	T	Τ	П	1	T	T	Г	П	1	T	Г	П		Т	П	П	T	Т	Τ	Г	П	T	Ţ		П	T	Τ	П	T	T	П	T	Т		Τ	Т	П	T	T	Г	П	Т	Ţ	П	Π	7	Τ	Γ
Ξ	H	6	П	П	П		П	Ι	П		Ι			Ţ	Т	П	T	T				T	П	٦	T	T	Τ		7	5	7	0	C	K	T	S	7	A	TL	S		1	RE	= 1	0	R	7	Τ			T	T		1	Τ		I	Π		Τ	Γ	П	T	Τ		П	Т	T	Γ	\Box	Ι	Ι	Γ
		7	П	П	\prod		П	Ι						Ι		П	Т	Τ	Γ		T	T			Τ	T	T		Ī	Τ	Ι	Γ		Ţ				I	Γ									Ι		Ţ				\mathbf{I}	Γ	\Box	Τ			Τ	Ι	П	Ι	Ι		П	Ι	Τ	Γ	Π	Ι	Ι	Γ
Ξ		8	П	\prod	П		П	Τ		\Box	Ι			Ι	Т	П	T	T	Ι	П	T	I			Τ	Ι	Τ			T	Т	Г		Ţ	T	Π		T	Т		П	T			Г		T	Τ			İ	Τ	П	Ţ	Τ		Τ			Τ	Ι	П	Ι	Ι			Т	Ţ	Γ	Π	Ι	Ι	
	Н	9	П	174	Μ	N	10					Π	1	Dε	5	c	R	IP	1	7	a	V		٦	T	Τ	Γ		Q	J	W	7	1	T	7		П	7	V							П		7	R	A	V	A	c	۲V	0	M	Τ	П	T	T.	Ι	M	П	V.		П					Ι	Ι	
_	H H	10	П	П	П	I	П	Ι			I	П		Ι			I	Ι	Γ	П	I	Ι			Ι	Ι	Ι		o	V	H	A	N	D	Ι	Γ			۱		0	R	DE	F	1		\Box	G	U	A	M	Ū	7	Y	Γ	П	I	Γ	I	Ι	I	B	A	4.		П	J	3 A	L		I	\perp	L
	- 1	11	П	Ш	П		П			\square	Ţ	П		Ι			\perp	Ĺ	L		Ι	I			Ι	Ι	Ĺ		I	Ι	Ι			_[Ι				I				I	Ι		П		L			1	I		1	L	Ц	1			Ţ	1	Ц	\perp	\perp			╝	\perp	\prod	Ц	1	\perp	L
_	D	12	Lì	(X)	W	ΧX				X)	XΙX	Ø	X)	Œ	(X	X	ΧĎ	(X		X	XD	Ø	X	Χľ	XI)	Φ	1			D	(X	Į,	X	XD	1				I	X	X	١	X)	$\langle \rangle$		Ш		1			-	Ţ	Ш			BAL- XXXX XXXX UNDER OR OVER																	
	D	13	П	\prod			П	Ι		1	SS	U	E	I			I	I	Γ		Τ	Γ			I	I			I	Ι	Ι	Ľ		Ĭ	Ι				QUANTITY TRAMSACTION MIM. MAX. ON ORDER QUANTITY BALL. XX,XXX XXXX XXX,XXX XXX,XXX XXX,XXX XXX,XXX INDER OR OVER (WEW ONORDER) Vines. XY page of the report.																																		
_	2	14	П	П	Ш		П	Ι		R	EC	ε	11	P 7	1		I		L	Ц	1	Ţ				ľ	QUANTITY QUANTITY TRANSACTION MIN. MAX. ON HAND ON ORDER QUANTITY BAL. XXI,XXX XXI,XXX XXXXX XXXX XXXX XXXX X																																														
		15].[П	Ш	Ц	П				\perp					Ц	I	Ī		\prod		1	Ц	╝	1	Ţ	ľ	Ц		I	I	Ĺ	Ц	1	1				\perp	L		1	1	1		Ц		1			- [1	1	П	1	L	Ц	1	Ĺ	1	1	1	Ц	4		Ц	Ц	1	1	Ţ	Ц	1	1	L
_	T		П	Ш	П	Ц	П	\perp			\perp	П	Ц	I			I	Ţ	L	Ц	1				1	I	L					X	X	Χį	۲X		Ц		X	X	Х	,	<u>(</u>)	XX	(*	×		Ţ	Ц	Ц	1	1	Ц		L	Ц	1	L		и	١N	D	Eŀ	₹_	0	R	_(<u>)</u>	E	R	1	1	Ţ
_		17	Ш	Ш	ĿĿ	Ш	Ш	1		Ш	\perp	Ш		Ĺ		Ш		1	L	Ш		L			1	1						L		┙	L	L.	L		(1	L	Ш	┙	L	Ŀ		┙	l	Ш	┙	L	Ц		Ш		\perp	L	Ц	1	\perp	Ц	Ц	Ц	\perp	Ľ	Ц	1	⊥	L
_		18	Ш	Ш	Ш	Ш		1			┸	Ш		1	L	Ш				Ш		L			\perp				<u> </u>	٧.	H	1	W	<u>Q</u> .	<u>)</u>	L				6	24	16	7/	31	\mathcal{H}	14	ر!	<u>'L</u>	Ш		i	L	Ш	\perp		Ш	1			1	L	Ш	┙	⊥	L	Ш	Ш	⊥	L	Ш	⊥	⊥	L
		19	П	П		\prod	П	Ι			Ι			Ţ			\Box					L				I				I	Ι			Ţ	1		Ш		I	L		j	ī	Γ			\perp	L				1	П				I	Ц		Ι	L	Ц	1	1	Ц	Ц	Ц	I	L	Ц	⊥	\perp	L
		20	П	П			П										floor					Ι				Ι				Ţ	I	L	Ц	1	İ	L	П	_		L	Ц		I	I		П		L	Ц				П	1	L	Ц	1.	Ц	4	1	L	Ц	1	1	L	Ц	4	\downarrow	Ľ	Ц	4	\downarrow	L
_		21	П	П				Ι	Γ		1				Ι				Ι	П	1	Ι				I	L	L			I		Ц		1	L	Ш		Ι	L	Ш		\perp	I	L	Ц		L	П	I	1			\perp	L	Ц	L	Ľ	\perp	⊥	1	Ц	1	1	Ц	Ц	Ц	1	L	Ц	1	1	L
_		22	Ц	П	Ц	Ц	Ц	1		Ц	\perp	Ц			L	Ц	I	1	L	Ц		L	L		1	1	1	L	Ц	1	1		Ц	\perp	1	L	Ц	_	1	L	Ц	1	1	1	Ļ	Ц	4	1	Ц	1	\perp	\perp	Ц	┵	Ļ	Ц	1	L	1	1	\downarrow	Ц	4	1	Ц	Ц	4	1	Ц	Ц	4	1	Ļ
_		23	Ш	Ц	Ц	Ц.	Ц	1	L	Ľ	1	Ц	Ц	1	L	Ц	2	1	1	19	zΖ	e	Ц	\$	k	26	4	4	2	1	4	ϵ	27	te	ZL,	44	Ц		Ż	Ľ	15	9	≴	ŧ	L	Ц		┸	Ц	Ц	1	\perp	Ц	4	L	Ц	1	L	1	1	\perp	Ц	4	1	L	Ц	4	┶	Ц	Ц	4	1	L
_	:	24	Ш			Ц	Ш	1			1	Ш	Ш	1	L		┙	l	L	1		1			<u>'</u>	1	L	L		l	L	L	Ш		1.			Ц	\perp	L		1	⊥		L	Ш		L	Ц		\perp	1	Ш	┙	L	Ц		Ш		⊥	⊥	Ц	┙	⊥	L	Ц	4	┵	Ц	Ц	1	\perp	L
	[:	25	П	П		Ц	П	I		П	I		П	Ι	L		\mathcal{D}	b	U	b	le	4		3	ķ	26	2	6	•	Ι	b	E	1	u	Je	1	2	7	1	V	e	te	2	iЦ		9	Ż	20	#	ŀ	4	Q:	te		1	4	ik	2	2	4٠	J	Ц	Т	\perp	Ц	Ц	1	┙	Ш	Ц	1	\perp	L
_	:	26	Ш	Ш	Ш	Ц	Ц	\perp	Ш	Ц	1	Ш	Ц	1		Ц	1	1	1	Ц		L		Ц	1	1	L	Ĺ	\perp	1	L	L	Ц	1	1	L	Ц		1	L	Ц	1	1	1		Ц	_	1	Ц		1	1.	Ц	1	┸	Ц	1	Ш	4	1	l	Ш	1	ļ	Ц	Ц	4	1	L	Ц	1	1	L
_		27	Ш	Ш	Ш	Ц	Ц	1	L	Ц	⊥	Ľ	Ц	\perp	L		\mathcal{H}	r	1	1	1	L	h	9	a	Ł	4	14	2	1	Ł	L	0	n	1	€	U	e	r	V	Ц	À	2/2	1	þ	4		10	2-	ϵ	ŀ	*	И	⇞	1	1	<u>ع:</u>	4	ع د	2 7	φ.	†	4	\downarrow	Ц	Ц	4	1	Ц	Ц	1	1	L
_	$\overline{}$	28	Ц	Ц	Ц	Ц	Ц	1	L	Ц	4	Ц	Ц	1	1	Ц	\perp	1	L	Ц	1	1	Ц	Ц	ļ	1	L	Ц	_	1	L	L	Ц	1	1	L	Ц	Ц	Ŀ	L	Ц	1	\downarrow	1	L	Ц	4	,	11	Ц	1	\downarrow	Ц	4	L	Ц,	1	Ц	4	╧	1	Ц	4	┰	Ц	Ц	4	1	Ľ	Ц	4	1	L
_		29	Ш	4	Ц	Ц	Ц	1	L	Ц	4	Ц	Ц	1	L	Ц	4	\$ 6	2	19	:\$	t	ę	d	4	\$,	Ł	٤	1	φ	6	Z	1	ø	η	a	8	4	2	6	킴	4	e	4	40	2	1	40	14	14	1	a	И	₫	╀	Μ	90	¥	γ	40	44	4	坤	10	4	٠	4	1	Ľ	Н	4	1	L
_	-	30	Щ	Ш	Ц	Ц	Ц	1	L	Ц	1	Ц	Ц	1	L	Ц	4	1	L	Ц	1	\downarrow	L	Ц	1	\downarrow	Ļ	Ц	_	\perp	L	L	4	1	1	L	Ц	4	\downarrow	L	Ц	1	1	1	L	Ц	4	1	Ц	4	1	1	Ц	1	Ļ	Ц	4	Ц	4	1	¥	Ц	4	╀	Ц	Ц	4	1	¥	Н	1	1	Ľ
_	_	31	Ц	Ш	Ц	Ц	Ц	1	\perp	Ц	1	Ц	Ц	1	L	Ц	Ţ	1	L	Ц	1	1	Ц	Ц	4	1	L	Ц	1	1	1	L	Ц	1	1	Ц	Ц	1	l	Ц	Ц	1	1	1	L	Ц	1	1	Ц	1	1	Ļ	Ц	1	Ļ	Ц	1	Ц	1	\perp	Ŧ	Ц	4	1	Ц	Ц	4	4	Ц	Ц	4	1	Ľ
_		32	Ц	Ц	Ц	Ц	Ц	1	L	Ц	4	Ц	Ц	1	L	Ц	Ц	\perp	1	Ц	1	1	L	Ц	_	1	1	Ц	4	1	L	L	Ц	4	1	L	Ц	Ц	1	L	Ц	1	1	1	L	Ц	4	1	Ц	4	1	1	Ц	4	Ļ	Ц	1	L	4	1	1	H	4	4	Ц	Н	4	+	Ľ	Н	4	4	L
_		33	Ц	44	Ц	Ц	Ц	4	L	Ц	4	Ц	Ц	1	\perp	Ц	Ц	1	ļ	Ц	1	1	Ц	Ц	1	1	1	Ц	_	1	Ļ	Ļ	Ц	4	1	Ц	Ц	Ц	1	L	Ц	4	1	1	L	Ц	4	1	Ц	1	1	L	Ц	1	Ļ	Ц	1	Ц	4	4	\perp	Ц	4	T.	Ц	Ц	4	4	Ľ	Ц	4	1	L
_	-	34	11	44	Ц	Ц	Ц	4	Ļ	Ц	4	Ц	Ц	4	L	Ц	Ц	1	1	Ц	4	1	Ц	Ц	4	1	Ļ	Ц	4	4	Ļ	L	Ц	4	1	L	Ц	Ц	1	L	Ц	4	1	4	L	Ц	4	1	Н	4	1	1	Ц	4	Ļ	Ц	4	Ц	1	+	╀	Н	4	Ŧ	Ц	Н	4	+	₽	Н	4	+	Ľ
	_	35	4	44	Ш	Ц	Ц	4	Ļ	Ц	4	\perp	Ц	+	1	Ц	4	4	1	Ц	4	1	Ц	Ц	4	1	1	Ц	4	4	1	L	Ц	4	1	L	Ц	4	1	L	Ц	4	1	+	L	Ц	4	4	Н	4	4	1	Ц	4	1	1	4-	Н	4	+	+	H	4	+	Н	Н	+	+	₽	H	+	+	Ľ
_		36	Ц	Ш	Ц	Ц	Ц	1	L	Ц	1	Ц	Ц	1	L	Ц	Ц	1	1	Ц	_	\perp	Ц	Ц	1	1	\perp	Ц	4	1	L	L	Ц	1	1	L	Ц	╛	1	Ц	Ц	4	1	1	Ц	Ц	4	1	Ц	4	1	╀	Ц	4	L	Ц	1	Ц	1	1	¥	Ц	4	T,	Ц	Ц	4	4	Ш	Н	4	1	Ц
_		37	Ш	11	Ц	Ц	Ц	1	1	Ц	1	П	Ц	1	L	Ц	Ц	1	1	Ц		1		Ц	1	1	1	Ц	1	1	1	L	Ц	1	1	L	Ц	Ц	1	Ц	Ц	4	1	1	L	Ц	4	1	Ц	4	1	1	Ц	4	1	Ц	1	Ц	4	1	1	Ļi	1	1	Ц	Н	4	1	ш	H	1	1	Ľ
_		38	Ц	Ш	Ц	Ц	Ц	1	L	Ц		Ш	Ц	1	L	Ц	Ц	1	ļ	Ц	1	Ţ	\perp	Ц	4	1	Ţ.	Ц	4	1	L	L	Ц	4	1	Ц	Ц		ļ	Ц	Ц	4	1	1	L	Ц	4	1	Ц	4	4	1	Ц	1	Ļ	Ц	4	Ц	1	\downarrow	¥	Ц	4	Ļ	Ц	Ц	4	+	₽	Н	4	4	Ц
	l:	39	П			H	11	ı		П	ı		1	1				1		H		1	ı	1	ı	1	1	П	-1	١	1	ı	Н	1	1			- 1	١		H	1	١	1	1		ı	ı			1	1		١		П		H	l	ı	١,	il	1		1	l	ı	İ		H	ı		1

JOB SPECIFICATIONS

TDM	International Business Machines Corporation Form X21-0092
IBM RPG CONTI	ROL CARD AND FILE DESCRIPTION SPECIFICATIONS
Date 1/10/7/ Program Stock Status Programmer La Donna Hoffmann	Punching Graphic Punch Page Page Program Identification Punch
Programmer La Donna Hoffmann	
111111111111111111111111111111111111111	Control Card Specifications
Line Core Size to Compile Core Size to Compile Core Size to Compile Core Size to Compile Core Size to Compile Core Size to Compile Core Size to Compile Core Size to Compile Core Size to Compile Core Size to Core Core Size to Core	Address to Start to Start and Start Start and Start Start and Start Start and Start Start and Start Start and Start Start and Start Start and Start Start and Start Start Start and Start
	File Description Specifications
File Type File Designation End of File Sequence File Format Out of File File Format Out of File File Format Out of File File Pout of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File Format Out of File File	Mode of Processing Length of Key Field or of Record Address Field Record Address Type Type of File Organization or Additional Area of Starting Wey Field Starting Key Field Starting Continuation Lines Key Field Starting Continuation Lines Continuation Lines Record Address Type Overflow Indicator or Additional Area of Starting Extent Exit for DAM Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind Continuation Lines Continuation Lines Continuation Lines Revind File Addition/Unordered Number of Tracks for DAM Number of Extents Tape Rewind Continuation Lines Symbolic Device Prile Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind Continuation Lines Symbolic Double Continuation Lines Symbolic Double Continuation Lines Symbolic Double Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Addition/Unordered Number of Tracks for Cylinder Overflow Number of Extents Tape Rewind File Additi
10141 Tel 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
IBW	International Business Machines Corporation Form X21-0094 Printed in U.S.A.
IBM, Date V10/71 Program Stock Status Programmer La Donna Hoffmann	
Date V10/71 Program Stock Status Programmer LaDonna Hoffmann	Printed in U.S.A. RPG INPUT SPECIFICATIONS Punching Graphic Program Identification STOKST Page 622 Program Identification STOKST
Date V10/71 Program Stock Status Programmer LaDonna Hoffmann Line Filename Filename Position	Printed in U.S.A. RPG INPUT SPECIFICATIONS Punching Instruction Graphic Punch Punc
Date V10/71 Program Stock Status Programmer La Donna Hoff mann Recc (N:1) adjust to position	Printed in U.S.A. RPG INPUT SPECIFICATIONS Punching Graphic Punching
Date V10/71 Program Stock Status Programmer La Donna Hoff mann Recc Columb Col	Printed in U.S.A. RPG INPUT SPECIFICATIONS Punching Graphic Punching
Date V10/71 Program Stock Status Programmer La Donna Hoff mann Recc 1	Printed in U.S.A. RPG INPUT SPECIFICATIONS Punching Graphic Punch Instruction Punch Position National Position Nation
Date VIO 71 Program Stock Status Programmer La Donna Hoffmann Reco 1	Printed in U.S.A. RPG INPUT SPECIFICATIONS Punching Graphic Punch Instruction Punch Position National Position Nation
Date VIO 71 Program Stock Status Programmer La Donna Hoffmann Recco Interpretation Properties Programmer Line Programmer Properties Programmer Properties Programmer Properties Programmer Programmer Properties Programmer Properties Programmer	Printed in U.S.A. RPG INPUT SPECIFICATIONS Punching Graphic Punch Instruction Punch Position National Position Nation
Date	Punching Graphic Punch
Date	Punching Graphic Punch
Date	Position
Date	Position
Date	Position
Date VIO 71 Program Stock Status Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction Programmer La Donna Hoffmann Reconstruction	Position

7517	· · · · · · · · · · · · · · · · · · ·		International Business Machines	`amaretian			Form X21-9093
IBW		RPG CA	ALCULATION SF	•	NC		Printed in U.S.A.
Date 1/10/71 Program Stock Status Programmer La Donna Hof		Punching Instruction	Graphic	ECIFICATIO	-	1 2 Progra Identi	75 76 77 78 79 80 am ification STOKST
Line Control Level (1,0,1,1), List State Not Not Not Not Not Not Not Not Not Not	Factor 1	Operation	Factor 2	Result Field	Field Length Field Length Field Length Field Length Field Length Field Length Field	rs ic Zero e Equal 1 - 2 or 2) is Equal	Comments
0 1 C 2 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 ADD	33 34 35 36 37 38 39 40 41	42 43 44 45 46 47 48 TOTISU	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
- - - - - - - - - 		ADD	TOTREC	TOTREC	5¢	FIND	TOTAL REC.
03 611			TOTISU	NEWONH	60	F (410	NEW TOTAL
	· · · · · · · · · · · · · · · · · · ·		TOTREC	NEWONH		DNH	AUD TOTAL
	 - - - - - - - - - -		TOTREC	NEWOND		FIND	MEW OND RDR
 - - - - - - - - - - - - - - - - - - 	 	- 1	MAX	MEMOND	99	IS ON	HAND > MAX
0 7 CL1	' 	COMP		 	7788		
	EMONH	COMP		 		I I IS ON	HAND K MIN
0 9 c	is in lines 01-05 for an explanation alculated, it is condicator 99 in condicator 90 in con	on of the mpared columns	ese entries. Afte to MAX to see 54-55 specifies	er new quan if it exceeds a test to det	tity on hand (N s the maximum termine whethe	IEWONH) limits set r Factor 1	
1114 101 1 1) is greater than		•	_	•		
	07, NEWONH i	-		ee if quanti	ty on hand is le	ss than the	
minimum se	et. If it is, indic	ator 88	is set on.				
С	of the same of the contract	COLUMN ST.	Seal Life Carry County	endigastari, reginistrari.	r regerges somewhat statement was	and programme and a second second	
C	 	\square					
C							
		ШШ					

IBM																		łn	ternati	ional	Busin	ess M	achine	es Cor	porati	on																				rm X: inted i		
. 1	/,	,/-	,,										R	PG	_	-	ou —	TF	דטי	·	FO	RN	۸A	Τ	SPI	ECI	FI	CA	TI	ON	IS				_	1	2									7 78		
Date	57 57		$\overline{\nu}$	<	4	, +		_									ching	<u>.</u>	Grap		4	4	-					_	4					Pa	ge	Ø.	7			Prog Iden	gram ntific	atio	2.	;7	0	K	5	7
Program E	<u> </u>	2001	200	<u>ر</u>	/ 44 · A	4	4,	<u>,</u>	4	2u /		- U			L				runc	;n			1					<u> </u>	ل																			
Programme	· C	νį		1,7	<u>u</u>		<i></i>			7 4	1						_				_																		_						_			ا_
						П	No.	pace	:8	Skip	ı	Ou	tput	Ind	icato	ors											\Rightarrow		-					Edi	it C	ode	:s		_	_			_	1				
						Н	Over	Т	-	Т	╁		Т		Т		\dashv					\cap			1	2		c	omm	as	Zero	Bala o Pri	ances nt	No	Sig	n	CR	-	-	x =		mov		1			terlin	g
Line		F	ilena	me		/T/E)	t/Fetc						And		And			Fiel	d Na	ame		9		nd osito:	,				Yes Yes			Yes			1 2	T	A B	J	ί.	Y =	D: Fi	ite eld E	-		ļ	Si Pi	ign ositio	n
	Ţ					0/H/0	ar Selec	After	Before	After	Not		Not		Not						ğ	k Afte		utpu		xed/b		L	No No			Yes No		ŀ	3 4.		C	N.		z =		ppre	ss]		l		
	ğ					ř	Stacke	1		1			1		-						1	Blar		ecoro	١	2								tant												l		
	6	7/17	10 1	1 12	13 14	4 15	16 1		19 2	0 21	22 23	1 1	25 26 D	27 2	8 29	30 3	1 32	33 :	34 35	36	37 38	39	40 4	1 42	43 4	4 45	46 4	7 48	49 5	50 51	52	53 5	54 55	56 5	57 5	8 59	60	61 (62 6	3 64	65	66 6	57 6	3 69	70	717	72 73	74
I 	0		H	4	C	Ŕ	Ħ	ľ		2	T	0	и	\forall	Ħ	H	╁	H	+	Н	\dagger	T	H	†	+	T		Ť	Ħ	+	T	Ħ	+	T	†	T	Ħ	1	+	†	t	П	Ť	T	$\dagger \dagger$	П	\dagger	Ш
I I I I I I	0	П	П	I			\Box	Ļ	Ц	\prod	1	П		П			1		Ţ		1	L		5	9	1	51	TO	2	4	L	Si	T	7	US	5		RI	E/	70	R	7	,	I		П	\perp	\prod
I - - - 	0	4	-	+		H	-	1	- -	+	- -	1		\vdash	\perp	1	╀		+	Н	+	╀	Н	+	\perp	\perp	\perp	+	Н	+	╀	\mathbb{H}	+	+	-	+	\sqcup	-	+	+	╀	\sqcup	+	+	\dashv	\dashv	+	H
 - - - 	0	+	H	+	0	Ή	Н	+	Н	+	╁	0	+	Н	+	\mathbb{H}	╁	H	╬	Н	+	+	H	2	9		, 7	-2	M	4	ın	,	+	H	+	+	Н	+	+	+	╁	Н	+	+	+	H	+	H
I I ─ ├ ─ } ─ 	0	++	\forall	╁	\vdash	H	\vdash	T	\vdash	$\dagger \dagger$	†	Ħ	+	Н	+	\forall	╁	H		Н	+	+	$\parallel \parallel$		8								10	W	,	╫	Н	+	+		t	H	+	+	\forall	H	+	Η.
0 8	0			T		\parallel		T		\Box	士	\parallel	I			I	\top				1	I		5	6	•	QL	I A	W	71	7	Y		N	(74	A	N	7	7	4	,		I	∄		1	
	0	\prod	П	F	Ц	Д	Д	F	П	Д	\perp	П	Γ	\prod	\Box	П	\perp	П	\prod	Ц	Ţ	Į	Ц	7	1	1	Ť	?A	M.	SA	C	7	10	M	1	ļ	Ц	\prod	1	\perp	1		1	\perp	ot	П		Ш
11-1-1-1	0	+	Н	+	H	1,	H	12	\sqcup	\mathbb{H}	+	1		Н	+	\dashv	╀	H	+	\vdash	+	╀	H	8	9		M	LN	\vdash	-	+	M	4 X	4-1	4	+	+	_	+	-	+	Н		+	\dashv	H	+	+
I I - - 		++	++	+)/ R	╟	2	╁┼	+	╁	0		\vdash	+	H	╁	H	+	+	+	╀	╢	+	+	+	+	╁	Н	+	+	Н	+	+	+	╁	\vdash	+	+	+	╁	Н		+	+	H	+	H
[6	$\dagger \dagger$	$\dagger\dagger$	\dagger	ľ		H	t	$\parallel \parallel$	$\dagger \dagger$	\dagger		+	H	+	\vdash	t	\forall	+	П	\dagger	t	H	5	6	•	01	v	H	AA	ID	\vdash	\dagger	Н	(%		0	RI	DE	R	•	+	+	$\dagger \dagger$	П	\dagger	H
1 4	0														İ							I		6	8		ŏ	JA	N	7/	7	Y	,	П							L	П	1	I	\Box	П		
┃ ┃	익	1	\sqcup	1	Ц	Ļ	Ц	1	Ц	Ш	4		↲	Ц	\perp	Ц	↓	Ц	_	Ц	Ļ	Ļ	Ц	8	9		BI	91	1.1	4	1	B	AL	4.	,	1	Ц		_			Ш	\perp	┵	Ц	Ц		Щ
1 	0		-	+		D	- -	1	-	+	+	1	P _	-	\perp	H	- -	4	C 84		↲	+	Н	-	8	\perp		-	Н	4	+		+	+												rm X inted		
11-1-1-1	計	+	\vdash	╁	-	+	╁	╁	╁┼	\mathbb{H}	+	${\mathbb H}$	╁	H	Н	H	ď		EM SC		4	╁	H	3		+	+	╁	H	+-	╁	-	+	+		1	2									7 78		
0 4	0	11	$ \cdot $	1		$\dagger \dagger$	\dagger	1		$\dagger \dagger$	†	Ħ	T	Ħ	Ħ	H	0	W	44	W	02	2	П	4	4	\dagger	\top	T	\sqcap	+	T	\Box	\top	Pa	age [Ø:	5			Prog	gran ntifi	catio	کاہ	37	0	K	S	7
0 5	0					П			П			П					0	N) R	D	_ 2	?		5	6			L	П					\coprod	1	1	Ш		1	1	1	Ц	1	1	Ш	Ц	\perp	Ц
	9	44	\sqcup	\downarrow	<u> </u>	Ш	4	╀	Ц	$\perp \mid$	╀	\sqcup	4	Ц	Ш	Ц	M	1	٧_	\sqcup	Ž	3	Щ	8		\perp		\perp	Н	1	\perp	Ц	4	11	4	1	Ш	\downarrow	4		1	П	4	\perp	Ц	Н		Ц.
I 1			╀	+	-	D	4	1	H	\mathbb{H}	+	2	_	H	+	\dashv	7	A.	X	\vdash	- 2	4	-	8	8	\perp	+	+	₩	+	-	Н	+	+	+	+	+	-	+	+	╀	-	-	+	+	H	+	
 - - - 	 	++-	╁┼	+-		۲	H	۴	╁	+	+	45	1	+	+	H	+	H	+-	Н	+	+	\vdash	1	7		/ <	55	U	-	+	Н	t	H	+	\dagger	H	+	+	+	+	H	+	+	+	H	+	H
1 0	0	$\dagger \dagger$	Ħ	t	Ħ	П	\parallel	T	H	\parallel	1	Ħ	†	H	Ħ	\parallel	1	5	SU	Έ	5 2	2	\parallel	6		T				_	T	H	Ť	П	1	Ť	П	1	Ť	Ť	T	П	7	†	$\dagger \dagger$	П	\dagger	Ħ
1 1	0	П				D		1				30	7		$oxed{\Box}$													I	П	I			I			I					L			\perp				Ш
 	0	+	H	-		Н	H	\perp	H	\mathbb{H}	1	+	-	Н	\perp		10				۵,	╧	H	1	9	 	KE	: C	$\boldsymbol{\varepsilon}$	1 /	17	ľ	+	Н	4	4	\sqcup		+	-		Н	4	+	u	H	+	\square
 	0	╂┼	╢	+-	-	7	1	12	\vdash	+	+	/	-	H	╁	-	K	=	CE	9	112	4_	╁	6	8	╁	+	╁	${\mathbb H}$	+	╁	Н	+	Н	+	+	Н	\dashv	\dashv	+	+	Н	+	+	+	Н	+	\mathbb{H}^{1}
1 	0	$\dagger \dagger$	\forall	\dagger		ľ	H	^	H	T	- -	Ħ	+	H	T	\vdash	W	Ε	NO	W	HZ	28	H	4	3	T	\vdash	+	$\dagger \dagger$	Ť	t	H		Ħ	\dagger	\dagger	\top	1	+	\dagger	t	H	Ť	+	\forall	Н	\dagger	H
	0	\prod						I					I		Τ		Τ					Ι	П	4	5	1	X	€'	П	Ţ	I			-							_		_			orm)		
	0	11	\prod	1		\perp	\coprod	1		$\perp \mid$	4	4	1	\prod	\perp	Ц	W	Ε	WO	N	0	28	\coprod	5	4					\downarrow	+		_	_		1	2							75 7				9 80
I	0	+	H	+	\vdash	+	${\mathbb H}$	+	H	+	+	8	0	\dashv	+	- -	+	H	+	+	+	+	H		8		*)		E	0	+	H	+	- Р	age	Ø	6			Pro	gran entif	n icatio	, E	3 7	71	2/4	(5	7
-	0	+	H	\dagger	+	\dagger		\dagger	\vdash	$\dagger \dagger$	+	9		H	†	\parallel	+	H	+	T	\dagger	\dagger	\dag	8	9		01			1	t	H	+	- 1	1	_ 	_ 		.		1			T	1		Ī	
	0	$\pm \dagger$		+	 	1		1		1	<u>+</u>	+-+	+	<u> </u>	1		1	<u> </u>			+	1	1-1-	. 🏲	11	1	<u> </u>	T	71	+	<u> </u>	$\frac{\Box}{\Box}$	+	1-1	+	+				<u> </u>	t		_	J	T		1	\prod
1 1 1 1 1	٥	\prod			F	Res	ult	in	g iı	ndi	cat	tor	s, v	vhi	ch	in	did	cat	e v	٧h	etl	hei	r tl	ne	ne	w	qua	ant	tity	y c	on	ha	and	d e	хс	ee	ds	tŀ	he	m	a	٠i-				Ц	\perp	Ц
$t \mapsto \to t$	0		\dashv		n	านเ	m d	or	is I	ess	th	nan	th	e r	nin	im	nur	n,	are	e u	ıse	d d	on	th	e C	Out	pu	ıt-l	Fo	rm	ıat	st	iee	et t	ю:	te	H١	٨ŀ	nei	า				-	╀	Н	+	H
-	0	+	Н	:	e	хc	ept	tio	na	l co	onc	itit	on	s sl	าดเ	uld	b	e r	ot	ed	. 1	No	tic	e i	n 1	the	to	ota	l li	ine	9, 1	the	w s	or	ds	U	N	DI	EF	?				1-	+	Н	+	H
1	0	++	$\dagger\dagger$	1								orir																														_		1	t	Н	\dagger	H
1 2	o	$\perp \!\!\! \perp$	\prod									ed y																														Н			T		\perp	
	0	\prod	\prod									nini					/EI	RΙ	pri	nt	s o	nly	y v	vh	en	99	is	or	1; t	ha	it i	is,	wl	ner	1 N	۱E	W	O	NI	ŀ i	S				\perp	Ц	\perp	\coprod
\cdots	0	+	\dashv	-	g	rea	ite	r t	naı	n tl	ıе	ma	xir	nu	m.																													-	+	H	+	H
 	0	+	+	-		, -	_	,	, .		,	1			1	1 '	1		,			1	1 1	1		1		,	, ,		ļ	1 7	-				_		1,4/%	Ţ	1			.	+	H	+	H
1444	0	++	$\dagger\dagger$	\dagger	\dagger		$\dag \dag$	\dagger	H	\dagger	+	\forall	+	H	+	H	+	H	+	+	\dashv	\dagger	$\dagger\dagger$	+	H	1.	\dagger	\dagger	$\dagger \dagger$	+	\dagger	H	\dagger	$\dagger \dagger$	+	\dagger	\dagger	H	\exists	\dagger	t	H	\dagger	\dagger	t	H	_	\dagger
	0			I		I	П	I									丁			L		I	П			I					I													I	I	\Box		
	0	IT	T		ļΤ		IT	T	Π			П		Π		ΙŢ					H		Π		ΙŢ		ΙŢ		\prod						I													

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.

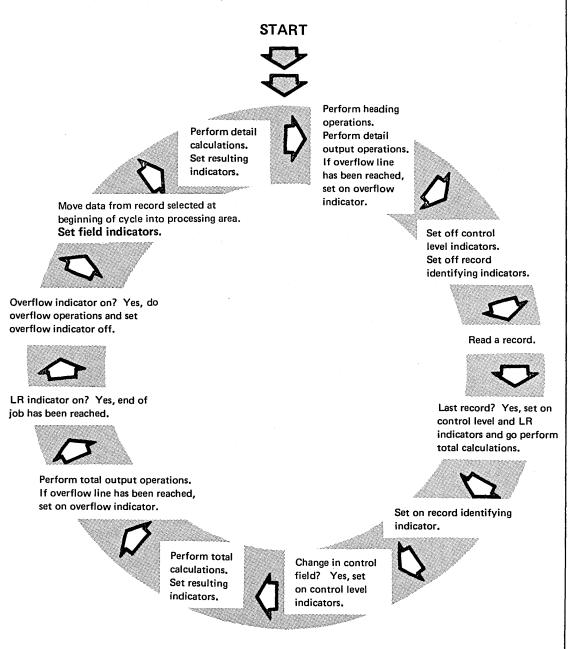


Figure 16. Program Cycle Operations for Field Indicators

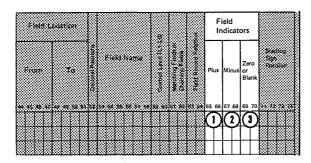
RPG II Specifications

Make these RPG II specifications when you use field indicators:

IBM	International Business Machines Corporation Form X21-909 Printed In U.S.
•	RPG INPUT SPECIFICATIONS
Date	Punching Graphic Page Program Identification
Program	Instruction Punch
Programmer	
Recor	rd Identification Codes Field Location Field Indicators
Frine Parameter Parametrial Parameter Parametrial Parameter Parametrial Parameter Parametrial Parameter Pa	
Line Filename (2) (3) (4) (5)	Position Not (N) Not (
Form Type Sequence Number (1- Option (0) Record Iden OZZ/D CAZ/D C	Position Not (N) Position Not (N) Position Po
	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
0 1 1 0 2 1	┤┤┤╏╏╏╏╏╏
	
	
IBM,	Internetional Business Machines Corporation Form X21-909 Printed in U.S. RPG CALCULATION SPECIFICATIONS
Date	1 2 75 76 77 78 79 8
Program	Instruction Punch Identification Ide
Programmer	
Indicators	Resulting Indicators Arithmetic
Line Long Long Long Long Long Long Long Long	Operation Factor 2 Result Field Field Length Fiel
1	Cookup Table (Factor 2) is High Low Equal R8 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 55 55 85 96 06 16 2 63 64 65 66 67 68 69 70 71 72 73 73 73 73 73 73 73 73 73 73 73 73 73
0 1 C	
0 2 (2)	
	
IBM	International Business Machines Corporation Form X21-9090 Printed In U.S.A
RPG	OUTPUT - FORMAT SPECIFICATIONS
Date	Punching Instruction Punch Page Program Identification
Programmer	
© §Space :Skip Output Ind	dicators Edit Codes
l localitation of the state of	Commas Zero Balances No Sign CR - X = Remove Sterling
Line Filename (1)	
	And Positon W Yes No 2 B K Field Edit No Yes 3 C L Z = Zero No No No No A D M Suppress
Form Type (H/O Stacker Sales After Before After Not Not Not	Record Record Constant or Edit Word.
2 4 5 6 7 8 9 10 11 12 13 14 15 15 17 19 19 20 21 22 24 25 26 27 2	28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 43 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 70 70 70 70 70 70 70 70 70 70 70 70 70
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	<u>┩╒</u> ╤┩┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼
0 2 0 0 0 3 0	╶╏╎╏╏╎╎╏╏╏╏╎╏╏╏╏╎╏╏╏╏╏
	╶┩┊╒┋╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒╒

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:

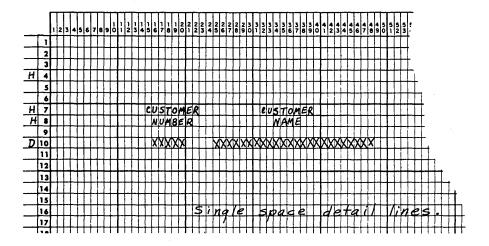
Card Name								97 98 99 100 101 102 103 10
1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16 17 18 19	29 30 31 32	33 34 35 36	37 38 39 40 41 42 43	44 45 46 47 48 49 50	51 52 53 54 55 56 57	58 59 60 61 62 63 64	65 66 67 68 69 70 71 7
	Print Line 1 Tier 1		<u> </u>		Print Line 2 Tier 2			1
E CUSTOMER NUMBER	CUSTOMER NAME	PA	ST YMENT YTE	CREDIT LIMIT	CURRENT CHARGES	30 DAYS OVERDUE	60 DAYS OVERDUE	90 DAYS OVER DUE
1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16 17 18 19 2	9 30 31 32	2 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

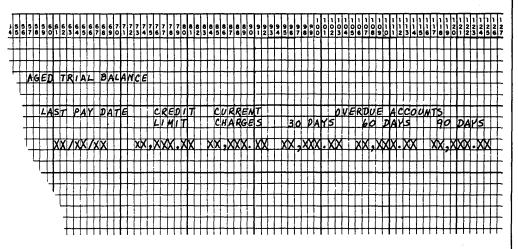
Processing: Check input balance due field for zero or credit balance (use field indicators).

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

		AGED TRIAL BALAN	NCE				
CUSTOMER NUMBER	CUSTOMER NAME	LAST PAY DATE	CREDIT LIMIT	CURRENT CHARGES	OV 30 DAYS	ERDUE ACCOUN 60 DAYS	NTS 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	90.4
24743	BERKLEY PAPER CO	2/21/70	6,300.00	1,185.50	2,652.45	1,400.05	51.0
25271	BEST DISTRIBUTION CO	10/06/69	1,000.00	3.25			762.1

This Printer Spacing Chart shows how the report is formatted:





JOB SPECIFICATIONS

IBM Internet	ional Business Machines Corporation Form X21-9092 Printed in U.S.A.
RPG CONTROL CARD AN	ID FILE DESCRIPTION SPECIFICATIONS
Programmer La Donna Hoffmann	
Program Hged Irial Balance Instruction Punc	
Programmer La Donna HOTTMANN	
Contro	l Card Specifications
Sterling Sterling	Model Mode
File Des	cription Specifications
Form Type Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Post Pendin Pendin Post Pendin Pen	dress Field dress Field Name of Number of Extents File Symbolic Label Exit
IBM Internation	onal Business Machines Corporation Form X21-9094 Printed in U.S.A.
•	Frinted in U.S., A.
RPG INP	PUT SPECIFICATIONS
Date 1/10/71	1 2 75 76 77 78 79 67
Date 1/10/71 Program Aged Trial Balance Punching Instruction Punch	Page 2 Program Identification A G E - T E
Date 1/10/71	Page 2 Program Identification A G E - T E
Date 1/10/71 Program Aged Trial Balance Programmer La Donna Hoff mann Record Identification Code 1 1 2 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	Page 2 Program Identification A G E - T E
Date 1/10/71 Program Aged Trial Balance Programmer La Donna Hoff mann Record Identification Code 1 1 2 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	Position Position
Date 1/10/71 Program Aged Trial Balance Program Aged Trial Balance Programmer La Donna Hoff mann Record Identification Cod 1 2 2 2 2 2 2 2 2 2	Position Position
Program Aged Trial Balance Program Aged Trial Balance Program Aged Trial Balance Programmer La Donna Hoff mann Record Identification Code Programmer La Donna Hoff mann Position Programmer La Donna Hoff mann Programmer La Donna Hof	Position State Page Page Program Program Program Program Program Page Page Program Program Page Page Program Page P

TD36								International Re	eineer M	achines Corporati	an					Fo	orm X21-9090
IBW					R	PG	0	UTPUT - F				IONS					inted in U.S.A.
Date 1/10	171					·	Punchi						Page	1 2	Program		7 78 79 80
Program Ag	ed Trial La Donna	Ba	lan	ce	_	Į	Instruc	g					rage	פט	Identificatio	466	- TB
Programmer	-a Donna	Ha	ff	ma	an												
		NO Sp	ace Si	kip	Output	Indicate	ors						Edit Co	ndee		_	
		Overfix	+	- 1	1	—т	_				Com		alances No Sign	7	X = Remov		Sterling
Line	Filename	t/Fetch			And	And		Field Name	(8)	End :		es Y	es 1	A J B K	Plus Si Y ■ Date Field E		Sign Position
Type	(H/Q)	r Seleci efore	After Before	After	Not	Not			Codes k After	in Output	oi i N		es 3	C L D M	Z = Zero Suppre	ess	
P P	<u>ع</u>	Stacke	1	1 1		.			Edit Blan	1	il .		Constant or E				
	8 9 10 11 12 13 14 15 EPORT H		18 19 20 3 Ø 6		23 24 25 26 1 P	27 28 29	30 (31)	32 33 34 35 36 37	38 39	40 41 42 43 4	4 45 46 47 48 49	50 51 52 5	3 54 55 56 57 58	59 60 61 62	63 64 65 66 6	57 68 69 70	71 72 73 74
0 2 0	OR		\prod	Ш	OV												
0 3 O O	H	H	1	H	1 P		\dashv		Н	74	AGEL	TRI	AL BA	LANC	<i>E</i> '	++	
0 5 0	OR				ov												
0 6 0		Ш		Ш	++1		\square			22	CUST	OMER	7			$\Box \Box \Box$	
0 7 0 0 0 8 0		Н	+	H	+++	+	+	++++	+	3 <i>9</i>	CUST	DA	DATE CURR ACCOU	,		+++	+++
9 0							$\perp \!\!\! \perp$			91	CRED	/ T	CURR	ENT'			
1 0 0	 	- -	4	\square			4		Ш	118	OVER	DUE	ACCOU	NTS'	+++1	444	
1 1 0 1 2 0	OR	+H	Z	Н	1P	++	+		\mathbb{H}				++++			+++	+++1
1 3 0										21 37	'NUME	ER					
1 4 O		H		\sqcup	++		+	++++	\mathbb{H}	37 91	WAME		CHAR		+++	+++	+++
		Ш		<u> </u>					<u> </u>		I ILI MII		LHAR	(at)			orm X21-9090
IBM					R	PG	O	UTPUT - F		achines Corporati		IONS					inted in U.S.A.
Date	<u>/71</u> ed Trial aDonna				•	 [Punchi	l cointre		1		10.10		1 2	Program		7 78 79 80
rogram <u>Ag</u>	ed Trial	Ŗ	2/a	nC	2		Instruc	g					rage		Program Identification	MOE	:-110
Programmer Z	a Donna	#0	offn	nal	n												
		low (F)	sce Si	kip	Output	Indicate	ors						Edit C	odes			
		Overf	-	T^{I}							Com		alances No Sign	1	X = Remov Plus Si	e	Sterling
Line	Filename 🗒	:t/Fetch			And	And		Field Name	(B)	End a	¥	es Y		A J B K	Y = Date Field E	- 1 1	Sign Position
m Type	(H/Q)	er Selec Sefore	After Before	After	Not	Not			t Codes	in Output	N N		es 3	C L	Z = Zero Suppre	ss	
3 4 5 6 7	8 9 10 11 12 13 14 15	Stack		, ,	22 24 25 26	27 20 20	20 21	22 24 25 26 27	Blar	Record	1 45 46 47 48 40	EA E1 E2 E	Constant or E		63 64 65 66 (27 69 60 70	71 70 70 74
0 1 0					23 24 25 20	27 28 29	30/31	32 33 34 33 36 37	36 39	102 124	130 L	AYS				7 68 69 70	1 12 /3 /4
0 2 0			1	\square	99	H-			\square	124	1'6Ø Z	AYS	94	DAY	5'		
0 4 0	OR		4	HH	98	++	+H	++++	H		H + H	+++	++++	+++	H + H	+++	H + H
0 5 0		LLI	山		97						 	1 1 1 1		1-1-1-	 	1111	
0 6 O O	OR	Ш		Ш	96		44	<u> </u>	Ш		Field in	ndicato	rs 96-99,	used to	o test tl	ne CH	ARGE
0 8 0		H	+	H	+++			CUSTNO NAME	\mathbb{H}	2¢ 48	1		ields for	-			
0 9 0								LASTPA		68	1		letail rec				
1 0 0		H	\bot	Ш	+++	4		LIMIT	2	81 92 1 6 3			one of th or 99 or 9				15,
1 1 0	+++++		+	H	+++	+	-	CHARGE OVER36	2	102	I		ord ident				;
								OVERLO	2	114	also use	ed to c	ondition	the det	ail reco	rd. Th	nis
1 3 0			\prod		Π			OVER90	2	125	1	•	vent prin	-			m
1 4 0	1 1 1 1 1 1 1					1 I I			 	1111	Largor	d type	other tha	an 01 /	+6		
1 4 O		Н	++	┼┼┤	+++	++	\dashv	++++	_	++++	I						
1 4 0											I		ally gotte				
1 4 0 1 1 5 0 0											I						- - -

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																			D	D.C.			<u> </u>		iternatik											
	Date																_			n	PG		Pun	chin	ng _	Grar Punk											
	Programn	ner _																																			
							10				rflow (F)	Spa	ace	: S	cip.			Οι	ıtp	ut	Ind	ica	itors	T		_											
	Line			F	iler	nan	ne			(H/D/T/E)	t/Fetch Over			•					A	ıd		An	d ,														
I		orm Type									Stacker Select/F	Before	After	Before	After		Not			Not			Not]	est e	This										the	r
Ì	3 4 5	6	, ,	8 9	10	. 11	12	13	14	15	ets Sta	17	18	19 20	21 :	22	23	24	25	26	27.43		200	1	8	indic	ato	r 1	P	or	O	V	is	on	•		
ı	0 1	٥	T	Τ	Τ	Γ			\Box	D			1	Т	П	1		1	P					7									,				
l	0 2	0		Τ	T	Γ			0	R				T	П	Ţ		5	V		F	7	П	T													
	0 3	0		Ι											П			1								·This	line	e is	w	rit	te	n	wh	en	ei	the	r
	0 4	0								D			1		П	1	1	1	Ø			I		7	(on.	indic	ato	r 1	0	or	20	0 i	s c	n,	or	wł	ner
I	0 5	0	T	Τ	7	Γ	Γ		0	R					П	1		2	Ø	_		7	7	T		both								,			
ı	0 6	0	T	T	Τ	Τ	Π		П	٦				\top	П	7	Ī	٦	٦		T	T		7		2000			• • •								

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:

I	BI	Ĺ				٠																	B	P(G		C.			-	al B		
	ograi	m					_										_		_ _							unch			-	aph	_		
			T					Ind	ica	itoi	's						_						T									_	
	Line	ļ		rol Level	il		A	nd 	,	Α	nd						Fa	ctc	r 1					Ор	era	atio	n				Fa	ct	
3	4	5 6		Control			11	TON 12		14	NO 15		17 1	8 1:	9 21	0 2	1 2	2 2	3 24	1 2	5 21	3 2	, 21	3 29	30	31	32	33	34	35 :	36 3	37	This operation is performed when 10 is on and 20 is not on.
_	1		-1	Ι		_	d	_	_	1	_	П	_	4		T	\prod					I	7	D	_	T	$\overline{}$	B	_		7		
0	2	d	;	Ι		Ι	ľ	Γ		I										Ι			Γ		Γ		Γ						This operation is performed
0	3	0	:	4	4	1	0		1	5		9	9/	4						T			5	U	B			D			4		when L1, 10, 15, and 99 are all on.
0	4		;			Ι	ľ			L		Ш						Ι	Ι			I		L									
0	5	ď	: 1	1	4			N	1	5		Ш	1	F)	C:	S.	T		1		۴	10	L	. T	1	T	A	X	+	+	This operation is performed
0	6	9	<u>:</u>	\perp	L		L	L	L	L	L					1					1	1					L	Ц					when L1 is on and 15 is not on.
0	7	1	:	1	L	1	L	L	L	L	L		1	1	1	1	4	1	1	1	1	1	1	ļ	L	L		Ц	Ц		1	4	

IBW	RPG	International Bu	
Date		Punching Graphic Instruction Punch	
Filename Filename Line Filename Line Filename Line Filename Line Holding Space Line Line Holding Line Line Line Line Line Line Line Line	· · · · · · · · · · · · · · · · · · ·	Field Name	This detail line is written when
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 1	19 20 21 22 23 24 25 26 27 28 2 1 5 1 Ø	89 30 31 32 33 34 35 20 7	15 and 10 are both on.
0 2 0 0 T 12		15	– This total line is written when L1, 10, and 15 are all on.
0 5 0	╏╎╏╏╏	 	

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:

1	В	N	Ę																						R	Р	G			C	טט												etion PE
D	ate	_							_																			ſ	Pı	ınçt	ina		G	rap	hic	:	Γ	T		T	_	Τ	Т
Pı	rogr	ran	n _																									l		stru		n	P	unc	:h		t	1	_	Ť		T	†
P	rogr	ran	nm	er_	 									_	Œ.	Γ	ace	 	: Sk	dip		Γ	_ 0	ut		: 11	ndi	cat	ors		Γ						T.						
	Li	ne		_			F	ile	na	me				/T/E)	Stacker Select/Fetch Overflow	-								_	Ind			und				Fie	eld	Na	m	e		F	T	En			= Binary
	_			⊕ Form Type	,	8	9							Type (H/D/T/E)				l	Before		After	Not	24	~	Not			Not				~~		•	•	•	Edit Codes	8 Blank After (B)	1	Re	tpu	d	P = Packed/B
H	. 1	1	5	0	ŕ	r	Ť	T	Ť	Ϊ	Ť	13		ה ה	۳	۲	1	٦	<u> </u>	ť	Ť	13	1	Q		1	1	23	1	2	32	Ü	Ť	39	Ť	T	38	339	Ť	Ť	T	T	H
1	0 :	2	-	0	-	_	H	t	t	+	†		A	-	ก	t	r	t	H	H	t	H	1	5	-	ľ	6		1		t	H	T	r		t	t	t	t	✝	✝	t	H
t	+	3	-	0	┢	H	H	t	t	\dagger	†	-	_	100	•	┢	t	H	H	H	t	H	1	-	╁	ľ		1	t	t	c	-	H	H	İ	1	t	t	t	t	4	ø	Н
l,	n I.	,	٦	<u>ا</u>	T	1	+	1	\dagger	+	†			Н	Г	t	t	H	t	1	1	┢	+	1	十	t	t	1	T	T	†=	Γ	T	r	T	t	t	t	t	t	Ť	1	Н

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.

I	B]	VI																												in	tern	atic	nal	Bus	ines	s Ma	chir	ies C	Corpe
		•																							RI	PC	;	(CA	٩L	.C	U	L	47	TIC	NC.	1 :	SP	E(
Da	te _							-																		Γ	Pu	nch	ing		G	rap	hic				_	Γ	Т
Pre	ogra	m_							_								_				-						Ins	tru	ctic	n	P	unc	h						Ī
Pr	ogra	mm	ner_																		-																		
			Γ					1	nd	ica	tor	s													Γ					Γ							_	_	٦
l				l		┝			_			Г	_	-																									ı
l	Line	e	l.	že.	LR, SR)			A	nd		A	l nd						F	act	tor	1				,	Οpi	era	tio	n				F	ac1	or	2			
			Type	Control Level	.9, L	L			L	_	_	L	_																										-
			Form	ខ្ញុ	(LO-L9,	ž			ž			Š																											ŀ
	4	5	6	ļ,	8	,	10	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
0	1	Γ	С	Γ		Г	$\overline{}$	5		3	ø		3				Γ		Γ			-		Γ	Г	Γ		Γ	Γ	Γ	Γ	Γ	Г	Г	Γ	Γ	Γ	П	П
0	2	Γ	С	A	N		9	Ø	N	9	1			П	C	0	s	T			Γ	Γ	Ī.		M	U	L	7		7	A	X							
0	3	ľ	С				Γ			1				П					Г						Γ		Ī		Г	ľ	Γ.	Γ.	Π	Γ	Γ				П
6	4	Τ	С	Γ	Г	Γ	Γ	Γ	Γ	Τ.	Γ	Г	Γ	П		Г	Π	Π	Γ	Γ	Γ	Γ	Γ	Γ	Г	T	Γ	Γ	Γ	Γ	Г	Γ	Π	Γ	Γ	Γ	Γ		П

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.

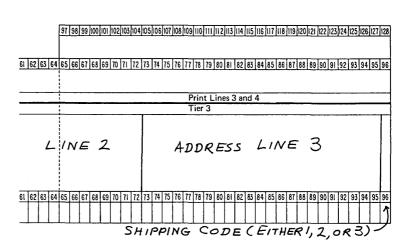
DETERMINE THE JOB REQUIREMENTS

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

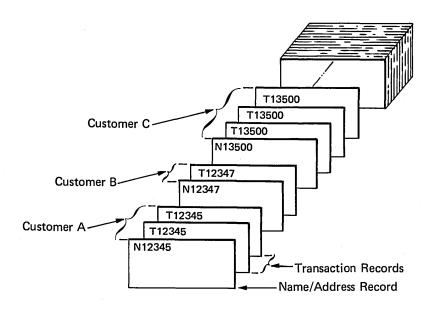
Print	ı	2	3	4	5	6	7	8	9	10	Ш	12	13	14	15	16	17	18	19	20	21	22	2	24	1 2	5 2	3 2	7 28	29	30	0 3	32	33	34	35	36	37	38	39	40	41	42	43	44 4	15 4	6 47	7 48	49	50	51	52	2 5	5	55	56	5 5	7 5	59	60
	F	_		_	_		_		_	_	_	_		rin ier	t L	in	e 1	_		_					_		_						-	_	_		_				_	_		_		int er 2		ne 2	2	_	_	_	_		_		_		_
Punch	CODE = N	1	co	: o	UN'	7									Ą	M	E						-						4	1	Di	Di	₹ <i>E</i>		3.5	5	۷		~	Ε		1					-							4	D	D	R	E	SS
Program Control Card	1	2	3	4	5	6	7	8	9	10	11	12	13	4	15	16	17	18	19	20	21	22	2 2	3 2	4 2	5 2	6 2	7 2	8 29	9 30	0 3	1 32	33	34	35	36	37	38	39	40	41	42	43	44	15 4	6 4	7 48	49	50	51	5	2 5	5 5	1 5	5 56	3 5	7 5	5	60



Card Name TRANSACTION RECORD

Print	1 2	3 4 5	6 7	[8]	9 10	[11]						19	20 2	21 22	23	24 2	5 26	27	28 29	30	31 3	32 3	34	35	36	37 3	8 39	40	41 4	2 43	44 4					51	52 5	53 54	55	56 5	57 58	59
	 								nt L	ine	1_											<u> </u>						_				_	_	ine	2		_					
	L-,		-					Tie	r 1													+	_	,				_				lie	er 2									
Punch	W	COUNT VMBER	1		Ι · Νυ		M BE				i	D £	S	c ƙ	?//	o 7	-/	٥٨	/	Q	VΑ	MT	ץזו	f	UN PR	// /C	Ē															
Program	1 2	3 4 5	6 7	8	9 10	11	12 1.	3 14	15	16 I	7 18	19	20 2	1 22	23	24 2	5 26	27	28 29	30	31	32 33	34	35	36 3	37 3	8 39	40	41 4	2 43	44	15 46	6 47	48 4	9 50	51	52	3 54	4 55	56 5	7 58	59
Control Card																																										

• 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.

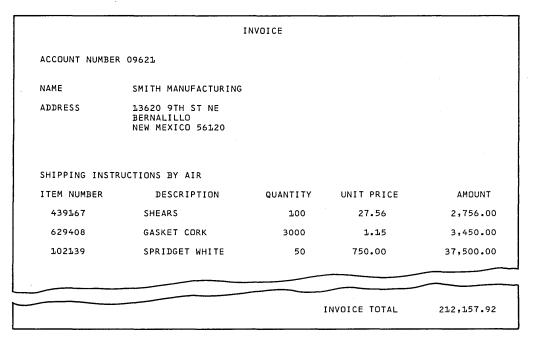


Figure 17. Sample Invoice

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

	Ţ,	2	3 4	5	<u></u>		او	ŀ	1	1 1	1	1	1 1	;	2	2	2 2	2	2	2	2	2 2	3	3	3	3	3	3	3	3	3	4	4	4 4	4	4	4	4 4	4 9	5	5	5	ş	5 5	5	5	5	5	6	6	6	6	6	6	6	5 6	7	,	7		7	7	7	7	7		9		8	8	8	9 5	8 8	٥	9	9 5
	iŤ	Ħ	+	Ħ	+	H	Ť	+	H	+	H	7	+	۲	H	+	+	Ŧ	H	7	7	+	+	ŀ	H	7	7	7	7	۲	-	H	H	+	T	H	7	Ť	۲	H	H	H	-	7	7	۴	۲	4	Ŧ	۲	۲	1	Н	۲	7	+	۲	H	-	+	۲	۲	۲	۴	-	7	+	۲	F	H	4	+	7	۴	H	7
	┇	H	$^{+}$	H	+	Н	+	t	Н	+	Ħ	+	+	+	H	+	+	t	Н	Н	+	$^{+}$	+	t	Н	+	+	+	$^{+}$	+	Н	Н	Н	+	+	Н	+	+	+	Н	Н	H	+	+	+	t	Н	+	+	+	╁	╁╴	Н	Н	+	+	۲	Н	+	t	t	Н	H	Н	+	+	t	╁	Н	H	+	$^{+}$	+	H	H	+
	3	H	Ť	H	+	H	$^{+}$	t	H	+	H	+	+	H	H	+	+	+	Н	1	+	+	+	H	Н	+	+	+	t	t	H	Н	\forall	+	+	Н	+	Ť	t	Н	Н	Н	+	+	$^{+}$	+	H	+	+	+	t	H	Н	Н	+	+	╁	Н	+	+	t	Н	Н	Н	+	†	t	t	Н	H	+	十	+	۲	Н	+
Н	4	Ħ	+	H	+	Ħ	+	t	Н	+	Ħ	†	+	۲	Н	十	+	$^{+}$	Н	+	+	+	t	t	Н	1	7	+	t	t	1	v	٧	0	t	E	+	$^{+}$	+	Н	H	Н	+	+	+	+	Н	+	+	+	t	t	П	Н	+	Ť	+	Н	+	†	t	Н	Н	Н	+	†	+	t	H	H	+	+	$^{+}$	۲	H	+
	5	11	+	Ħ	†	Ħ	+	t	П	+	Ħ	T	t	T	П	†	†	T	Ħ	1	7	\dagger	t	t	П	1	7	+	t	t	Ť	Ή	1	1	۲	П	ナ	T	†	H	H	Н	7	+	+	t	Н	T	†	T	t	t	H		†	t	+	H	7	Ť	✝	H	Н	Н	+	†	t	t	H	H	1	+	+	۲	H	+
	6	Ħ	†	H	†	H	†	t	Н	†	Ħ	7	t	۲	Н	+	+	t	H	H	1	†	t	t	H	7	7	+	†	t	H	Н	H	+	t	Н	+	+	t	Н	Н		+	+	+	+	H	+	+	+	t	t	Н	Н	+	+	$^{+}$	H	+	†	t	H	Н	Н	+	†	+	t	Н	Н	+	$^{+}$	+	t	Н	+
H :	扌	Ħ	A	d	cla	u	Μī	オ־	W	υN	18	ΕÍ	₹	k	V	χİ:	χĺ	ď	Ħ	1	A	1	1,	V	Ы	Ť	1	†	t	t	Н	Н	П	Ť	Ť	Н	†	+	Ť	Н	Н	П	7	†	Ť	t	П	1	t	Ť	t	t	Н	П	†	Ť	t	П	┪	t	t	Ħ	Н	П	1	†	Ť	t	H	П	1	†	†	Ħ	H	+
	8	П	T	Ī	Ť	Ħ	Τ	T	Ħ	Ť	T	Ť	1	۳,	M	Ť	Ť	Ť	П	۲	٦	٦	Ť	r		7	7	+	Ť	T	П	П	П	†	T	Ħ	1	Ť	T	П	П	П	7	1	Ť	T	П		T	+	T	T	П	П	1	- -	T	П	1	†	T	Т	П	П	1	1	t	t	П	Π	†	†	†	۲	П	†
	9	Ħ	Т	П	T	Ħ	7	T	П	T	11	7	T	Т	Π	T	Ť	Τ	П	٦	7	Ť	T	٢	П	1	1	1	Ť	T		П	П	1	T	П	T	T	t	П	П	П	1	1	T	t	П	П	Ť	Ť	T	T	П		7	T	Τ	П	7	Ť	T	Т	М	П	T	†	Ť	t	Ħ	Π	7	1	T	М	П	+
H 10	0	П	W	A	MI		T	T		T	П	Т	T	Τ	ΧĪ	ΧĪ	Ø	ላ×	X	X	X	Ø	Ø	X	X	X	X	K.	φ	ĺχ	X	П		(N	17	1	М	Ē	7	П	П	П	1	1	1	T	П	П	1	1	T	T	П		7	†	1	П	7	Ť	T	Т	Г	П		7	T	T	П	П	7	T	T	Т	П	+
1	1	П	1	Π	Ι	Π	T	Т	П	T	П	П	T	Γ	П	T	T	T	Π		٦	Ť	Ť	Γ			1	T	T	ľ		П		T	Γ	П	.1	7	7			Π	7	T	1	Τ			T	Τ	T	T	П		1	1		П	1	T	Γ	Γ	Г			1	1	T	Г	П	1	T	T	Г	П	T
H 12	2	П	A	1	DA	Έ	5	1	П	T	П	Т	T	П	X	办	()	₫X	X	X	X	Ø	Œ	k	M	K	X	X)		X	X	Ø	XI.	V	Т	1	Ā	D	D	R	E	3	3	7	4	۸/	16		1	1	Ţ	T	П		7	T	Τ	П	7	T	Т	Г	П			1	T	T	П	Π	7	T	T	П	П	T
H_{1}		\prod	Ι	П		П	T	Ι		T		П	T		M	X.	W	Œ	X	Х	X	W	Ø	X	X	XI	X	X	()	X	K	Ø	XI)	X.	Ι	a	Ā	D.	Þ	R	E	3	3	- 1	4	1/	7	Ξ	Ţ	Ž.	D	Γ	П			Τ			1	Τ	Г	Г	Г			·T	Т	Γ	П	П	T	Т	Τ	П	П	Τ
H_{1}	4			П	T		Т	Τ	П	Τ	П	П	T	Γ	Х	ΧĎ	$\langle 1 \rangle$	ØX	X	X	X	X	Œ	X	X	X	X	X)	()	(X	X	X	X	1	Т	7	A	Ž.	D	R	£	3	\$	7	4	11	V 4	=	Ŀ	3	7	Γ	П		T	1	Τ		7	T	Г	Г	Г		П	T	T	T	П	П	1	T	T	П	П	T
1:	5	П	Ι	П	Τ	Π		Τ		Ι	П	Π	T	Г	П	Τ	Ī	Τ			٦	T	T	Γ				Ī	T	Τ	Γ		П	T	Γ		T	T	Ι					Т	Τ	Τ			Τ	I	Ι	Γ			1	T	Γ			Ī	Ι	Г	Г			Ι	Τ	Τ		Π	1	T	Τ	П	П	Τ
110	6	П		П	Ţ	\prod	I	Γ		Ι			Ι		П	I	Τ	Ι				I	T	Γ					I	Ι			П	Ι	Γ		\Box	T	Τ				I	Ι	T	L			Ι	I	Γ	Ι			1	Ι	L		I	Ι	Ι	Γ	Г			Ι	T	T	Π	Π	Ţ	Ι	Ι	Γ	\Box	Ι
1:	7	П		L	Ι			Ι		Ι	П		I			I	I	Ι	L			Ι	I					1	Ι	Ι	L			Ι	L			1							I	Γ			1		Ι	L			1	1	L			I	L	Γ				I	Ι	I		Π		I	1	Γ	\prod	Ι
1	8	П	Ι	П				Ι		Ι		П	T		Π	Ι	Ι	T				Ι	Ι	Γ				Ī	I					I	Γ	П		T	T					Ι	Ι	Ι	П		Ι	Ι	Γ					Ι				Ţ	Ι	Γ	Г			Ι	I	I		П	I	Ι	Ι		П	Ι
H	9		S	H	11	P	11	VG	П	1/4	13	7/	90	ic	ਜ	7	0/	٧s		X)	X	Κþ	()	X	X	Х	٦	T	()	B	Y		A	7	R	,	2	3	7		7		र	u.	¢	K	1	₹	가	R	1	В	Ч	,	Í	₹.	A	1	4	J	Σ.	П	Г			Ι	I			Π		Τ	Τ	П	П	Ι
20		П	I	Π	Ι			Τ	П	T	П		T	Ī	П	I		Τ	Γ			I	Τ	Γ				Ī	Τ	T			П		Γ		T	Τ	Τ				I	Ι	Τ	Ι			Ι						I	Ι				Ī	Ι	Г	Г			Ι	Ι	Ι		Π		I	Ι	Γ	П	Ι
H 2	1	П	1	r	ΕV	1	Νl	W	B	Ef	₹		Ι			I	Ι	Đ	E	5	4	RI	P	7	1	a	N		Ţ	Τ			П	G	U	A	M	11	7	4				Ι	L	W	1	T	1	PF	1	C	E		1	l	L			Ι	Γ	A	М	0	U	V.	Ĺ	I		П		I	Ţ.	Γ	\Box	Ι
2:	2	П		П	1			L	П		П		I			Ц		I	L			I	Ι	L				J	I	I	L		Ц	1	L			1	L					\perp	1	L			1	1	L	L			\perp	1					I	L	L	L		1	I	I	L	Ц		\perp	\perp	\perp	П	Ι
D 2	3	Ц	1	Ц	ΧŅ	W	Δ	(X	Ш	1	Ш		⊥	L	Ц	_1	X)	(X	X	M	N	<u>()</u>	Ø	X	X	X	X	X)	(L		Ц	1	L	X	Χþ	Ø	(X	Ш	Ц	Ц	1	┙	1	L	X	X	Ц.	. X	Ω	L	Ц		1	1	Ц	Ц	1	2	X		X	X	M	.₽	(Z	L	Ц	Ц	1	1	1	Ц	Ш	\perp
2	4	Ш	┙	41		Ľ		L	Ш		Ц		1		П	┙			L					L	Ш			J,	L				Ш	1	L	اما			L										l	\perp										ı.	L	Ĺ	Ľ			1	1			Ц		⊥	⊥	Ц	Ц	⊥
2	5	Ц	1	1	7	16	: /	V	Δ	10	2		Ι	L		I		Ι	C	Z	2	=	₹	4	L	R		2	L	L	L		Ц		1	4	<u>એ</u>	Ť	4	11		Ц		I	K	U	1	2	3	1	(1	٥)			L	Ц	I	1	A	Λ	10	2	Ų	V	7	12		Ц		Ţ	\perp	Ц	П	Ι
2	6	Ш			1	Ш		Ľ	П		Ĺ		1		Ц	┙			Ľ				1	L				1	1	L	L		Ш	1			Ì.	1	Ľ	Ĺ	L	Ц		1	1	L	Ш		1		L		П		1	L	L	Ц	1	1	L	L	L			1	Ţ.	1	L	Ц	1	1	\perp	Ц	Ц	L
2		Ш	Ш	Ш	1	Ш		L	Ц	\perp	Ш		l		Ц	Ц	_		L				I	L				I		Ţ	L		Ц	1	L			1	1	Ц	L	Ц	Ĺ	1	1	Ĺ	П		1	1	L	L		Ц	1	1	L	Ц	1	1	L	L	L		Ц	1	1	L		Ц		\perp	\perp	Ц	Ц	1
T 2	8	Ц	1	Ц	4	Ш	Ц	L	Ц	1	Ц	Ц	1		Ц	1	1	1	L	Ц			1	L	L		Ц	1	1	L	L	Ц	Ц	1	L		Ц	1	L	L	L	Ц	1	M١	1	1	9	E	_17		17	Α	4	Ц	1	┸	L	М	X.	2	X	X	٠	X,	X)	Υ.	<u>. X</u>	1X	L	Ц	1	1	1	Ц	Ц	\perp
2	9	Ц		Ц	1	Ц	Ц	1	Ц	1	Ш	Ц	l	L	Ц	┙	1	⊥	L				L	L	L	Ц		┙		L	L	Ц	Ц	1	L	Ц		_	1	Ц	L	Ц	1	1	L	L	Ц		1	1	L	L	Ц	Ц	1	1	Ц	Ц	_	1	T.	L	Ĺ	Ц	Ц	1	L	Ł	L	Ц	┙	1	1	Ц	Ц	\perp
30	0	П		Ш		\perp	Ш	L	Ц		\perp			L	Ц	Ц	1		L	Ц		Ţ	Ĺ	L	\sqcup			\int	Ī		L	Ц	Ц		L	Ц			1	L	L	Ц	1	1		L	Ц		┙	\perp		L	Ц	Ц	1	1	L	᠘		1	¥	V	Ц		2	1	I,	L	Ц	Ц	1	1	1	L	Ц	\perp
3	1	П		Ш	4			L	П	\perp		П	\perp	L	П	П	I	Ĺ				\perp		Ĺ				J			Ĺ		Ц		L	П	I	\perp	L			Ц	\perp		1	L	Ц		1	1	L	L	Ц		1	1	L	Ц		1	L	L	Ц	Ц		1	ľ	L		Ц	1	1	1	L	Ц	\perp
3:	2	П		\Box	1	\perp	Ц	Ĺ	Ц	Ł	4	Ц	I	1	IJ	ال	1	L	L			I	Ι	Ĺ		Ц	J	J	I	Ĺ	1	Ц	J	1	L		4	1	1	ŀ	L	Ц		Ĺ	1	1	Ц	Ц	1	1	L	L	Ц	Ц	1	1	L		1	1	L	L	Ľ	Ц	Ц	1	L	L	L	Ц	1	1	1	L	Ц	\perp
3:		Ц	1	Ц	1	L	Ц	L	Ц	4	Ł	Ľ	2	Ÿ	Ø	4	₹	1	L	1	5	I	1	Ł	9	4	9	1	1	6	16	2	7	4	11	Ц	1	1	1	1	(19	3	5	٧.	1	Ц		1	1	L	L	Ц	Ц	1	1	L	Ц	1	1	L	L	Ц	Ц	Ц	1	1	L	L	Ľ	1	1	1	Ц	Ц	\perp
3	_	Ц	1	Ц	4	Ц	Ц	1.	Ц	1	Ц	Ц	4	L	Ц	Ц	1	1	L	Ц	_	1	L	L	Ц	Ц	Ц	1	1	1	L	Ц	Ц	1	L	Ц	1	1	1	Ц	Ц	Ц	1	1	1	1	Ц	4	4	1	L	L	Ц		4	1	Ц	Ц	4	1	L	Ц	Ц	Ц	Ц	4	1	L	Ц	Ц	1	4	4	Ц	Ц	1
3		Ц	\perp	Ц	1	\perp	Ц.	1	Ц	\downarrow	\perp	Ц	1	1	Ц	Ц	1	1	L	Ц	4	1	1	L	L	Ц	Ц	4	1	1	L	Ц	Ц	1	L	Ц	1	1	L	L	L	Ц	1	1	1	┸	Ц	4	4	1	L	L	Ц	4	4	4	Ц	Ц	4	1	L	L	Ш	Ц	4	4	\downarrow	L	Ц	Ц	1	4	4	Ц	Ц	\perp
1_		1 1	Т	11	- 1			1	ı i	- 1	1	ıl	- 1	1		. 1	- 1	- 1	1	1 1	ı	- 1	i	1	ıl	1	- 1	- 1	1	1	1		1 1	1	ı	: 1	- 1	- 1	1	1		ıl	- 1	4	1	1	ιi	. 1	ı	1	1	1	, 1		- 1	1	1	ıŧ	- 1	1	1	. '	1 1	ıl	- 1		1	1	1		- 1	- 1	1	1	1	1

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.

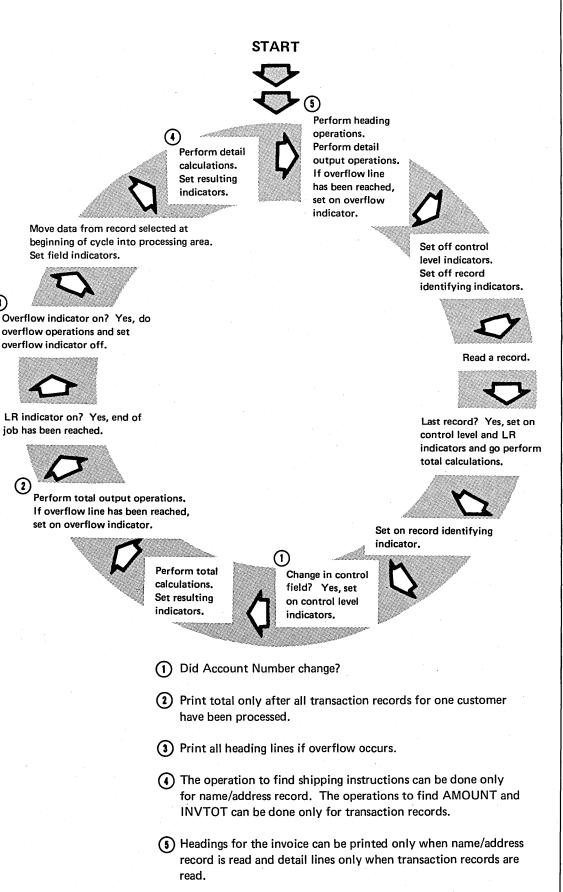


Figure 18. Program Cycle Operations for Sample Job

(3)

2

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 CONTI	ROL CARD AND FILE DESCRIPTION SPECIF	ICATIONS 1 2 75 76 77 78 79 80
Date 1/10/71 Program 1NVOICE Preparation Programmer La Donna Hoffmann	Punching Instruction Punch	Page Program I W V O C E
Programmer La Donna Hoffmann	Taken I was a same a	
	Control Card Specifications	
Stertling Ster	Model 20 Model 20	Refer to the specific System Reference Library manual for actual entries. 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
	File Description Specifications	
File Type	Mode of Processing	File Addition/Unordered
Line Filename File Designation End of File Sequence File Format O(1)(1)(1)(2)(2)(3)(3)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)(4)	Length of Key Field or of Record Address Field Record Address Type Type of File Organization or Additional Area of Overflow Indicator of the control of th	Name of Label Exit Core Index Solution Index Core Index Core Index Number of Tracks for Cylinder Overflow Number of Exents Tape Rewind 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 2	Starting X Location X Starting X Location X Starting X Location X Starting X Location X Starting X Star	Continuation Lines K Option Entry 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 FWAMADD VP 90 0 3 FVMVO/CE 0 120	OV PRINTER	
IBM	. International Business Machines Corporation	Form X21-9094 Printed in U.S.A.
		Frinted in U.S.A.
1/10/71	RPG INPUT SPECIFICATIONS	1 2 75 76 77 78 79 80
Date 1/10/71 Program Invoice Preparation	Punching Graphic Instruction Punch	Page Program Identification 75 76 77 78 79 80
Date 1/10/71 Program Invoice Preparation Programmer La Donna Hoffmann	Punching Graphic	Page 62 Program
Program Invoice Preparation Programmer La Donna Hoffmann	Punching Instruction Punch Punch Punch Field Location	Program Identification / NVOCE
Program Invoice Preparation Programmer La Donna Hoffmann	Punching Graphic Instruction Punch Punch	Program Identification // NVOCE
Program Invoice Preparation Programmer La Donna Hoffmann Recomply September 1 Line Filename No. 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 25 27 27 28 29 27 27 28 28 27 28 28 27 28 28	Punching Instruction Punch Punch Punch Field Location	Page 62. Program Identification // NVOCE
Program Invoice Preparation Programmer La Donna Hoffmann Recording to the programmer La Donna Hoffmann Recording to t	Punching Instruction Punch Pun	Program Identification Progra
Program Invoice Preparation Programmer La Donna Hoffmann Rect Comparation Programmer La Donna Hoffmann Programmer La	Punching Instruction Punch Pun	Program Identification Progra
Program Invoice Preparation Programmer La Donna Hoffmann Rect 1	Punching Instruction Punch Pun	Program Identification Progra
Program Invoice Preparation Programmer La Donna Hoffmann Rect I I I I I I I I I	Punching Instruction Punch Pun	Program Identification Progra
Program Invoice Preparation Programmer La Donna Hoffmann Record Invoice Programmer La Donna Hoffmann Invoice Programmer La Donna Hoffmann Invoice Programmer La Donna Hoffmann Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Inv	Punching Instruction Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Position (2) 10 10 10 10 10 10 10 10 10 10 10 10 10	Field Name Field Name Field Name Field Indicators Sterling Sign Position Field Name Field Name Field Indicators Sterling Sign Position Flus Minus or Blank ACCNO L1 MAME ADDR3 Page Page Field Indicators Sterling Sign Position Blank Field Indicators 7 2 70 71 72 73 74
Program Invoice Preparation Programmer La Donna Hoffmann Rect 1 1 1 1 1 1 1 1 1	Punching Instruction Punch Pun	Page 22 Program Identification
Program Invoice Preparation Programmer La Donna Hoffmann Rect Invoice Programmer La Donna Hoffmann Rect Invoice Programmer La Donna Hoffmann Rect Invoice Programmer La Donna Hoffmann Rect Invoice Invoice Invoice Rect Invoice	Punching Instruction Punch Pun	Field Name Field Name Field Indicators Field Indicators Field Indicators Field Sign Position Field Sign Position Field Sign Position Field Indicators Sterling Sign Position Flux Minus or Blank A CCNO L1 NAME ADDR1 ADDR2 ADDR3 SHPCD
Program Invoice Preparation Programmer La Donna Hoffmann Record Invoice Programmer La Donna Hoffmann Invoice Preparation Programmer La Donna Hoffmann Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice	Punching Instruction Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Position Punch Position Punch Position Punch Position Punch Pun	Field Name Field Name Field Indicators Field Indicators Sterling Sign Position Field Name Field Indicators Sterling Sign Position Plus Minus or Blank ACCNO L1 ADDR1 ADDR2 ADDR3 SHPCD
	Punching Instruction Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Punch Position Punch Position Punch Position Punch Position Punch Pun	Field Name Field Name Field Indicators Field Indicators Sterling Sign Position Field Name Field Indicators Sterling Sign Position Plus Minus or Blank ACCNO L1 ADDR1 ADDR2 ADDR3 SHPCD
Program Invoice Preparation Programmer La Donna Hoffmann Record Invoice Programmer La Donna Hoffmann Invoice Preparation Programmer La Donna Hoffmann Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice Preparation Invoice	Punching Instruction Punch Pun	Field Name Field Name Field Indicators Field Indicators Field Indicators Field Sign Position Field Sign Position Field Sign Position Field Indicators Sterling Sign Position Flux Minus or Blank A CCNO L1 NAME ADDR1 ADDR2 ADDR3 SHPCD

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

IBW	International Business Machines Corporation Form X2 Printed in	
Date VIO/71	RPG CALCULATION SPECIFICATIONS 1 2 Punching Graphic Program Identification Iden	
Programmer LaDonna Hoffmann	Punch Punch 1	
Indicators	Resulting Indicators Arithmetic	
Line GES And And Factor 1	Operation Factor 2 Result Field Field Length Secondary Seconda	
Form Ty Control I (LO La, u) Not Not	G Table (Factor 2) is	
o 1 c 1d SHPCD	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	3
0 2 C 20 QTY	MULT UPRICE AMOUNT 72 FIND ITEM TOTAL	AL
0 4 C	ADD TRVICT TRVICT TOTAL	
0 5 C	╿╒ ╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫╫	
0 7 C		
0 8 C	╎┧╏┞┩┦╏┊╎╌┼┼┼┼╏╏┈╏╏┈╏╏╏╏╏ ┼┼ ╏╏╏╏╏╏	+
1 0 C		耳
1	╎╸	\dashv
1 3 C		\Box
1 4 C 1 1 5 C 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		$\pm \parallel$
IBM	International Business Machines Corporation Form X2	
IBM,	RPG OUTPUT - FORMAT SPECIFICATIONS	U.S.A. 79 80
Date 1/10/71	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78	U.S.A. 79 80
*	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78	U.S.A. 79 80
Date 1/10/71 Program Invoice Preparation Programmer La Donna Hoffmann	RPG OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78	U.S.A. 79 80
Program Invoice Preparation Programmer La Donna Hoffmann Space Skip Output Output	Printed in Printed in	U.S.A. 79 80 CE
Date 1/10/71 Program Invoice Preparation Programmer La Donna Hoffmann Line Filename 13/15	Printed in RPG OUTPUT - FORMAT SPECIFICATIONS Punching Graphic Instruction Punch Punch P	U.S.A. 79 80 CE
Date 1/10/71 Program Invoice Preparation Programmer La Donna Hoffmann Line Filename (3/10/13/15/15/15/15/15/15/15/15/15/15/15/15/15/	Printed in Plant Specifications 1 2 Punching Graphic Instruction Punch Punch Sign Punch Sign CR - X = Remove Print Sign CR - Yes Ves 1 A J Y - Date Posts Sign Printed In Plant Sign Printed In Plan	U.S.A. 79 80 CE
Date 1/10/71 Program Invoice Preparation Programmer La Donna Hoffmann Line Filename (1) Space : Skip Outp	Printed in Planching Graphic Instruction Punch P	U.S.A. 79 80 2 E
Date 1/10/71 Program Invoice Preparation Programmer La Donna Hoffmann Line Filename (1) 1/10/11 a 1/10 a	Printed in Page OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 Punching Graphic Punch	U.S.A. 79 80
Date	Printed in Page OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 Punching Graphic Punch	U.S.A. 79 80 2 E
Date	Printed in Program Progr	11 (1) (1) (1) (1) (1) (1) (1) (1) (1) (
Date	Printed in Program Progr	11 (1) (1) (1) (1) (1) (1) (1) (1) (1) (
Date	Printed in Page OUTPUT - FORMAT SPECIFICATIONS 1 2 75 76 77 78 Punching Graphic Punch	10.5.A. 79 80
Date	Printed in RPG OUTPUT - FORMAT SPECIFICATIONS Punching Instruction Graphic Instruction Punch	U.S.A. 79 80 2 E
Date	Printed in Punching Instruction Punch Punc	10.5.A. 79 80
Date	Punching Punching Instruction Punch Pu	10.5.A. 79 80
Date	Punching Punching Instruction Punch Pu	U.S.A. 79 80 2 E
Date	Punching Punching Instruction Punch Pu	U.S.A. 79 80 2 E
Date	Punching Punching Instruction Punch Pu	U.S.A. 79 80 2 E

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

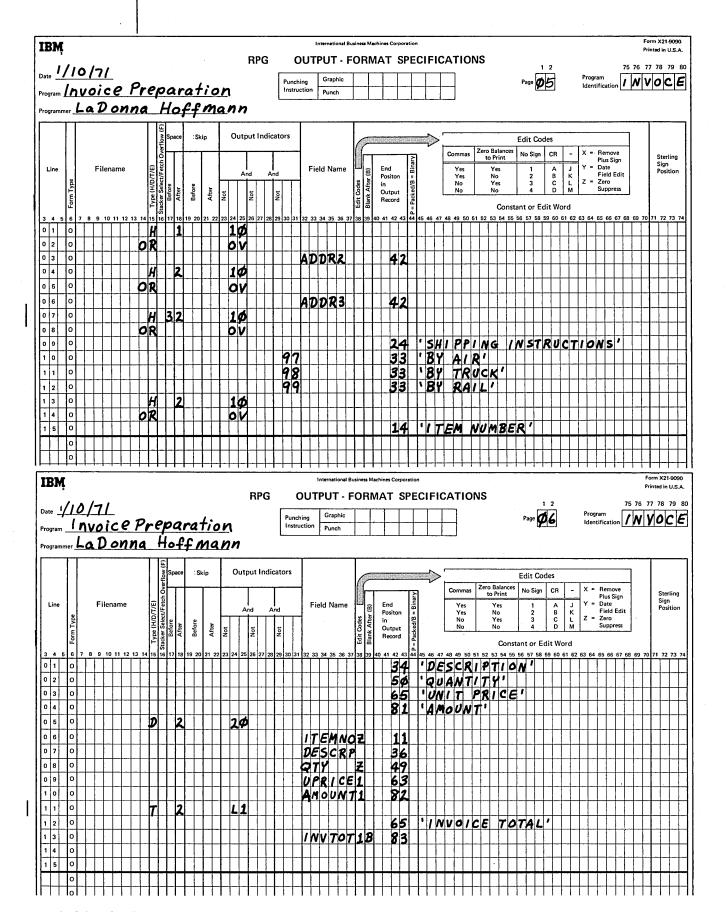


Figure 19 (Part 3 of 3). Specifications for Sample Invoice Program

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:

75 76 77 78 79 80
75 76 77 78 79 80
i cators
Sterling Sign Zero Position or Blank
68 69 70 71 72 73 74
INE CAN
* * *

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:

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

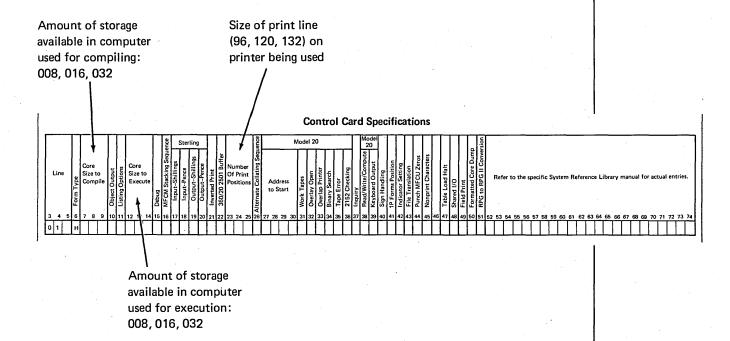
Punching	Graphic				
Instruction	Punch				

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 0 and the number 2 and the letter Z. You may, therefore, devise a graphic symbol that you use for certain letters. Some people use \emptyset for zero, Z 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:



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				_					•												-						ı	nter	natio	nal I	Busi	ness	Mac	hine	Co	peri	tion									_	<u> </u>							_						_	_				I- 9 0	
TDW	RPG										OUTPUT - FORMAT SPECIFICATIONS																Printed in I							U.S	.А.																															
Date																						_	1		_	_	_			_	_	_	_	 T	_		_	_	· ·	1		1								г	1	2				Prog	0531	_		75	5 70 T	6 7	7	78	79	80
	am																										ider			ion	╧			\Box																																
Program _		_											_						,				i	-					31101	_			L	۰		L_			L_	丄	-	J																								
Programm	ner .	_			_			_	_	_																																																								
											į	Now (F)	nac		<u> </u>	Skip		Π	O	ıth	ut I	ndi	icat	ors															7							_	_	_		_	_			_	_		_			_	 1		Γ			٦
	ŀ									1	П	ĒL	pac			SKIP	_	L													1	P							v	⊳ ۱				70	ro E	tala	nce	. T	dit		$\overline{}$		_	_	Ť		_			_	ł		l			-
l	ļ			_							١	ó	l	١												١.						r	T	_			Jary			ļ		mm	_		to	Prir		1	No	_	<u> </u>	CR	_	_	4	x =	Ρ		Sign		l			Ste	rling n	,
Line	l.			Fi	ler	nan	ne			į	, LE	t/Fe		١		1		L		An	d	,	And	Field Nam			ield Name					Positon							Yes Yes			Yes No				1 2			A B	- 1	K	1	Y =	F	Date Field Edit Zero		lit	l			Position					
	Ž	١									/D/H) ed/	Before	١	N I I I	Before		After	Not		-	Not		Not									š :	5	in Ou	itpu	t	ked/B			L		No No		L		es lo				3 4.		D		M					oress	s						- 1
	Forn	l								1	Š.	a ke	۱	١	ĕ		ď	Ž	1	-	Ž		Ž								ŀ			Re	cor	į	= Pac									c	ons	star	nt c	or I	Edi	t V	Vo	rd												-
3 4 5	6	1,								_	_				19 2		1 2	2 23		25	26 2	7 2	B 29	30	31	32	33	34	35	36	37 3	8 3	9 40	0 41	42	43	44	45	46	47	48	49 5	50 5	1 5	2 5										2 6	3 64	6	5 6	6 67	68	3 69	3 70	71	72	73	74
0 1	٥	1	N	V	0	,	(: {		1		1	3	3	Ø.	1	1	L	Ø	1	_	1	1	Ľ	L	L	_			4	1	1	1	1	Ļ		Ц			Ц	_	1	1	1	1	1	1	1	1	1	1	L	1	\perp	1	\perp	1	1	┸	L	1	\perp	L	L	L	Ц
	٥	L		L		L	1	ļ	C	1	R	1	ļ	1	\perp	1	1	1	0	۷	\perp	1	L	L	L	L				_	1	4	1	1	L		Ц	j	Ц	Ц	_		1	1	1	1	1	1	1	1		ļ	1	\perp	1	╧	1	1	_	L	1	\perp	L	L	L	Ц
++	٥	L	L	L		L	1	1	1	1	1	1	Ļ	1	\downarrow	1	1	L		4	4	1	ļ	L		L				4	1	1	1	1	4	5	Ц	•	I	N	V	0	1	: (5 '	1	1	1	1	1	1	ļ	1	\downarrow	4	\perp	1	1	1	\perp	1	\downarrow	L	L	L	Ц
0 4	0	L	L	L	L	L	1	1	1	1		1	3	3	4	1	1	L	Ø	1	4	1	Ļ	L	L	L				4	4	4	1	1	L		Ц		Ш	Ц	_	4	1	1	1	1	1	1	1	ļ	1	Ļ	1	1	4	1	1	1	1	Ļ	\downarrow	1	L		L	Ц
0 5	٥	L		L		L	L	ļ	C	الا	R	1	1	1	\perp	1	1	L	٥		_	1	1	L	L					_	1	1	1	1	L	L	Ц		Ц	Ц		\perp	1	1	1	1	\perp		1	↓	\perp	L	1	\downarrow	1	╧	1	1	\perp	l	1	\perp	L			Ц
0 6	0	L		L		L		L	\perp	1	4	╧	į	2	\perp	1	1	L	Ø	1				L	L	L					1	\perp						-		Ц	_	╛						Ĺ			_			\perp	┙	╧	╧	\perp		L	\perp	Ţ	L	Ŀ		Ц
0 7	0	L		L				L	C)	R		╽	╛	\perp	1	1		0	۷	_[_	L	L					╛	\perp	1	\perp	L		Ш			Ц		\perp	1			ł						1.	1	\perp	╧	╧	╧	╧		L	1		L	·	L	Ш
0 8	0	L	L	L		L	L	ļ	1	1	1	┸	ļ	1	\perp	1	1	⇂	Ш	_	\perp	_	L	L	L	Ŀ				\perp	1	┵	1	1	1	7	Ц	'	N	A	M	E	'	1	1	1	1		1	1	1	1	1	\perp	\perp	\perp	1	1	\downarrow	\perp	1	\perp	L	L		Ц
0 9	0	L	L	L		L	L	1	1	1	1	1	1	1	4	1	1	\perp			1	1	Ļ	L	L	N	A	M	E	4	1	1	1	1	3	8	Ц			Ц		1	_	1	_	1	1	1		L	_	\perp	1	\perp	1	1	ļ	ļ	↓	L	1	퇶	L	L	L	Ц
1 0	0	L	L	L	L		L	1		1	4	1	1	4		1	1	L	Ø		_	_	1	L	L			Ц			1	1	1	1			Ц	Ц		Ц		\perp	\perp	1	1	1			1		1	1	1	\perp	1	\perp	1	\perp	\perp	Ļ	1	\perp	L	L		Ц
1 1	0	L	L	L	L	L	L	L	Ç	2	₹		ļ	1	_	1	1		0	V		1	L	L	L					_	1	\perp		1	L		Ш			Ц		_	1	1	1	1				L			1	\perp	1	╧	1	╧	\perp	L	1	┸	L	L	L	Ц
1 2	0	L	<u>L</u>	L	L			1	1	1	1	\perp	l	1		┸	1	┸	Ц		_		L	L							1	4	1	1	1	φ	Ц	•	Α	D	D	R	€!	S	5	1	1	\perp	1	L	_	L	1	1	\perp	1	1	\downarrow	\perp	1	\perp	\perp	L	L		Ц
1 3	0	L		L		L	L	L	1	1	1	┸	l	1	4	1	1	L	Ц	_	\perp		┸	Ļ	L	A	D	D	R	1	1	1	1	1	4	3	Ц		Ĺ	Ц		1	\perp	1	\perp	1			1		\perp	1		1	_	\perp	1	\perp	1	\perp	\perp	\downarrow	L	L	L	Ц
\rightarrow	٥	┺	L	L	L	L	L	1	1	1	1	4	1	4	4	1	1	L		4	4	1	Ļ	Ļ	L	L				4	4	4	1	1	L	Ľ	Ц	Ц		Ц	4	_	4	1	_	1	1	\downarrow	1	1	1	Ļ	1	4	4	\downarrow	1	1	\perp	1	\downarrow	\downarrow	Ļ		L	Ц
	0	L	L	L	L	L	L	1	1	1	┙	⊥	L	1		┸	1	L	L		_	_	L	L	L	L					_	1	1	_			Ц			Ц	_	_	4	1	1	1	_	1	1			Ļ	\downarrow	┵	┵	╧	1	1	Ļ	╧	┵	╧	L	L	L	Ц
Ø51		L	L	L	L	L	1	1	1	1	1	┸		1	1	1.	1	L	Ц	4	_	1	L	L	L	L		Ц		4	4	1	1	1	-	7	Ц	_	A	C	C	O	21		1	٨	V	1 1	1/2	3 4	= 1	₹′	1	1	1	1	\downarrow	4		\perp	1	1	L	L	L	Ц
Ø52	٥	L	Ļ	L	L	L	1	1	1	1	1	1	1	4	1	1	1	┸	Ц	_	_	\downarrow	Ļ	L		A	C	C	N	O	1	4	1	L	2	4	Ц	Ц		Ц		_	_	1	1	ļ	1	1	\downarrow	1	1	\downarrow	1	1	4	1	\perp	4	\perp	\downarrow	1	1	Ļ	L	L	Ц
	0	L	L	L	L	L	1	1	1	4	Γ		_			_	-												_		_								_	_	_													_			ì	1	1	\downarrow	1	\downarrow	L	Ļ	L	Ц
\coprod	0	L	1	L	L	L	1		\downarrow	╛	l.															L	L	Ц																																						
	o	L		L	L	L				╛														_															_																			1		\perp	\perp	上	L		L	Ш
									<u> </u>	_		1	missing specifications. Notice that columns 3-5 tell where the specifications																																																					
									-			1	be	ele	or	ıg.	٠	Th	e	lir	ne	nı	un	nb	er	's	0	51	a	ın	d I	05	52	ir	١d	ic	at	е	th	nat	t t	:he	e s	зp	ec	if	ic	at	ic	n	s	be	ele	or	١g											
			between lines 05 and 06.																																																															
Ø52	0											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																																																						

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.
- B 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.
- (D) A list of all indicators used in the program.
 - * Note that the sample listing shown is the program listing for the invoice job.

```
0101 +008 008
                                                                                 INVOCE
0001 0102 FNAMADD IP
                             96
                                             MFCU1
                                                                                 INVOCE
0002 0104 FINVOICE 0
                             120
                                             PRINTER
                                                                                 INVUCE
0003 0201 INAMADD 011 10
                                                                                 INVOCE
0004 0202 I
                                                       60ACCNO L1
                                                                                 INVOCE
0005 0203 1
                                                                                 INVUCE
                                                       26 NAME
                                                   27
                                                                                 INVOCE
0006 0204 I
                                                       49 ADDR1
0007 0205 I
                                                                                 INVOCE
                                                   50 72 ADDR2
                                                                                 INVOCE
0008 0206 I
                                                   73
                                                       95 ADDR3
0009 0207 I
                                                   96 960SHPCD
                                                                                 INVOCE
                   02N 20
                                                                                 INVOCE
0010 0208 I
                            1 CT
0011 0209 1
                                                    2
                                                       60ACCNO L1
                                                                                 INVOCE
0012 0210 I
                                                    9 1401TEMNO
                                                                                 INVOCE
0013 0211 I
                                                   15 29 DESCRP
                                                                                 INVOCE
0014 0212 1
                                                   30
                                                       340QTY
                                                                                 INVOCE
0015 0213 1
                                                       392UPRICE
                                                                                 INVOCE
                                                                                 INVOCE
0041 0508 0
                  UΒ
                            ΩV
                                               24 *SHIPPING INSTRUCTIONS*
                                                                                 INVOCE
0042 0509 G
                                               33 *BY AIR*
                                                                                 INVOCE
0043 0510 G
                                   97
                                               33 'BY TRUCK'
0044 0511 0
                                   98
                                                                                 INVOCE
0045 0512 0
                                   99
                                               33 *BY RAIL*
                                                                                 INVOCE
0046 0513 N
                             10
                                                                                 INVOCE
                          DIAGNOSTICS
   ERRUR
             STMT.
             NO.
             *0017*
   ERROR SEVERITY
                                                   TEXT
   NOTE
                   RESULT FIELD LENGTH MAY NOT BE LARGE ENOUGH.
                          TABLES AND FIELDS
                                   DECIMAL LENGTH
POSITIONS IN BYTES
                                                              TYPE
         ADDRESS
                       NAME
         0212
                       NAME
                                               019
                                                          ALPHAMERIC
                       ADDR 1
                                               024
                                                           ALPHAMERIC
         022A
         0241
                       ADDR 2
                                               023
                                                           ALPHAMERIC
         0258
                       ADDR 3
                                               023
                                                           ALPHAMERIC
                                                           ALPHAMERIC
                       DESCRP
                                               015
         0267
                                                             NUMERIC
                                               005
                        ACCNO
                                       0
         026C
                                                              NUMERIC
                                               001
         026C
                        SHPCD
           :
                                                               •
                        "UNIT PRICE"
                                                             CONSTANT
         0303
                                                             CONSTANT
                        * AMOUNT *
         0309
                        'INVOICE TOTAL'
                                                            CONSTANT
         0316
                                                            EDIT CODE
         0310
                        • 0•
                                                                          ...
         0327
                            , 0.
                                                           EDIT CODE
                           ADDRESS OF ASSIGNED INDICATOR
                1ST 2ND 3RD 4TH 5TH 6TH 7TH 8TH
                LR L1 L2 L3 L4 L5 L6 L7
L8 L9 10 20 1P
         0365
         0366
                             99 98 97
         0367
                αv
```

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





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 = W).

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.

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.

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, *, +, ϕ , %). 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 \$\\$\\$57 (\beta\ represents one blank space).

address 1	glossary 127
alphameric constants, rules for 89	
alphameric field 20	half-adjusting 40
AND relationship 104	heading records 23
arithmetic/logic section of computer 2	
antimiotio/10gic bootion of compater 2	indicators 45
coloulation aparations	
calculation operations	input 1
description 35	input devices 1
program cycle operations 35	input/output operations
RPG II specifications 37	description 12
sample job 41	program cycle operations 12
Calculation sheet 36	RPG II specifications 13
coding 4	sample job 31
compilation 5	input record description 16
compiler 5	Input sheet 16
computer	input shoot 10
description 1	last record indicator
input devices 1	definition 64
output devices 1	program cycle operations 65
processing unit 1	RPG II specifications 66
constants	sample job 67
alphameric, rules for 89	LR indicator (see last record indicator)
definition 37	L1-L9 indicators (see control level indicators)
headings, rules for 57	
numeric, rules for 37, 89	machine language 3
control break 46	
Control Card sheet 119	numeric constants, rules for 37, 89
control field 46	numeric field 20
	numeric field 20
control level indicators	04.00.0071.11.4
definition 45	OA-OG, OV indicators (see overflow indicators)
program cycle operations 47	object program 3
RPG II specifications 48	OR relationship 64
sample job 51	output 1
control section of computer 2	output devices 1
	Output-Format sheet 21
detail records 23	output record description 21
diagnostic messages 124	overflow 59
documentation 117	overflow handling 59
	overflow indicators
edit codes 29	definition 59
editing 28	program cycle operations 61
error messages 124	RPG II specifications 62
enormessages 124	-
C 11 0	sample job 67
field 9	overflow line 59
field indicators	overflow page 59
definition 97	
program cycle operations 97	printed reports 25
RPG II specifications 98	printer spacing chart 11
sample job 100	processing unit 1
field names, rules for 18	arithmetic/logic 2
file 9	control 2
File Description sheet 13	storage 1
file names, rules for 14	program 3
first page indicator	
	program cycle 7
definition 55	program listing 122
program cycle operations 56	programming language 3
RPG II specifications 57	programming terms, basic 9
sample job 67	

```
record 9
record description
   input 16
   output 21
record identifying indicators
   definition 71
   program cycle operations 72
   RPG II specifications 73
   sample job 78
Record Layout Form 10
reports, printed 25
resulting indicators
   definition 84
   program cycle operations 84
   RPG II specifications 86
   sample job 91
RPG II program cycle 7
RPG II programming language 3
RPG II specification sheets 4
rounding 40
specification sheet order 118
specifications 4
storage section of computer 1
source program
   compilation 5
   definition 3
   description 3
   translation 5
   writing 3
total records 23
1P indicator (see first page indicator)
01-99 indicators (see field indicators; record identifying indicators;
resulting indicators)
```

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)