

NOTATION

TECO-8	refers to OS/8 TECO V5
TECO-10	refers to TOPS-10 TECO V24
TECO-11	refers to TECO-11 V29
RSX-11	encompasses RSX-11/D, RSX-11/M, IAS and VAX
\$	represents an ESCAPE (ASCII code 33, which is traditionally referred to as ALTMODE)
$\uparrow x$	represents a single control character (CONTROL/x)
$\uparrow x$	represents either $\uparrow x$ or the two-character sequence CARET-x (^x)
n	represents a numeric value specified by a digit string or TECO command that returns a value

Color Coding

black	represents features of all DEC TECO's [exceptions noted in brackets]
red	represents features of TECO-10 only
blue	represents features of TECO-11 only

Order No. AV-D530A-TK

Copyright © 1978 by Digital Equipment Corporation

INVOKING TECO

Effective Initial Command

OS/8

R TECO

RUN dev:TECO

MAKE filespec

EWfilespec\$

TECO filespec

EBfilespec\$Y

TECO filespec1 < filespec2

ERfilespec2\$EWfilespec1\$Y

TECO

[note 1]

MUNG filespec

ERfilespec\$YHXYHKMY

MUNG filespec,text

ERfilespec\$YHXYHKltext\$MY

RT-11

R TECO

RUN dev:TECO

EDIT/TECO filespec

EBfilespec\$Y [note 2]

Switches on EDIT Command:

/CREATE

EWfilespec\$

/INSPECT

ERfilespec\$Y

/OUTPUT:filespec2

ERfilespec\$EWfilespec2\$

/EXECUTE

ERfilespec\$YHXZHKMZ

/EXECUTE:text

ERfilespec\$YHXZHKltext\$MZ

RSTS/E

TECO

[note 1]

RUN \$TECO

RUN dev:TECO[p,pn]

TECO filespec

EBfilespec\$Y

TECO filespec1=filespec2

ERfilespec2\$EWfilespec1\$Y

TECO/INSPECT filespec

ERfilespec\$Y

MAKE filespec

EWfilespec\$

MUNG filespec

Elfilespec\$

MUNG filespec,text

ltext\$Elfilespec\$

RSX-11

TECO

TECO filespec

EBfilespec\$Y

TECO filespec1=filespec2

ERfilespec2\$EWfilespec1\$Y

MAKE filespec

EWfilespec\$

MUNG filespec

Elfilespec\$

MUNG filespec,text

ltext\$Elfilespec\$

TOPS-10

R TECO

RUN dev:TECO [p,pn]

[note 3]

MAKE filespec

EWfilespec\$

TECO filespec

EBfilespec\$Y

TECO filespec1=filespec2

EBfilespec1\$Y

MAKE filespec1=filespec2

ERfilespec2\$EWfilespec1\$Y

TECO

[note 1]

MAKE

[note 4]

IMMEDIATE MODE EDITING COMMANDS

\$\$	Executes command string
<rubout>	Deletes last character in command string
(↑C)	Exits from TECO [note 5]
(↑U)	Deletes current line in command string
(↑G) (↑G)	Deletes the entire command string
(↑G) <space>	Types current line of command string
(↑G) *	Types entire command string [not in TECO-10]

IMMEDIATE MODE AIDS

You must type these immediate mode commands as the first command after TECO's prompt.

?	Types command string up to command that caused an error [note 6]
/	Types detailed explanation of error
*q	Stores previous command string in Q-register q [note 7] [not in TECO-8]

EXECUTION-TIME COMMANDS

(↑C)	Stops command execution [note 8]
(↑O)	Turns off type out; nullified by a second (↑O)
(↑S)	Freezes terminal output; nullified by a (↑Q) [note 9]

FILE SPECIFICATION COMMANDS

ERfilespec\$	Opens file for input
EWfilespec\$	Opens file for output [note 10]
EBfilespec\$	Opens file for input and output with backup protection
EZfilespec\$	Zeros (non-disk) directory and opens file for output

nEM

Positions Magtape:

- EM Rewinds
- 1EM Rewinds
- 3EM Writes EOF
- 6EM Skips one record
- 7EM Backspaces one record
- 8EM Skips to EOT
- 9EM Rewinds and unloads
- 11EM Writes 3 inches blank tape
- 14EM Advances one file
- 15EM Backspaces one file

- ENfilespec\$ Sets up wildcard file specification [not in RT-11]
- EN\$ Gets next wildcard match into filespec buffer. [not in RT-11]
- :ERfilespec\$ Same as ERfilespec\$ but returns value: - 1 if file found, 0 if file not found
- :EBfilespec\$ Same as EBfilespec\$ but returns value: - 1 if file found, 0 if file not found
- :EN\$ Same as EN\$ but returns value: - 1 if next file found, 0 if no more files match

FILE TERMINATION COMMANDS AND EXIT

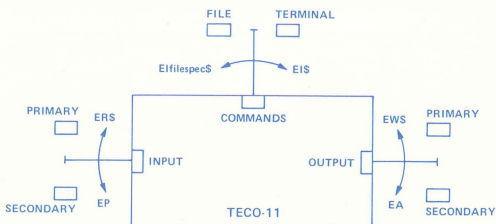
- EF Closes the output file
- EC Moves the remainder of the input file to the output file, then closes the output file [not in TECO-10]
- EX Same as EC but exits from TECO
- EK Kills the output file [not in TECO-10]
- EG\$ Same as EX but re-executes the last compile-class command [note 11]
- EGcommand\$ Same as EX but executes the operating system command specified [not in TECO-10]
- ↑C Exits from TECO [note 12] [not in TECO-10]
- ↑Z Closes output file and exits from TECO

AUXILIARY FILE SUPPORT

[not in RT-11]

(See illustration below)

- EP Switches to secondary input stream
- ER\$ Switches to primary input stream
- EA Switches to secondary output stream
- EW\$ Switches to primary output stream
- EIfilespec\$ Opens file for command input; effective at end of current command string
- EIS Switches command input to terminal and discards remainder of command file after current command string



FILE SPECIFICATIONS [note 3]

System	Input	Output
OS/8	dev:name.ex	dev:name.ex
TOPS-10	dev:name.ext[p,pn]	dev:name.ext[p,pn] <prot>
RT-11	dev:name.ext	dev:name.ext[<i>len</i>]
RSTS/E	dev: [p,pn] name.ext	dev: [p,pn] name.ext<prot>
RSX-11	dev: [p,pn] name.ext;ver	dev: [p,pn] name.ext;ver

FILE SPECIFICATION SWITCHES

TOPS-10	/GENLSN	Generates line sequence numbers
	/SUPLSN	Suppresses line sequence numbers
RSTS/E	/MODE:n	Uses nonstandard open mode
	/CLUSTER SIZE:n	Specifies output file cluster size
	/B+	Edits a BASIC-PLUS file
	/B2	Edits a BASIC-PLUS-2 file (same as /72)
	/n	Handles BASIC-PLUS continuation character (&) in column n
RSX-11	/RW	Rewinds before opening file [magtape only]
	/CR	Implied carriage control
	/- CR	No (internal) carriage control
	/FT	FORTTRAN carriage control

INVOCATION SWITCHES

RSTS/E	/SIZE:n	starts with nK word editing area
	/SIZE:+n	starts with extra nK words of editing area
RSX-11	/INC=n	starts with extra nK words of editing area

PAGE MANIPULATION COMMANDS

A	Appends next page of input file to text buffer
:A	Appends next line of input file to text buffer
Y	Same as HKA
PW	Outputs current page and appends a form feed
nPW	Same as n<PW>
m,nPW	Outputs the characters between text buffer pointer positions m and n (without a form feed)
P	Outputs the text buffer, clears the buffer, and reads the next page of the input file into the buffer; a form feed is appended to the output file if the last read was terminated with a form feed
nP	Same as n<P>
m,nP	Same as m,nPW

BUFFER POINTER POSITIONING COMMANDS

nJ	Moves pointer to immediately following the nth character in the text buffer
OJ	Moves pointer to beginning of buffer
J	Same as OJ
nC	Advances pointer n positions
C	Same as 1C
- C	Same as - 1C
nR	Backs up pointer n positions (same as - nC)

R	Same as 1R
- R	Same as - 1R
nL	Moves pointer to beginning of nth line from current pointer position
L	Same as 1L
- L	Same as - 1L

INSERTION COMMANDS

The pointer is positioned after the inserted text.

ltext\$	Inserts the text specified
nI\$	Inserts character whose ASCII code is n
<tab>text\$	Inserts specified text preceded by a tab
↑ltext\$	Inserts specified text preceded by a tab
n\	Inserts current radix representation of n in text buffer (signed if current radix is decimal, unsigned if octal) [note 13]
FRtext\$	Same as ↑SDltext\$

ARITHMETIC OPERATORS

Operators are evaluated from left to right.

+n	Same as n
- n	Negative of n
m+n	Addition [note 14]
m-n	Subtraction
m*n	Multiplication
m/n	Division, remainder discarded
m&n	Bitwise logical AND
m#n	Bitwise logical OR
(m)	Performs enclosed operations first
n↑_	One's complement of n

ARITHMETIC PRECISION

TECO-8	$1 - 2^{12} \leq n \leq 2^{12} - 1$
TECO-11	$- 2^{15} \leq n \leq 2^{15} - 1$
TECO-10	$- 2^{35} \leq n \leq 2^{35} - 1$

TYPE-OUT COMMANDS

nT	Types out all the text from the current pointer position to the beginning of the nth line from the current pointer position
T	Same as 1T
- T	Same as - 1T
m,nT	Types the characters between pointer positions m and n
nV	Same as (1-n)TnT
V	Same as OTT
↑Atext (↑A)	Types the text specified
n=	Types value of n in decimal (signed) followed by <CR> <LF>
n==	Types value of n in octal [note 15] followed by <CR> <LF>
n:=	Types value of n in decimal (signed) [not in TECO-10]
n:==	Types value of n in octal (unsigned) [not in TECO-10]
n↑T	Types out the character whose ASCII code is n [not in TECO-10]
↑L	Types a form feed [not in TECO-8]

DELETION COMMANDS

nD	Deletes n characters following the current pointer position
-nD	Deletes n characters preceding the current pointer position
D	Same as 1D
-D	Same as -D
m,nD	Same as m,nK
nK	Deletes all characters in the text buffer from the current pointer position to the beginning of the nth line from the current pointer position.
K	Same as 1K
-K	Same as -1K
m,nK	Deletes all characters in the text buffer between pointer positions m and n; the pointer moves to position m

BUFFER POSITIONS

B	First pointer position; same as 0
Z	Last pointer position; equal to number of characters in text buffer
.	Current pointer position; equal to number of characters to the left of the pointer
m,n	(m+1)st through nth characters in the text buffer
H	Same as B, Z
↑Y	Same as .↑S,.

SEARCH COMMANDS

Searches that fail leave the pointer at position 0
Successful searches leave pointer after text found

nStext\$	Searches for the nth occurrence of the specified text (following the pointer), but does not go beyond the end of the current text buffer
Stext\$	Same as 1Stext\$
-nStext\$	Same as nStext\$ but the search proceeds backwards
nNtext\$	Same as nStext\$, except that if the text is not found on the current page, pages are input and output until it is found
Ntext\$	Same as 1Ntext\$
n_text\$	Same as nNtext\$, except that it does input only, no output
_text\$	Same as 1_text\$
nFStext1\$text2\$	Executes the nStext1\$ command, then deletes text1 and replaces it with text2
FStext1\$text2\$	Same as 1FStext1\$text2\$
nFNtext1\$text2\$	Executes the nNtext1\$ command, then deletes text1 and replaces it with text2
FNtext1\$text2\$	Same as 1FNtext1\$text2\$
nF_text1\$text2\$	Executes the n_text1\$ command, then deletes text1 and replaces it with text2
F_text1\$text2\$	Same as 1F_text1\$text2\$

BOUNDED SEARCH

Bounded searches that fail leave the pointer unchanged

- `m,nStext$` Same as `nStext$`, but the search does not continue across more than $|m| - 1$ characters.
- `0,nStext$` Same as `nStext$`, but the pointer position remains unchanged on search string failure
- `m,Stext$` Same as `m,1Stext$`
- `m,-Stext$` Same as `m,-1Stext$`
- `::Stext$` Same as `1,1:Stext$` (anchored search)

COLON MODIFIED SEARCHES

Any search command preceded by a colon (:) modifier returns a value as follows:

- 1 If the search succeeds
- 0 If the search fails

NUMERIC QUANTITIES

- `0A` ASCII code for character following pointer
- `nA` ASCII code for the $(. + n + 1)$ th character in the buffer [note 16] [not in TECO-10]
- `\` Equal to the value of the digit string (if any) that follows the pointer in the text buffer (interpreted in the current radix [note 13], and optionally signed); pointer is positioned after the digit string. Equal to 0 if no digit string is present.
- `↑B` Operating-system dependent encoding of current date
 - RT-11: $((\text{month} * 32) + \text{day}) * 32 + \text{year} - 1972$
 - RSTS/E: $((\text{year} - 1970) * 1000) + \text{day}$ within year
 - RSX-11: $((\text{year} - 1900) * 16 + \text{month}) * 32 + \text{day}$
- `↑F` Value of console switch register
- `↑H` Operating system dependent encoding of time of day
 - OS/8: 0
 - RT-11: 0
 - RSTS/E: minutes until midnight
 - RSX-11: $(\text{seconds since midnight}) / 2$
 - TOPS-10: 60ths of a second since midnight
- `:Qq` Number of characters stored in Q-register q
- `n↑Q` Number of characters between the buffer pointer and the nth line separator from the current pointer position
- `↑Q` Same as `1↑Q`
- `↑S` Negative of length of last insert, string found, or string inserted with G command, whichever occurred last
- `↑T` ASCII code for the next character typed at the terminal
- `↑↑x` ASCII code for the character (x) specified

ITERATION AND FLOW CONTROL

<code>n<commands></code>	Performs the enclosed command string n times; skips the commands if $n \leq 0$
<code><commands></code>	Performs the enclosed command string an indefinite number of times
<code>n;</code>	If $n \geq 0$, jumps out of the current iteration field
<code>;</code>	Jumps out of the current iteration field if the last search executed failed [note 17]
<code>!tag!</code>	Defines a position in the command string with the name, tag
<code>Otag\$</code>	Jumps to the position defined by <code>!tag!</code> ; must be in same macro level [note 18]

@ MODIFIER

@	Modifies the next command that takes a text argument to use the delimiter form; text is surrounded by a delimiter (shown here as /) that cannot appear inside the text
---	--

The format of the commands becomes:

@nS/text/	
@nN/text/	
@n_/text/	
@nFS/text1/text2/	
@nFN/text1/text2/	
@I/text/	
@nI//	
@ER/filespec/	[not in TECO-10]
@EW/filespec/	[not in TECO-10]
@EB/filespec/	[not in TECO-10]
@↑Uq/text/	[not in TECO-10]
@EG/text/	[not in TECO-10]
@EG//	[not in TECO-10]
@EN/filespec/	
@EI/filespec/	
@FR/text/	
@↑A/text/	
@!/tag/	
@O/tag/	
@n↑Uq//	

CONDITIONAL COMMANDS

<code>n''Acommands'</code>	Executes the commands if n is the ASCII code for an alphabetic character (A-Z or a-z) [not in TECO-8]
<code>n''Ccommands'</code>	Executes the commands if n is the ASCII code for a symbol constituent TECO-8: A-Z, a-z, 0-9 TECO-11: A-Z, a-z, 0-9, ., \$ TECO-10: A-Z, a-z, 0-9, ., \$, %
<code>n''Dcommands'</code>	Executes the commands if n is the ASCII code for a digit (0-9) [not in TECO-8]
<code>n''Ecommands'</code>	Executes the commands if $n=0$
<code>n''Fcommands'</code>	Executes the commands if n represents FALSE; same as <code>n''Ecommands'</code>
<code>n''Gcommands'</code>	Executes the commands if $n > 0$
<code>n''Lcommands'</code>	Executes the commands if $n < 0$
<code>n''Ncommands'</code>	Executes the commands if $n \neq 0$
<code>n''Rcommands'</code>	Executes the commands if n is the ASCII code for an alphanumeric (A-Z, a-z, 0-9) [not in TECO-10]

n''Scommands'	Executes the commands if n represents SUCCESS; same as n''Lcommands'
n''Tcommands'	Executes the commands if n represents TRUE; same as n''Lcommands'
n''Ucommands'	Executes the commands if n represents UNSUCCESSFUL; same as n''Ecommands'
n''Vcommands'	Executes the commands if n is the ASCII code for a lower case alphabetic (a-z) [not in TECO-8]
n''Wcommands'	Executes the commands if n is the ASCII code for an upper case alphabetic (A-Z) [not in TECO-8]
n''< commands'	Same as n''Lcommands' [not in TECO-10]
n''> commands'	Same as n''Gcommands' [not in TECO-10]

Q-REGISTER LOADING [note 19]

nUq	Stores integer n in Q register q
%q	Adds 1 to the number in Q-register q and returns this value
n%q	Adds n to the number in Q-register q and returns this value [note 20] [not in TECO-10]
nXq	Stores, in Q-register q, all characters from the current pointer position to the beginning of the nth line from the current pointer position
m,nXq	Stores, in Q-register q, all characters between text buffer pointer positions m and n
Xq	Same as 1Xq
-Xq	Same as -1Xq
n:Xq	Same as nXq, but appends to Q-register
m,n:Xq	Same as m,nXq, but appends to Q-register
:Xq	Same as 1:Xq
:-Xq	Same as -1:Xq
↑Uqtext\$	Stores the specified text in Q-register q [not in TECO-10]
:↑Uqtext\$	Same as ↑Uqtext\$, but appends to Q-register
n↑Uq\$	Stores character whose ASCII code is n in Q-register q
n:↑Uq\$	Appends character whose ASCII code is n to contents of Q-register q
]q	Pops from the Q-register push-down list into Q-register q [note 21] [not in TECO-8]
:]q	Same as]q, but returns value (-1 if successful, 0 if push down stack was empty)

Q-REGISTER RETRIEVAL [note 19]

Qq	Returns the numeric value stored in Q-register q
Gq	Places the text in Q-register q into the text buffer before the current pointer position
:Gq	Types the contents of the text stored in Q-register q on the terminal [not in TECO-10]
Mq	Executes the text in Q-register q as a command string
nMq	Executes the text in Q-register q as a command string and uses n as a numeric argument for the first command executed

m,nMq	Executes the text in Q-register q as a command string and uses m,n as a pair of numeric arguments for the first command executed
[q	Pushes the contents of Q-register q onto the Q-register push-down list [note 21]
G*	Places the text from the filespec buffer into the text buffer before the current pointer position
:G*	Types the contents of the filespec buffer on the terminal
G_	Places the text from the search string buffer into the text buffer before the current pointer position
:G_	Types the contents of the search string buffer on the terminal

RETRIEVING MODE CONTROL FLAGS

Flag	Name	Initial Value
↑E	Form feed flag	0
↑N	End-of-file flag	0
↑X	Search Mode Flag	0
ED	Edit level flag	0 [RT-11] 0 [RSTS/E] 1 [RSX-11]
EH	Help level flag	2 [TECO-10] 0 [other systems]
EO	Version number flag	2 [TECO-10] 5 [TECO-8] 29 [TECO-11]
ES	Search verification flag	0
ET	Terminal mode flag	

TECO-10: 0
TECO-8: 0 if SET TTY NOSCOPE
2 if SET TTY SCOPE

TECO-11:

Bit Value	Initial Value
1	0
2	1 if terminal is a scope 0 if terminal is not a scope
4	1 if terminal has lower case input 0 if terminal does not have lower case input
8	0
16	0
32	0
64	0
128	1 before TECO's prompt 0 after TECO's prompt
256	0
512	1 if terminal is VT52 and VT52 support is available 0 if VT52 support is not available
1024	1 if VT11 is present and VT11 support is available 0 if VT11 support is not available

EU	Upper/lower case flag	
	- 1	if terminal supports lower case
	0	if terminal does not support lower case [note 22]
EV	Edit verify flag	0

SETTING MODE CONTROL FLAGS

Command	Bit Value	Meaning
nED	1	0 ↑ in search arguments means next character is CTRL
		1 ↑ in search arguments means ↑
	2	0 disables Y or ___ if data would be lost
		1 no Yank protection
nEH	1,2	0 same as 2
		1 types 3-letter error code only
		2 types error code and short message
		3 types error code and long message
	4	1 Simulates ? after error
nEO		Allows EO version n features to work
m*256+nES if n=0, nothing is typed out after searches; if n=-1, an effective mV command is executed when a search is completed. If n is between 1 and 31, <LF> represents the pointer; if n is between 32 and 126, it is ASCII code for character to represent the pointer. [TECO-10 ignores m]. If m ≤ 0, m is treated as 1.		
nET		Sets terminal characteristics [note 23]
	Bit Value	Meaning if 1
	1	type-out is in image mode
	2	terminal is a scope
	4	reads lower case
	8	↑T reads with no echo
	16	cancels (↑O) on type-out
	32	↑T reads with no wait (↑T returns -1 if no input)
	64	detached flag
	128	aborts on error
	256	truncates output to terminal width
	512	terminal is VT52 and VT52 support is available
	1024	VT11 is present and VT11 support is available
	32768	Traps (↑C)'s (Reset if (↑C) typed)
nEU	0	flags lower case characters
	>0	flags upper case characters
	<0	no case flagging on output [note 22]

nEV		Same as nES but lines to be viewed appear before TECO's prompt
n↑X	0	either case matches in searches
	1	exact case matches required
		[not in TECO-8]

RADIX CONTROL [note 13]

↑O	Switches to octal radix [note 24]
↑D	Switches to decimal radix [not in TECO-10]

TRACE CONTROL

?	Enters trace mode; nullified by a 2nd ?
---	---

VT52 SCOPE CONTROL

-1W	Updates window into text buffer
0W	Resets cursor line to default position
nW	If n>0, sets cursor to line n
	If n<-1, "forgets" top n -1 lines of screen

VR12, VT11, or VS60 SCOPE CONTROL

[not in TECO-10]

W	Updates window into text buffer
nW	Sets window display to n lines on either side of pointer

CASE CONTROL

↑V	Enters translate to lower case mode
↑W	Enters translate to upper case mode

MATCH CONTROL CONSTRUCTS

(Match control constructs are valid inside search strings)

(↑X)	Accepts any character
(↑S)	Accepts any non-alphanumeric separator [note 25]
(↑N)	Accepts any character except a character matched by the following character (or match control construct)
(↑E) A	Accepts any alphabetic character (A-Z or a-z)
(↑E) C	Accepts any symbol constituent TECO-11: A-Z, a-z, 0-9, ., \$
(↑E) D	Accepts any digit (0-9) [not in TECO-8]
(↑E) Gq	Accepts any character in Q-register q
(↑E) L	Accepts any line terminator (<LF>, <VT>, <FF>) [not in TECO-8]
(↑E) R	Accepts any alphanumeric (A-Z, 0-9)
(↑E) S	Accepts any non-null string of spaces and/or tabs [not in TECO-8]
(↑E) V	Accepts any lower case alphabetic (a-z) [not in TECO-8]

- (↑E) W Accepts any upper case alphabetic (A-Z) [not in TECO-8]
- (↑E) X Same as (↑X)
- (↑E) <nnn> Accepts the character whose ASCII code (in octal) is nnn
- (↑E) [a,b,c,...] Accepts exclusive OR of the characters or match control constructs specified

SEARCH STRING BUILDING CHARACTERS

The following characters have special meaning when used inside a search string.

- (↑Q) Uses the next character literally [note 26]
- (↑V) Converts next character, if alphabetic, to lower case [not in TECO-8]
- (↑W) Converts next character, if alphabetic, to upper case [not in TECO-8]
- (↑↑) Converts next character (if it is @, [, \,], ↑, or _) to lower case
- (↑R) Uses the next character literally
- (↑V) (↑V) Converts all following alphabetic characters to lower case
- (↑W) (↑W) Converts all following alphabetic characters to upper case
- (↑T) Treats all succeeding control characters except (↑R) and (↑T) as text; nullified by a second (↑T)
- (↑\) Accepts either case for all succeeding alphabetic characters; nullified by a second (↑\)
- ↑ If ED&1=0, converts the following alphabetic character to the corresponding control character
- (↑E) Qq Uses the string stored in Q-register q at this position
- (↑E) Q* Uses the string stored in the filespec buffer at this position
- (↑E) Q_ Uses the string stored in the search string buffer at this position

INSERT STRING BUILDING CHARACTERS

The following characters have special meaning when used inside insert strings on TECO-10. They have the same meaning that they have when used inside search strings.

- | | |
|-----------|------|
| (↑V) | (↑↑) |
| (↑W) | (↑R) |
| (↑V) (↑V) | (↑T) |
| (↑W) (↑W) | |

ERROR MESSAGES

Code	Explanation
?ARG	Improper arguments
?BAK	Cannot delete old backup file
?BNI	> not in an iteration
?COR	Same as ?MEM
?CCL	CCL not found or EG argument too big
?COS	Contradictory output switches
?CPQ	Cannot pop Q-register
?DEV	Invalid device
?DIO	Directory I/O error
?DTB	Delete too big
?EBD	Bad EB device
?EBF	Bad EB filename
?EBO	Same as ?OFO
?EBP	EB illegal because of protection
?EEE	Unable to read error message file
?EMA	EM with illegal argument
?EMD	EM with no input device open
?ENT	Enter error
?EOA	Bad argument to EO
?ERR	RSTS/E monitor call failed
?FER	File error
?FNF	File not found
?FUL	Output command would have overflowed device (In TECO-10, same as ?NRO)
?IAB	Invalid nesting of < ... > or (...)
?ICE	Illegal CTRL/E command in search argument
?ICT	Illegal control character in text argument
?IDV	Bad input device
?IEC	Illegal character after E
?IEM	Re-init failure after EM
?IFC	Illegal character after F
?IFN	Bad filename
?IIA	Illegal insert argument
?ILL	Illegal command
?ILN	Same as ?OCT
?ILR	Cannot lookup input file to rename it
?INP	Input error
?IOS	Illegal character in I/O switch
?IQC	Illegal character after "
?IQN	Illegal Q-register name
?IRB	Cannot rename input filename
?IRN	Failure in rename process
?ISA	Illegal search argument
?ISS	Illegal search string
?MAP	Missing '
?MEE	Macro ending with E
?MEF	Macro ending with F
?MEM	Storage capacity exceeded
?MEO	Macro ending with O
?MEQ	Macro ending with "
?MEU	Macro ending with U
?MFN	Missing filename
?MIQ	Missing Q-register name
?MLA	Same as ?BNI
?MLP	Missing (
?MRA	Missing >
?MRP	Missing)
?MUU	Macro ending with ↑↑
?NAB	No argument before ↑_
?NAC	TECO-11: No argument before , TECO-8: Negative argument to ,
?NAE	No argument before =
?NAI	No altmode after nl

ERROR MESSAGES (Cont.)

Code	Explanation
?NAP	TECO-11: Same as ?MLP TECO-8: Negative or 0 argument to P
?NAQ	No argument before "
?NAS	TECO-11: No argument before ; TECO-8: Negative or 0 argument to search
?NAU	No argument before U
?NAY	Numeric argument with Y
?NCS	No command string seen prior to *q
?NFI	No file for input
?NFO	No file for output
?NNQ	No number in Q-register
?NRO	No room for output
?NTQ	No text in Q-register
?NYI	Command not yet implemented
?OCT	8 or 9 in octal digit string
?ODV	Output device not available
?OFO	Output file already open
?OLR	Cannot lookup output file to rename it
?OUT	Output error
?PAR	Confused use of parentheses
?PDO	Push-down list overflow
?POP	Attempt to move pointer off page
?PPN	Illegal PPN
?QMO	Q-register memory overflow
?RNO	Cannot rename output file
?SAL	Second argument less than first
?SNA	Initial search with no argument
?SNI	; not in an iteration
?SRH	Search failure
?STC	Search string has too many characters
?STL	Search string too long
?TAG	Missing tag
?TAL	Two arguments with L
?TTL	Tag too long
?TTY	Illegal TTY device
?UCA	Unterminated ↑A command
?UEA	Unimplemented EA
?UEI	Unimplemented EI
?UEN	Unimplemented EN
?UEP	Unimplemented EP
?UFS	Macro ending with unterminated file select
?UIN	Unterminated insert command
?UIS	Undefined I/O switch
?USR	Unterminated search command
?UTC	Unterminated command
?UTG	Unterminated tag
?UTM	Unterminated macro
?UZO	Illegal UZO
?WLO	System device is write locked
?XAB	Execution aborted by ↑C
?YCA	Y command aborted
?nnn	RSX-11 monitor call failed

OCTAL AND DECIMAL ASCII CODES

chr	OCT	DEC	chr	OCT	DEC
nul	000	000	@	100	064
↑A	001	001	A	101	065
↑B	002	002	B	102	066
↑C	003	003	C	103	067
↑D	004	004	D	104	068
↑E	005	005	E	105	069
↑F	006	006	F	106	070
↑G	007	007	G	107	071
↑H	010	008	H	110	072
tab	011	009	I	111	073
lf	012	010	J	112	074
vt	013	011	K	113	075
ff	014	012	L	114	076
cr	015	013	M	115	077
↑N	016	014	N	116	078
↑O	017	015	O	117	079
↑P	020	016	P	120	080
↑Q	021	017	Q	121	081
↑R	022	018	R	122	082
↑S	023	019	S	123	083
↑T	024	020	T	124	084
↑U	025	021	U	125	085
↑V	026	022	V	126	086
↑W	027	023	W	127	087
↑X	030	024	X	130	088
↑Y	031	025	Y	131	089
↑Z	032	026	Z	132	090
esc	033	027	[133	091
fs	034	028	\	134	092
gs	035	029]	135	093
rs	036	030	↑	136	094
us	037	031	—	137	095
sp	040	032	'	140	096
!	041	033	a	141	097
"	042	034	b	142	098
#	043	035	c	143	099
\$	044	036	d	144	100
%	045	037	e	145	101
&	046	038	f	146	102
'	047	039	g	147	103
(050	040	h	150	104
)	051	041	i	151	105
*	052	042	j	152	106
+	053	043	k	153	107
,	054	044	l	154	108
-	055	045	m	155	109
.	056	046	n	156	110
/	057	047	o	157	111
0	060	048	p	160	112
1	061	049	q	161	113
2	062	050	r	162	114
3	063	051	s	163	115
4	064	052	t	164	116
5	065	053	u	165	117
6	066	054	v	166	118
7	067	055	w	167	119
8	070	056	x	170	120
9	071	057	y	171	121
:	072	058	z	172	122
;	073	059	{	173	123
<	074	060	}	174	124
=	075	061	~	175	125
>	076	062	del	176	126
?	077	063		177	127

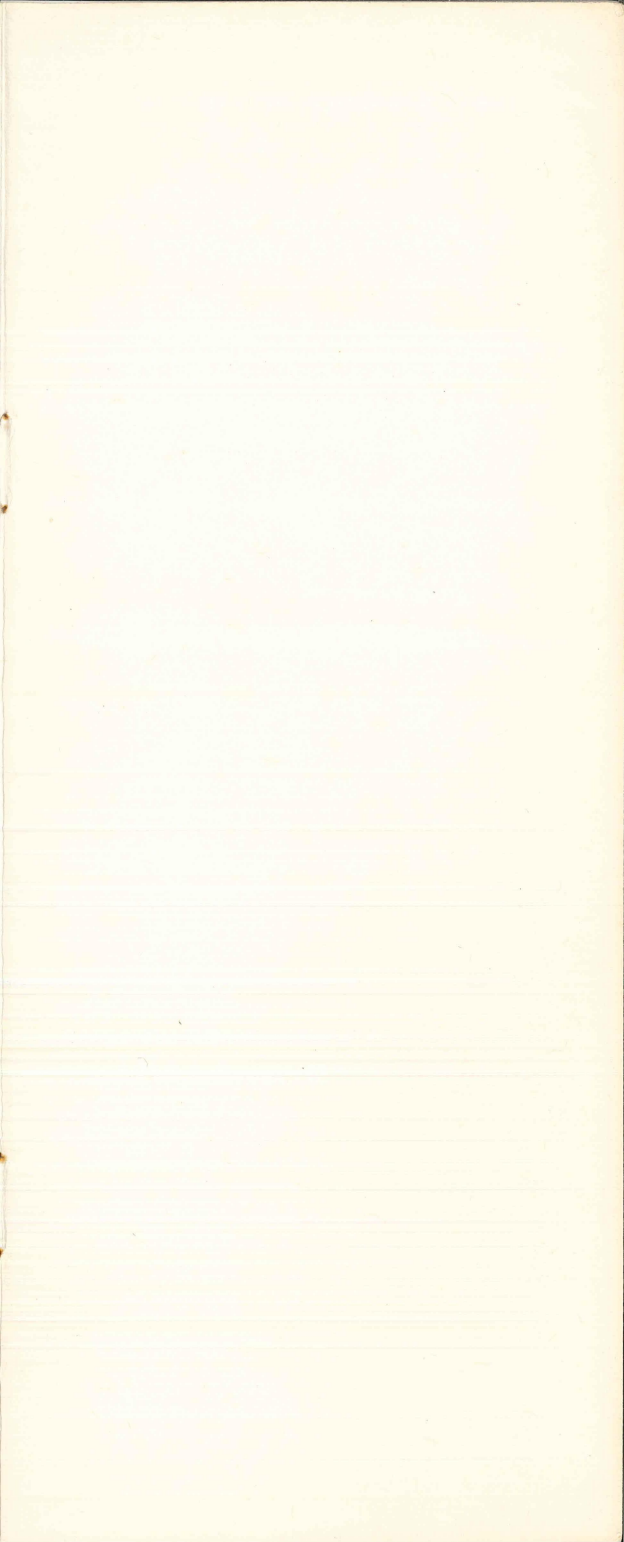
NOTES

- [1] Re-executes last TECO command. Ignores filespec2, if present on previous command
- [2] /TECO switch not necessary if SET EDITOR TECO command has been previously specified
- [3] SFD's allowed in TOPS-10
- [4] Re-executes last MAKE command
- [5] TECO under RSTS/E and OS/8 requires two $\uparrow C$ s. $\uparrow C$ is not an immediate mode command in RT-11 and RSX-11
- [6] TECO-10 types the last 10 characters of the command string before the error occurred
- [7] Not an immediate mode command in TECO-10 (execution is deferred until next wake-up character). In TECO-8, the immediate mode command $\uparrow S$ is the same as *Z
- [8] Two $\uparrow C$ s may be necessary on systems that support type-ahead
- [9] Only if feature is enabled in the operating system
- [10] TECO-10 closes any previous file.
- [11] Not in RSTS. TECO-10 does not require \$
- [12] On TECO-11, $\uparrow C$ in a macro returns control to TECO's prompt
- [13] TECO-10 has no general octal radix mode
- [14] In TECO-10, a space between two numbers also acts as an addition
- [15] TECO-10 prints signed result; TECO-8 and TECO-11 print unsigned result
- [16] On TECO-10, nA is the same as 0A
- [17] Must immediately follow search in TECO-11
- [18] Cannot jump to before start of current iteration in TECO-8 or TECO-11
- [19] TECO-10 stores either text or numeric data (but not both simultaneously) in a single area within the Q-register
- [20] Same as 1%q in TECO-10
- [21] Arguments "pass through" this command in TECO-11
- [22] On TECO-10, no case flagging occurs if the terminal supports lower case, regardless of state of EU flag
- [23] TECO-10 uses only bit value 1, TECO-8 uses only bit values 1, 2, and 8
- [24] In TECO-10, $\uparrow O$ must be followed by a digit string and the octal radix mode applies only to this string
- [25] TECO-10 allows begin of buffer, ., \$, and % to match
- [26] Cannot quote \$ on TECO-10

NOTES

NOTES

NOTES



NOVICE SUBSET OF TECO

These commands allow you to do most common editing jobs on all versions of TECO.

MAKE file	Creates a new file (under RT-11, use CREATE command)
TECO file	Edits an existing file (under RT-11, use EDIT command)
J	Moves pointer to beginning of buffer
ZJ	Moves pointer to end of buffer
L	Advances pointer to beginning of next line
nL	Advances pointer n lines
-nL	Moves pointer back n lines
OL	Moves pointer to beginning of current line
C	Advances pointer across one character
nC	Advances pointer across n characters
-nC	Moves pointer backwards across n characters
T	Types text from current pointer position to end of current line
nT	Types n lines on terminal
HT	Types entire content of text buffer
OTT	Types current line
ltext\$	Inserts text before current pointer position
K	Deletes text from current pointer position to end of line
nK	Deletes n lines of text
HK	Deletes entire content of text buffer
D	Deletes the character following the pointer
nD	Deletes n characters after the pointer
-nD	Deletes n characters before the pointer
P	Outputs the current page and reads in the next page
nP	Executes the P command n times
A	Append the next input page to the end of the buffer
EX	Moves the remainder of the input file to the output file, closes the output file, and exits from TECO
Stext\$	Beginning at the current pointer position, searches for the specified text on the current page. The buffer pointer is positioned after the specified text if it is found. If the text is not found, an error message is given and the buffer pointer is left at the beginning of the page
Ntext\$	Same as Stext\$ except that the search continues across page boundaries, if necessary, until the string or the end of the input file is encountered
FStext1\$text2\$	Searches for text1 and, if found, replaces it with text2