RCSL No: 43-GL 9609

Edition: October 1979

Author: Stig Møllgaard
Erik Jeppesen

Title:

RC BASIC/COMAL
Corrections (no. 3) to the RC BASIC Programming Guide

A/S REGNECENTRALEN af 1979

Keywords:

RC BASIC, RC 3600, RC 7000, Corrections.

Abstract:

This manual contains corrections to the RC BASIC Programming Guide (RCSL: 42-i 0671).
The manual, which replaces two previous correction manuals (no. 1: RCSL 43-GL 6940 and no. 2: RCSL 43-GL 7914), updates the Programming Guide to a point corresponding to rev. 2.0 of the RC BASIC system.
(53 printed pages)

42-i 1286

CONTENTS PAGE

1. INTRODUCTION 1.

This manual describes corrections to the first edition of the RC BASIC PROGRAMMING GUIDE (RCSL No 42-i 0671).

Some corrections are due to printing errors, most of the changes are, however, a consequence of new features that have been implemented in the RC BASIC system.

The most important new features are:

- STRING ARRAYS: It is now possible to work with two-dimensional strings.

- LOWBOUND: The lower bound of arrays can be set to zero or one.

- RESET: The effect of an ON ERR/ESC-statement can be cancelled.

- IMPROVED FILE SYSTEM: On systems with moving head disc, subcatalogs are used instead of logical discs. Several users can have write-access to the same logical disc/subcatalog, and one logical disc/subcatalog can be used as a program-library accessable from all terminals.

- LOGON/LOGOFF: On configurations with some kind of disc, the system will make an attempt to load a program from the disc when a terminal is logged on/off. This makes it possible for the user to implement an account-system.

- PROTECT: It is possible (by means of the above mentioned facility) to prevent a user from changing subcatalogs, saving programs into existing files etc.

- CALL ROUTINES: It is now possible to call assembler-coded subroutines from RC BASIC programs.

- EXTENDED PRECISION: The system can be delivered with extended precision-arithmetic corresponding to 10 decimals digits in output. The number of digits to be output can be defined by the user by means of the DIGITS-statement.

Many of the corrections in this manual have been described in the previous manuals: "Corrections (no 1 (2)) to the RC BASIC Programming Guide", RCSL No 43-GL6940 (7914) which are replaced by the present manual.

Changes or new information are marked with a vertical bar in the margin.

Most of the corrections are mentioned by a page number and a line number. A positive line number means that lines should be counted from the top of the page, a negative line number means that lines should be counted from the buttom.

With these corrections included, the Programming Guide corresponds to rev. 02.00 of the RC BASIC System.

2. CORRECTIONS 2.

In the list of reserved words on the second page:

word	use	section
CALL	S	App. H
DIGITS	C,S	3.6.A
LOWBOUND	C,S	9.10.A
PROTECT	C,S	3.26.A
RESET	C,S	3.30.A

✓

CONTENTS, Last page, replace line -3 by:

✓ G LOGON/LOGOF FUNCTIONS.

CONTENTS, Last page, insert after line -2:

✓ I EXTENDED PRECISION IN RC BASIC.

Page 13, insert after line -6:

✓ <subcatname>: The name of a subcatalog.

Page 17, line 4:

✓ The lower bound of a dimension is usually 1, it may, however, be set to 0 by means of the LOWBOUND statement (see Ch. 9).

Page 17, line 10:

✓ 32 767 -> 16 380.

Page 21, line -5 through -1:

1. A string variable, e.g. ANSWER$
2. An element of a string array, e.g. TEXTS$(5)

3. A part of a string variable or a part of an element of a string array.
4. A string literal, e.g. "PETER"
5. The CHR(X) function, e.g. CHR(65)
6. A concatenation of the above items, e.g. "JOHN SMITH", ADDRESS$

Page 22, line -7:

block -> blocks.

Page 25, line -10, -8, -6:

```
0250  LET MONTH$="OCTOBER"; DAYS=31
0270  LET MONTH$="NOVEMBER"; DAYS=30
0290  LET MONTH$="DECEMBER"; DAYS=31
```

Page 29, line 6 through 9:

2. If the SAVEd program is on disc, the system searches the logical disc/subcatalog to which the terminal is connected for <filename> (see Ch.8). If <filename> is not found then the system searches for a logical disc called LIB and if this is found then a search is made in this logical disc for <filename>. If <filename> is not found, the system outputs the error message 0100: FILE UNKNOWN.

Page 29, line 12 and 13:

The statement: "If <filename> is not found ..." should be deleted.

Page 29, insert after line -12:

7. When the CHAINed program is executed, it is done without clearing the state of the program. This means that all variables have the values they had, when the program was

SAVEd. If a CHAINed program should be executed as if the user had started it by means of the command RUN, then the user must bring the program into "neutral" state before it is SAVEd:

```
V       1 stop                   ; insert STOP as the first
                                   statement
        RUN                      ; execute the program
        1                        ; delete STOP-statement
        SAVE "<filename>"        ; SAVE the program
```

Page 30, insert after line -4:

4. The number of items in each DATA-statement is limited. Each list may contain 22 (in extended precision systems 15) numeric items. If the list contains string elements the maximal number of items depends on the length of the
V strings.
   If the list has too many elements, error message no. 0009: LINE TOO LONG will be output when the DATA-statement is entered.

Page 31, line -3:

more that -> more than.

Page 33, insert after line 14:

3.6A DIGITS

Format

DIGITS = <expr>

<expr>: a numeric expression which evaluates to an integer, 1 <= <expr> <= 10.

Use

As a statement or command to define the maximal number of digits output in PRINT-statements in systems with extended precision arithmetic.

Remarks

1. The default value of DIGITS is 6.
2. The statement/command has no effect in systems with normal precision arithmetic.
3. If <expr> evaluates to a number smaller than 1 or greater than 10, DIGITS is set equal to 1 respectively 10.
4. For further remarks, see appendix I.

Page 33, line -12 through -1:

$$\text{DIM}\left\{\begin{array}{l}\langle\text{svar}\rangle(\langle\text{l}\rangle)\\ \langle\text{sarray}\rangle(\langle\text{n}\rangle,\langle\text{l}\rangle)\\ \langle\text{array}\rangle(\langle\text{m}\rangle)\\ \langle\text{array}\rangle(\langle\text{row}\rangle,\langle\text{col}\rangle)\end{array}\right\}\left[\;,\;\left\{\begin{array}{l}\langle\text{svar}\rangle(\langle\text{l}\rangle)\\ \langle\text{sarray}\rangle(\langle\text{n}\rangle,\langle\text{l}\rangle)\\ \langle\text{array}\rangle(\langle\text{m}\rangle)\\ \langle\text{array}\rangle(\langle\text{row}\rangle,\langle\text{col}\rangle)\end{array}\right\}\right]\ldots$$

<svar>:	a string variable
<l>:	a numeric expression, which evaluates to the length of a string variable or the length of each element in a string array.
<sarray>:	a string array name.
<n>:	a numeric expression, which evaluates to the number of elements in a string array.
<array>:	an array name
<m>:	a numeric expression, which evaluates to the number of the last element in a one-dimensional array.
<row>:	a numeric expression, which evaluates to the number of the last row in a two-dimensional array.
<col>:	a numeric expression, which evaluates to the number of the last column in a two-dimensional array.

Page 34, line 1 through 4:

Use

As a statement or command to define explicitly the size of one or more numeric variable arrays, string variables or string arrays. (For the dimensioning of string variables, see Chapter 5).

Page 34, line 15:

✓ 1 <= value -> 0 <= value

Page 34, line -8:

✓ Less than 1 -> less than the lower bound
(as defined by means of the LOWBOUND-statement, see Chapter 9.)

Page 35, line -9:

✓ 32767 -> 16380

Page 54, insert after line -13:

✓ 3.18A: LOWBOUND
For description, see Chapter 9.

Page 55, insert after line 8:

✓ 3. Execution of a RESET ERR statement (see sect. 3.30A) will restore normal error action.

Page 56, insert after last line:

✓ 6. Execution of a RESET ESC statement (see sect. 3.30A) will restore the normal ESCape key function.

Page 73, insert after line -3:

3.26A PROTECT

<u>Format</u>

PROTECT = <expr>

    <expr>: a numeric expression specifying a
            protection mask.

<u>Use</u>

As a statement or command to protect certain statements or commands from being executed on the terminal. An attempt to execute a protected facility will cause an error message to be output.

<u>Remarks</u>

√ 1. Protection codes:

1.	PROTECT
2.	CONNECT
4.	RELEASE
8.	COPY
16.	COPY to existing file
32.	LIST in existing file
64.	SAVE in existing file

2. If PROTECT=1 has been executed, the protection-mask cannot be changed. The mask will be cleared when the terminal is logged off.

3. The PROTECT statement is especially useful in LOGON-programs. For further details see APP. G.

Example

PROTECT = 1+2+4

The CONNECT and RELEASE statements and commands cannot be executed on the terminal.

Page 79, insert after line 19:

3.30A RESET

Format

$$\text{RESET}\begin{Bmatrix}\text{ESC}\\ \\ \text{ERR}\end{Bmatrix}$$

Use

As a statement to cancel the effect of an ON-ESC or ON-ERR statement.

Remarks

1. After execution of a RESET statement the normal action is restored.
   See ON-ERR, sect. 3.20 and ON-ESC, sect. 3.21.

Page 92, line -10:

2 16 -> 2 15

Page 103, insert after line -13:

5.1.8 <u>String arrays</u>

The array-concept can be used in connection with strings. Dimensioning of a string array is accomplished by means of the DIM statement, (see Chapter 3), for example:

```
DIM TEXT$(20,40)
```

The string array TEXT$ consists of 20 string-elements each 40 characters long. The i'th element can be referenced as TEXT$(i). TEXT$(i,j) is a reference to the j'th character in the i'th element, and TEXT$(i,j,k) points our character number j through k in the i'th element of the string array.
The lower bound of a string array (i.e. the number of the first element) is usually 1. By means of the LOW-BOUND-statement (see Chapter 9) the lower bound may, however, be set to zero. The first character in a string element will always be number 1.

```
Example                                        Comment

0010 TAB=4                                     The 5 names are

0020 DIM NAMES$(5,8)                            stored in the string
0030 LEFT I=1                                   array NAMES$
0040 REPEAT
0050   READ NAMES (I)
0060   LEFT I=I+1
0070 UNTIL I>5
0080 FOR I=1 TO 5
0090   PRINT NAMES$(I)
0100 NEXT I
0110 DATA "PETER,"JOHN","ROBERT","ROBERTA","DIANA"

PETER   JOHN    ROBERT  ROBERTA DIANA
```

<u>Example</u>   <u>Comment</u>

```
0010 DIM TEXT$(6,5)
0020 FOR I=1 TO 6
0030   LET TEXT$(I)="TEXT",CHR(48+I)
0040 NEXT I
0050 FOR I=6 TO 2 STEP -1
0060   PRINT TEXT$(I)                 print string element 6 to 2
0070 NEXT I
0080 FOR I=1 TO 5
0090   PRINT TEXT$(1,I);              print string element 1, cha-
0100 NEXT I                           racter by character
0110 PRINT
0120 PRINT TEXT$(2,3,LEN(TEXT$(2)))   print string element 2,
                                      from character no. 3.
```

```
TEXT6
TEXT5
TEXT4
TEXT3
TEXT2
TEXT1
XT2
```

Page 108, insert after line 1:

If the lower bound of arrays has been set to zero (by means of the LOWBOUND-statement, see Chapter 9), then ✓ MATRIXA will have 11 rows (no. 0 through 10) and 21 columns (0 through 20).

Page 124, line 1 and 2:

✓

7. Logical Discs/Subcatalogs and Related Commands.
7.1. Introduction.
7.1.1. Logical Discs.

Page 126, line 5:

✓ protection key -> protection key (>0)

Page 126, insert before line 14:

7.1.2. Subcatalogs.

The logical discs described in Sect. 7.1.1. are always used on systems with flexible discs. On systems with moving-head disc (systems running under the DOMUS operating system) RC BASIC will usually use subcatalogs in stead of logical discs. The subcatalogs can be considered as being a part of the DOMUS system, and therefore they will not be described in this manual. The reader is referred to ref. [1] and [2].

Once a subcatalog has been created (by means of the DOMUS utility program SUBCA, see ref. [1]) it may from the BASIC-users point of view be looked at as a logical disc, except for a few minor differences that will be mentioned in the following:

- The name of a subcatalog consists of max. 5 characters.

- The name of a file contained in a subcatalog consists of max. 5 characters.

- A subcatalog does not have a fixed size. Files may be created within the subcatalog as long as there is space on the disc holding the subcatalog.

- The INIT-, LOCK- and USERS-command (Sect. 7.4, 7.5 and 7.8) are blind, i.e. they have no effect when used in subcatalog-systems.

- The concept "exclusive user" does not exist in connection with subcatalogs. When a user connects his terminal to a subcatalog and he specifies a correct protection key, he will be allowed to perform all kinds of operations on the files contained in the subcatalog. This does not mean, that other users are excluded from the actual subcatalogs. In other words: all users may perform all kinds of operations at the same time. It is, however, not possible for two or more users to write into or delete the same file at the same time.

- If a subcatalog does not have a protection key, a user is allowed to perform all operations (including "DELETE") if he knows the name of the subcatalog.

- The main catalog of the system can be looked at as a subcatalog named CAT with protection key equal to zero. If a user has not connected his terminal to a subcatalog, he will be connected to CAT. The user is then allowed to read all files described in CAT, but he may only change files, that have been created from BASIC. This means, that all system-files are protected against destruction.

Page 126, line -10:

Remarks (logical disc-systems)

1. If the user CONNECT's to a logical disc which has a protection key (<>0) without specifying this, he may only read from the logical disc.

2. If the user specifies a protection key (<>0) and the value is correct, he becomes the exclusive user of <ldname> and may now write to as well as read from <ldname>, whereas no other user may connect his terminal to <ldname>.

3. If a CONNECT-command is given from a terminal which is already connected to a logical disc, a RELEASE-command (see Sect. 7.7) will automatically be executed.

4. If the protection key of a logical disc is equal to zero, this logical disc can be used by several users at the same time. The users will be able to CREATE, RENAME, read from and write to files on the logical disc, if they do not specify a protection key when CONNECTing. If a file is to be DELETE'd, this can only be done by a user, who is exclusive user of the logical disc. One becomes exclusive user of a logical disc with protection key equal to zero by specifying any protection key not equal to zero when CONNECTing.

The following table shows which kind of access the user will have depending on the protection key of the logical disc and the key specified in the CONNECT-command.

Key specified in CONNECT-command / Protection key of the logical disc	=0 i.e. no key	>0
=0	Write-access: all kind of access is allowed except DELETE	Exclusive use: all kind of access is allowed
>0	Read-access only	Exclusive use: all kind of access is allowed.

Remarks (subcatalog-systems)

1. If the user CONNECT's to a subcatalog which has a protection key (<>0) without specifying this, he may only read from files contained in the subcatalog.

2. If the user specifies a protection key and the value is correct, he is allowed to perform all kinds of operations on files contained in the subcatalog.

3. If a CONNECT-command is given from a terminal which is already conncted to a subcatalog, a RELEASE-command (see Sect. 7.7) will automatically be executed.

4. If a subcatalog does not have a protection key, a user is allowed to perform all kinds of operations on files contained in the subcatalog, if he CONNECT's to the subcatalog.

The following table shows which kind of access the user will have depending on the protection key of the subcatalog and the key specified in the CONNECT-command.

Key specified in CONNECT-command / Protection key of the subcatalog	=0 i.e. no key	>0
=0	all kinds of access is allowed	all kinds of access is allowed
>0	read-access only	all kinds of access is allowed (if the key is correct)

Page 127-132:

✓ <ldname> -> <ldname>/<subcatname>

logical disc -> logical disc/subcatalog

Page 127, line 14:

✓ user"s -> user's

Page 127, line -11:

1. A file can be copied to a logical disc/subcatalog only if the user has write-access to the files of that logical disc/subcatalog (see Sect. 7.2).

Page 128, insert after line -12:

✓ 4. The INIT-command has no effect in subcatalog-systems.

Page 129, insert after line 13:

✓ 5. The LOCK-command has no effect in subcatalog-systems.

Page 130, Remarks to LOOKUP.

Remarks 1-6 only applies to logical-disc systems.

In subcatalog-systems the listing of files will look as follows: (The headings will only be printed, if the LOOKUP "$LPT" form of the command is used).

```
NAME  ATTRIBUTE  LENGTH  INDEX  RLENGTH  LBYTE    LBLOCK  SEQ
                                         RECSIZE  NO.REC  RAN

PROG1 ........V    3      576      6      133       3      S
DATA1 ........F    7      666     12       80      42      R
```

LENGTH, INDEX and RLENGTH are usually of no interest for the user of BASIC. For further information, see ref. [2].

NAME:	The name of file.
ATTRIBUTES:	
V:	The file is extendable.
F:	The file has a fixed length i.e. it is not extendable.
LENGTH:	The length of the file.
INDEX:	The number of the sector, where the indexblock of the file is placed.
RLENGTH:	The reserved length of the file.
LBYTE:	(sequential files). The number of bytes used in the last block written.
LBLOCK:	(sequential files). The number of the last block written.
RECSIZE:	(random access files). The length (in bytes) of the record.
NO.REC:	(random access files). The number of records in the file.
SEQ/RAN:	
S:	a sequential file.
R:	a random access file.

Only the files that have been created from BASIC will be listed although other files will be accessable for reading.

Page 132, insert after line 6:

✓ 3. The USERS-command has no effect in subcatalog-systems.

Page 133, line 6:

✓ *Logical discs* -> *logical discs/subcatalogs*.

Page 133-150:

✓ Logical disc -> logical disc/subcatalog.

Page 133, insert after line -10:

✓ Serial printer ($SP)

Page 134, line 2-4:

In logical disc-systems a RC BASIC file comprises a number of consecutive blocks in a logical disc. Each file is described separately by an entry in the subcatalog of the logical disc (see Sect. 7.1). In subcatalog-systems an RC BASIC file consists of a number of *slices* that can be placed anywhere on the disc. A slice is a number of consecutive blocks; the number depends on how the DOMUS-disc has been generated, but it is usually 6 (see also ref. [2].)

Page 134, line 6:

Logical disc -> logical disc/subcatalogs.

Page 134, line 13-14:

In logical disc-systems the name consists of 1 to 8 characters, and in subcatalog-systems 1 to 5 character. Only characters with decimal values between 32 and 127 (see appendix D) are legal.

Page 135, line 13 and
Page 137, line -16 and
Page 149, line 2:

Replace "correctly specified the protection key of the logical disc/subcatalog in the CONNECT-command" by "has write-access to the files of the logical disc/subcatalog to which the terminal is connected"

Page 135:

Replace lines -5 through -3 by
A discfile can only be CREATEd, RENAMEd or written into if
V the user has write-access to the logical disc/subcatalog
(see Chapter 7, CONNECT).
A discfile can only be DELETEd if the user is exclusive
user of the logical disc/subcatalog.

Page 137, line 4:

<filename>: the name of the disc file to be created, expressed as a string literal or by means of a variable. In logical disc-systems the name consists of 1 to 8 characters, and in subcatalog-systems 1 to 5 character. Only characters with decimal value between 32 and 127 (see appendix D) are legal.

Page 137, line -9 - -1:

3. In subcatalog-systems a file that is created with size equal to 0 will be extendable. This means that the file will be extended if the user writes more data into the file, than can be held in the blocks actually allocated to the file. If <size> is >0 the length of the file will be fixed, i.e. the file can not be extended.

4. In logical disc-system a file that is created with <size> equal to 0 will be given a length corresponding to the number of free blocks in the logical disc. The file can then be used for output, and when it is CLOSEd (see Sect. 8.2), the system will truncate it. No more than one file (in logical disc-systems) created with <size> equal to 0 can be used, unless the files in question have already been CLOSEd once.

5. <size> must be positive, if the file is a random access file.

Page 138, line 1:

✓ 5 -> 6

Page 138, line 7:

✓ 6 -> 7

Page 138, line -5 through -3:

✓ 1. A file can be deleted only if the user is exclusive user of the logical disc containing the file to be deleted (see Chapter 7, CONNECT).

Page 138, DELETE:

✓ Remarks 1-3 only applies to logical disc-system. In subcatalog-systems a file can be deleted, if the user has write-access to the subcatalog.

Page 143, line 5:

✓ mode 0 or 3 -> mode 0, 2 or 3.

Page 149, line -13:

✓ mode 0 or 3 -> mode 0, 2 or 3.

Page 160, line -8 and
Page 170, line -3:

✓ disc file -> disc file (only possible if the user has write-access to a logical disc/subcatalog).

Page 161, insert after last line, and
Page 170, insert after line 5:

3. If the SAVEd program is on disc, the system searches the logical disc/subcatalog to which the terminal is connected for <filename> (see Chapter 8). If <filename> is not found then the system searches for a logical disc/ subcatalog called LIB and if this is found then a search is made in this logical disc/subcatalog for <filename>. If <filename> is not found, the system outputs the error message 0100: FILE UNKNOWN.

Page 162, insert before line 5:

9.10A LOWBOUND

Format

LOWBOUND=<expr>

<expr>: a numeric expression which evaluates to 0 or 1

Use

As a command or statement to change the lower bound of numeric and string arrays.

Remark

1. The default lower bound is 1 (i.e. the number of the first element in an array is one). By means of the LOWBOUND-statement the lower bound may, however, be set to zero.

2. If lowerbound is zero, the row and column no. zero is included in all matrix-operations.

3. The LOWBOUND-statement can be placed anywhere in an RC BASIC program. Each time an array element is referenced the current lower bound will be taken as the first element of the array.

Example	Comment
0010 LOWBOUND =1	
0020 DIM A(5)	
0030 FOR I=1 TO 5	
0040 LET A(I)=I	
0050 NEXT I	
0060 FOR I=1 TO 5	lower bound of A-array is 1
0070 PRINT A(I);	
0080 NEXT I	
0090 LOWBOUND =0	
0100 DIM B(5)	
0110 PRINT	
0120 FOR I=0 TO 4	now lower bound of A-array is 0
0130 PRINT A(I)	
0140 NEXT I	
0150 PRINT	
0160 FOR I=0 to 5	B-array has 6 elements (0 to 5)
0170 PRINT B(I);	
0180 NEXT I	

```
1  2  3  4  5
1  2  3  4  5
0  0  0  0  0  0
```

Page 165, line 3, 8 and 9:

NUL -> STX

New Error Messages (page 176-187)

0047: PARAMETER ERROR

The actual parameters used in a call of an assembler-coded subroutine do not correspond to the formal parameters as specified in the subroutine.

0049: FACILITY PROTECTED

The statement or command is protected and cannot be executed at the terminal.

0090: USER CALL ERROR 1

0091: USER CALL ERROR 2

These messages can be used by the programmer of assembler-coded subroutines if an error is detected during the execution of the subroutine.

0098: PAGING ERROR.

This error will only occur on a system running under the DOMUS-operating system. The reason is, that (part of) the virtual storage cannot be read from the disc. The user's current program is destroyed. If the error occurs regularly the reason is probably malfunctioning hardware.

0099: STACK OVERFLOW.

A stack overflow may occur, if the user has recursive functions or procedures in his program, for instance 10 DEF FNA(X) = FNA(X). If this is not the reason, then please contact RC.

No 0105, 0110, 0115:

LD -> LD/SUBCATALOG

0160: CATALOG I/O ERROR

Some kind of inconsistence was discovered in the catalog. Contact RC.

0161: SUBCATALOG UNKNOWN

The user attempted to connect his terminal to a subcatalog which does not exist or to which no link has been made (see ref. [1]).

0165: DISC WRITEPROTECTED

An attempt has been made to write data on a (flexible) disc which is writeprotected.

0166: ILLEGAL OPERATION

1) A wrong protection key was specified or
2) The disc-unit containing a subcatalog has not been initialized or
3) An attempt has been made to connect to a file which is not a subcatalog.

0167:

This error message means that NO FREE AREAPROCESSes are available for the time being (see ref. [3] and [2]). As error messages are fetched from a discfile, and as this operation requires an areaprocess, the text cannot be fetched. Therefore the text that is written in connection with error no. 0167 is random.

0172: INDEXBLOCK FULL

In connection with each file an indexblock is used. This indexblock can be filled up, if a file is very large. Usually the disc must be reorganized.

0180: Same as 0160

0181: DISC OFF-LINE

An attempt has been made to access a disc, which was not ON-LINE.

0185: Same as 0165.

0186: ILLEGAL OPERATION ON FILE
The file is reserved by another user i.e. by a program running simultaneously with the BASIC-system.

0187: DISC FULL
The disc on which a file is created or extended is full.

0192: Same as 0172.

As it can be seen, many of the messages 0160-0172 and 0180-0192 are alike each other. In general it can be said, that errors 0160-0172 occur in connection with operations on (sub)catalogs, while 0180-0172 occur in connection with operations on files.

Page 189, line 5:

Disc write-protected -> Skip block

Page 211:

Insert DIGITS, LOWBOUND, PROTECT and RESET as reserved words.

Page 212, insert after line 4:

$$\text{CALL} \quad \left\{\begin{array}{l}\text{<svar>}\\ \\ \text{<slit>}\end{array}\right\} \left[\ ,\ \left\{\begin{array}{l}\text{<var>}\\ \text{<svar>}\\ \text{<mvar>}\\ \text{<sarray>}\\ \text{<slit>}\\ \text{<expr>}\end{array}\right\}\right] \ \ldots \qquad \text{H}$$

Invokes the execution of an assembler-coded subroutine. STATEMENT

Page 213, insert before line 4:

DIGITS = <expr> 3.6A

Specifies the maximal number digits output in PRINT-statements in systems with extended precision. STATEMENT or COMMAND

Page 213, line 4 through 8:

$$\text{DIM} \quad \left\{\begin{array}{l}\text{<svar> (<l>)}\\ \text{<sarray> (<n>,<l>)}\\ \text{<array> (<m>)}\\ \text{<array> (<row>,<col>)}\end{array}\right\} \left[\ ,\ \left\{\begin{array}{l}\text{<svar> (<l>)}\\ \text{<sarray> (<n>,<l>)}\\ \text{<array> (<m>)}\\ \text{<array> (<row>,<col>)}\end{array}\right\}\right] \quad 3.7$$

Defines the size of string variables, string arrays or numeric arrays. STATEMENT or COMMAND

Page 215, insert after line -10:

PROTECT = <expr>	3.26A
Protects certain statements and commands so that they cannot be executed at the terminal.	STATEMENT

Page 216, insert after line 12:

$\text{RESET}\left\{\begin{matrix}\text{ERR}\\ \\ \text{ESC}\end{matrix}\right\}$	3.30A
Restores normal function of the ESCape key and normal error action.	STATEMENT or COMMAND

Page 225, insert efter line -5:

LOWBOUND = <expr>	9.10.A
Sets the lower bound of arrays to 0 or 1	COMMAND or STATEMENT

Corrections to the Index (page 230-239)

Page 231:

U CALL, H

Page 232:

U DIGITS, 3.6A, I

Page 234:

LIB, 3.3, 9.10, 9.16
Logical discs
- write access, 7.2
LOGON/LOGOF functions, G
LOWBOUND, 2.4, 3.7, 5.1, 6.2, 9.10A

Page 237

PROTECT, 3.26A, G
RESET, 3.20, 3.21, 3.30A

Page 238:

Strings
- arrays 3.7, 5.1, 9.10A

Page 238:

line -14 is replaced by:
subcatalog, see Logical disc.
subcatalog (in a logical disc), see Catalog.

RETURN LETTER

Title: RC BASIC/COMAL Corrections (no. 3) to the RC BASIC Programming Guide

RCSL No.: 43-GL 9609

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: ______________ Title: ______________

Company: ______________

Address: ______________

Date: ________

Thank you

42-i 1288

. . . . Fold here . . . .

. . . . Do not tear - Fold here and staple . . . .

Affix
postage
here

A/S REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark