COBOL VERSION 4
INSTANT MANUAL

CONTROL DATA
CYBER 170 SERIES
CYBER 70 SERIES
MODELS 72, 73, 74
6000 SERIES
COMPUTER SYSTEMS
<table>
<thead>
<tr>
<th>REVISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Original printing.</td>
</tr>
<tr>
<td>(2-15-76)</td>
<td></td>
</tr>
</tbody>
</table>

Publication No. 60497000

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
215 Moffett Park Drive
Sunnyvale, California 94086

or use Comment Sheet in the back of this manual

© 1976
Control Data Corporation
Printed in the United States of America
INTRODUCTION

This instant outlines the COBOL Version 4.5 language as designed to run under NOS/BE 1 and NOS 1 operating systems for the CONTROL DATA® CYBER 170, CYBER 70 Model 72, 73, 74, and 6000 Series Computer Systems.

COBOL Version 4.5 is compatible with the American National Standard COBOL, X3.23-1968, and contains extensions to the standard. Extensions are indicated in this instant by shading.

Special Features

Mass storage input and output including indexed sequential, actual key, and direct access file processing

SORT verb sorts files within COBOL program

Automatic table search using index names and the SEARCH and SET statements

Report Writer produces printed reports automatically, or user may produce report page with LINAGE clause and WRITE statement

Full arithmetic facility including:
  18-digit operands
  DIVIDE with REMAINDER
  COMPUTE with exponentiation
  CORRESPONDING option with ADD and SUBTRACT

Segmentation and overlay of object program

Inter-program communication with separately compiled COBOL programs as well as with FORTRAN or COMPASS programs

Access to COBOL source library

Memory dumps with restart at specified checkpoints

Remote interactive capability for remote terminal input/output
Multiple-index processing capability
Ability to trace paragraphs
EBCDIC file processing

PROGRAM EFFICIENCY HINTS

To reduce keypunching:

Use abbreviations where permitted.
Use PIC clause rather than SIZE, CLASS, USAGE clauses.

To increase compilation efficiency:

Restrict data and paragraph names to 9 characters or less.
Eliminate unnecessary paragraph names.
Reduce forward references.

To increase execution efficiency:

Use same size sending and receiving fields.
Make table and item sizes a multiple of 10 characters.
Reduce subscripting.
Subscript with literals instead of variables.
Use COMPUTATIONAL-1 items or index-names as subscripts.
Use COMPUTATIONAL-1 items as arithmetic variables.
Restrict arithmetic items to 9 digits or less.
Use SYNCHRONIZED RIGHT clause for data frequently referenced.
Use SAME RECORD AREA to save moves; SAME AREA to save space.
Enclosed elements are optional.

Only one element must be selected

Repeat preceding bracketed material as needed.

Entire phrase may be repeated.

COBOL words have preassigned meanings and appear in capitals.

COBOL words not underlined may be omitted.

Terms in small letters are words supplied by the programmer.

Punctuation and special characters are required where shown.

**COBOL LANGUAGE ELEMENTS**

**Word**
Sequence of up to 30 alphanumeric characters including embedded hyphens.

**Identifier**
Word that may be qualified or subscripted.

**Literal**
String of characters whose value is exactly represented by the characters; numeric literal may be 0-9, +, -, and decimal point; non-numeric literal must be enclosed in quotes, may be any alphanumeric character except quotes.

**Statement**
Procedure Division verb with associated options.

**Sentence**
One or more statements terminated by period.

**Paragraph**
Procedure Division sentences, Identification and Environment Division entries introduced by paragraph name, terminated by period.

**Paragraph Name**
Word terminated by period used to introduce paragraph; user defined in Procedure Division, pre-defined in Identification and Environment Divisions.

**Section**
Paragraphs may be included in sections introduced by section name.

**Section Name**
Word followed by SECTION and terminated by period; user defined in Procedure Division, pre-defined in Identification, Environment, and Data Divisions.

**Entry**
Unit of description in Data Division; must be terminated by period.
IDENTIFICATION DIVISION

\{ ID \}
\{ IDENTIFICATION \} D\{IV\}ISION

PROGRAM-ID. program-name.
[AUTHOR. [comment-entry.]]
[INSTALLATION. [comment-entry.]]
[DATE-WRITTEN. [comment-entry.]]
[DATE-COMPILED. [current-date supplied by compiler.]]
[SECURITY. [comment-entry.]]
[REMARKS. [comment-entry.]]

ENVIRONMENT DIVISION

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.†

format 1:

SOURCE-COMPUTER. COPY library-name

\[
\text{REPLACING} \begin{cases}
\text{literal-1} \\
\text{word-1} \\
\text{identifier-1}
\end{cases}
\text{BY} \begin{cases}
\text{literal-2} \\
\text{word-2} \\
\text{identifier-2}
\end{cases}
\]
\[
\text{REPLACING} \begin{cases}
\text{literal-3} \\
\text{word-3} \\
\text{identifier-3}
\end{cases}
\text{BY} \begin{cases}
\text{literal-4} \\
\text{word-4} \\
\text{identifier-4}
\end{cases}
\ldots
\]

format 2:

SOURCE-COMPUTER. computer-name.

format 1:

OBJECT-COMPUTER. COPY library-name

\[
\text{REPLACING} \begin{cases}
\text{literal-1} \\
\text{word-1} \\
\text{identifier-1}
\end{cases}
\text{BY} \begin{cases}
\text{literal-2} \\
\text{word-2} \\
\text{identifier-2}
\end{cases}
\]
\[
\text{REPLACING} \begin{cases}
\text{literal-3} \\
\text{word-3} \\
\text{identifier-3}
\end{cases}
\text{BY} \begin{cases}
\text{literal-4} \\
\text{word-4} \\
\text{identifier-4}
\end{cases}
\ldots
\]

†CONFIGURATION SECTION, SOURCE-COMPUTER, and OBJECT-COMPUTER are optional.
format 2:

OBJECT-COMPUTER. computer-name

[SEGMENT-LIMIT IS priority-number]

[MEMORY SIZE integer { WORDS
CHARACTERS
MODULES }] .

format 1:

SPECIAL-NAMES. COPY library-name

[REPLACING { literal-1
word-1
identifier-1 } BY { literal-2
word-2
identifier-2 }]

[ literal-3
word-3
identifier-3 ] BY { literal-4
word-4
identifier-4 } ... ].

format 2:

SPECIAL-NAMES.

[SWITCH integer-1

IS mnemonic-name-1

[ON STATUS IS condition-name-1
[OFF STATUS IS condition-name-2 ] ]

IS mnemonic-name-2

[OFF STATUS IS condition-name-3
[ON STATUS IS condition-name-4 ] ]

ON STATUS IS condition-name-5

[OFF STATUS IS condition-name-6]

OFF STATUS IS condition-name-7

[ON STATUS IS condition-name-8]

[ implementor-name-1 IS mnemonic-name-1
implementor-name-2 IS mnemonic-name-2 . . . ]

[ non-numeric-literal-1 IS mnemonic-name-1
non-numeric-literal-2 IS mnemonic-name-2 . . . ]

[CURRENCY SIGN IS literal]

[DECIMAL-POINT IS COMMA]
[CONSOLE IS mnemonic-name]
[TERMINAL IS mnemonic-name]
[SUB-SHEMA IS mnemonic-name].

INPUT-OUTPUT SECTION.

format 1:

FILE-CONTROL. COPY library-name

[ REPLACING { literal-1
          word-1
          identifier-1 } BY { literal-2
          word-2
          identifier-2 } ]

[ { literal-3
    word-3
    identifier-3 } BY { literal-4
    word-4
    identifier-4 } ]...

format 2:

FILE-CONTROL.

{ SELECT [OPTIONAL] file-name-1[RENAMEING file-name-2]

ASSIGN TO [integer] implementor-name-1 [implementor-name-2]

... [OR implementor-name-3 [implementor-name-4] ...]

[ FOR MULTIPLE { REEL UNIT } ] [ERROR FILE IS file-name]

[ RESERVE { NO integer } ALTERNATE { AREA ( AREAS ) } ]

[ { FILE-LIMIT IS
    FILE-LIMITS ARE
    THRU
    THROUGH
    data-name-1
    literal-1
    data-name-2
    literal-2
    data-name-3
    THRU
    THROUGH
    literal-3
    literal-4
    data-name-4
    literal-4
    data-name-5
    literal-5 } ]...
ORGANIZATION IS
  { SEQUENTIAL
    STANDARD
    DIRECT
    INDEXED SEQUENTIAL
    RELATIVE
    ACTUAL KEY
  }

ACCESS MODE IS
  { SEQUENTIAL
    RANDOM
  }

PROCESSING MODE IS SEQUENTIAL

ACTUAL KEY IS data-name-1
SYMBOLIC KEY IS data-name-2 [WITH DUPLICATES]
RECORD KEY IS data-name-3 [WITH DUPLICATES]

ALTERNATE RECORD KEY IS data-name-3
  [WITH DUPLICATES [INDEXED]]

NUMBER OF BLOCKS IS
  { data-name
    integer
  }

{ INDEX-LEVEL IS
  INDEX-LEVELS ARE
  integer
}

INDEX-BLOCK CONTAINS integer CHARACTERS

RECORD-BLOCK CONTAINS integer
  { RECORDS
    CHARACTERS
  }

INDEX-PADDING IS integer PERCENT

DATA-PADDING IS integer PERCENT

format 1:

I-O-CONTROL, COPY library-name

REPLACING
  { literal-1
    word-1
    identifier-1
  }
BY
  { literal-2
    word-2
    identifier-2
  }

{ literal-3
  word-3
  identifier-3
}
BY
  { literal-4
    word-4
    identifier-4
  } ...

7
format 2:

I-O CONTROL

\[
\text{RERUN [ ON \{ file-name-1 implementor-name \} ]}
\]

\[
\text{EVERY \{ \{ [END OF] REEL UNIT \} \{ OF file-name-2 \} }\]
\]

\[
\text{SAME AREA FOR file-name-1 \{ file-name-2 \} ...}
\]

\[
\text{SAME RECORD AREA FOR file-name-1 \{ file-name-2 \} ...}
\]

\[
\text{MULTIPLE FILE TAPE CONTAINS file-name-1}
\]

\[
\text{POSITION integer-1 \{ file-name-2}
\]

\[
\text{POSITION integer-2] ... ] ... .}
\]
PICTURE DESCRIPTION CODES

Data Characters

A  Alphabetic character
X  Alphanumeric character
9  Numeric character

Operation Symbols

S  Signed
V  Assumed decimal point location
P  Assumed decimal point scaling position

Replacement Characters

Z  Leading zeros replaced by blanks
*  Leading zeros replaced by * (check protection symbol)

Insertion Characters

$  Dollar sign; floating when more than one (dollar sign may be replaced by currency sign defined in SPECIAL-NAMES)
,  Comma
/  Slash (instead of comma)
.  Actual decimal point
B  Blank
0  Zero
-  Minus sign when item is negative, blank when positive; floating when more than one
+  Plus sign when item is positive, minus when negative; floating when more than one
CR  Credit symbol when item is negative, blank when positive
DB  Debit symbol when item is negative, blank when positive
# DATA SPECIFICATIONS

<table>
<thead>
<tr>
<th></th>
<th>File Section</th>
<th>Common and Working Storage Sections</th>
<th>Constant Section</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>01 group elem</td>
<td>77 01 group elem</td>
<td>77 01 group elem</td>
</tr>
<tr>
<td>REDEFINES</td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIZE</td>
<td>R R R R R R R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USAGE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS</td>
<td>R R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCCURS</td>
<td>I I I I I I I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINT LOCATION</td>
<td>J I J I J I J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGNED</td>
<td>J I J I J I J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JUSTIFIED</td>
<td>J I J I J I J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYNCHRONIZED</td>
<td>J I J I J I J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PICTURE</td>
<td>J I J I J I J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editing Clauses</td>
<td>J I J I J I J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUE</td>
<td>K K C V V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILLER</td>
<td>I I I I I I I</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**C** Legal only in defining values for condition names

**I** Illegal

**R** Required if PICTURE is not used

**blank** Optional

**V** Required

**J** Legal only on elementary 01 items

**K** Documentary only

01 items may be elementary or group, depending on the program
DATA DIVISION.

[FILE SECTION.]

[COMMON-STORAGE SECTION.]

[WORKING-STORAGE SECTION.]

[CONSTANT SECTION.]

[LINKAGE SECTION.]

[REPORT SECTION.]

File Description Entry (File Section Only)

A Sort File Description (SD) entry may contain only DATA RECORD, RECORD CONTAINS, and FILE CONTAINS clauses; any or all may be omitted from an SD entry.

format 1:

\[
\{ \text{SD} \} \text{ file-name COPY library-name} \\
\{ \text{FD} \} \\
\left\{ \text{REPLACING \{ literal-1 \} \{ word-1 \} \{ identifier-1 \} \} \text{ BY \{ literal-2 \} \{ word-2 \} \{ identifier-2 \} \} \right. \\
\left. \{ \text{literal-3 \} \{ word-3 \} \{ identifier-3 \} \} \text{ BY \{ literal-4 \} \{ word-4 \} \{ identifier-4 \} \} \ldots \right\}
\]

format 2:

\[
\{ \text{SD} \} \text{ file-name} \\
\{ \text{FD} \} \\
\left[ \text{BLOCK CONTAINS [integer-1 TO integer-2 \{ RECORDS \{ CHARACTERS \} \} \} } \right. \\
\left[ \text{DATA \{ RECORD IS \{ RECORDS ARE \} data-name-1 [data-name-2] \ldots \} \right. \\
\left. \{ REPORT IS \{ REPORTS ARE \} report-name-1 [report-name-2] \ldots \} \right. \\
\left. \{ \text{FILE CONTAINS ABOUT integer RECORDS} \} \right.
\]
LABEL \{ RECORDS ARE \} \{ STANDARD OMITTED \data-name-1 [data-name-2] ... \}

If label records are STANDARD:

\[ \text{VALUE OF} \{ \begin{align*}
\{ \text{ID IDENTIFICATION} \} \text{ IS } \{ \text{literal-1} \} \\
\{ \text{DATE-WRITTEN} \} \text{ IS } \{ \text{literal-2} \} \\
\{ \text{EDITION-NUMBER} \} \text{ IS } \{ \text{literal-3} \} \\
\{ \text{REEL-NUMBER} \} \text{ IS } \{ \text{literal-4} \} \\
\{ \text{RETENTION-CYCLE} \} \text{ IS } \{ \text{literal-5} \}
\end{align*} \} \]

If label records are a data-name:

\[ \text{VALUE OF data-name-3 IS } \{ \text{literal-1} \} \\
\text{data-name-5 IS } \{ \text{literal-2} \} \ldots \]

\[ \text{VALUE OF ENDING-TAPE-LABEL-IDENTIFIER} \text{ IS } \{ \text{literal-3} \} \\
\text{LINAGE IS } \{ \text{identifier} \} \text{ LINES} \]

\[ \text{RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS} \]

\[ \text{DEPENDING ON } \{ \text{RECORD-MARK} \} \text{ data-name-1} \]

\[ \text{RECORDING MODE IS } \{ \begin{align*}
\text{BINARY} \\
\text{DECIMAL (EBCDIC)} \\
\{ \text{HIGH LOW HYPER} \} \text{ DENSITY} 
\end{align*} \}

\[ \text{SEQUENCED ON data-name-1 [data-name-2] ...} \]
Record Description Entry (File, Common-Storage, Working-Storage, Constant and Linkage Sections)

format 1:

{ 01 } data-name COPY library-name [FROM LIBRARY]

[ REPLACING { word-1 identifier-1 } BY { word-2 identifier-2 } ]

{ word-3 identifier-3 } BY { word-4 identifier-4 } ] ... ]

format 2:

level-number data-name-1 [REDEFINES identifier]
COPY data-name-2 FROM SOURCE.

format 3:

level-number { data-name-1 [REDEFINES identifier] } FILLER

[ BWZ BLANK WHEN ZERO ]

[ CHECK PROTECT FLOAT DOLLAR SIGN FLOAT CURRENCY SIGN ]
[ ZERO SUPPRESS ]

[ LEAVING integer PLACES ]

[ CLASS IS ]

[ ALPHABETIC NUMERIC ALPHANUMERIC ]

[ JUSTIFIED RIGHT ]
OCCURS integer-1 [TO integer-2] TIMES

[DEPENDING ON data-name-1]

[ { ASCENDING } KEY IS data-name-2 [data-name-3] ... ] ...

[INDEXED BY index-name-1 [index-name-2] ...]

[ { PIC PICTURE } IS character-string ]

POINT LOCATION IS { LEFT RIGHT } integer PLACES

RANGE IS literal-1 { THRU THROUGH } literal-2

{ SIGNED
  { SIGN IS data-name } }

SIZE IS integer [ { CHARACTERS } ]

[ { SYNC SYNCHRONIZED } { LEFT RIGHT } ]

[ USAGE IS ]

( COMP COMPUTATIONAL
  COMP-1 COMPUTATIONAL-1
  COMP-2 COMPUTATIONAL-2
  DISPLAY INDEX

[ VALUE IS literal ]

format 4:

66 data-name RENAMES identifier-1 [ { THRU THROUGH } identifier-2 ].

format 5:

88 condition-name { VALUE IS VALUES ARE } literal-1

[ { THRU THROUGH } literal-2 [ literal-3 [ { THRU THROUGH } literal-4 ] ] ...

14
Report Description Entry (Report Section Only)

format 1:

RD report-name [WITH CODE mnemonic-name]

COPY library-name [REPLACING {literal-1} {word-1} {identifier-1}]

{literal-2}
{word-2}
{identifier-2}

{literal-3}
{word-3}
{identifier-3}

BY {literal-4}
{word-4}
{identifier-4} ...

format 2:

RD report-name [WITH CODE mnemonic-name]

{CONTROL IS}
{CONTROLS ARE}

{identifier-1[identifier-2] ...}

{FINAL}

{FINAL identifier-1[identifier-2] ...}

PAGE {LIMIT IS}
{LIMITS ARE}

integer-1 {LINE}
{LINES}

[HEADING integer-2] [FIRST DETAIL integer-3]

[LAST DETAIL integer-4 [FOOTING integer-5]]

Report Group Description Entry (Report Section Only)

format 1:

01 [data-name] COPY library-name [FROM LIBRARY]

[REPLACING {literal-1} {word-1} {identifier-1}]

{literal-2}
{word-2}
{identifier-2}

{literal-3}
{word-3}
{identifier-3}

BY {literal-4}
{word-4}
{identifier-4} ...
format 2:

01 data-name-1 [REDEFINES identifier]
   COPY data-name-2 FROM SOURCE.

format 3:

01 [data-name]

   [CLASS IS] { ALPHABETIC
                  NUMERIC
                  ALPHANUMERIC
                  AN
                  }

   [LINE NUMBER IS { integer-1
                    PLUS integer-2
                    NEXT PAGE
                    }
                ]

   [NEXT GROUP IS { integer-1
                    PLUS integer-2
                    NEXT PAGE
                    }
                ]

   [SIZE IS integer { CHARACTERS
                     DIGITS
                     }
            ]

   { REPORT HEADING
     RH
     PAGE HEADING
     PH
     OVERFLOW HEADING
     OH
     { CONTROL HEADING
       CH
       }
     { identifier-1
       FINAL
       }
     DETAIL
     DE
     { CONTROL FOOTING
       CF
       }
     { identifier-2
       FINAL
       }
     OVERFLOW FOOTING
     OV
     PAGE FOOTING
     PF
     REPORT FOOTING
     RF
     [ USAGE IS] DISPLAY ]
Report Element Description (Report Section Only)

**TYPE** clause allowed if level 01

**NEXT** **GROUP** clause allowed if level 01

level-number [data-name]

```
{ BLANK WHEN ZERO }
{ BWZ }

{ CHECK PROTECT }
{ FLOAT DOLLAR SIGN }
{ FLOAT CURRENCY SIGN }
{ ZERO SUPPRESS }

[LEAVING integer PLACES]
```

[CLASS IS]

```
{ ALPHABETIC }
{ NUMERIC }
{ ALPHANUMERIC }
{ AN }
```

[COLUMN NUMBER IS integer]

**GROUP** **INDICATE**

```
{ JUSTIFIED }
{ JUST }

RIGHT
```

```
LINE NUMBER IS { integer-1 }
{ PLUS integer-2 }
{ NEXT PAGE }
```

```
{ PIC }
{ PICTURE }

IS character-string
```

```
POINT LOCATION IS { LEFT }
{ RIGHT }

integer PLACES
```

```
RESET ON { identifier }
{ FINAL }
```

17
\[
\left\{
\begin{array}{l}
\text{SIGNED} \\
\text{SIGN IS data-name}
\end{array}
\right\}
\]
\[
\left\{
\begin{array}{l}
\text{SIZE IS integer} \\
\text{CHARACTERS} \\
\text{DIGITS}
\end{array}
\right\}
\]

\[
\left\{
\begin{array}{l}
\text{SOURCE IS} \\
\text{[SELECTED] identifier} \\
\text{LINE-COUNTER} \\
\text{PAGE-COUNTER} \\
\text{TODAYS-DATE}
\end{array}
\right\}
\]

\[
\text{SUM identifier-1}[\text{identifier-2}] \ldots [\text{UPON data-name}]
\]

\[
\text{VALUE IS literal}
\]

\[
[\text{USAGE IS} \text{ DISPLAY}].
\]
# USAGE SPECIFICATIONS

<table>
<thead>
<tr>
<th>Element</th>
<th>Upper Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>data-name</td>
<td>30 characters</td>
</tr>
<tr>
<td>literal</td>
<td>255 characters/digits</td>
</tr>
<tr>
<td>PERFORM nesting</td>
<td>no limit</td>
</tr>
<tr>
<td>level numbers</td>
<td>01-49, 66, 77, 88, FD, RD, SD</td>
</tr>
<tr>
<td>OCCURS...DEPENDING ON</td>
<td>1 per record description</td>
</tr>
<tr>
<td>library copies</td>
<td>5 levels of nesting</td>
</tr>
<tr>
<td>ACCEPT items</td>
<td>80 characters; 40 characters from console</td>
</tr>
<tr>
<td>PICTURE clause</td>
<td>30 symbols</td>
</tr>
<tr>
<td>arithmetic operand</td>
<td>18 digits</td>
</tr>
<tr>
<td>GO TO statement</td>
<td>no limit</td>
</tr>
<tr>
<td>ALTER statement</td>
<td>100 procedure names</td>
</tr>
<tr>
<td>DISPLAY items</td>
<td>no limit</td>
</tr>
<tr>
<td>ENTER parameters</td>
<td>no limit</td>
</tr>
<tr>
<td>Total files, I/O devices, and reports</td>
<td>53</td>
</tr>
<tr>
<td>Total procedure names</td>
<td>depends on field length</td>
</tr>
<tr>
<td>Total external references</td>
<td>depends on field length</td>
</tr>
<tr>
<td>index names</td>
<td>9</td>
</tr>
<tr>
<td>report group</td>
<td>127 detail lines</td>
</tr>
</tbody>
</table>

19
## VALID MOVE OPERATIONS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Elem. Alpha</td>
<td>AN</td>
<td>TD AN</td>
<td>AN</td>
<td>X</td>
<td>AN-Edit</td>
<td>AN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elem. BCD Num.</td>
<td>Conv. Bin.</td>
<td>TD AN</td>
<td>Num. AN†</td>
<td>Edit†</td>
<td>AN-Edit</td>
<td>AN†</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elem. AN</td>
<td>X</td>
<td>TD AN</td>
<td>Num. AN</td>
<td>Edit</td>
<td>AN-Edit</td>
<td>AN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elem. Edit Num.</td>
<td>X</td>
<td>TD AN</td>
<td>X</td>
<td>AN</td>
<td>X</td>
<td>AN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elem. Edit AN</td>
<td>X</td>
<td>TD AN</td>
<td>X</td>
<td>AN</td>
<td>X</td>
<td>AN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group AN</td>
<td>TD AN</td>
<td>TD AN</td>
<td>TD AN</td>
<td>AN</td>
<td>X</td>
<td>AN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Binary &amp; Mixed</td>
<td>TD AN</td>
<td>TD AN</td>
<td>TD AN</td>
<td>TD AN</td>
<td>X</td>
<td>TD AN-Moves</td>
<td>TD AN</td>
<td></td>
</tr>
<tr>
<td>Zero</td>
<td>Num. Bin.</td>
<td>X</td>
<td>Num. AN</td>
<td>Edit</td>
<td>AN-Edit</td>
<td>AN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Literal &amp; Fig. Cons. AN</td>
<td>X</td>
<td>TD AN</td>
<td>X</td>
<td>AN</td>
<td>Edit</td>
<td>AN-Edit</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td>Literal Num.</td>
<td>Conv. Bin.</td>
<td>X</td>
<td>Num. AN†</td>
<td>Edit</td>
<td>AN-Edit</td>
<td>AN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

† Valid only when source is integer; others TD.

Any move to a binary or mixed group is treated as an alphanumeric move; a precautionary diagnostic is issued.

A move to a figurative constant or literal is illegal.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Illegal</td>
</tr>
<tr>
<td>AN</td>
<td>Alphanumeric</td>
</tr>
<tr>
<td>AN-Edit</td>
<td>Alphanumeric edited</td>
</tr>
<tr>
<td>Conv.</td>
<td>Conversion prior to move</td>
</tr>
<tr>
<td>Edit</td>
<td>Numeric edited</td>
</tr>
<tr>
<td>Num.</td>
<td>Numeric</td>
</tr>
<tr>
<td>Num. Bin.</td>
<td>Numeric binary</td>
</tr>
<tr>
<td>TD</td>
<td>Trivial diagnostic issued</td>
</tr>
<tr>
<td>AN-MOVES</td>
<td>Alphanumeric moves</td>
</tr>
</tbody>
</table>
PROCEDURE DIVISION

DECLARATIVES.

section-name SECTION, declarative-sentence.

paragraph-name. { sentence. } ... } ... } ...

END DECLARATIVES.

section-name SECTION [priority-number].

paragraph-name. { sentence. } ... } ... } ...

ACCEPT identifier 
FROM { TIME 
DATE 
DAY 
mnemonic-name. }

ADD \{ identifier-1 \} \{ identifier-2 \} ... 
identifier-3 [ROUNDED] 

[ON SIZE ERROR imperative-statement]

ADD \{ identifier-1 \} \{ identifier-2 \} \{ identifier-3 \} ... TO 
identifier-3 [ROUNDED] identifier-4 [ROUNDED] ... 

[ON SIZE ERROR imperative-statement]

ADD \{ identifier-1 \} \{ identifier-2 \} \{ identifier-3 \} \{ literal-3 \} 

GIVING identifier-4 [ROUNDED] identifier-5 [ROUNDED] 

[ON SIZE ERROR imperative-statement]

ADD \{ CORR 
CORRESPONDING \} identifier-1 

TO identifier-2 [ROUNDED] identifier-3 [ROUNDED] ... 

[ON SIZE ERROR imperative-statement]
ALTER procedure-name-1 TO [PROCEED TO] procedure-name-2

[procedure-name-3 TO [PROCEED TO] procedure-name-4] ...

CALL [language-name]routine-name

CLOSE file-name-1

[ {'UNIT'}
     {'REEL'} ]

[ WITH {'NO REWIND'}
     {'LOCK'} ]

file-name-2

[ {'UNIT'}
     {'REEL'} ]

[ WITH {'NO REWIND'}
     {'LOCK'} ]

...

COMPUTE identifier-1 [ROUNDED] identifier-2 [ROUNDED] ...

{ FROM }

{ literal
     arithmetic-expression
     equal
     identifier-3
     EQUALS
}

[ON SIZE ERROR imperative-statement]

{ COPY
     INCLUDE }

library-name [FROM LIBRARY]

[REPLACING

{ literal-1
     word-1
     identifier-1
}

BY

{ literal-2
     word-2
     identifier-2
}

{ literal-3
     word-3
     identifier-3
}

BY

{ literal-4
     word-4
     identifier-4
}

... ]

DELETE [LAST]RECORD FROM file-name

[INVALID KEY imperative-statement]

DISPLAY { literal-1
     identifier-1
     literal-2
     identifier-2
}

[UPON mnemonic-name]

DIVIDE { literal-1
     identifier-1
}

INTO identifier-2 [ROUNDED]

[identifier-3 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]
DIVIDE \{ identifier-1 \} \{ BY \} \{ identifier-2 \}
    \{ literal-1 \} \{ INTO \} \{ literal-2 \}

GIVING identifier-3 [ROUNDED] \{ identifier-4 \{ROUNDED\} \}...

[ON SIZE ERROR imperative-statement]

DIVIDE \{ identifier-1 \} \{ BY \} \{ identifier-2 \}
    \{ literal-1 \} \{ INTO \} \{ literal-2 \}

GIVING identifier-3 [ROUNDED]

REMAINDER identifier-4

[ON SIZE ERROR imperative-statement]

ENTER \{ language-name \} routine-name \{ USING parameter-list \}.

ENTER COBOL.

ENTER LINKAGE.

ENTRY routine-name \{ USING parameter-list \}.

EXAMINE identifier

\{ TALLYING \{ ALL LEADING \} \{ UNTIL FIRST \} \}
    \{ \{ REPLACING BY \} \{ literal-2 \} \}
    \{ \{ literal-1 \} \}

EXIT.

\{ EXIT PROGRAM \}
    \{ RETURN \}

GENERATE identifier

GO TO \{ procedure-name \}

GO TO procedure-name-1 \{ procedure-name-2 \}...

DEPENDING ON identifier
IF conditional-expression \{ **THEN** \} \{ statement-1 \} \{ NEXT SENTENCE \} \\
[ **THEN** ] \{ **OTHERWISE** \} \{ statement-2 \} \{ NEXT SENTENCE \} \\
Conditional expressions include:

- GREATER THAN \( \text{GR} \gt \) \\
- LESS THAN \( \text{LS} \lt \) \\
- GREATER-EQUAL TO \( \text{GQ} \geq \) \\
- LESS-EQUAL TO \( \text{LQ} \leq \) \\
- EQUAL TO \( \text{EQ} \equiv \) \\
- IS \{ **NOT** \} UNEQUAL TO \( \text{EU} \neq \) \\
- EXCEEDS \( \text{EX} \) \\
- IS NO \( \text{IN} \) \\
- IS NGR \( \text{NR} \) \\
- IS NLS \( \text{NS} \) \\
- (identifier) \{ **NOT** \} IS UNEQUAL TO \( \text{EU} \neq \) \\
- POSITIVE \( \text{P} \) \\
- NEGATIVE \( \text{N} \) \\
- ZERO \( \text{Z} \) \\
- identifier IS \{ **NOT** \} NUMERIC \( \text{NUM} \) \\
- ALPHABETIC \( \text{ALPH} \) \\
- [ **NOT** ] \{ condition-name \} \{ switch-status-name \} \\
- INITIATE \{ report-name-1 \{ report-name-2 \} \ldots \} \{ ALL \} \\
- MOVE \{ \{ CORR \} CORRESPONDING \} identifier-1 \\
- (literal-1 \{ identifier-1 \}) TO \\
- identifier-2[identifier-3] ...
MULTIPLY \{ identifier-1 \} \{ literal \} \{ identifier-2 \} [ROUNDED] 

[identifier-3 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

MULTIPLY \{ identifier-1 \} \{ literal-1 \} \{ identifier-2 \} \{ literal-2 \}

GIVING identifier-3 [ROUNDED]

[identifier-4 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

NOTE character-string.

EXTEND file-name-1 [file-name-2] ...

INPUT file-name-1 \{ REVERSED \{ WITH NO REWIND \} \}

OPEN \{ file-name-2 \{ REVERSED \{ WITH NO REWIND \} \} \} ...

OUTPUT file-name-1 [WITH NO REWIND]

[file-name-2 [WITH NO REWIND]] ...

{ INPUT-OUTPUT \{ I-O \} file-name-1 [file-name-2] ...

PERFORM procedure-name-1 \{ THRU \{ THROUGH \} procedure-name-2 \}

\{ \{ identifier \} TIMES \} 

\{ \{ integer \} UNTIL condition \}
PERFORM procedure-name-1 [ { THRU
    THROUGH } procedure-name-2 ]

VARYING { index-name-1 }
FROM { index-name-2 }
    BY identifier-2
    UNTIL condition-1
    [ AFTER { index-name-3 }
           identifier-4 ]

FROM { index-name-4 }
BY { literal-4 }
    UNTIL condition-2

FROM { index-name-5 }
BY { literal-5 }
    UNTIL condition-3

READ file-name[NEXT] RECORD [ INTO identifier]

AT END imperative-statement
[ MAJOR KEY IS data-name ]
    INVALID KEY imperative-statement

RELEASE record-name [ FROM identifier]

RETURN file-name RECORD [ INTO identifier] AT END
    imperative-statement

REWITE [ LAST ] record-name [ FROM identifier]
    [ INVALID KEY imperative-statement]

SEARCH identifier-1 [ VARYING { index-name }]

[ AT END imperative-statement-1 ]

WHEN condition-1 { imperative-statement-2 }
    NEXT SENTENCE

[ WHEN condition-2 { imperative-statement-3 } ] ...

26
SEARCH ALL identifier [AT END imperative-statement-1]

WHEN condition

{ imperative-statement-2 }

NEXT SENTENCE

SEEK file-name RECORD [WITH KEY CONVERSION]

SET { index-name-1 [index-name-2] ... }

{ identifier-1 [identifier-2] ... }

TO { index-name-3 }

{ identifier-3 }

{ literal }

SET index-name-1 [index-name-2] ...

{ UP BY }

{ DOWN BY }

{ identifier }

{ literal }

SKIP { literal } { data-name } RECORDS ON file-name

SORT file-name-1 ON { DESCENDING }

{ ASCENDING }

KEY data-name-1 [data-name-2] ...

[ ON { DESCENDING }

{ ASCENDING }

KEY data-name-3 [data-name-4] ... ]...

INPUT PROCEDURE IS section-name-1

{ [ THRU }

{ THROUGH }

section-name-2 ]

USING file-name-2

OUTPUT PROCEDURE IS section-name-3

{ [ THRU }

{ THROUGH }

section-name-4 ]

GIVING file-name-3

27
START file-name KEY 
\[
\{\text{IS EQUAL TO} \quad \text{IS} = \quad \text{IS GREATER THAN} \quad \text{IS} > \quad \text{IS NOT LESS THAN} \quad \text{IS NOT} < \}\] data-name

INVALID KEY imperative-statement

STOP \{ literal \} \{ RUN \} .

SUBTRACT \{ identifier-1 \} \{ identifier-2 \} ... FROM identifier-3 
\[\text{[ROUNDED]} \quad \text{identifier-4[ROUNDED]} \]\...

\[\text{[ON SIZE ERROR imperative-statement]}\]

SUBTRACT \{ identifier-1 \} \{ identifier-2 \} ... FROM \{ identifier-3 \} \{ literal-3 \}

GIVING identifier-4 [ROUNDED] 
\[\text{identifier-5 [ROUNDED]} \] . . .

\[\text{[ON SIZE ERROR imperative-statement]}\]

SUBTRACT \{ CORR \} \{ CORRESPONDING \} identifier-1 FROM 

\[\text{[ON SIZE ERROR imperative-statement]}\]

TERMINATE \{ report-name-1 [report-name-2] ... \} \{ ALL \}

USE AFTER STANDARD ERROR PROCEDURE ON 
\{ file-name-1 [file-name-2] ... \}

INPUT
OUTPUT
INPUT-OUTPUT
I-O
EXTEND

28
USE {BEFORE} {STANDARD} {BEGINNING} {ENDING} 

{REEL} {FILE} {UNIT} 

LABEL {PROCEDURE} {PROCEDURES} ON {file-name-1} {file-name-2} ... 

INPUT 
OUTPUT 
INPUT-OUTPUT 
I-O 

USE BEFORE REPORTING identifier-1 [identifier-2] ...

USE FOR HASHING ON {ALL} {file-name-1} {file-name-2} ...

USE FOR DUPLICATE KEY ON {ALL} {file-name-1} {file-name-2} ...

USE FOR KEY CONVERSION ON {ALL} {file-name-1} {file-name-2} ...

WRITE record-name [FROM identifier-1]

{BEFORE} {AFTER} ADVANCING {identifier-2} {LINES} {integer} {LINES} {mnemonic-name}

AT {END-OF-PAGE} {EOP} {imperative-statement}

WRITE [NEXT] record-name [FROM identifier]

[INVALID KEY imperative-statement]
COBOL CONTROL STATEMENT

Parameters are used to select compilation options. All are optional and may be specified in any order. Each is separated from the other by a comma. The list may be enclosed in parentheses (as shown) or it may be separated from the word COBOL by a comma and terminated by a period.

COBOL.
   COBOL (parameter-list)  [comments]

A (Blank Conversion)  A  treats leading blanks as zeros

B (Binary Output)     absent  relocatable binary file on file
           B  LGO
           B = fn  binary output on file fn
           B = 0  suppress binary output

BUF (Buffer Size)     BUF  selects buffer size by method of version 3.0 COBOL

C (Copy Default)      C  uses version 3.0 COPY mode; to copy from library, FROM LIBRARY must be specified

D (Execution Abort)    D  prevents execution of program if E diagnostic occurs

DB (Subscript Limit Check)  DB  checks if subscript item within range of OCCURS for that item

DB1 (Paragraph Trace)  DB1  allows a program’s flow to be traced by paragraphs. Used in conjunction with trace directives:

          ENTER 'D.ONTR’, ENTER 'D.OFFTR’, or ENTER 'D.STPTR’.

E (EDITLIB)           E = prog  generates OVERLAY (COBCODE, O, O) statement;
                          selects V parameter

30
F (Computational Modification)  F
interprets COMPUTATIONAL items as COMPUTATIONAL-1

H (Sort File)  H
allocates buffer area for a sort

I (Source Input)  absent
 I
I=INPUT

I=fn  source input on file fn

K (Sub-Schema File)  K=fn
sub-schema on file fn

absent
K=O

no sub-schema files for this compile

L (List)  absent
 L
L=OUTPUT

normal listing on OUTPUT

any combination of following options:

L=fn  Output on file fn

L=O  suppress list output

LX  extended diagnostics

LR  cross reference pointers

LC  copy from library

LO  object code in octal

LM  data map

N (Non-ANSI Diagnostic)  N
diagnoses any non-ANSI feature

OB (Overlay Binary)  OB=fn
binary output from overlay segments put on file fn

P (ANSI execution)  P
allows non-ANSI reserved words; selects N parameter

S (Source Library)  absent
 S
S=COLIB

source library from file COLIB

S=fn  from file fn
<table>
<thead>
<tr>
<th>SUBM (Main Subcompile)</th>
<th>SUB</th>
<th>suppresses all Data Division binary output except from Working and Constant Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>T (Tape Sort)</td>
<td>T</td>
<td>sort requests tape sort</td>
</tr>
<tr>
<td>U (Alternate Collating)</td>
<td>U</td>
<td>uses alternate collating sequence</td>
</tr>
<tr>
<td>V (Sort Overlay)</td>
<td>V</td>
<td>saves loaded program, using NOGO</td>
</tr>
<tr>
<td>W (Initialize Overlays)</td>
<td>W</td>
<td>uses version 3.0 method of treating independent segments; they are available in last used state</td>
</tr>
<tr>
<td>Z (3.0 Compatibility)</td>
<td>Z</td>
<td>provides compatibility with version 3.0 COBOL; selects parameters C, and W</td>
</tr>
</tbody>
</table>
## COBOL CODING FORMAT

<table>
<thead>
<tr>
<th>Column</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 6</td>
<td>Sequence number</td>
</tr>
<tr>
<td>7</td>
<td>Hyphen, slash, or asterisk</td>
</tr>
</tbody>
</table>
| 8      | Division name  
Section name  
Paragraph name  
File description  
Record description level number |
| 12     | Record description data name  
First sentence of a paragraph  
File name  
Continuation of a data description or a sentence |
| 73 – 80| Identification (optional) |

**Sequence number**
Optional, checked by the processor if used

**Hyphen**
Indicates continuation of a word or literal from the preceding line

**Slash or asterisk**
Remainder of line is treated as comment and skips to new page

**Division name**
Terminated by period, remainder of line is blank

**Section name**
Followed by optional priority number, terminated by period, remainder is blank

**Paragraph name**
Terminated by period, followed by at least one blank before text begins

**File Description**
FD or SD followed by file name and at least one blank

**Record Description**
Level number followed by at least one blank and data name

**First Sentence**
Begin in or after column 12. Spaces may be used freely to avoid splitting a word or literal. If a word or literal is split, a hyphen must appear in column 7 of the next line.
COBOL compilation

COBOL source deck

COBOL.
ACCOUNT, account_number, password
(job statement)

†If required by operating system
COMPILATION AND EXECUTION

†If required by operating system
EXECUTION WITH SUBCOMPiled PROGRAM

empty record = end of loader information

If required by operating system

†
<table>
<thead>
<tr>
<th>COBOL RESERVED WORD LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABOUT</td>
</tr>
<tr>
<td>ACCEPT</td>
</tr>
<tr>
<td>ACCESS</td>
</tr>
<tr>
<td>ACTUAL</td>
</tr>
<tr>
<td>ADD</td>
</tr>
<tr>
<td>*ADDRESS</td>
</tr>
<tr>
<td>ADVANCING</td>
</tr>
<tr>
<td>AFTER</td>
</tr>
<tr>
<td>ALL</td>
</tr>
<tr>
<td>ALPHABETIC</td>
</tr>
<tr>
<td>ALPHANUMERIC</td>
</tr>
<tr>
<td>ALTER</td>
</tr>
<tr>
<td>ALTERNATE</td>
</tr>
<tr>
<td>AN</td>
</tr>
<tr>
<td>AND</td>
</tr>
<tr>
<td>*ANSIB</td>
</tr>
<tr>
<td>*APPLY</td>
</tr>
<tr>
<td>ARE</td>
</tr>
<tr>
<td>AREA</td>
</tr>
<tr>
<td>AREAS</td>
</tr>
<tr>
<td>ASCENDING</td>
</tr>
<tr>
<td>ASSIGN</td>
</tr>
<tr>
<td>AT</td>
</tr>
<tr>
<td>AUTHOR</td>
</tr>
<tr>
<td>*BCD</td>
</tr>
<tr>
<td>BEFORE</td>
</tr>
<tr>
<td>BEGINNING</td>
</tr>
<tr>
<td>BEGINNING-FILE-LABEL</td>
</tr>
<tr>
<td>BEGINNING-TAPE-LABEL</td>
</tr>
<tr>
<td>BINARY</td>
</tr>
<tr>
<td>*BITS</td>
</tr>
<tr>
<td>BLANK</td>
</tr>
<tr>
<td>BLOCK</td>
</tr>
<tr>
<td>BLOCKS</td>
</tr>
<tr>
<td>BWZ</td>
</tr>
<tr>
<td>BY</td>
</tr>
<tr>
<td>CALL</td>
</tr>
<tr>
<td>CANCEL</td>
</tr>
<tr>
<td>*CD</td>
</tr>
<tr>
<td>CF</td>
</tr>
<tr>
<td>CH</td>
</tr>
<tr>
<td>CHARACTER</td>
</tr>
<tr>
<td>CHARACTERS</td>
</tr>
<tr>
<td>CHECK</td>
</tr>
<tr>
<td>CLASS</td>
</tr>
<tr>
<td>CLOCK-UNITS</td>
</tr>
<tr>
<td>CLOSE</td>
</tr>
<tr>
<td>COBOL</td>
</tr>
<tr>
<td>CODE</td>
</tr>
<tr>
<td>COLUMN</td>
</tr>
<tr>
<td>COMMA</td>
</tr>
<tr>
<td>COMMON-STORAGE</td>
</tr>
<tr>
<td>COMP</td>
</tr>
<tr>
<td>COMP-1</td>
</tr>
<tr>
<td>COMP-2</td>
</tr>
<tr>
<td>COMP-3</td>
</tr>
<tr>
<td>COMPUTATIONAL</td>
</tr>
<tr>
<td>COMPUTATIONAL-1</td>
</tr>
<tr>
<td>COMPUTATIONAL-2</td>
</tr>
<tr>
<td>COMPUTATIONAL-3</td>
</tr>
<tr>
<td>COMPUTE</td>
</tr>
<tr>
<td>CONFIGURATION</td>
</tr>
<tr>
<td>CONSOLE</td>
</tr>
<tr>
<td>CONSTANT</td>
</tr>
<tr>
<td>CONTAINS</td>
</tr>
<tr>
<td>CONTROL</td>
</tr>
<tr>
<td>CONTROLS</td>
</tr>
<tr>
<td>CONVERSION</td>
</tr>
<tr>
<td>COPY</td>
</tr>
<tr>
<td>CORR</td>
</tr>
<tr>
<td>CORRESPONDING</td>
</tr>
<tr>
<td>*COUNT</td>
</tr>
<tr>
<td>CREATE</td>
</tr>
<tr>
<td>CURRENCY</td>
</tr>
<tr>
<td>DATA</td>
</tr>
<tr>
<td>DATA-PADDING</td>
</tr>
<tr>
<td>DATE</td>
</tr>
<tr>
<td>DATE-COMPiled</td>
</tr>
<tr>
<td>DATE-WRITTEN</td>
</tr>
<tr>
<td>DAY</td>
</tr>
<tr>
<td>DE</td>
</tr>
<tr>
<td>DECIMAL</td>
</tr>
<tr>
<td>DECIMAL-POINT</td>
</tr>
<tr>
<td>DECLARATIVES</td>
</tr>
<tr>
<td>DELETE</td>
</tr>
<tr>
<td>*DELMITER</td>
</tr>
<tr>
<td>*DELMITED</td>
</tr>
</tbody>
</table>
DENSITY
DEPENDING
*DEPTH
DESCENDING
*DESTINATION
DETAIL
DIGIT
DIGITS
DIRECT
*DISABLE
DISPLAY
DIVIDE
DIVIDED
DIVISION
DOLLAR
DOWN
DUPLICATE
DUPLICATES

EBCDIC
EDITION-NUMBER
ELSE
*EMI
*ENABLE
END
END-OF-PAGE
ENDING
ENDING-FILE-LABEL
ENDING-TAPE-LABEL
ENDING-TAPE-LABEL-IDENTIFIER
ENTER
ENTRY
ENVIRONMENT
EOP
EQ
EQUAL
EQUALS
ERROR
ERROR-CODE
*ESI
*ETI
EVERY
EXAMINE
EXCEEDS
EXIT
EXPONENTIATED
EXTEND
*EXTERNAL

FD
FILE
FILE-CONTROL
FILE-LABEL
FILE-LIMIT
FILE-LIMITS
FILLER
FINAL
*FIND
FIRST
FLOAT
FOOTING
FOR
*FORMAT
FROM

GENERATE
GIVING
GO
GQ
GR
GREATER
GREATER-EQUAL
GROUP

*HASHED
HASHED-VALUE
HASHING
HEADING
HIGH
HIGH-VALUE
HIGH-VALUES
*HOLD
HYPER

ID
IDENTIFICATION
IF
IN
INCLUDE
INDEX
INDEX-BLOCK
INDEX-LEVEL
INDEX-LEVELS
INDEX-PADDING
INDEXED
INDicate
INITIate
INPUT
INPUT-OUTPUT
INSTALLATION
INTO
INVALID
I-O
I-O-CONTROL
IS

*LOWER-BOUNDS
LO
LS

MAJOR
MEMORY
*MESSAGE
MINUS
MODE
MODULES
MOVE
MULTIPLE
MULTIPLIED
MULTIPLY

JUST
JUSTIFIED

KEY
KEYS

LABEL
LAST
LEADING
LEAVING
LEFT
LESS
LESS-EQUAL
LIBRARY
LIMIT
LIMITS
LINAGE
LINAGE-COUNTER
LINE
LINE-COUNTER
LINES
LINKAGE
LOCATION
LOCK
LOW
LOW-VALUE
LOW-VALUES
*LOWER-BOUND

NEGATIVE
NEXT
NGR
NLS
NO
NOT
NOTE
NO
NUMBER
NUMERIC

OBJECT-COMPUTER
OCCURS
OF
OFF
OH
OMITTED
ON
OPEN
OPTIONAL
OR
ORGANIZATION
OTHERWISE
OUTPUT
OV
OVERFLOW
*OWNER
<table>
<thead>
<tr>
<th>PAGE</th>
<th>PAGE-COUNTER</th>
<th>REMAINDER</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERCENT</td>
<td>PERFORM</td>
<td>RENAMES</td>
<td>RENAMING</td>
</tr>
<tr>
<td>PF</td>
<td>PH</td>
<td>REPLACING</td>
<td>REPORT</td>
</tr>
<tr>
<td>PIC</td>
<td>PICTURE</td>
<td>REPORTING</td>
<td>REPORTS</td>
</tr>
<tr>
<td>PLACES</td>
<td>PLUS</td>
<td>RERUN</td>
<td>RESERVE</td>
</tr>
<tr>
<td>POINT</td>
<td>POSITION</td>
<td>RESET</td>
<td>RETURN</td>
</tr>
<tr>
<td>POSITIVE</td>
<td>*PREPARED</td>
<td>REVERSED</td>
<td>REWIND</td>
</tr>
<tr>
<td>*PRINT-SWITCH</td>
<td>PRIORITY</td>
<td>REWRITE</td>
<td>RF</td>
</tr>
<tr>
<td>PROCEDURE</td>
<td>PROCEDURES</td>
<td>RH</td>
<td>RIGHT</td>
</tr>
<tr>
<td>PROCEED</td>
<td>*PROCESS</td>
<td>ROUNDED</td>
<td>RUN</td>
</tr>
<tr>
<td>PROCESSING</td>
<td>PROGRAM</td>
<td>*SA</td>
<td>SAME</td>
</tr>
<tr>
<td>PROGRAM-ID</td>
<td>PROTECT</td>
<td>SD</td>
<td>SEARCH</td>
</tr>
<tr>
<td>*QUEUE</td>
<td>QUOTE</td>
<td>SECTION</td>
<td>SECURITY</td>
</tr>
<tr>
<td>QUOTES</td>
<td>RANDOM</td>
<td>SEEK</td>
<td>SEGMENT-LIMIT</td>
</tr>
<tr>
<td>RANGE</td>
<td>RD</td>
<td>SELECT</td>
<td>SELECTED</td>
</tr>
<tr>
<td>READ</td>
<td>*RECEIVE</td>
<td>*SEND</td>
<td>SENTENCE</td>
</tr>
<tr>
<td>RECORD</td>
<td>RECORD-BLOCK</td>
<td>SEQUENCED</td>
<td>SEQUENTIAL</td>
</tr>
<tr>
<td>RECORD-MARK</td>
<td>RECORDING</td>
<td>SET</td>
<td>SIGN</td>
</tr>
<tr>
<td>RECORDS</td>
<td>REDEFINES</td>
<td>SIGNED</td>
<td>SIZE</td>
</tr>
<tr>
<td>REEL</td>
<td>REEL-NUMBER</td>
<td>SKIP</td>
<td>SORT</td>
</tr>
<tr>
<td>*REFERENCES</td>
<td>REFERENCE</td>
<td>SOURCE</td>
<td>SOURCE-COMPUTER</td>
</tr>
<tr>
<td>RELATIVE</td>
<td>RELEASE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SPACE
SPACES
SPECIAL-_NAMES
STANDARD
START
STATUS
STOP
*STRING
*SUB-QUEUE-1
*SUB-QUEUE-2
*SUB-QUEUE-3
SUB-Schema
SUBTRACT
SUM
*SUPERVISOR
SUPPRESS
SYMBOLIC
SYNC
SYNCHRONIZED

*TABLE
TALLY
TALLYING
TAPE
TERMINAL
TERMINATE
*TEST
*TEXT
THAN
THEN
THROUGH
THRU
TIME
TIMES

TO
TODAYS-DATE
TYPE

UNEQUAL
UNIT
*UNSTRING
UNTIL
UP
UPON
*UPPER-BOUND
*UPPER-BOUNDS
USAGE
USE
USING

VALUE
VALUES
VARYING
*VOLUME

WHEN
WITH
*WORDS
WORKING-STORAGE
WRITE

ZERO
ZEROES
ZEROS
### 64-CHARACTER SET COLLATING SEQUENCE

<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>COBOL Character</th>
<th>Display Code</th>
<th>Hollerith Punch (026)</th>
<th>Hollerith Punch (029)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>blank</td>
<td>55</td>
<td>no punch</td>
<td>no punch</td>
</tr>
<tr>
<td>01</td>
<td>(&lt;)</td>
<td>74</td>
<td>8-5</td>
<td>12-8-4</td>
</tr>
<tr>
<td>02</td>
<td>%</td>
<td>63</td>
<td>8-6</td>
<td>0-8-4</td>
</tr>
<tr>
<td>03</td>
<td>[*]</td>
<td>61</td>
<td>8-7</td>
<td>8-5</td>
</tr>
<tr>
<td>04</td>
<td>(\rightarrow)</td>
<td>65</td>
<td>0-8-5</td>
<td>0-8-5</td>
</tr>
<tr>
<td>05</td>
<td>(\equiv)</td>
<td>60</td>
<td>0-8-6</td>
<td>8-3</td>
</tr>
<tr>
<td>06</td>
<td>(\wedge)</td>
<td>67</td>
<td>0-8-7</td>
<td>12</td>
</tr>
<tr>
<td>07</td>
<td>(\dagger)</td>
<td>70</td>
<td>11-8-5</td>
<td>8-4</td>
</tr>
<tr>
<td>08</td>
<td>(\dagger)</td>
<td>71</td>
<td>11-8-6</td>
<td>0-8-7</td>
</tr>
<tr>
<td>09</td>
<td>(&gt;)</td>
<td>73</td>
<td>11-8-7</td>
<td>0-8-6</td>
</tr>
<tr>
<td>10</td>
<td>(\geq)</td>
<td>75</td>
<td>12-8-5</td>
<td>0-8-2</td>
</tr>
<tr>
<td>11</td>
<td>(\neg)</td>
<td>76</td>
<td>12-8-6</td>
<td>11-8-7</td>
</tr>
<tr>
<td>12</td>
<td>.</td>
<td>57</td>
<td>12-8-3</td>
<td>12-8-3</td>
</tr>
<tr>
<td>13</td>
<td>)</td>
<td>52</td>
<td>12-8-4</td>
<td>11-8-5</td>
</tr>
<tr>
<td>14</td>
<td>;</td>
<td>77</td>
<td>12-8-7</td>
<td>11-8-6</td>
</tr>
<tr>
<td>15</td>
<td>+</td>
<td>45</td>
<td>12</td>
<td>12-8-6</td>
</tr>
<tr>
<td>16</td>
<td>$</td>
<td>53</td>
<td>11-8-3</td>
<td>11-8-3</td>
</tr>
<tr>
<td>17</td>
<td>*</td>
<td>47</td>
<td>11-8-4</td>
<td>11-8-4</td>
</tr>
<tr>
<td>18</td>
<td>-</td>
<td>46</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>/</td>
<td>50</td>
<td>0-1</td>
<td>0-1</td>
</tr>
<tr>
<td>20</td>
<td>,</td>
<td>56</td>
<td>0-8-3</td>
<td>0-8-3</td>
</tr>
<tr>
<td>21</td>
<td>{</td>
<td>51</td>
<td>0-8-4</td>
<td>12-8-5</td>
</tr>
<tr>
<td>22</td>
<td>=</td>
<td>54</td>
<td>8-3</td>
<td>8-6</td>
</tr>
<tr>
<td>23</td>
<td>(\neq)\dagger</td>
<td>64</td>
<td>8-4</td>
<td>8-7</td>
</tr>
<tr>
<td>24</td>
<td>(&lt;)</td>
<td>72</td>
<td>12-0</td>
<td>12-8-2</td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>01</td>
<td>12-1</td>
<td>12-1</td>
</tr>
<tr>
<td>26</td>
<td>B</td>
<td>02</td>
<td>12-2</td>
<td>12-2</td>
</tr>
<tr>
<td>27</td>
<td>C</td>
<td>03</td>
<td>12-3</td>
<td>12-3</td>
</tr>
<tr>
<td>28</td>
<td>D</td>
<td>04</td>
<td>12-4</td>
<td>12-4</td>
</tr>
<tr>
<td>29</td>
<td>E</td>
<td>05</td>
<td>12-5</td>
<td>12-5</td>
</tr>
<tr>
<td>30</td>
<td>F</td>
<td>06</td>
<td>12-6</td>
<td>12-6</td>
</tr>
<tr>
<td>31</td>
<td>G</td>
<td>07</td>
<td>12-7</td>
<td>12-7</td>
</tr>
</tbody>
</table>

*Not in COBOL character set; may be present in data

\[\dagger\]COBOL quote character (\"\") is output on printer as \#
<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>COBOL Character</th>
<th>Display Code</th>
<th>Hollerith Punch (026)</th>
<th>Hollerith Punch (029)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>H</td>
<td>10</td>
<td>12-8</td>
<td>12-8</td>
</tr>
<tr>
<td>33</td>
<td>I</td>
<td>11</td>
<td>12-9</td>
<td>12-9</td>
</tr>
<tr>
<td>34</td>
<td>V</td>
<td>66</td>
<td>11-0</td>
<td>11-8-2</td>
</tr>
<tr>
<td>35</td>
<td>J</td>
<td>12</td>
<td>11-1</td>
<td>11-1</td>
</tr>
<tr>
<td>36</td>
<td>K</td>
<td>13</td>
<td>11-2</td>
<td>11-2</td>
</tr>
<tr>
<td>37</td>
<td>L</td>
<td>14</td>
<td>11-3</td>
<td>11-3</td>
</tr>
<tr>
<td>38</td>
<td>M</td>
<td>15</td>
<td>11-4</td>
<td>11-4</td>
</tr>
<tr>
<td>39</td>
<td>N</td>
<td>16</td>
<td>11-5</td>
<td>11-5</td>
</tr>
<tr>
<td>40</td>
<td>O</td>
<td>17</td>
<td>11-6</td>
<td>11-6</td>
</tr>
<tr>
<td>41</td>
<td>P</td>
<td>20</td>
<td>11-7</td>
<td>11-7</td>
</tr>
<tr>
<td>42</td>
<td>Q</td>
<td>21</td>
<td>11-8</td>
<td>11-8</td>
</tr>
<tr>
<td>43</td>
<td>R</td>
<td>22</td>
<td>11-9</td>
<td>11-9</td>
</tr>
<tr>
<td>44</td>
<td>*</td>
<td>62</td>
<td>0-8-2</td>
<td>12-8-7</td>
</tr>
<tr>
<td>45</td>
<td>S</td>
<td>23</td>
<td>0-2</td>
<td>0-2</td>
</tr>
<tr>
<td>46</td>
<td>T</td>
<td>24</td>
<td>0-3</td>
<td>0-3</td>
</tr>
<tr>
<td>47</td>
<td>U</td>
<td>25</td>
<td>0-4</td>
<td>0-4</td>
</tr>
<tr>
<td>48</td>
<td>V</td>
<td>26</td>
<td>0-5</td>
<td>0-5</td>
</tr>
<tr>
<td>49</td>
<td>W</td>
<td>27</td>
<td>0-6</td>
<td>0-6</td>
</tr>
<tr>
<td>50</td>
<td>X</td>
<td>30</td>
<td>0-7</td>
<td>0-7</td>
</tr>
<tr>
<td>51</td>
<td>Y</td>
<td>31</td>
<td>0-8</td>
<td>0-8</td>
</tr>
<tr>
<td>52</td>
<td>Z</td>
<td>32</td>
<td>0-9</td>
<td>0-9</td>
</tr>
<tr>
<td>53</td>
<td>:*</td>
<td>00</td>
<td>8-2</td>
<td>8-2</td>
</tr>
<tr>
<td>54</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>34</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>56</td>
<td>2</td>
<td>35</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>57</td>
<td>3</td>
<td>36</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>58</td>
<td>4</td>
<td>37</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>59</td>
<td>5</td>
<td>40</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>60</td>
<td>6</td>
<td>41</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>61</td>
<td>7</td>
<td>42</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>62</td>
<td>8</td>
<td>43</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>63</td>
<td>9</td>
<td>44</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

*Not in COBOL character set

††COBOL record mark
## ASCII COLLATING SEQUENCE

<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>Character</th>
<th>Display Code</th>
<th>Hollerith Punch (026)</th>
<th>Hollerith Punch (029)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>blank</td>
<td>55</td>
<td>no punch</td>
<td>no punch</td>
</tr>
<tr>
<td>01</td>
<td>*</td>
<td>62</td>
<td>0-8-2</td>
<td>12-8-7</td>
</tr>
<tr>
<td>02</td>
<td>&quot;</td>
<td>64</td>
<td>8-4</td>
<td>8-7</td>
</tr>
<tr>
<td>03</td>
<td>#</td>
<td>60</td>
<td>0-8-6</td>
<td>8-3</td>
</tr>
<tr>
<td>04</td>
<td>$</td>
<td>53</td>
<td>11-8-3</td>
<td>11-8-3</td>
</tr>
<tr>
<td>05</td>
<td>%</td>
<td>63</td>
<td>8-6</td>
<td>0-8-4</td>
</tr>
<tr>
<td>06</td>
<td>&amp;</td>
<td>67</td>
<td>0-8-7</td>
<td>12</td>
</tr>
<tr>
<td>07</td>
<td>'</td>
<td>61</td>
<td>8-7</td>
<td>8-5</td>
</tr>
<tr>
<td>08</td>
<td>(</td>
<td>51</td>
<td>0-8-4</td>
<td>12-8-5</td>
</tr>
<tr>
<td>09</td>
<td>)</td>
<td>52</td>
<td>12-8-4</td>
<td>11-8-5</td>
</tr>
<tr>
<td>10</td>
<td>*</td>
<td>47</td>
<td>11-8-4</td>
<td>11-8-4</td>
</tr>
<tr>
<td>11</td>
<td>+</td>
<td>45</td>
<td>12</td>
<td>12-8-6</td>
</tr>
<tr>
<td>12</td>
<td>,</td>
<td>56</td>
<td>0-8-3</td>
<td>0-8-3</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>46</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>14</td>
<td>.</td>
<td>57</td>
<td>12-8-3</td>
<td>12-8-3</td>
</tr>
<tr>
<td>15</td>
<td>/</td>
<td>50</td>
<td>0-1</td>
<td>0-1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>34</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>35</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>36</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>37</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>5</td>
<td>40</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>6</td>
<td>41</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>23</td>
<td>7</td>
<td>42</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>43</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>9</td>
<td>44</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>26</td>
<td>;</td>
<td>00</td>
<td>8-2</td>
<td>8-2</td>
</tr>
<tr>
<td>27</td>
<td>:</td>
<td>77</td>
<td>12-8-7</td>
<td>11-8-6</td>
</tr>
<tr>
<td>28</td>
<td>&lt;</td>
<td>74</td>
<td>8-5</td>
<td>12-8-4</td>
</tr>
<tr>
<td>29</td>
<td>=</td>
<td>54</td>
<td>8-3</td>
<td>8-6</td>
</tr>
<tr>
<td>30</td>
<td>&gt;</td>
<td>73</td>
<td>11-8-7</td>
<td>0-8-6</td>
</tr>
<tr>
<td>31</td>
<td>?</td>
<td>71</td>
<td>11-8-6</td>
<td>0-8-7</td>
</tr>
<tr>
<td>Collating Sequence</td>
<td>Character</td>
<td>Display Code</td>
<td>Hollerith Punch (026)</td>
<td>Hollerith Punch (029)</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>--------------</td>
<td>-----------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>32</td>
<td>@</td>
<td>70</td>
<td>11-8-5</td>
<td>8-4</td>
</tr>
<tr>
<td>33</td>
<td>A</td>
<td>01</td>
<td>12-1</td>
<td>12-1</td>
</tr>
<tr>
<td>34</td>
<td>B</td>
<td>02</td>
<td>12-2</td>
<td>12-2</td>
</tr>
<tr>
<td>35</td>
<td>C</td>
<td>03</td>
<td>12-3</td>
<td>12-3</td>
</tr>
<tr>
<td>36</td>
<td>D</td>
<td>04</td>
<td>12-4</td>
<td>12-4</td>
</tr>
<tr>
<td>37</td>
<td>E</td>
<td>05</td>
<td>12-5</td>
<td>12-5</td>
</tr>
<tr>
<td>38</td>
<td>F</td>
<td>06</td>
<td>12-6</td>
<td>12-6</td>
</tr>
<tr>
<td>39</td>
<td>G</td>
<td>07</td>
<td>12-7</td>
<td>12-7</td>
</tr>
<tr>
<td>40</td>
<td>H</td>
<td>10</td>
<td>12-8</td>
<td>12-8</td>
</tr>
<tr>
<td>41</td>
<td>I</td>
<td>11</td>
<td>12-9</td>
<td>12-9</td>
</tr>
<tr>
<td>42</td>
<td>J</td>
<td>12</td>
<td>11-1</td>
<td>11-1</td>
</tr>
<tr>
<td>43</td>
<td>K</td>
<td>13</td>
<td>11-2</td>
<td>11-2</td>
</tr>
<tr>
<td>44</td>
<td>L</td>
<td>14</td>
<td>11-3</td>
<td>11-3</td>
</tr>
<tr>
<td>45</td>
<td>M</td>
<td>15</td>
<td>11-4</td>
<td>11-4</td>
</tr>
<tr>
<td>46</td>
<td>N</td>
<td>16</td>
<td>11-5</td>
<td>11-5</td>
</tr>
<tr>
<td>47</td>
<td>O</td>
<td>17</td>
<td>11-6</td>
<td>11-6</td>
</tr>
<tr>
<td>48</td>
<td>P</td>
<td>20</td>
<td>11-7</td>
<td>11-7</td>
</tr>
<tr>
<td>49</td>
<td>Q</td>
<td>21</td>
<td>11-8</td>
<td>11-8</td>
</tr>
<tr>
<td>50</td>
<td>R</td>
<td>22</td>
<td>11-9</td>
<td>11-9</td>
</tr>
<tr>
<td>51</td>
<td>S</td>
<td>23</td>
<td>0-2</td>
<td>0-2</td>
</tr>
<tr>
<td>52</td>
<td>T</td>
<td>24</td>
<td>0-3</td>
<td>0-3</td>
</tr>
<tr>
<td>53</td>
<td>U</td>
<td>25</td>
<td>0-4</td>
<td>0-4</td>
</tr>
<tr>
<td>54</td>
<td>V</td>
<td>26</td>
<td>0-5</td>
<td>0-5</td>
</tr>
<tr>
<td>55</td>
<td>W</td>
<td>27</td>
<td>0-6</td>
<td>0-6</td>
</tr>
<tr>
<td>56</td>
<td>X</td>
<td>30</td>
<td>0-7</td>
<td>0-7</td>
</tr>
<tr>
<td>57</td>
<td>Y</td>
<td>31</td>
<td>0-8</td>
<td>0-8</td>
</tr>
<tr>
<td>58</td>
<td>Z</td>
<td>32</td>
<td>0-9</td>
<td>0-9</td>
</tr>
<tr>
<td>59</td>
<td>[</td>
<td>72</td>
<td>12-0 or 12-8-2</td>
<td>12-8-2 or 8-2</td>
</tr>
<tr>
<td>60</td>
<td>/</td>
<td>75</td>
<td>12-8-5</td>
<td>0-8-2</td>
</tr>
<tr>
<td>61</td>
<td>]</td>
<td>66</td>
<td>11-0 or 11-8-2</td>
<td>11-8-2 or 8-2</td>
</tr>
<tr>
<td>62</td>
<td>~</td>
<td>76</td>
<td>12-8-6</td>
<td>11-8-7</td>
</tr>
<tr>
<td>63</td>
<td>–</td>
<td>65</td>
<td>0-8-5</td>
<td>0-8-5</td>
</tr>
</tbody>
</table>
### 63-Character Set Collating Sequence

<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>COBOL Character</th>
<th>Display Code</th>
<th>Hollerith Punch (026)</th>
<th>Hollerith Punch (029)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>blank</td>
<td>55</td>
<td>no punch</td>
<td>no punch</td>
</tr>
<tr>
<td>01</td>
<td>≤</td>
<td>74</td>
<td>8-5</td>
<td>12-8-4</td>
</tr>
<tr>
<td>02</td>
<td>[</td>
<td>61</td>
<td>8-7</td>
<td>0-8-4</td>
</tr>
<tr>
<td>03</td>
<td>→</td>
<td>65</td>
<td>0-8-5</td>
<td>0-8-5</td>
</tr>
<tr>
<td>04</td>
<td>≡</td>
<td>60</td>
<td>0-8-6</td>
<td>8-3</td>
</tr>
<tr>
<td>05</td>
<td>∧</td>
<td>67</td>
<td>0-8-7</td>
<td>12</td>
</tr>
<tr>
<td>06</td>
<td>↑</td>
<td>70</td>
<td>11-8-5</td>
<td>8-4</td>
</tr>
<tr>
<td>07</td>
<td>↓</td>
<td>71</td>
<td>11-8-6</td>
<td>0-8-7</td>
</tr>
<tr>
<td>08</td>
<td>&gt;</td>
<td>73</td>
<td>11-8-7</td>
<td>0-8-6</td>
</tr>
<tr>
<td>09</td>
<td>≥</td>
<td>75</td>
<td>12-8-5</td>
<td>0-8-2</td>
</tr>
<tr>
<td>10</td>
<td>⊼</td>
<td>76</td>
<td>12-8-6</td>
<td>11-8-7</td>
</tr>
<tr>
<td>11</td>
<td>.</td>
<td>57</td>
<td>12-8-3</td>
<td>12-8-3</td>
</tr>
<tr>
<td>12</td>
<td>)</td>
<td>52</td>
<td>12-8-4</td>
<td>11-8-5</td>
</tr>
<tr>
<td>13</td>
<td>;</td>
<td>77</td>
<td>12-8-7</td>
<td>11-8-6</td>
</tr>
<tr>
<td>14</td>
<td>+</td>
<td>45</td>
<td>12</td>
<td>12-8-6</td>
</tr>
<tr>
<td>15</td>
<td>$</td>
<td>53</td>
<td>11-8-3</td>
<td>11-8-3</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>47</td>
<td>11-8-4</td>
<td>11-8-4</td>
</tr>
<tr>
<td>17</td>
<td>‐</td>
<td>46</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>18</td>
<td>/</td>
<td>50</td>
<td>0-1</td>
<td>0-1</td>
</tr>
<tr>
<td>19</td>
<td>,</td>
<td>56</td>
<td>0-8-3</td>
<td>0-8-3</td>
</tr>
<tr>
<td>20</td>
<td>(</td>
<td>51</td>
<td>0-8-4</td>
<td>12-8-5</td>
</tr>
<tr>
<td>21</td>
<td>=</td>
<td>54</td>
<td>8-3</td>
<td>8-6</td>
</tr>
<tr>
<td>22</td>
<td>≠↑</td>
<td>64</td>
<td>8-4</td>
<td>8-7</td>
</tr>
<tr>
<td>23</td>
<td>&lt;</td>
<td>72</td>
<td>12-0</td>
<td>12-8-2</td>
</tr>
<tr>
<td>24</td>
<td>A</td>
<td>01</td>
<td>12-1</td>
<td>12-1</td>
</tr>
<tr>
<td>25</td>
<td>B</td>
<td>02</td>
<td>12-2</td>
<td>12-2</td>
</tr>
<tr>
<td>26</td>
<td>C</td>
<td>03</td>
<td>12-3</td>
<td>12-3</td>
</tr>
<tr>
<td>27</td>
<td>D</td>
<td>04</td>
<td>12-4</td>
<td>12-4</td>
</tr>
<tr>
<td>28</td>
<td>E</td>
<td>05</td>
<td>12-5</td>
<td>12-5</td>
</tr>
<tr>
<td>29</td>
<td>F</td>
<td>06</td>
<td>12-6</td>
<td>12-6</td>
</tr>
<tr>
<td>30</td>
<td>G</td>
<td>07</td>
<td>12-7</td>
<td>12-7</td>
</tr>
<tr>
<td>31</td>
<td>H</td>
<td>10</td>
<td>12-8</td>
<td>12-8</td>
</tr>
</tbody>
</table>

*Not in COBOL character set; may be present in data
†COBOL quote character (""’) is output on printer as ≠
<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>COBOL Character</th>
<th>Display Code</th>
<th>Hollerith Punch (026)</th>
<th>Hollerith Punch (029)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>I</td>
<td>11</td>
<td>12-9</td>
<td>12-9</td>
</tr>
<tr>
<td>33</td>
<td>V</td>
<td>66</td>
<td>11-0</td>
<td>11-8-2</td>
</tr>
<tr>
<td>34</td>
<td>J</td>
<td>12</td>
<td>11-1</td>
<td>11-1</td>
</tr>
<tr>
<td>35</td>
<td>K</td>
<td>13</td>
<td>11-2</td>
<td>11-2</td>
</tr>
<tr>
<td>36</td>
<td>L</td>
<td>14</td>
<td>11-3</td>
<td>11-3</td>
</tr>
<tr>
<td>37</td>
<td>M</td>
<td>15</td>
<td>11-4</td>
<td>11-4</td>
</tr>
<tr>
<td>38</td>
<td>N</td>
<td>16</td>
<td>11-5</td>
<td>11-5</td>
</tr>
<tr>
<td>39</td>
<td>O</td>
<td>17</td>
<td>11-6</td>
<td>11-6</td>
</tr>
<tr>
<td>40</td>
<td>P</td>
<td>20</td>
<td>11-7</td>
<td>11-7</td>
</tr>
<tr>
<td>41</td>
<td>Q</td>
<td>50</td>
<td>11-8</td>
<td>11-8</td>
</tr>
<tr>
<td>42</td>
<td>R</td>
<td>22</td>
<td>11-9</td>
<td>11-9</td>
</tr>
<tr>
<td>43</td>
<td>]</td>
<td>62</td>
<td>0-8-2</td>
<td>12-8-7</td>
</tr>
<tr>
<td>44</td>
<td>S</td>
<td>23</td>
<td>0-2</td>
<td>0-2</td>
</tr>
<tr>
<td>45</td>
<td>T</td>
<td>24</td>
<td>0-3</td>
<td>0-3</td>
</tr>
<tr>
<td>46</td>
<td>U</td>
<td>25</td>
<td>0-4</td>
<td>0-4</td>
</tr>
<tr>
<td>47</td>
<td>V</td>
<td>26</td>
<td>0-5</td>
<td>0-5</td>
</tr>
<tr>
<td>48</td>
<td>W</td>
<td>27</td>
<td>0-6</td>
<td>0-6</td>
</tr>
<tr>
<td>49</td>
<td>X</td>
<td>30</td>
<td>0-7</td>
<td>0-7</td>
</tr>
<tr>
<td>50</td>
<td>Y</td>
<td>31</td>
<td>0-8</td>
<td>0-8</td>
</tr>
<tr>
<td>51</td>
<td>Z</td>
<td>32</td>
<td>0-9</td>
<td>0-9</td>
</tr>
<tr>
<td>52</td>
<td>:</td>
<td>63</td>
<td>8-2</td>
<td>8-2</td>
</tr>
<tr>
<td>53</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>1</td>
<td>34</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>35</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>56</td>
<td>3</td>
<td>36</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>57</td>
<td>4</td>
<td>37</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>58</td>
<td>5</td>
<td>40</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>59</td>
<td>6</td>
<td>41</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>60</td>
<td>7</td>
<td>42</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>61</td>
<td>8</td>
<td>43</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>62</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

*Not in COBOL character set
††COBOL record mark