Chapter 9
The Edit Instruction

Objectives

Upon completion of this chapter you will be able to:

- Define an edit mask to suppress leading zeroes,
- Define an edit mask to include commas where appropriate,
- Define an edit mask to include a decimal point where appropriate,
- Define an edit mask to include a sign,
- Define an edit mask to print CR for credit or DB for debit,
- Define an edit mask for check protection, and
- Use the edit instruction with a mask to print a packed number in the desired format.

Introduction

In chapter seven we introduced packed decimal arithmetic operations. In that chapter we produced two reports for Cogsworth Industries: the Sales Recap and Inventory Discrepancies reports. Those reports appeared as follows:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345678901234567890123456789012345678901234567890</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COGSWORTH INDUSTRIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales Recap</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>Calif</td>
<td>Ill</td>
<td>Utah</td>
<td>Misc</td>
<td>TOTAL</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>GIZMOS</td>
<td>020</td>
<td>030</td>
<td>020</td>
<td>020</td>
<td>090</td>
</tr>
<tr>
<td>WIDGETS</td>
<td>015</td>
<td>010</td>
<td>010</td>
<td>002</td>
<td>037</td>
</tr>
<tr>
<td>JUNQUE</td>
<td>025</td>
<td>015</td>
<td>015</td>
<td>018</td>
<td>073</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234567890123456789012345678901234567890123456</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COGSWORTH INDUSTRIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inventory Discrepancies Report</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>Begin + Purch - Sales - Expect</td>
<td>Actual</td>
<td>Result</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------</td>
<td>--------</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GIZMOS</td>
<td>017</td>
<td>099</td>
<td>090</td>
<td>026</td>
<td>023</td>
</tr>
<tr>
<td>WIDGETS</td>
<td>022</td>
<td>034</td>
<td>037</td>
<td>019</td>
<td>019</td>
</tr>
<tr>
<td>JUNQUE</td>
<td>030</td>
<td>052</td>
<td>073</td>
<td>009</td>
<td>010</td>
</tr>
</tbody>
</table>

003 records processed.
001 indicate shortage.
001 indicate overage.

We used the UNPK operation to move the results of an arithmetic operation to an output field, and we used the MVZ operation to remove the sign. (We saw that a signed number would sometimes print as a letter.) Recall from our discussion in that chapter that there are two problems with this technique:

- leading zeroes are not suppressed, and
In this chapter we introduce the edit instruction, which will enable us to suppress leading zeroes and print a sign, as well as to perform all of the formatting operations which you have seen in other languages, such as inserting commas, check protection, etc.
CHAPTER 9
THE EDIT INSTRUCTION

9.3

Problems with UNPK and MVZ Revisited

Assume we are given FLDA defined as PL4, with a value of +123,456:

```
FLDA DC PL4'123456'
```

To print this field, we could use UNPK and MVZ as follows:

```
UNPK WK7,FLDA
MVZ WK7+6(1),=X'F0'
```

But what if FLDA had been negative? We would have lost the sign, as follows:

```
FLDA DC PL4'-123456'
```

```
UNPK WK7,FLDA
MVZ WK7+6(1),=X'F0'
```

In both cases, WK7 has the same value after the UNPK and MVZ.

Similarly, what if FLDA represented dollars and cents; that is, what if we wanted to see 1234.56?

Given the field definitions:

```
AMOUNT   DS 0CL8
DOLLARS  DS CL5
DECIMAL  DS CL1
CENTS    DS CL2
```

...we could do the following:

```
MVC DOLLARS,WK7
MV1 DECIMAL,'.'
MVC CENTS,WK7+5
```

But we still have the problem with the sign. And what about suppressing leading zeroes? And inserting commas? To accomplish these things in BAL, we use the edit (ED) instruction. This gives us the same capabilities found in other languages. For example, in COBOL we could code:

```
WORKING-STORAGE SECTION.
  01 MISC.
    05 FLDA PIC S9(5)V99 COMP-3 VALUE +1234.56.
    05 PRTA  PIC ZZ,ZZZ.99.

PROCEDURE DIVISION.
  MOVE FLDA TO PRTA.
```
...and in BASIC we could code:

10 LET FLDA = 1234.56
20 LET MASK$ = "##,###.##"
30 PRINT USING MASK$; FLDA

In both cases, the result is "1,234.56".

**Defining the Edit Mask**

There are three steps to editing a number:
1. Define an edit mask,
2. Move the mask to the target field, and
3. Edit a packed number over that target field.

To edit a number, you **must** define a mask indicating the desired format. The first byte of the mask will always contain a fill character indicating the character with which leading zeroes will be replaced. This will usually be a blank (X'40') but may instead be an asterisk (X'5C') for check protection. The fill character is followed by as many X'20's as there are digits in the packed number being edited. We continue with the previous example. Given FLDA defined as PL4, with a value of +123,456:

```
FLDA  DC  PL4'123456'
```

We want to print this field with leading zeroes suppressed. Given the following field definition:

```
MASK  DC  X'4020202020202020'
```

...we code the following:

```
WK8
MVC   WK8,MASK
ED    WK8,FLDA
```

(Recall that when I use a label of the form WKn, I am implying that a work field of n bytes in length has been defined; for example, WK8 DS CL8.)

Notice that the mask has a fill character (X'40') indicating that all leading zeroes are to be replaced with blanks. Also, there are seven X'20's, because a field defined as PL4 has seven digits. Hence the mask is a total of eight bytes in length. Of course, I could have coded the following instead:

```
MVC   WK8,-X'4020202020202020'
ED    WK8,FLDA
```
The literal (\texttt{=X'4020202020202020'}) would then appear after the \texttt{LTORG} in the assembly listing. Of course, if \texttt{FLDA} had a value of zero, \texttt{WK8} would be all blanks. For example, given \texttt{FLDB} and \texttt{MASK2} defined as follows:

\begin{verbatim}
FLDB DC PL3'0'
MASK2 DC X'402020202020'
\end{verbatim}

...we code the following:

\begin{verbatim}
  MVC WK6,MASK2
   40 20 20 20 20
  ED WK6,FLDB
   40 40 40 40 40
\end{verbatim}

Note that the mask must be moved to the output area each time a number will be edited because the edit instruction destroys the mask. Finally, the edit instruction is an SS-type instruction similar to the MVC; that is, the length of the edit is determined by the length of the receiving field only. The edit continues for as many digits as are represented in the receiving field.

\textbf{You Try It...}

Given \texttt{X DC PL3'1234'} and \texttt{MASK DC XL6'402020202020'}, show the value of \texttt{WK6} or \texttt{WK4} after each of the following:

1. \texttt{UNPK WK6,X}
2. \texttt{UNPK WK6,X}
   \texttt{MVZ WK6+5(1),=X'0F0'}
3. \texttt{MVC WK6,MASK}
   \texttt{ED WK6,X}
4. \texttt{MVC WK4,MASK} \quad \textit{Careful!}
   \texttt{ED WK4,X+1}

\textbf{Indicating Significance}

If we want to stop suppression of leading zeroes, so as to force at least one zero to print, we \textit{replace a single X'20' with a X'21'}. The X'21' is the last digit which will be replaced by the fill character. In other words, \textit{zeros to the right of the X'21' will be printed}. For example, given:

\begin{verbatim}
FLDB DC PL3'0'
MASK3 DC X'4020202020202020'
\end{verbatim}

...we code the following:

\begin{verbatim}
  MVC WK6,MASK3
   40 20 20 20 21 20
  ED WK6,FLDB
   40 40 40 40 40 F0
\end{verbatim}
CHAPTER 9
THE EDIT INSTRUCTION

Note: the total number of X'20's and X'21's will always be odd!

You Try It...

Given PJS DC PL3'-49' write the instruction(s) to move PJS to...

5. ...WK6 such that WK6 will be C'kkkkk049'.
6. ...WK6 such that WK6 will be C'kkkkk0049'.
7. ...WK4 such that WK4 will be C'kkkkk49'.
8. ...WK4 such that WK4 will be C'004R'. (Hint: C'R'=X'D9')

Printing Decimal Points

What about decimal points? We didn't even mention decimal points in our discussion of packed decimal arithmetic. That's because there aren't any in BAL; that is, all arithmetic in BAL is integer arithmetic. It's up to you as the programmer to keep track of where the implied decimal is. This complicates any arithmetic (as we will see in a later chapter) and formatting. To print a decimal point, we add a period (X'4B') in the appropriate position within the mask. For example, given our earlier definitions for FLDA and FLDB:

```
MVC   WK9,=X'4020202020214B2020'
ED    WK9,FLDA
MVC   WK7,=X'402020214B2020'
ED    WK7,FLDB
```

...gives WK9 equal to C'kkkkk1234.56' and WK7 equal to C'kkkkk00.00'. Note that the masks are getting longer....

How you will show zeroes when printing dollars and cents is a matter of personal preference. For example, you may prefer to have the dollars portion blank (as in WK7 above), or you may prefer to show one zero in the dollars portion. In the latter case, the mask would be changed with the X'21' moved one position to the left. For example:

```
MVC   WK7,=X'402021204B2020'
ED    WK7,FLDB
```

...gives WK7 equal to C'kkkkk00.00'.

You Try It...

Given H DC PL3'6' write the instruction(s) to move H to...

9. ...WK7 such that WK7 will be C'kkkkk6'.
10. ...WK7 such that WK7 will be C'kkkkk00.6'.
11. ...WK7 such that WK7 will be C'kkkkk0.06'.
Printing Commas

In the same way that we add periods to the output, we can add commas to the output. To print a comma, we add a comma (X'6B') in the appropriate position(s) within the mask. Continuing with our earlier example:

```
MVC WK10,-X'4020206B2020214B2020'
ED WK10,FLDA
```

...gives WK10 equal to C'bb1,234.56'. Note that our target field and mask have grown from eight bytes to ten bytes! Note also that the total number of X'20's and X'21's is still odd (seven): one for each digit in FLDA.

You Try It...

Given W DC PL5'6301982' write the instruction(s) to move W to...

12. ...WK10 such that WK10 will be C'bb6301982'.
13. ...WK12 such that WK12 will be C'bb63,019,982'.
14. ...WK12 such that WK12 will be C'bb63,019.82'.
15. ...WK12 such that WK12 will be C'bb63,019.82'.
16. ...WK7 such that WK7 will be C'bb1,982'. *(this one is tricky!)*

Given C DC PL4'72384' write the instruction(s) to move C to...

17. ...WK9 such that WK9 will be C'bb72,384'.
18. ...WK10 such that WK10 will be C'bb7,238.4'.

Printing the Sign

Consider the following example. Given:

```
POS DC PL3'+123'          A positive number
NEG DC PL3'-123'          A negative number
MASK4 DC X'402021204B2020'
```

...if we code the following instructions we get the same results:

```
MVC WK7,MASK4   WK7
ED WK7,POS       40 20 21 20 4B 20 20
MVC WK7,MASK4   WK7
ED WK7,NEG       40 20 21 20 4B 20 20
```

Copyright © 2009 by Bill Qualls – All Rights Reserved
We see that both \texttt{POS} and \texttt{NEG} will be printed as \texttt{C'bbb1.23'}. The \textit{ed} instruction removes the sign. To correct this problem, we can add a hyphen (\texttt{X'60'}) to the end of the mask. This hyphen will be replaced by the fill character if the number is not negative. For example, given:

\begin{verbatim}
MASK5 DC X'402021204B202060'
\end{verbatim}

...if we code the following instructions we get different (correct) results:

\begin{verbatim}
WK8
MVC WK8,MASK5
ED WK8,POS
MVC WK8,MASK5
ED WK8,NEG
\end{verbatim}

We see that \texttt{POS} will be printed as \texttt{C'bbb1.23b'} whereas \texttt{NEG} will be printed as \texttt{C'bbb1.23-'} (The \texttt{EDMK} instruction can be used to print a leading sign as opposed to a trailing sign: it will be discussed in a later chapter.)

Note that when I changed the mask, I had to increase the length of the receiving field (I used \texttt{WK8} instead of \texttt{WK7}). Failure to do so is the source of many programming errors for beginning BAL programmers. Remember: the length of the receiving field must be the same as the length of the mask, and that mask must have as many \texttt{X'20'}s and \texttt{X'21'}s as there are digits in the packed field being edited!

\textbf{You Try It...}

Given \texttt{E DC PL2'-4'} write the instruction(s) to move \texttt{E} to...

19. ...\texttt{WK5} such that \texttt{WK5} will be \texttt{C'bb4-'}.
20. ...\texttt{WK6} such that \texttt{WK6} will be \texttt{C'bb.04-'}.

\textbf{Printing CR (credit) or DB (debit)}

Similarly, if we are working on an accounting application, we can add \texttt{CR} (\texttt{X'C3D9'}) or \texttt{DB} (\texttt{X'C4C2'}) to our mask to indicate a \textbf{CREDIT} or \textbf{DEBIT} respectively. The \texttt{CR} (or \texttt{DB}) will be printed if the number is negative, otherwise it is replaced by the fill character. For example, given:

\begin{verbatim}
POS DC PL3'+123'
NEG DC PL3'-123'
CR DC X'402021204B2020C3D9'
DB DC X'402021204B2020C4C2'
\end{verbatim}

...the following instructions will give the results indicated:
CHAPTER 9  
THE EDIT INSTRUCTION

You Try It...

Given \( B \ DC \ PL2'{-38}' \), write the instruction(s) to move \( B \) to...

21. ...WK6 such that WK6 will be \( C'{38DB}' \).
22. ...WK7 such that WK7 will be \( C'{38CR}' \).
23. ...WK7 such that WK7 will be \( C'{3.8bb}' \).

Check Protection

Thus far we have used a blank as the fill character in all of our masks. For check protection, we can use an asterisk (\( X'5C' \)). For example, given our earlier definitions of \( FLDA \) and \( FLDB \), and the following definitions for \( CHKA \) and \( CHKB \):

\[
\begin{align*}
CHKA & \ DC \ X'5C20206B2021204B2020' \\
CHKB & \ DC \ X'5C2021204B2020'
\end{align*}
\]

...the following instructions will give the results indicated:

You Try It...

Given \( DUE \ DC \ PL4'6591' \), write the instruction(s) to move \( DUE \) to...

24. ...WK10 such that WK10 will be \( C'**6,591' \).
25. ...WK9 such that WK9 will be \( C'**6,591' \).
26. ...WK11 such that WK11 will be \( C'**65.91b' \).
Documenting the Edit Mask

To simplify maintenance of the program, I like to "document" the print masks by showing a character representation of the hex fields used in the mask. I use a B to indicate a blank as the leading fill character, a Z to indicate digit positions where leading zeroes will be suppressed, and a 9 to indicate digit positions where leading zeroes will not be suppressed. For example, I would document the print masks which we have used thus far as:

```
MASK     DC    X'4020202020202020'       BZZZZZZZ
MASK2    DC    X'402020202020'           BZZZZZ
MASK3    DC    X'402021204B2020'         BZZ9.99
MASK4    DC    X'402021204B202060'       BZZ9.99-
CR       DC    X'402021204B2020C3D9'     BZZ9.99CR
DB       DC    X'402021204B2020C4C2'     BZZ9.99DB
CHKA     DC    X'5C20206B2021204B2020'   ***,**9.99
CHKB     DC    X'5C2021204B2020'         ***9.99
```

With the exception of the leading B, COBOL programmers will recognize this notation as that used in COBOL's PIC clauses. Henceforth, I will use this same notation in the print specifications for all exercises.

Sample Program

The following program, EDITS.MLC, illustrates the examples we have used thus far. The WTO (write to operator) macro was used to show the results of the edits. In addition to illustrating the use of the ED instruction, I hope this program illustrates how you can use the WTO command to experiment with this and other instructions as you attempt to learn BAL.

```
PRINT NOGEN
****************************************************************
*                  FILENAME: EDITS9.MLC                        *
*                  AUTHOR  : Bill Qualls                      *
*                  SYSTEM  : PC/370 R4.2                      *
*                  REMARKS : Demonstrate the edit instruction by *
                      * implementing examples shown in chapter 9. *
****************************************************************
START 0
BEGIN    BEGIN
****************************************************************
WTO   'SEE PAGE 9.3'
****************************************************************
MVC   WK8,MASK
ED    WK8,FLDA
WTO   WK8
MVC   WK8,=X'402020202020202020'
ED    WK8,FLDA
WTO   WK8
```

(continued)
CHAPTER 9
THE EDIT INSTRUCTION

****************************************************************
WTO   'SEE PAGE 9.4'
****************************************************************
MVC   WK6, MASK2
ED    WK6, FLDK
WTO   WK6
MVC   WK6, MASK3
ED    WK6, FLDK
WTO   WK6
****************************************************************
WTO   'SEE PAGE 9.5'
****************************************************************
MVC   WK9, =X'4020202020214B202020214B2020'
ED    WK9, FLDA
WTO   WK9
MVC   WK7, =X'402020214B202020214B2020'
ED    WK7, FLDK
WTO   WK7
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
(continued)
CHAPTER 9
THE EDIT INSTRUCTION

MVC WK7,CHKB
ED WK7,FLDB
WTO WK7

WTO 'ALL DONE...
RETURN

* Literals, if any, will go here
* LTORG

* Other field definitions

WK6  DS  CL6
WK7  DS  CL7
WK8  DS  CL8
WK9  DS  CL9
WK10 DS  CL10
FLDA DC  PL4'123456'
FLDB DC  PL3'0'
POS DC  PL3'+123'
NEG DC  PL3'123'

MASK DC  X'4020202020202020'  BZZZZZZZ
MASK2 DC  X'4020202020202020' BZZZZZ
MASK3 DC  X'4020202021202020' BZZZ9
MASK4 DC  X'402021204B202020' BZZ9.99

CR DC  X'402021204B2020C3D9' BZZ9.99CR

DB DC  X'402021204B2020C4C2' BZZ9.99DB

CHKA DC  X'5C20202021204B2020' ***9.99

CHKB DC  X'5C2021204B2020' ***9.99

END  BEGIN

A:\MIN>edits9
SEE PAGE 9.3
123456
123456
SEE PAGE 9.4

0
SEE PAGE 9.5
1234.56
0.00
0.00
SEE PAGE 9.6
1,234.56
1.23
1.23
SEE PAGE 9.7
1.23
1.23
SEE PAGE 9.8
1.23
1.23CR
1.23
1.23DB
**1,234.56
**0.00
ALL DONE...

Copyright © 2009 by Bill Qualls – All Rights Reserved
We now return to the programming problems which motivated this discussion: Cogsworth's Sales Recap and Inventory Discrepancies reports.

The following print layout for the Sales Recap has been modified to use edited output. Recall that a layout field of \texttt{BZZ9} corresponds to a mask of \texttt{X'40202120'}. These amount fields were previously defined as three bytes long. Now they are four bytes long so the output record definition will need to be changed accordingly.

Previously we used \texttt{MVC} (only) to move the sales by state to their respective output fields. We packed those fields only so we could total them. Now, these packed fields will be used so we can display the output with leading zeroes suppressed:

\begin{verbatim}
PACK WCALIF,ICALIF      Each product's sales must
PACK WILL,IILL           be packed so they can be
PACK WUTAH,IUTAH          added to total for this
PACK WWISC,IWISC

MVC   OCALIF,WMASK
ED    OCALIF,WCALIF
MVC   OILL,WMASK
ED    OILL,WILL
MVC   OUTAH,WMASK
ED    OUTAH,WUTAH
MVC   OWISC,WMASK
ED    OWISC,WWISC
\end{verbatim}

where:

\begin{verbatim}
WMASK   DC   X'40202120'   BZZ9
\end{verbatim}

Similarly, the record count will be shown as follows:

\begin{verbatim}
MVC   OREC(23),=CL23'BZZ9 records processed.'
MVC   OREC(4),WMASK
ED    OREC(4),#IN    Count
\end{verbatim}

The complete program and its output follow.
CHAPTER 9  9.14
THE EDIT INSTRUCTION

****************************************************************
*        FILENAME:  COGS9A.MLC                                 *
*        AUTHOR  :  Bill Qualls                                *
*        SYSTEM  :  PC/370 R4.2                                *
*        REMARKS :  Produce report for COGSWORTH INDUSTRIES    *
*                   showing sales by state.                    *
*                   Modify COGS7A.MLC to use ED instruction.   *
****************************************************************

START 0
REGS
BEGIN    BEGIN
WTO   'COGS9A ... Begin execution'
BAL   R10,SETUP
MAIN     EQU   *
CLI   EOFSW,C'Y'
BE    EOJ
BAL   R10,PROCESS
B     MAIN
EOJ      EQU   *
BAL   R10,WRAPUP
WTO   'COGS9A ... Normal end of program'
RETURN
****************************************************************
*        SETUP - Those things which happen one time only,      *
*                before any records are processed.             *
****************************************************************
SETUP    EQU   *
ST    R10,SVSETUP
OI    INVENTRY+10,X'08'  PC/370 ONLY - Convert all
*                                 input from ASCII to EBCDIC
OI    REPORT+10,X'08'    PC/370 ONLY - Convert all
*                                 output from EBCDIC to ASCII
OPEN  INVENTRY
OPEN  REPORT
BAL   R10,HDGS
BAL   R10,READ
L     R10,SVSETUP
BR    R10
****************************************************************
*        HDGS - Print headings.                                *
****************************************************************
HDGS     EQU   *
ST    R10,SVHDGS
PUT   REPORT,HD1
PUT   REPORT,HD2
PUT   REPORT,HD3
PUT   REPORT,HD4
PUT   REPORT,HD5
L     R10,SVHDGS
BR    R10
****************************************************************
*        PROCESS - Those things which happen once per record.  *
****************************************************************
PROCESS  EQU   *
ST    R10,SVPROC
BAL   R10,FORMAT
BAL   R10,WRITE
BAL   R10,READ
L     R10,SVPROC
BR    R10
(continued)
**READ** - Read a record.

```
* READ EQU *
ST R10,SVREAD
GET INVENTORY,IREC Read a single product record
AP #IN,'P'1' Increment record count
B READX
* ATEND EQU *
MVI EOFSW,C'Y'
READX EQU *
L R10,SVREAD
BR R10
```

**FORMAT** - Format a single detail line.

```
* FORMAT EQU *
ST R10,SVFORM
MVC OREC,BLANKS
MVC ODESC,IDESC
PACK WCALIF,ICALIF Each product's sales must
PACK WILL,IILL be packed so they can be
PACK WUTAH,IUTAH added to total for this
PACK WWISC,IWISC product...
MVC OCALLIF,WMASK
ED OCALLIF,WCALIF
ED OCALLIF,WCALIF
ED OILL,WMASK
ED OILL,WILL
MVC OUTAH,WMASK
ED OUTAH,WUTAH
MVC OWISC,WMASK
ED OWISC,WWISC
ZAP WTOTAL,'0'
AP WTOTAL,WCALIF Initialize the total to zero
AP WTOTAL,WILL and start adding...
AP WTOTAL,WUTAH
AP WTOTAL,WWISC
MVC OTOTAL,WMASK
ED OTOTAL,WTOTAL
MVC OCRLF,WCRLF PC/370 only.
L R10,SVFORM
BR R10
```

**WRITE** - Write a single detail line.

```
* WRITE EQU *
ST R10,SVWRITE
PUT REPORT,OREC Write report line
L R10,SVWRITE
BR R10
```

**WRAPUP** - Those things which happen one time only, after all records have been processed.

```
* WRAPUP EQU *
ST R10,SVWRAP
MVC OREC,BLANKS
MVC OCRLF,WCRLF PC/370 only.
BAL R10,WRITE Skip a line.
(continued)
CHAPTER 9
THE EDIT INSTRUCTION

MVC OREC(23),=CL23'BZZ9 records processed.'
MVC OREC(4),WMASK
ED OREC(4),#IN   Count
BAL R10,WRITE
CLOSE INVENTRY
CLOSE REPORT
WTO 'COGS9A ... Sales recap on REPORT.TXT'
L   R10,SVWRAP
BR   R10
********************************************************************************
*        Literals, if any, will go here                                        *
********************************************************************************
LTORG
********************************************************************************
*        File definitions                                                      *
********************************************************************************
INVENTRY DCB   LRECL=41,RECFM=F,MACRF=G,EODAD=ATEND,
           DDNAME='COGS.DAT'
REPORT DCB   LRECL=62,RECFM=F,MACRF=P,
           DDNAME='REPORT.TXT'
********************************************************************************
*        RETURN ADDRESSES                                                      *
********************************************************************************
SVSETUP  DC    F'0'             SETUP
SVHDGS   DC    F'0'             HDGS
SVPROC   DC    F'0'             PROCESS
SVREAD   DC    F'0'             READ
SVFORM   DC    F'0'             FORMAT
SVWRITE  DC    F'0'             WRITE
SVWRAP   DC    F'0'             WRAPUP
********************************************************************************
*        Miscellaneous field definitions                                       *
********************************************************************************
WCRLF    DC    X'0D25'        PC/370 ONLY - EBCDIC CR/LF
EOFSW    DC    CL1'N'        End of file? (Y/N)
BLANKS   DC    CL62' '       Blank character
WCALIF   DC    PL2'0'       Units sold in Calif
WILL     DC    PL2'0'       Units sold in Illinois
WUTAH    DC    PL2'0'       Units sold in Utah
WWISC    DC    PL2'0'       Units sold in Wisconsin
WTOTAL   DC    PL2'0'       Units sold in all states
#IN      DC    PL2'0'       Input record count
WMASK    DC    X'40202120' BZZ9
********************************************************************************
*        Input record definition                                              *
********************************************************************************
IREC     DS    0CL41        1-41 Inventory record
IDESC    DS    CL10         1-10 Product description
ICALIF   DS    CL3         11-13 Units sold in Calif
IILL     DS    CL3         14-16 Units sold in Illinois
IUTAH    DS    CL3         17-19 Units sold in Utah
IWISC    DS    CL3         20-22 Units sold in Wisconsin
IBEGIN   DS    CL3         23-25 Beginning inventory
IPURCH   DS    CL3         26-28 Purchases throughout year
IQOH     DS    CL3         29-31 Actual quantity on hand
ICOST    DS    CL4         32-35 Cost (each) 99V99
ISELL    DS    CL4         36-39 Sell for (each) 99V99
ICRLF    DS    CL2         40-41 PC/370 only - CR/LF
********************************************************************************
*        Output (line) definition                                             *
********************************************************************************
(continued)
Cogsworth Reports Revisited: The Inventory Discrepancies Report

The new Inventory Discrepancies report will appear as follows:
COGSWORTH INDUSTRIES
Inventory Discrepancies Report

Product | Begin + Purch - Sales - Expect | Actual | Diff
---------- | ----------------------------- | ------ | -----
XXXXXXXXXX | BZZ9 BZZ9 BZZ9 BZZ9 BZZ9 BZZZ- |
XXXXXXXXXX | BZZ9 BZZ9 BZZ9 BZZ9 BZZ9 BZZZ- |
XXXXXXXXXX | BZZ9 BZZ9 BZZ9 BZZ9 BZZ9 BZZZ- |

BZZ9 records processed.
BZZ9 indicate shortage.
BZZ9 indicate overage.

The complete program and its output follow.

```assembly
PRINT NOGEN
****************************************************************
*        FILENAME:  COGS9B.MLC                                 *
*        AUTHOR  :  Bill Qualls                                *
*        SYSTEM  :  PC/370 R4.2                                *
*        REMARKS :  Produce report for COGSWORTH INDUSTRIES    *
*                   showing inventory discrepancies.            *
*                   Modify COGS7B.MLC to use ED instruction.    *
****************************************************************
START 0
REGS
BEGIN

WTO 'COGS9B ... Begin execution'
BAL R10,SETUP
MAIN EQU *
CLI EOFSW,C'Y'
BE EOJ
BAL R10,PROCESS
B MAIN
EOJ EQU *
BAL R10,WRAPUP

WTO 'COGS9B ... Normal end of program'
RETURN
****************************************************************
*        SETUP - Those things which happen one time only,      *
*                before any records are processed.               *
****************************************************************
SETUP EQU *
ST R10,SVSETUP
OI INVENTORY+10,X'08' PC/370 ONLY - Convert all
    input from ASCII to EBCDIC
OI REPORT+10,X'08' PC/370 ONLY - Convert all
    output from EBCDIC to ASCII
OPEN INVENTORY
OPEN REPORT
BAL R10,HDGS
BAL R10,READ
L R10,SVSETUP
BR R10
```

(continued)
* HDGS - Print headings.
* **************************************************************************
HDGS EQU *
ST R10,SVHDGS
PUT REPORT,HD1
PUT REPORT,HD2
PUT REPORT,HD3
PUT REPORT,HD4
PUT REPORT,HD5
L R10,SVHDGS
BR R10
******************************************************************************
* PROCESS - Those things which happen once per record. *
* **************************************************************************
PROCESS EQU *
ST R10,SVPROC
BAL R10,FORMAT
BAL R10,WRITE
BAL R10,READ
L R10,SVPROC
BR R10
******************************************************************************
* READ - Read a record. *
* **************************************************************************
READ EQU *
ST R10,SVREAD
GET INVENTRY,IREC Read a single product record
AP #IN,=P'1'
B READX
ATEND EQU *
MVI EOFSW,C'Y'
READX EQU *
L R10,SVREAD
BR R10
******************************************************************************
* FORMAT - Format a single detail line. *
* **************************************************************************
FORMAT EQU *
ST R10,SVFORM
MVC OREC,BLANKS
MVC ODESC,IDESC Description
PACK WBEGIN,IBEGIN Beginning inventory
MVC OBEGIN,WMASK
ED OBEGIN,WBEGIN
PACK WPURCH,IPURCH Purchases
MVC OPURCH,WMASK
ED OPURCH,WPURCH
PACK WCALIF,ICALIF Each product's sales must be packed so they can be added to total for this product...
PACK WWISC,IWISC
ZAP WTOTAL,-P'0' Initialize the total to zero
AP WTOTAL,WCALIF and start adding...
AP WTOTAL,WILL
AP WTOTAL,WUTAH
AP WTOTAL,WWISC
MVC OSALES,WMASK
ED OSALES,WTOTAL

(continued)
CHAPTER 9
THE EDIT INSTRUCTION

9.20

ZAP WENDING,WBEGIN Ending inventory =
AP WENDING,WPURCH Beginning + Purchases
SP WENDING,WTOTAL = Sales

MVC ENDING,WMASK
ED ENDING,ENDING
PACK WQOH,IQOH Actual ending inventory

MVC QQOH,WMASK
ED QQOH,WQOH

MVC QCRLF,WCRLE PC/370 only.
CP WQOH,ENDING Compare actual vs. expected
BE DODIFF
BL SHORT
AP #OVER,=P'1' Count overages
B DODIFF

SHORT EQU *
AP #SHORT,=P'1' Count shortages
DODIFF EQU *
ZAP WDIFF,ENDING Difference = Expected - Actual
SP WDIFF,WQOH

MVC ODIFF,WMASK2
ED ODIFF,WDIFF

FORMATX EQU *
L R10,SVFORM
BR R10

WRITE - Write a single detail line.
*
WRITE EQU *
ST R10,SVWRITE
PUT REPORT,OREC Write report line
L R10,SVWRITE
BR R10

WRAPUP EQU *
ST R10,SVWRAP
MVC OREC,BLANKS
MVC QCRLF,WCRLE PC/370 only.
BAL R10,WRITE Skip a line.

MVC OREC(23),=CL23'BZZ9 records processed.'
MVC OREC(4),WMASK
ED OREC(4),#IN Count all
BAL R10,WRITE

MVC OREC(23),=CL23'BZZ9 indicate shortage.'
MVC OREC(4),WMASK
ED OREC(4),#SHORT Count shortages
BAL R10,WRITE

MVC OREC(23),=CL23'BZZ9 indicate overage.'
MVC OREC(4),WMASK
ED OREC(4),#OVER Count overages
BAL R10,WRITE
CLOSE INVENTORY
CLOSE REPORT

WTO 'COGS9B ... Discrepancies report on REPORT.TXT'
L R10,SVWRAP
BR R10

(continued)
**CHAPTER 9**  
**THE EDIT INSTRUCTION**

* Literals, if any, will go here

**LTORG**

* File definitions

**INVENTORY** DCB  LRECL=41,RECFM=F,MACRF=G,EODAD=ATEND,
DDNAME='COGS.DAT'

**REPORT** DCB LRECL=67,RECFM=F,MACRF=P,
DDNAME='REPORT.TXT'

* RETURN ADDRESSES

**SVSETUP** DC '0' SETUP
**SVHDGS** DC '0' HDGS
**SVPROC** DC '0' PROCESS
**SVREAD** DC '0' READ
**SVFORM** DC '0' FORMAT
**SVWRITE** DC '0' WRITE
**SVWRAP** DC '0' WRAPUP

* Miscellaneous field definitions

**WCRLF** DC X'0D25' PC/370 ONLY - EBCDIC CR/LF
**EOFSW** DC CL1'N' End of file? (Y/N)
**BLANKS** DC CL67' '
**WCALIF** DC CL3 Units sold in Calif
**WILL** DC CL3 Units sold in Illinois
**WUTAH** DC CL3 Units sold in Utah
**WWISC** DC CL3 Units sold in Wisconsin
**WTOTAL** DC CL3 Units sold in all states
**WBEGIN** DC CL3 Beginning inventory
**IPURCH** DC CL3 Purchases throughout year
**IQOH** DC CL3 Actual quantity on hand
**IBEGIN** DC CL3 Beginning inventory
**IN** DC CL2 Input record count
**OVER** DC CL2 Records showing overage
**SHORT** DC CL2 Records showing shortage

**WMASK** DC X'40202120' BZZ9
**WMASK2** DC X'4020202060' BZZZ-

* Input record definition

**IREC** DS OCl41 Inventory record
**IDESC** DS CL10 Product description
**ICALIF** DS CL3 Units sold in Calif
**IILL** DS CL3 Units sold in Illinois
**IUTAH** DS CL3 Units sold in Utah
**IWISC** DS CL3 Units sold in Wisconsin
**IBEGIN** DS CL3 Beginning inventory
**IPURCH** DS CL3 Purchases throughout year
**IQOH** DS CL3 Actual quantity on hand
**ICOST** DS CL4 Cost (each) 99V99
**ISELL** DS CL4 Sell for (each) 99V99
**ICRLF** DS CL2 PC/370 only - CR/LF

* Output (line) definition

(continued)
CHAPTER 9
THE EDIT INSTRUCTION

9.22

THE EDIT INSTRUCTION

<table>
<thead>
<tr>
<th>OREC</th>
<th>DS</th>
<th>0CL67</th>
<th>1-67</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODESC</td>
<td>DS</td>
<td>CL10</td>
<td>1-10</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>CL3</td>
<td>11-13</td>
</tr>
<tr>
<td>OBEGIN</td>
<td>DS</td>
<td>CL4</td>
<td>14-17</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>CL4</td>
<td>18-21</td>
</tr>
<tr>
<td>OPURCH</td>
<td>DS</td>
<td>CL4</td>
<td>22-25</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>CL4</td>
<td>26-29</td>
</tr>
<tr>
<td>OSALES</td>
<td>DS</td>
<td>CL4</td>
<td>30-33</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>CL5</td>
<td>34-38</td>
</tr>
<tr>
<td>OENDING</td>
<td>DS</td>
<td>CL4</td>
<td>39-42</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>CL4</td>
<td>43-46</td>
</tr>
<tr>
<td>OQOH</td>
<td>DS</td>
<td>CL4</td>
<td>47-50</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>CL4</td>
<td>51-54</td>
</tr>
<tr>
<td>ODIFF</td>
<td>DS</td>
<td>CL5</td>
<td>55-59</td>
</tr>
<tr>
<td></td>
<td>DS</td>
<td>CL6</td>
<td>60-65</td>
</tr>
<tr>
<td>OCRLF</td>
<td>DS</td>
<td>CL2</td>
<td>66-67</td>
</tr>
</tbody>
</table>

*Headings definitions*

<table>
<thead>
<tr>
<th>HD1</th>
<th>DS</th>
<th>0CL67</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>CL40</td>
<td>COGSWORTH INDUSTRIES'</td>
</tr>
<tr>
<td>DC</td>
<td>CL25</td>
<td>'</td>
</tr>
<tr>
<td>DC</td>
<td>XLS '0D25'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HD2</th>
<th>DS</th>
<th>0CL67</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>CL40</td>
<td>Inventory Discrepancies R'</td>
</tr>
<tr>
<td>DC</td>
<td>CL25</td>
<td>'eport'</td>
</tr>
<tr>
<td>DC</td>
<td>XLS '0D25'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HD3</th>
<th>DS</th>
<th>0CL67</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>CL65</td>
<td>'</td>
</tr>
<tr>
<td>DC</td>
<td>XLS '0D25'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HD4</th>
<th>DS</th>
<th>0CL67</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>CL40</td>
<td>'Product Begin + Purch - Sales = Expect'</td>
</tr>
<tr>
<td>DC</td>
<td>CL25</td>
<td>'ect Actual Diff'</td>
</tr>
<tr>
<td>DC</td>
<td>XLS '0D25'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HD5</th>
<th>DS</th>
<th>0CL67</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>CL40</td>
<td>'---------- ----- ------ ---- ---'</td>
</tr>
<tr>
<td>DC</td>
<td>XLS '0D25'</td>
<td></td>
</tr>
</tbody>
</table>

A:\MIN>cogs9b
COGS9B ... Begin execution
COGS9B ... Discrepancies report on REPORT.TXT
COGS9B ... Normal end of program

A:\MIN>type report.txt
COGSWORTH INDUSTRIES
Inventory Discrepancies Report

<table>
<thead>
<tr>
<th>Product</th>
<th>Begin + Purch</th>
<th>Sales</th>
<th>Expect</th>
<th>Actual</th>
<th>Diff</th>
</tr>
</thead>
</table>
| GIZMOS  | 17           | 99    | 90     | 26     | 23   | 3
| WIDGETS | 22           | 34    | 37     | 19     | 19   |
| JUNQUE  | 30           | 52    | 73     | 9      | 10   | 1

3 records processed.
1 indicate shortage.
1 indicate overage.
Summary

A number must be in packed decimal format to be edited (formatted).

To edit a number, you must define a mask. This mask:

- The mask must have the same length as the target field.
- The first byte of the mask is always the fill character. This fill character (which replaces leading zeroes) is usually a blank (X'40'). Use an asterisk (X'5C') for check protection.
- The mask may include commas (X'6B') if desired and where appropriate.
- The mask may include a decimal point (X'4B').
- The mask must include a X'20' or X'21' for each digit in the packed field being edited. The total number of X'20's and X'21's will *always* be an odd number.
- The mask may include a trailing sign (X'60') or CR (X'C3D9') or DB (X'C4C2') which will be replaced by the fill character if the number is not negative.
CHAPTER 9
THE EDIT INSTRUCTION

Exercises

1. True or false.

   T  F  a. If a literal is used as an edit mask, that literal will appear in the LTORG.
   T  F  b. The total length of an edit mask must be an odd number.
   T  F  c. The edit instruction, like PACK and UNPK, is an SS-type instruction which allows
            a length operator on both operands.
   T  F  d. An output field must be "refreshed" prior to subsequent edits even if that field
            was defined with a DC and an edit mask.
   T  F  e. The X'21' in the mask indicates the last byte where leading zeroes will be
            replaced by the fill character.
   T  F  f. If an edit mask ends with X'C3D9', then negative numbers will print with CR
            and positive numbers will print with DB.
   T  F  g. If an edit mask ends with X'60', then negative numbers will print with a trailing
            sign.
   T  F  h. An edit mask must have a X'4B' for each comma.
   T  F  i. If the field being edited is defined as PL3, then the edit mask must contain three
            X'20's, or two X'20's and one X'21'.
   T  F  j. Commas can be used with check protection in the edit mask.
   T  F  k. The mask documentation technique shown in this chapter is similar COBOL's
            PIC clauses.
   T  F  l. The mask X'402021204B202020' would be documented as BZ99.99.
   T  F  m. It is up to the programmer to determine the placement of decimal points since
            all packed decimal arithmetic in BAL is integer arithmetic.

2. What is wrong with the following edit masks?

   a. X'202020'
   b. X'402020'
   c. X'40202121'
   d. X'4020204B2020'
   e. X'4020204B2020206B2120'
   f. X'4020206B2020206B20212060'

3. Show the changes you would make to the HDGS routine shown in chapter 8 so as to
   suppress leading zeroes on page counts.

4. Determine the correct length for the output field and packed field for each of the following
   edit masks. Then show the result if a packed field of the proper length with a value of -123
   was edited using the indicated mask.
Exercises

<table>
<thead>
<tr>
<th>MASK</th>
<th>Output Field Length</th>
<th>Packed Field Length</th>
<th>Result of editing</th>
</tr>
</thead>
<tbody>
<tr>
<td>x. X'402020202120'</td>
<td>6</td>
<td>3</td>
<td>BBB123</td>
</tr>
<tr>
<td>a. X'40202120'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. X'4020206B202120'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. X'5C20206B2020214B2020'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. X'40202120C4C2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. X'40202020202020202060'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f. X'5C20202020202020C3D9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g. X'40206B2020206B202120'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h. X'4020204B202020C4C2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i. X'5C2020202020202060'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j. X'40204B202020202020'</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. For each of the edit masks shown in exercise 4 above, show the result if a packed field of the proper length and with a value of zero was edited. Repeat for a packed field with a value of -1.

6. Given the following field definitions:

```plaintext
FLDA     DC    PL3'-12'
WK6      DS    CL6
WK7      DS    CL7
WK8      DS    CL8
```

...show the results of the following edits:

a. MVC WK6,=X'402020202020'  
   ED WK6, FLDA
b. MVC WK6,=X'402020202120'  
   ED WK6, FLDA
c. MVC WK6,=X'402021202020'  
   ED WK6, FLDA
d. MVC WK6,=X'5C2020202120'  
   ED WK6, FLDA
e. MVC WK6,=X'5C2021202020'  
   ED WK6, FLDA
f. MVC WK7,=X'40202020202060'  
   ED WK7, FLDA
g. MVC WK7,=X'5C202020212060'  
   ED WK7, FLDA
h. MVC WK7,=X'4020206B202120'  
   ED WK7, FLDA
Exercises

i.  MVC        WK8,=X'4020206B20212060'  
    ED         WK8,FLDA

j.  MVC        WK8,=X'40206B2021204B20'  
    ED         WK8,FLDA

k.  MVC        WK8,=X'5C2021204B202060'  
    ED         WK8,FLDA

l.  MVC        WK8,=X'402021202020C3D9'  
    ED         WK8,FLDA

7.  Write a program which will verify your answers to exercise 6 above. (Hint: See EDITS9.MLC in this chapter.)

8.  Given the following field definitions:

    MASK     DC    X'4020206B2021204B202060'
    WK11     DS    CL11

...show the results of the following edits:

a.  MVC        WK11,MASK
    ED         WK11,=PL4'0'

b.  MVC        WK11,MASK
    ED         WK11,=PL4'1'

c.  MVC        WK11,MASK
    ED         WK11,=PL4'-1'

d.  MVC        WK11,MASK
    ED         WK11,=PL4'12'

e.  MVC        WK11,MASK
    ED         WK11,=PL4'-12'

f.  MVC        WK11,MASK
    ED         WK11,=PL4'1234'

g.  MVC        WK11,MASK
    ED         WK11,=PL4'-1234'

h.  MVC        WK11,MASK
    ED         WK11,=PL4'12345'

i.  MVC        WK11,MASK
    ED         WK11,=PL4'123456'

j.  MVC        WK11,MASK
    ED         WK11,=PL4'-1234567'
Exercises

9. Write a program which will verify your answers to exercise 8 above. (Hint: See EDITS9.MLC in this chapter.)

10. (This exercise is similar to exercise 12 of chapter 7.) Modify program COGS9A.MLC in this chapter to include totals by state; that is, your output should appear as follows:

```
123456789012345678901234567890123456789012345678901234567890
COGSWORTH INDUSTRIES
Sales Recap
Product       Calif     Ill      Utah     Wisc    TOTAL
----------    -----    -----    -----    -----    -----    -------
GIZMOS          20       30       20       20       90
WIDGETS         15       10       10        2       37
JUNQUE          25       15       15       18       73
----------    -----    -----    -----    -----    -----    -------
TOTAL           60       55       45       40      200
```

3 records processed.

11. Modify the program from exercise 13 of chapter 7 to use ED instead of UNPK and MVZ.

12. Modify the program from exercise 14 of chapter 7 to use ED instead of UNPK and MVZ.

13. (Refer to the Small Town Payroll database in More Datasets.) Use the HISTORY file to produce a payroll register for the pay period ending 1/2/93. Include totals and count as shown. The report should appear as follows:

```
12345678901234567890123456789012345678901234567890
SMALL TOWN PAYROLL
Payroll Register for PPED 1/2/93
Employee    Hours        Gross
--------   -------    -----------
--------    -------    -----------
```

There were BZZ9 checks printed.