CONTROL DATA®
CYBER 70
COMPUTER SYSTEMS
MODELS 72, 73, 74, 76
7600 COMPUTER SYSTEM
6000 COMPUTER SYSTEMS

COBOL INSTANT
MODELS 72, 73, 74 VERSION 4
MODEL 76 VERSION 1
7600 VERSION 1
6000 VERSION 4
CONTROL DATA®
CYBER 70
COMPUTER SYSTEMS
MODELS 72, 73, 74, 76
7600 COMPUTER SYSTEM
6000 COMPUTER SYSTEMS

COBOL INSTANT
MODELS 72, 73, 74 VERSION 4
MODEL 76 VERSION 1
7600 VERSION 1
6000 VERSION 4
New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

<table>
<thead>
<tr>
<th>REVISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Original printing.</td>
</tr>
<tr>
<td>(12-18-71)</td>
<td></td>
</tr>
</tbody>
</table>

Publication No.
60328400

Additional copies of this manual may be obtained from the nearest Control Data Corporation sales office.

© 1971
Control Data Corporation
Printed in the United States of America

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Software Documentation
215 MOFFETT PARK DRIVE
SUNNYVALE, CALIFORNIA 94086
CONTROL DATA CYBER 70, 6000 SERIES, 7600 COBOL

The COBOL language is designed to simplify the programming of business data processing operations; it produces easily modifiable source programs that result in shorter program development time and low program conversion costs. COBOL source and object programs run under the control of the SCOPE operating system.

This version of COBOL is designed for the CONTROL DATA® CYBER 70, 6000 Series and 7600 computers. It is upwards compatible with the COBOL developed by the American National Standards Institute (ANSI). This version provides many features in addition to all ANSI features. ANSI formats are printed in black.

Extensions to ANSI are printed in color:

- blue: All CDC extensions
- green: CDC 6000 Series and CDC CYBER 70/Models 72, 73, 74
- red: CDC 7600 and CDC CYBER 70/Model 76

Special Features:

- Mass storage input and output including indexed sequential 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
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.
COBOL NOTATION

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 verb 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 }

DIVISION

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

[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:

SOURCE-COMPUTER. computer-name.

format 1:

OBJECT-COMPUTER. 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:

OBJECT-COMPUTER: computer-name

[SEGMENT-LIMIT IS priority-number]

MEMORY SIZE integer { WORDS  \integers  \text{CHARACTERS} \integers \text{MODULES} \integers \}.

format 1:

SPECIAL- NAMES, COPY library-name

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

\{ \text{literal}-3 \} \text{word}-3 \{ \text{identifier}-3 \} \text{BY} \{ \text{literal}-4 \} \text{word}-4 \{ \text{identifier}-4 \} \ldots \}.

format 2:

SPECIAL- NAMES

SWITCH integer-1

\{ \text{IS \ mnemonic-name}-1 \[ \text{ON \ STATUS \ IS \ condition-name}-1 \]
[ \text{OFF \ STATUS \ IS \ condition-name}-2 \] \}

\{ \text{IS \ mnemonic-name}-2 \[ \text{OFF \ STATUS \ IS \ condition-name}-3 \]
[ \text{ON \ STATUS \ IS \ condition-name}-4 \] \}

{ \text{ON \ STATUS \ IS \ condition-name}-5
[ \text{OFF \ STATUS \ IS \ condition-name}-6 \] \}

\{ \text{OFF \ STATUS \ IS \ condition-name}-7
[ \text{ON \ STATUS \ IS \ condition-name}-8 \] \}

[non-numeric-literal \text{IS \ mnemonic-name}-1] \ldots

[implementor-name \text{IS \ mnemonic-name}-1] \ldots

[CURRENCY SIGN \text{IS \ literal}]

[DECIMAL-POINT \text{IS \ COMMA}]

[CONSOLE \text{IS \ mnemonic-name}]

[TERMINAL \text{IS \ mnemonic-name}].
INPUT-OUTPUT SECTION.

format 1:

FILE-CONTROL, COPY library-name

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

format 2:

FILE-CONTROL.

\{ SELECT [OPTIONAL] file-name-1 [RENAMEING file-name-2] \\
\text{ASSIGN TO [integer] implementor-name-1 [implementor-name-2] \\
\ldots [OR implementor-name-3 [implementor-name-4] \ldots] \\
\text{FOR MULTIPLE \{ REEL \} UNIT \} \\
\text{ERROR FILE IS file-name] \\
\text{RESERVE \{ NO integer \} ALTERNATE \{ \text{AREA \} \text{AREAS} \} ] \\
\text{FILE-LIMIT IS \{ data-name-1 \} literal-1 \\
\text{FILE-LIMITS ARE \{ data-name-2 \} literal-2 \\
\text{THRU \{ data-name-3 \} THRU \{ data-name-4 \} THROUGH \{ literal-3 \} THROUGH \{ literal-4 \} \ldots \\
\text{SEQUENTIAL STANDARD DIRECT } \\
\text{INDEXED SEQUENTIAL RELATIVE } \\
\text{ACCESS MODE IS \{ SEQUENTIAL RANDOM \} ] \\
\text{PROCESSING MODE IS SEQUENTIAL] 

6
[\{ACTUAL\} KEY IS data-name ]
[\{SYMBOLIC\}]
[\{NUMBER OF BLOCKS IS \} (data-name \} integer \}
[\{INDEX-LEVEL IS \} integer \}
[\{INDEX-LEVELS ARE \}]
[\{INDEX-BLOCK CONTAINS integer RECORDS \}]
[\{RECORD-BLOCK CONTAINS integer \} (RECORDS CHARACTERS \}]
[\{INDEX-PADDING IS integer PERCENT \}]
[\{DATA-PADDING IS integer PERCENT \} . ] ...

format 1:

I-O-CONTROL. COPY library-name

\[\{REPLACING \} (literal-1 \} word-1 \} identifier-1 \} BY \} (literal-2 \} word-2 \} identifier-2 \}
[\{literal-3 \} word-3 \} identifier-3 \} BY \} (literal-4 \} word-4 \} identifier-4 \} ... \].

format 2:

I-O CONTROL.

\[RERUN \} ON \{ file-name-1 \} implementor-name \}
\[ EVERY \} [\{END OF \} [\{REEL \} REEL UNIT \} OF file-name-2 \}
\} [\{integer-1 RECORDS \} integer-1 RECORDS \}
\} [\{integer-2 CLOCK-UNITS \} integer-2 CLOCK-UNITS \}
\} [\{condition-name \} condition-name \}
\[SAME \} [\{SORT \} SORT \}
\} AREA FOR file-name-1 \{ file-name-2 \} ... \]
\[MULTIPLE FILE TAPE CONTAINS file-name-1 \}
\} [\{POSITION integer-1 \} file-name-2 \}
\} [\{POSITION integer-2 \} ... \].
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</td>
<td>77 group</td>
<td>77 group</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</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</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POINT LOCATION</td>
<td>J I</td>
<td>J I</td>
<td>J I</td>
</tr>
<tr>
<td>SIGNED</td>
<td>J I</td>
<td>J I</td>
<td>J I</td>
</tr>
<tr>
<td>JUSTIFIED</td>
<td>J I</td>
<td>J I</td>
<td>J I</td>
</tr>
<tr>
<td>SYNCHRONIZED</td>
<td>J I</td>
<td>J I</td>
<td>J I</td>
</tr>
<tr>
<td>PICTURE</td>
<td>J I</td>
<td>J I</td>
<td>J I</td>
</tr>
<tr>
<td>Editing Clauses</td>
<td>J I</td>
<td>J I</td>
<td>J I</td>
</tr>
<tr>
<td>COPY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUE</td>
<td>K K C</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>FILLER</td>
<td>I I I</td>
<td>I I</td>
<td>I I</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 |
DATA DIVISION

[FILE SECTION.]

[COMMON-STORAGE SECTION.]

[WORKING-STORAGE SECTION.]

[SECONDARY-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:

\{
SD
FD
\}
file-name COPY library-name

REPLACING

\{
\}

literal-1
word-1
identifier-1

BY

\{
\}

literal-2
word-2
identifier-2

Replacingspace

\{
\}

literal-3
word-3
identifier-3

BY

\{
\}

literal-4
word-4
identifier-4

...  

format 2:

\{
SD
FD
\}
file-name

BLOCK CONTAINS [integer-1 TO] integer-2

\{
\}

RECORDS
CHARACTERS

\{
\}

DATA

RECORD IS

RECORDS ARE

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

\{
\}

REPORT IS

REPORTS ARE

report-name-1 [report-name-2] ...

[FILE CONTAINS ABOUT integer RECORDS]
LABEL { RECORDS ARE } { STANDARD OMITTED }
{ data-name-1 [data-name-2] ... }

If label records are STANDARD:

[ VALUE OF [ { ID IDENTIFICATION } IS { literal-1 data-name-1 } ]

[ DATE-WRITTEN IS { literal-2 data-name-2 } ]

[ EDITION-NUMBER IS { literal-3 data-name-3 } ]

[ REEL-NUMBER IS { literal-4 data-name-4 } ]

[ RETENTION-CYCLE IS { literal-5 data-name-5 } ] ]

If label records are a data-name:

[ VALUE OF data-name-3 IS { literal-1 data-name-4 }

[ data-name-5 IS { literal-2 data-name-6 } ] ... ]

[ VALUE OF ENDING-TAPE-LABEL-IDENTIFIER IS { literal-3 data-name-7 } ]

[ LINAGE IS { integer identifier } LINES ]

[ RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS

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

[ RECORDING MODE IS { BINARY DECIMAL } [ { HIGH LOW } HYPER DENSITY ] ]

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

format 1:

\[
\begin{align*}
\{01\} & \text{ data-name COPY library-name [FROM LIBRARY]} \\
\{02-49\} & \\
\end{align*}
\]

REPLACING \[
\begin{align*}
\{\text{word-1} \} & \text{ BY } \{\text{word-2} \} \\
\{\text{identifier-1} \} & \text{ literal-1} \\
\{\text{identifier-2} \} & \text{ literal-2} \\
\end{align*}
\]

\[
\begin{align*}
\{\text{word-3} \} & \text{ BY } \{\text{word-4} \} \\
\{\text{identifier-3} \} & \text{ literal-3} \\
\{\text{identifier-4} \} & \text{ literal-4} \\
\end{align*}
\]

... .

format 2:

level-number data-name-1 [REDEFINES identifier]

COPY data-name-2 FROM SOURCE.

format 3:

level-number \[
\begin{align*}
\{\text{data-name [REDEFINES identifier]} \} \\
\{\text{FILLER} \} \\
\end{align*}
\]

\[
\begin{align*}
\{\text{BWZ} \} & \\
\{\text{BLANK WHEN ZERO} \} \\
\end{align*}
\]

\[
\begin{align*}
\{\text{CHECK PROTECT} \} & \\
\{\text{FLOAT DOLLAR SIGN} \} & \\
\{\text{FLOAT CURRENCY SIGN} \} & \\
\{\text{ZERO SUPPRESS} \} & \text{ [LEAVING integer PLACES]} \\
\{\text{CLASS IS} \} & \\
\{\text{ALPHABETIC} \} & \\
\{\text{NUMERIC} \} & \\
\{\text{ALPHANUMERIC} \} & \\
\{\text{AN} \} & \\
\end{align*}
\]

\[
\begin{align*}
\{\text{JUST} \} & \\
\{\text{JUSTIFIED} \} & \text{ RIGHT} \\
\end{align*}
\]
OCCURS integer-1 [ TO integer-2 ] TIMES

[ DEPENDING ON data-name-1 ]

\[
\begin{cases}
\text{ASCENDING} \\
\text{DESCENDING}
\end{cases}
\] KEY IS data-name-2 [ data-name-3 ] ...

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

\[
\{ \text{PIC} \\
\{ \text{PICTURE} \} \} \text{ IS character-string}
\]

POINT LOCATION IS \{ \text{LEFT} \\
\{ \text{RIGHT} \} \} integer PLACES

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

\[
\begin{cases}
\text{SIGNED} \\
\{ \text{SIGN IS data-name} \}
\end{cases}
\]

SIZE IS integer \[
\begin{cases}
\{ \text{CHARACTERS} \\
\{ \text{DIGITS} \}
\end{cases}
\]

\[
\begin{cases}
\text{SYNC} \\
\{ \text{SYNCHRONIZED} \}
\end{cases}
\]

\[
\begin{cases}
\text{LEFT} \\
\{ \text{RIGHT} \}
\end{cases}
\]

[ USAGE IS ]

\[
\begin{cases}
\text{COMP} \\
\text{COMP-1} \\
\text{COMP-2} \\
\text{COMP-2} \\
\text{COMP-2} \\
\text{DISPLAY} \\
\text{INDEX}
\end{cases}
\]

[ VALUE IS literal ].

format 4:

66 data-name RENAMES identifier-1 \[
\begin{cases}
\text{THRU} \\
\{ \text{THROUGH} \}
\end{cases}
\] identifier-2.

format 5:

88 condition-name \[
\begin{cases}
\text{VALUE IS} \\
\{ \text{VALUES ARE} \}
\end{cases}
\] literal-1

\[
\begin{cases}
\text{THRU} \\
\{ \text{THROUGH} \}
\end{cases}
\] literal-2 \[
\begin{cases}
\text{THRU} \\
\{ \text{THROUGH} \}
\end{cases}
\] literal-3 \[
\begin{cases}
\text{THRU} \\
\{ \text{THROUGH} \}
\end{cases}
\] literal-4 ...
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 } BY {
{l literal-2 word-2 identifier-2 } { literal-3 word-3 identifier-3 } {
\qquad \text{BY } \{ \text{literal-4 word-4 identifier-4} \} \ldots \}
}

format 2:

RD report-name [WITH CODE mnemonic-name]

\[
\begin{array}{l}
\{ \text{CONTROL IS} \} \{ \text{CONTROL ARE} \} \{ \text{FINAL} \} \{ \text{FINAL identifier-1[identifier-2]} \ldots \} \\
\{ \text{LIMIT IS} \} \{ \text{LIMIT ARE} \} \{ \text{LINE} \} \{ \text{LINES} \} \\
\{ \text{HEADING integer-2} \} \{ \text{FIRST DETAIL integer-3} \} \\
\{ \text{LAST DETAIL integer-4} \} \{ \text{FOOTING integer-5} \}
\end{array}
\]

Report Group Description Entry (Report Section Only)

format 1:

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

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

\[\begin{array}{l}
\{ \text{literal-3 word-3 identifier-3} \} \text{BY } \{ \text{literal-4 word-4 identifier-4} \} \ldots \}
\end{array}\]
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

15
Report Element Description (Report Section Only)

level-number [data-name]

[  { BLANK WHEN ZERO } ]
  BWZ

[  { CHECK PROTECT }
      { FLOAT DOLLAR SIGN }
      { FLOAT CURRENCY SIGN }
      [ LEAVING integer PLACES ]
      ZERO SUPPRESS
  ]

[  { 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 }

[SIGNED]
  SIGN IS data-name

[SIZE IS integer { CHARACTERS }
                 DIGITS ]
SOURCE IS
{ [SELECTED] identifier
  LINE-COUNTER
  PAGE-COUNTER
  TODAYS-DATE
}

SUM identifier-1[identifier-2] . . . [UPON data-name]
VALUE IS literal

[[USAGE IS] DISPLAY].

TYPE clause allowed if level 01

NEXT GROUP clause allowed if level 01
## USAGE SPECIFICATIONS

<table>
<thead>
<tr>
<th>Element</th>
<th>Upper Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>data-name</td>
<td>30 characters, 5 levels of qualifications</td>
</tr>
<tr>
<td>elementary item/literal</td>
<td>255 characters/digits</td>
</tr>
<tr>
<td>PERFORM nesting</td>
<td>15 levels in separate overlays, no limit in main overlay</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>100 procedure names</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>
</tbody>
</table>
### 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>AN†</td>
<td>AN</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>AN†</td>
<td>AN†</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>AN</td>
<td>AN</td>
</tr>
<tr>
<td>Elem. Edit Num.</td>
<td>X</td>
<td>TD AN</td>
<td>X AN</td>
<td>X</td>
<td>AN-Edit</td>
<td>AN</td>
<td>AN</td>
<td>AN</td>
</tr>
<tr>
<td>Elem. Edit AN</td>
<td>X</td>
<td>TD AN</td>
<td>X AN</td>
<td>AN†</td>
<td>Edit</td>
<td>AN-Edit</td>
<td>AN-Edit</td>
<td>AN</td>
</tr>
<tr>
<td>Group AN</td>
<td>TD AN</td>
<td>TD AN</td>
<td>TD AN</td>
<td>AN</td>
<td>AN-Edit</td>
<td>AN-Edit</td>
<td>AN-Edit</td>
<td>AN</td>
</tr>
<tr>
<td>Group Binary &amp; Mixed</td>
<td>TD AN</td>
<td>TD AN</td>
<td>TD AN</td>
<td>AN</td>
<td>AN-Edit</td>
<td>AN-Edit</td>
<td>AN-Edit</td>
<td>AN</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>AN</td>
<td>AN</td>
</tr>
<tr>
<td>Literal &amp; Fig. Cons. AN</td>
<td>X</td>
<td>TD AN</td>
<td>X AN</td>
<td>X</td>
<td>AN-Edit</td>
<td>AN</td>
<td>AN</td>
<td>AN</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>AN</td>
<td>AN</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.

- X Illegal
- AN Alphanumeric
- AN-Edit Alphanumeric edited
- Conv. Conversion prior to move
- Edit Numeric edited
- Num. Numeric
- Num. Bin. Numeric binary
- TD Trivial diagnostic issued
PROCEDURE DIVISION.

PROCEDURE DIVISION. [USING parameter-list].

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 } ] ...
  literal-1

identifier-3 [ROUNDED]

[ON SIZE ERROR imperative-statement]

ADD { identifier-1 }
  [ { identifier-2 } ] ... TO
  literal-1

identifier-3 [ROUNDED] [identifier-4 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

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

  literal-1

  literal-2

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 { routine-name } [USING identifier-1 [identifier-2] ...].

CLOSE file-name-1 [ { UNIT } [ WITH { NO REWIND } ] ]

file-name-2 [ { UNIT } [ WITH { NO REWIND } ] ] ...

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

{ FROM = EQUALS } { literal arithmetic-expression } identifier-3

[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 RECORD FROM file-name

[INVALID KEY imperative-statement]

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

[UPON mnemonic-name]

DIVIDE { identifier-1 } INTO identifier-2 [ROUNDED]

[identifier-3 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

21
DIVIDE \{ identifier-1 \} \{ BY \} \{ identifier-2 \} \{ INTO \} \{ literal-1 \} \{ literal-2 \}

GIVING identifier-3 [ROUNDED] [identifier-4 [ROUNDED]] ...

[ON SIZE ERROR imperative-statement]

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

| ENTER \{ CALL \} \{ [language-name] routine-name \{ USING parameter-list \}. \}

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

EXAMINE identifier

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

\{ REPLACING \{ ALL \{ LEADING \{ [UNTIL] FIRST \} \} \} \{ literal-3 \{ BY literal-4 \} \}

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:

\[
\begin{align*}
\text{GREATER THAN} & \quad \text{GR} \\
\text{LESS THAN} & \quad \text{LS} \\
\text{IS \{ NOT \}} & \quad \text{GREATER-EQUAL TO} \\
\text{GO} & \quad \text{LESS-EQUAL TO} \\
\text{EQ} & \quad \text{EQUAL \{ TO \}} \\
\text{IS UNEQUAL TO} & \quad \text{EQUALS} \\
\text{EXCEEDS} & \quad \text{IS NO} \\
\text{IS NGR} & \quad \text{IS NLS} \\
\{ \text{identifier-1} \} & \quad \{ \text{identifier-2} \} \\
\{ \text{literal-1} \} & \quad \{ \text{literal-2} \} \\
\{ \text{formula-1} \} & \quad \{ \text{formula-2} \} \\
\{ \text{identifier} \} & \quad \{ \text{identifier} \} \\
\{ \text{formula} \} & \quad \text{IS \{ NOT \}} \\
\{ \text{POSITIVE} \} & \quad \text{NEGATIVE} \\
\text{ZERO} & \quad \text{NUMERIC} \\
\text{ALPHABETIC} & \quad \text{[NOT]} \\
\{ \text{condition-name} \} & \quad \{ \text{switch-status-name} \} \\
\text{INITIATE} & \quad \{ \text{report-name-1} \} \quad \{ \text{report-name-2} \} \ldots \\
\{ \text{ALL} \} & \quad \text{MOVE} \\
\{ \text{CORR} \} \quad \{ \text{CORRESPONDING} \} \quad \{ \text{identifier-1} \} \\
\{ \text{literal-1} \} \quad \{ \text{identifier-1} \} \quad \{ \text{TO} \} \\
\text{identifier-2} \{ \text{identifier-3} \} \ldots
\end{align*}
\]
MULTIPLY \{identifier-1\} BY identifier-2 [ROUNDED]

[identifier-3 [ROUNDED] ...]

[ON SIZE ERROR imperative-statement]

MULTIPLY \{identifier-1\} BY \{identifier-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

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

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

\{identifier integer\} TIMES

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

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

VARYING { index-name-1 } FROM { literal-1 index-name-2 } BY
{ identifier-1 identifier-2 } UNTIL condition-1

{ literal-2 identifier-3 } AFTER { index-name-3 identifier-4 }

FROM { literal-3 index-name-4 } BY { literal-4 identifier-5 }
{ index-name-4 identifier-6 } UNTIL condition-2

[ AFTER { index-name-5 identifier-7 } FROM { literal-5 index-name-6 } BY
{ identifier-7 identifier-8 } UNTIL condition-3 ]

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

READ file-name RECORD [INTO identifier]
[MAJOR KEY IS data-name] INVALID KEY imperative-statement

RELEASE record-name [FROM identifier]

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

REWRITE record-name [FROM identifier]

[INVALID KEY imperative-statement]

SEARCH identifier-1 [ VARYING { index-name identifier-2 } ]

[AT END imperative-statement-1]

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

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

\{ 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] ...

\[ \{ DESCENDING \}
\{ ASCENDING \} \]
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

STOP \{ literal \}
\{ RUN \}
SUBTRACT \{identifier-1\} \[\{identifier-2\}\] ... FROM identifier-3

[ROUNDED] [identifier-4 [ROUNDED]] ... [ON SIZE ERROR imperative-statement]

SUBTRACT \{identifier-1\} \[\{identifier-2\}\] ... FROM \{identifier-3\} \[\{liter-3\}\]

GIVING identifier-4 [ROUNDED]

(identifier-5 [ROUNDED]) ... [ON SIZE ERROR imperative-statement]

SUBTRACT \{CORR CORRESPONDING\} identifier-1 FROM identifier-2 [ROUNDED] [identifier-3 [ROUNDED]] ...

[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

USE \{BEFORE \{REEL \{FILE \{UNIT \} \} \} \} STANDARD \{BEGINNING \{ENDING \} \}

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]

\[
\begin{bmatrix}
\{ \text{BEFORE} \} & \text{ADVANCING} & \{ \text{identifier}-2 \text{ LINES} \} \\
\{ \text{AFTER} \} & \text{integer LINES} & \text{mnemonic-name} \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
\{ \text{AT} \} & \{ \text{END-OF-PAGE} \} & \text{imperative-statement} \\
\{ \text{EOP} \} & \text{imperative-statement} \\
\end{bmatrix}
\]

WRITE record-name [FROM identifier]

\[
\begin{bmatrix}
\{ \text{INVALID KEY} \} & \text{imperative-statement} \\
\end{bmatrix}
\]
### COBOL CONTROL CARD

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (Blank Conversion)</td>
<td>A</td>
<td>treats leading blanks as zeros</td>
</tr>
<tr>
<td>B (Binary Output)</td>
<td>absent</td>
<td>relocatable binary file on file LGO</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>binary output on file fn</td>
</tr>
<tr>
<td></td>
<td>B = LGO</td>
<td>suppress binary output</td>
</tr>
<tr>
<td>BUF (Buffer Size)</td>
<td>BUF</td>
<td>selects buffer size by method of version 3.0 COBOL</td>
</tr>
<tr>
<td>C (Copy Default)</td>
<td>C</td>
<td>uses version 3.0 COPY mode; to copy from library, FROM LIBRARY must be specified</td>
</tr>
<tr>
<td>D (Execution Abort)</td>
<td>D</td>
<td>prevents execution of program if E diagnostic occurs</td>
</tr>
<tr>
<td>E (EDITLIB)</td>
<td>E = fn</td>
<td>using EDITLIB, add object code to system library</td>
</tr>
<tr>
<td>F (Computational Modification)</td>
<td>F</td>
<td>interprets COMPUTATIONAL items as COMPUTATIONAL-1</td>
</tr>
<tr>
<td>H (BCOMMON)</td>
<td>H</td>
<td>BCOMMON replaces blank common as buffer area</td>
</tr>
<tr>
<td>I (Source Input)</td>
<td>absent</td>
<td>INPUT assumed</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>source input on file fn</td>
</tr>
<tr>
<td></td>
<td>I = INPUT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I = fn</td>
<td></td>
</tr>
</tbody>
</table>
L (List)

absent L

LX
LR
LC
LO
LM
L = fn
L = 0

normal listing on OUTPUT
extended diagnostics
cross reference pointers

copy from library
object code in octal
data map
output on file fn
suppress list output

N (Non-ANSI Diagnostic)

N
diagnoses any non-ANSI feature

O (Compiler Options)

O = X
O = R
O = O
O = C
O = M
extended diagnostics
cross reference pointers
object code in octal
copy from library
data map

OB (Overlay Binary)

OB
OB = LG02
OB = fn
binary output on LG02
binary output from overlay segments put on file fn

OL (Optimizer Level)

OL = 0
OL = 1
OL = 2
OL = 3

no optimizations
program flow optimization

machine instruction optimization
all optimizations
P (ANSI execution)  P  allows non-ANSI reserved words; selects N parameter

S (Source Library)  absent  
  S  
  S = COLIB  
  S = fn  from file fn

SUB (Subcompile)  SUB  suppresses all data division binary output except from working and constant storage

T (Tape Sort)  T  sort requests tape sort

U (ASCII Collating)  U  use ASCII collating sequence

W (Initialize Overlays)  W  uses version 3.0 method of treating independent segments: they are available in last used state

Z (3.0 Compatibility)  Z  provides compatibility with version 3.0 COBOL: selects parameters BUF, C, and W
# 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>
<tr>
<td>8</td>
<td>Division name&lt;br&gt;Section name&lt;br&gt;Paragraph name&lt;br&gt;File description&lt;br&gt;Record description level number</td>
</tr>
<tr>
<td>12</td>
<td>Record description data name&lt;br&gt;First sentence of a paragraph&lt;br&gt;File name&lt;br&gt;Continuation of a data description or a sentence</td>
</tr>
<tr>
<td>73 – 80</td>
<td>Identification (optional)</td>
</tr>
</tbody>
</table>

- **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, and 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**: Begins 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 SOURCE DECKS

PROCEDURE DIVISION.

DATA DIVISION.

ENVIRONMENT DIVISION.

IDENTIFICATION DIVISION.
COBOL COMPILATION

(end of job)

COBOL source deck

(end of record)

COBOL.

(job card)
EXECUTION

6 (end of job)
7
8
9

data deck
7 (end of record)
8
9
COBOL source deck
7 (end of record)
8
9
LGO.
COBOL.
(job card)
<table>
<thead>
<tr>
<th>COBOL RESERVED WORD LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Indicates word not implemented.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ABOUT</th>
<th>CHARACTERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT</td>
<td>CHECK</td>
</tr>
<tr>
<td>ACCESS</td>
<td>CLASS</td>
</tr>
<tr>
<td>ACTUAL</td>
<td>CLOCK-UNITS</td>
</tr>
<tr>
<td>ADD</td>
<td>CLOSE</td>
</tr>
<tr>
<td>*ADDRESS</td>
<td>COBOL</td>
</tr>
<tr>
<td>ADVANCING</td>
<td>CODE</td>
</tr>
<tr>
<td>AFTER</td>
<td>COLUMN</td>
</tr>
<tr>
<td>ALL</td>
<td>COMMA</td>
</tr>
<tr>
<td>ALPHABETIC</td>
<td>COMMON-STORAGE</td>
</tr>
<tr>
<td>ALPHANUMERIC</td>
<td>COMP</td>
</tr>
<tr>
<td>ALTER</td>
<td>COMP-1</td>
</tr>
<tr>
<td>ALTERNATE</td>
<td>COMP-2</td>
</tr>
<tr>
<td>AN</td>
<td>COMPASS</td>
</tr>
<tr>
<td>AND</td>
<td>COMPUTATIONAL</td>
</tr>
<tr>
<td>*ANSIB</td>
<td>COMPUTATIONAL-1</td>
</tr>
<tr>
<td>*APPLY</td>
<td>COMPUTATIONAL-2</td>
</tr>
<tr>
<td>ARE</td>
<td>COMPUTE</td>
</tr>
<tr>
<td>AREA</td>
<td>CONFIGURATION</td>
</tr>
<tr>
<td>AREAS</td>
<td>CONSOLE</td>
</tr>
<tr>
<td>ASCENDING</td>
<td>CONSTANT</td>
</tr>
<tr>
<td>AREAS</td>
<td>CONTAINS</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>CONTROL</td>
</tr>
<tr>
<td>AT</td>
<td>CONTROLS</td>
</tr>
<tr>
<td>AUTHOR</td>
<td>CONVERSION</td>
</tr>
<tr>
<td>**BCD</td>
<td>COPY</td>
</tr>
<tr>
<td>BEFORE</td>
<td>CORR</td>
</tr>
<tr>
<td>BEGINNING</td>
<td>CORRESPONDING</td>
</tr>
<tr>
<td>BEGINNING-FILE-LABEL</td>
<td>*COUNT</td>
</tr>
<tr>
<td>BEGINNING-TAPE-LABEL</td>
<td>CREATE</td>
</tr>
<tr>
<td>BINARY</td>
<td>CURRENCY</td>
</tr>
<tr>
<td>**BITS</td>
<td>DATA</td>
</tr>
<tr>
<td>BLANK</td>
<td>DATA-PADDING</td>
</tr>
<tr>
<td>BLOCK</td>
<td>DATE</td>
</tr>
<tr>
<td>BLOCKS</td>
<td>DATE-COMPILED</td>
</tr>
<tr>
<td>BWZ</td>
<td>DATE-WRITTEN</td>
</tr>
<tr>
<td>BY</td>
<td>DAY</td>
</tr>
<tr>
<td>CALL</td>
<td>DE</td>
</tr>
<tr>
<td>CANCEL</td>
<td>DECIMAL</td>
</tr>
<tr>
<td>**CD</td>
<td>DECIMAL-POINT</td>
</tr>
<tr>
<td>CF</td>
<td>DECLARATIVES</td>
</tr>
<tr>
<td>CH</td>
<td>DELETE</td>
</tr>
<tr>
<td>CHARACTER</td>
<td>*DELEMITER</td>
</tr>
<tr>
<td>**DELEMITED</td>
<td></td>
</tr>
</tbody>
</table>
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
LQ
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
NQ
NUMBER
NUMERIC

OBJECT-COMPUTER
OCCURS
OF
OFF
OH
OMITTED
ON
OPEN
OPTIONAL
OR
ORGANIZATION
OTHERWISE
OUTPUT
OV
OVERFLOW
*OWNER
PAGE
PAGE-COUNTER
PERCENT
PERFORM
PF
PH
PIC
PICTURE
PLACES
PLUS
POINT
POSITION
POSITIVE
*PREPARED
*PRINT-SWITCH
PRIORITY
PROCEDURE
PROCEDURES
PROCEED
*PROCESS
PROCESSING
PROGRAM
PROGRAM-ID
PROTECT

*QUEUE
QUOTE
QUOTES

RANDOM
RANGE
RD
READ
*RECEIVE
RECORD
RECORD-BLOCK
RECORD-MARK
RECORDING
RECORDS
REDEFINES
REEL
REEL-NUMBER
*REFERENCES
RELATIVE
RELEASE

REMAINDER
REMARKS
RENAMES
RENAMEING
REPLACING
REPORT
REPORTING
REPORTS
RENEW
RESERVE
RESET
RETENTION
RETENTION-CYCLE
RETURN
REVERSED
REWIND
REWRITE
RF
RH
RIGHT
ROUNDED
RUN

*SA
SAME
SD
SEARCH
SECONDARY-STORAGE
SECTION
SECURITY
SEEK
SEGMENT-LIMIT
SELECT
SELECTED
*SEND
SENTENCE
SEQUENCED
SEQUENTIAL
SET
SIGN
SIGNED
SIZE
SKIP
SORT
SOURCE
SOURCE-COMPUTER
SPACE
SPACES
SPECIAL- NAMES
STANDARD
STATUS
STOP
*STRING
*SUB-QUEUE-1
*SUB-QUEUE-2
*SUB-QUEUE-3
SUBTRACT
SUM
*SUPERVISOR
SUPPRESS
SWITCH
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

41
<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>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>→*</td>
<td>65</td>
<td>0-8-5</td>
<td>0-8-5</td>
</tr>
<tr>
<td>05</td>
<td>≡*</td>
<td>60</td>
<td>0-8-6</td>
<td>8-3</td>
</tr>
<tr>
<td>06</td>
<td>∧*</td>
<td>67</td>
<td>0-8-7</td>
<td>12</td>
</tr>
<tr>
<td>07</td>
<td>↑*</td>
<td>70</td>
<td>11-8-5</td>
<td>8-4</td>
</tr>
<tr>
<td>08</td>
<td>↓*</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>≥*</td>
<td>75</td>
<td>12-8-5</td>
<td>0-8-2</td>
</tr>
<tr>
<td>11</td>
<td>¬*</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>≠†</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
†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>J</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
<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>
</tbody>
</table>
## ASCII COLLATING SEQUENCE (continued)

<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>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</td>
<td>or 12-8-2</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</td>
<td>or 11-8-2</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 ≠
### 63-CHARACTER SET COLLATING SEQUENCE (continued)

<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