

REMODEL & PROLOAD

REnumber - MOve - DElete
PROgram - LOADer

Ein TRS-80 BASIC Dienstprogramm

I N H A L T

| | | |
|----------|-----------------------------------|----|
| REMODEL | Einfuehrung..... | 2 |
| | Arbeitsweise..... | 2 |
| | Laden und Ausfuehren..... | 3 |
| | Befehls-Eingabe..... | 4 |
| | Beispiele - Korrekte Arbeit..... | 7 |
| | Beispiele - Unrichtige Eingaben.. | 10 |
| | Hinweise und Vorschlaege..... | 13 |
| PROLOAD | Einfuehrung..... | 14 |
| | Arbeitsweise..... | 15 |
| | Laden und Ausfuehren..... | 15 |
| | Befehlseingabe..... | 16 |
| | Beispiele - Korrekte Eingaben.... | 19 |
| | Beispiele - Falsche Eingaben..... | 22 |
| | Hinweise und Vorschlaege..... | 24 |
| ANHANG A | Benutzung der Bandkassette..... | 25 |
| ANHANG B | REMODEL/PROLOAD unter DOS..... | 26 |
| ANHANG C | REMODEL/PROLOAD mit TRSDOS 2.2/3. | 27 |

Autoren: T. S. JOHNSTON und T. D. JOHNSTON
Herausgeber: RACET computes, ORANGE, KALIFORNIEN

Uebersetzung: REINHARD ENGEL
Vertrieb: ELSOFT, Koeln

© Copyright 1979 by RACET computes, ORANGE, KALIFORNIEN.
Alle Rechte vorbehalten.

NAMENSERKLAERUNG: REMODEL = REnumber - MOVe - DELete
(Umnumerieren - Verschieben - Loeschen)

EINFUEHRUNG

REMODEL ist ein leistungsfahiges Dienstprogramm, mit dem der TRS-80 Benutzer BASIC Programme effizient entwickeln und pflegen kann. Mit REMODEL sind jetzt fuer den LEVEL II TRS-80 Funktionen verfuegbar, die frueher nur auf grossen Systemen zu finden waren. Die professionellen Programmierer von RACET haben darueber hinaus dafuer gesorgt, das REMODEL effizient, verlaesslich und leicht benutzbar ist.

Folgende einzigartigen FUNKTIONEN bietet Ihnen nur REMODEL:

- o NUMERIEREN Sie ein ganzes BASIC Programm oder Teile davon neu. Jeder Bezug auf eine geaenderte Zeilennummer wird in in allen anderen Befehlen entsprechend geaendert.
- o VERSCHIEBEN Sie beliebige Teile eines BASIC Programms von einer Stelle im Programm an eine andere. Befehle werden automatisch wie erfordert neu nummeriert.
- o LOESCHEN sie einen beliebigen Teil eines BASIC Programms.

REMODEL bietet Ihnen damit erstmalig folgende VORTEILE:

- o Durch Umgruppierung und Neunummerierung von Programmteilen koennen Sie durchorganisierte Programme erzeugen. Sie erhoehen damit die Lesbarkeit Ihrer Programme und erleichtern sich die Programmdokumentation.
- o Sie koennen den Abstand zwischen der ersten und letzten Zeilennummer des Programms vergruessern. Sie schaffen sich damit mehr Raum fuer Ihre Programmzeilen und vermeiden dadurch die Notwendigkeit mehrerer Befehle pro Zeile und anderer schlechter Programmiergewohnheiten.

Diese Funktionen und Vorteile werden von Professionellen als unbedingt notwendig fuer eine effiziente Programm-entwicklung und -pflege angesehen. Sie sind jetzt fuer alle Benutzer eines TRS-80 Systems verfuegbar.

ARBEITSWEISE

REMODEL ist ein Systemprogramm, das den oberen Speicherbereich belegt. Normalerweise belegen die BASIC Programme, die Sie bearbeiten wollen, den unteren Speicherbereich. REMODEL-Befehle koennen beliebig oft mit Standard-BASIC TRS-80-Funktionen wie CLOAD, CSAVE und EDIT gemischt werden, ohne dass REMODEL neu geladen werden muesste.

REMODEL erfordert etwa 2K Speicher. Sogar bei sehr langen BASIC Programmen ist es moeglich, zu CLOADen und REMODEL-Befehle auszufuehren. Es kann jedoch vorkommen, dass eine Ausfuehrung des BASIC Programms mit RUN nicht moeglich ist, weil das Programm dabei groessere Speicherbereiche fuer Daten, Variable, Arrays oder Strings braucht. In diesem Falle muss das Programm nach allen REMODEL-Operationen zunaechst mit CSAVE abgespeichert und danach ohne REMODEL mit CLOAD wieder eingelesen werden.

LADEN UND AUSFUEHREN

REMODEL wird mit dem TRS-80 SYSTEM-Befehl geladen. Der obere Speicherbereich muss beim Einschalten geschuetzt werden. Bitte folgen Sie den folgenden Schritten, um REMODEL zu laden und auszufuehren. Beachten Sie: Ihre Eingaben sind unterstrichen.

1. Schalten Sie Ihr System ein. Bei Systemen ohne Expansion-Interface koennen Sie das Einschalten auch durch folgende Eingaben simulieren:

SYSTEM
*?/0

2. Setzen Sie jetzt die Speichergroesse:

MEMORY SIZE? mmmmm 29400 (16k-Auft.)

Bitte entnehmen Sie den richtigen Wert 'mmmmm' fuer Ihr System dem Anhang A.

3. Bereiten Sie den Rekorder mit dem REMODEL-Band vor.

4. Geben Sie folgende Befehle ein:

| | |
|---------------|---------------------------------------|
| <u>SYSTEM</u> | (Sie geben ein 'SYSTEM'.) |
| *?REM | (Sie geben ein 'REM'. REMODEL laedt.) |
| *?/ | (Wenn fertig, geben Sie ein '/'.) |
| READY | (Das System reagiert mit 'READY'.) |
| <u>CLEAR</u> | (Geben Sie ein 'CLEAR'.) |

5. Geben Sie Ihre BASIC Programme ueber die Tastatur oder von Band oder Diskette ein.

6. Rufen Sie REMODEL so oft wie erfordert durch folgenden Befehl auf:

?USR(0) (ENTER)

... muss ohne Leeraste eingegeben werden

7. Arbeiten Sie mit REMODEL, wie im folgenden Abschnitt dieses Manuals im Detail beschrieben.

8. Wenn Sie REMODEL beenden, kehren Sie in den READY-Zustand zurueck. Sie koennen dann mit den normalen BASIC-Befehlen

weiterarbeiten. Durch Wiederholung von Schritt (6) oben koennen Sie REMODEL beliebig oft wieder aufrufen.

Sie koennen REMODEL mit Schritt (1) oben aus dem Speicher loeschen. Dadurch wird auch das BASIC Programm geloescht. Wenn Sie dieses Programm nicht verlieren wollen, muessen sie es vorher auf Band oder Diskette abspeichern.

BEFEHLS-EINGABE

Wenn REMODEL durch '?USR(0)' aufgerufen wird, erscheinen die beiden folgenden Zeilen:

```
REMODEL V3.0 - ENTER DATA
START?_   END?       NEW?       BY?       MODE?
```

(Bei Ihnen erscheint natuerlich Ihre Version statt 'V3.0'.) REMODEL ist nun bereit zur Eingabe. Der blinkende Cursor steht am Anfang des ersten Feldes in der zweiten Zeile. Die Tafeln (1-2) geben eine Uebersicht ueber die Benutzung der Tastatur und den Zweck der einzelnen Felder waehrend des INPUT-Zustands von REMODEL:

| Taste | Funktion |
|------------|--|
| → | Cursor springt an den Anfang des naechsten Feldes und ueberspringt vorhandene Zeichen. |
| ← | Cursor geht um eine Stelle innerhalb eines Feldes oder ans Ende des vorhergehenden Feldes zurueck. Uebersprungene Buchstaben werden nicht geaendert. |
| CLEAR | Alle Felder werden geloescht. Cursor springt an Anfang des ersten Feldes zurueck. |
| BREAK | REMODEL wird beendet. System kehrt in den READY-Zustand von BASIC zurueck. |
| LEER-TASTE | Eine Leerstelle wird an der gegenwaertigen Stelle des Cursors eingefuegt. Vorhandene Zeichen an dieser Stelle werden geloescht. |
| ENTER | Pruefung der eingegebenen Werte wird veranlasst, wobei die tatsaechlichen Werte und der gewaehlte Modus angezeigt werden. Der EXECUTE-Zustand - in Tabelle (3) beschrieben - wird aufgerufen. Das System wartet auf die nochmalige Bestaetigung der der eingegebenen Daten durch den Benutzer. |
| andere | Das entsprechende Zeichen wird an der Stelle des Cursors eingefuegt. |

Tabelle (1). Tastaturfunktionen im INPUT-Zustand

Nachdem Sie die erforderlichen Daten in die Felder eingegeben und (ENTER) gedrueckt haben, prueft REMODEL alle Felder. Dabei werden Fehler gesucht, Leerwerte (defaults) eingesetzt und/oder die tatsaechlichen Zeilennummern eingesetzt. Diese Werte, die REMODEL benutzt, ersetzen die Werte, die Sie eingegeben haben.

REMODEL beginnt nicht mit einer Operation, die Fehler wie z.B. eine Zeilenueberlagerung waehrend einer Blockverschiebung verursacht. Bei Entdeckung einer moeglichen Fehlerquelle wird der INPUT-Zustand wieder aufgerufen, so dass Sie das Problem beseitigen koennen.

Ist alles in Ordnung, wird der EXECUTE-Zustand (execute=ausfuehren) aufgerufen. Der Cursor steht dabei auf der ersten Zeile der Anzeige. Das System wartet auf Ihre endgueltige Bestaetigung der angezeigten Werte:

REMODEL V3.0 - OK(Y/N)? _

Wenn Sie 'N' (Nein, nicht in Ordnung) eingeben, springt das System in den INPUT-Zustand zurueck.

Wenn Sie 'Y' (Ja, alles in Ordnung) eingeben, fuehrt das System die festgelegte Operation aus. Dabei zeigt es auf der ersten Zeile an:

REMODEL V3.0 - OK(Y/N)?YES - EXECUTING nnnnn

Der Wert 'nnnnn' zeigt die Nummer der Zeile an, die gerade verarbeitet wird. Wenn die Aufgabe beendet ist, erscheint:

REMODEL V3.0 - ENTER DATA

Das System befindet sich wieder im INPUT-Zustand und wartet auf Ihre weiteren Eingaben. Mit BREAK koennen Sie REMODEL verlassen und nach BASIC zurueckspringen.

| FELD | BEDEUTUNG | LEERWERT (DEFAULT) |
|-------|--|--|
| START | Zeilennummer, mit der die gewünschte Operation beginnen soll. Wenn es diese Nummer im Programm nicht gibt, wird die nächst grössere vorhandene Nummer benutzt. | Erste Zeilennummer des Programms |
| END | Zeilennummer, mit der die gewünschte Operation enden soll. Wenn es diese Nummer im Programm nicht gibt, wird die nächst kleinere vorhandene Nummer benutzt. | Letzte Zeilennummer des Programms |
| NEW | Die neue Zeilennummer, mit der ein Programmblock beginnen soll, der neu nummeriert oder verschoben werden soll | Zeilennummer, die im START-Feld angegeben oder als Leerwert eingesetzt wurde |
| BY | Abstand zwischen den Zeilen des Programmblocks | 10 |
| MODE | Kode, der die gewünschte Operation definiert. Muss nur beim Löschen mit 'DEL' angegeben werden. | Beim Numerieren und Verschieben wird der Modus abhängig von den Werten START, END und NEW automatisch gewählt. |

Tabelle (2). Beschreibung der INPUT-Felder

| TASTE | FUNKTION |
|--------|---|
| Y | Festgelegte Operation wird gestartet. |
| BREAK | Ausführung wird unterbrochen. REMODEL springt in den INPUT-Zustand zurück und wartet auf weitere Eingaben gemäss Tafel (1) und (2). Drücken Sie BREAK ein zweites Mal, um REMODEL zu beenden. |
| andere | Ausführung wird unterbrochen. REMODEL springt in den INPUT-Zustand zurück und wartet auf weitere Eingaben. |

Tabelle (3). Tastaturfunktionen im EXECUTE-Zustand

BEISPIELE - KORREKTE ARBEIT MIT REMODEL

Die folgenden Beispiele Zeigen, wie Programme mit REMODEL auf korrekte Weise manipuliert werden koennen. Der Zustand vor und nach einer bestimmten Operation wird an einem trivialen Beispielprogramm gezeigt. Ihre Eingaben sind unterstrichen. Die geaenderten Teile des Programms sind schattiert.

1. WUNSCH: Das ganze Programm soll neu nummeriert werden. Es mit Zeile 10 beginnen. Der Zeilenabstand soll 10 betragen.

Benutzer: START? END? NEW?10 BY? MODE?
 Benutzt: START?5 END?125 NEW?10 BY?10 MODE?RES

PROGRAMM VORHER

```
5  PRINT 1: GOTO 24
24 PRINT 2: GOTO 125
25 PRINT 3: GOTO 63
50 PRINT 4: END
63 PRINT 5: GOTO 50
125 PRINT 6: GOTO 25
```

PROGRAMM NACHHER

```
10 PRINT 1: GOTO 20
20 PRINT 2: GOTO 60
30 PRINT 3: GOTO 50
40 PRINT 4: END
50 PRINT 5: GOTO 40
60 PRINT 6: GOTO 30
```

KOMMENTAR: Beachten Sie, dass die Leerwerte automatisch in die leeren Felder eingesetzt wurden.

2. WUNSCH: Der Programmteil von Zeile 24 bis 25 soll neu nummeriert werden. Die neuen Zeilennummern sollen 20 und 30 sein.

Benutzer: START?15 END?31 NEW?20 BY? MODE?
 Benutzt: START?24 END?25 NEW?20 BY?10 MODE?RES

PROGRAMM VORHER

```
5  PRINT 1: GOTO 24
24 PRINT 2: GOTO 125
25 PRINT 3: GOTO 63
50 PRINT 4: END
63 PRINT 5: GOTO 50
125 PRINT 6: GOTO 25
```

PROGRAMM NACHHER

```
5  PRINT 1: GOTO 20
20 PRINT 2: GOTO 125
30 PRINT 3: GOTO 63
50 PRINT 4: END
63 PRINT 5: GOTO 50
125 PRINT 6: GOTO 30
```

KOMMENTAR: Die genauen Werte brauchen nicht bekannt sein. Die erste Zahl nach 15(24) und die erste vor 31(25) wurden gewaehlt, weil sie die urspruengliche Definition des Bereichs erfuellen.

3. WUNSCH: Die Zeilen 24 - 50 sollen ans Ende des Programms gesetzt und neu numeriert werden. Der Start sei Zeile 200, der Zeilenabstand 100.

Benutzer: START?24 END?50 NEW?200 BY?100 MODE?
 Benutzt: START?24 END?50 NEW?200 BY?200 MODE?MOV

PROGRAMM VORHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 24 | PRINT 2: GOTO 125 |
| 25 | PRINT 3: GOTO 63 |
| 50 | PRINT 4: END |
| 63 | PRINT 5: GOTO 50 |
| 125 | PRINT 6: GOTO 25 |

PROGRAMM NACHHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 200 |
| 63 | PRINT 5: GOTO 400 |
| 125 | PRINT 6: GOTO 300 |
| 200 | PRINT 2: GOTO 125 |
| 300 | PRINT 3: GOTO 63 |
| 400 | PRINT 4: END |

KOMMENTAR: Die Verantwortung dafuer, dass das Ergebnis der Operation logisch wuenschenswert ist, liegt bei Ihnen.

4. WUNSCH: Die Zeilen 50 - 63 sollen zwischen die Zeilen 5 und 24 geschoben werden. Beginnend mit 10 sollen sie mit einem Abstand von 10 neu numeriert werden.

Benutzer: START?50 END?100 NEW?10 BY? MODE?
 Benutzt: START?50 END?63 NEW?10 BY?10 MODE?MOV

PROGRAMM VORHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 24 | PRINT 2: GOTO 125 |
| 25 | PRINT 3: GOTO 63 |
| 50 | PRINT 4: END |
| 63 | PRINT 5: GOTO 50 |
| 125 | PRINT 6: GOTO 25 |

PROGRAMM NACHHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 10 | PRINT 4: END |
| 20 | PRINT 5: GOTO 10 |
| 24 | PRINT 2: GOTO 125 |
| 25 | PRINT 3: GOTO 20 |
| 125 | PRINT 6: GOTO 25 |

KOMMENTAR: Programmblöcke können an eine beliebige neue Stelle geschoben werden, solange - wie oben - genügend Platz zwischen zwei aufeinanderfolgenden Zeilennummern vorhanden ist.

5. WUNSCH: Die Zeilen 50 - 63 sollen zwischen die Zeilen 24 und 25 geschoben werden. Zuvor soll dafuer Platz gemacht werden.

Benutzer: START?25 END?25 NEW?40 BY? MODE?
 Benutzt: START?25 END?25 NEW?40 BY?10 MODE?RES

Benutzer: START?50 END?63 NEW?30 BY?5 MODE?
 Benutzt: START?50 END?63 NEW?30 BY?5 MODE?

PROGRAMM VORHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 24 | PRINT 2: GOTO 125 |
| 25 | PRINT 3: GOTO 63 |
| 50 | PRINT 4: END |
| 63 | PRINT 5: GOTO 50 |
| 125 | PRINT 6: GOTO 25 |

PROGRAMM NACHHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 24 | PRINT 2: GOTO 125 |
| 30 | PRINT 4: END |
| 35 | PRINT 5: GOTO 30 |
| 40 | PRINT 3: GOTO 35 |
| 125 | PRINT 6: GOTO 40 |



KOMMENTAR: Im diesem Beispiel war zwischen den Zeilen 24 und 25 kein Platz vorhanden, um einen Block dazwischenzuschieben. Deshalb sind zwei Schritte notwendig. Der erste numeriert Zeile 25 um in Zeile 40, der zweite schiebt die Zeilen 50 - 63 zwischen 24 und 40 und erzeugt so das gewünschte Resultat.

6. WUNSCH: Die Zeilen 50 - 63 sollen geloescht werden.

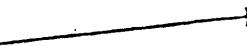
Benutzer: START?50 END?63 NEW? BY? MODE?D
 Benutzt: START?50 END?63 NEW? BY? MODE?DEL

PROGRAMM VORHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 24 | PRINT 2: GOTO 125 |
| 25 | PRINT 3: GOTO 63 |
| 50 | PRINT 4: END |
| 63 | PRINT 5: GOTO 50 |
| 125 | PRINT 6: GOTO 25 |

PROGRAMM NACHHER

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 24 | PRINT 2: GOTO 125 |
| 25 | PRINT 3: GOTO 63 |
| 125 | PRINT 6: GOTO 25 |



KOMMENTAR: Beachten Sie, dass die Felder NEW und BY nicht benutzt wurden. Sie brauchen auch nur 'D' fuer 'DEL' einzugeben. Sie sind auch dafuer verantwortlich, dass die gewünschten Aenderungen korrekt sind. In diesem Falle gibt es in Zeile 25 immer noch einen Bezug auf die geloeschte Zeile 63!

BEISPIELE - UNRICHTIGE EINGABEN

Unten finden Sie mehrere Beispiele fuer ungueltige Aenderungswuensche. Sie werden am gleichen Beispielprogramm erlaeutert:

| | |
|-----|-------------------|
| 5 | PRINT 1: GOTO 24 |
| 24 | PRINT 2: GOTO 125 |
| 25 | PRINT 3: GOTO 63 |
| 50 | PRINT 4: END |
| 63 | PRINT 5: GOTO 50 |
| 125 | PRINT 6: GOTO 25 |

REMODEL diagnostiziert das Problem, zeigt die entsprechende Meldung auf der ersten Zeile an und geht dann in den INPUT-Zustand zurueck. Sie koennen dann den Fehler korrigieren.

1. PROBLEM: Ein nicht-numerisches Zeichen wird in das START, END, NEW oder BY Feld eingeben.

Benutzer: START?2R END?25 NEW?20 BY?10 MODE?

Meldung: REMODEL V3.0 - DIGIT BAD - REDO

KOMMENTAR: (DIGIT BAD = Zahl falsch) Korrigieren Sie die falsche Zahl und druecken Sie wieder (ENTER).

2. PROBLEM: Die Zeilennummern werden in der umgekehrten Reihenfolge eingegeben.

Benutzer: START?25 END?24 NEW?20 BY?10 MODE?

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich) Der Wert im START Feld muss gleich oder kleiner dem Wert im END Feld sein.

3. PROBLEM: Die eingegebenen Parameter fuehren zu einer Ueberlagerung der Zeilennummern.

Benutzer: START?24 END?25 NEW?40 BY? MODE?

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich) Der Leerwert (default) 10 fuer BY haette zur Folge, dass die Zeilen 24 und 25 in 40 und 50 umnumeriert wuerden. Da jedoch Zeile 50 schon vorhanden ist, kaeme es zu einer Ueberlagerung.

4. PROBLEM: Ein Programmblock soll an eine Stelle mit unzureichendem Platz verschoben werden.

Benutzer: START?50 END?125 NEW?10 BY? MODE?

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich) Die Zeilen 50 bis 125 bekaemen die Nummern 10 bis 30. Die letzte Nummer (30) ueberlagert jedoch die vorhandenen Zeile 24 und 25.

5. PROBLEM: Die angegebenen Parameter enthalten eine Zeilennummer, die groesser als die hoechste zulaessige ist.

Benutzer: START? END? NEW?20000 BY?10000 MODE?

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich) Die groesste zulaessige Zeilennummer beim TRS-80 ist 65535. Hier wird versucht die Zeilen 5 bis 125 in 20000 bis 70000 umzunummerieren.

6. PROBLEM: Ein ungueltige Zeilennummer wird eingegeben.

Benutzer: START? END?83000 NEW? BY? MODE?

Meldung: REMODEL V3.0 - NUMBER BAD - REDO

KOMMENTAR: (NUMBER BAD = falsche Nummer)

7. PROBLEM: Eine ungueltige Zeilennummer wird eingegeben.

Benutzer: START? END? NEW?0 BY? MODE?

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich) Beim TRS-80 ist die Zeilennummer 0 nicht erlaubt.

8. PROBLEM: Der Zeilenabstand soll 0 betragen.

Benutzer: START? END? NEW? BY?0 MODE?

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich) Jeder Zeile die gleiche Nummer zu geben, ist offensichtlich sinnlos.

9. PROBLEM: Es sollen Zeilen geloescht werden, die es nicht gibt.

Benutzer: START?200 END?300 NEW? BY? MODE?DEL

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich) Der START Wert ist groesser als die groesste vorhandene Nummer.

10. PROBLEM: Es soll ein Bereich neu numeriert werden, den es gar nicht gibt.

Benutzer: START?2 END?3 NEW?1 BY?1 MODE?

Meldung: REMODEL V3.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = falscher Bereich)

HILFREICHE HINWEISE UND VORSCHLAEGE

1. Benutzen Sie nicht den GOTO Befehl, um Ihr BASIC Programm ausfuehren, unmittelbar nachdem Sie REMODEL benutzt haben. Starten Sie Ihr Programm mit 'RUN'.
2. Benutzen Sie NICHT den RESET Knopf, waehrend REMODEL laeuft. Dadurch wird normalerweise das BASIC Programm im Speicher zerstoert.
3. REMODEL fuegt eine Leerstelle vor jeder Zeilennummer ein und braucht mehr Speicherplatz, wenn die Zahl der Ziffern in einer Zeilennummer groesser wird. Bei sehr langen Programmen kann es deshalb vorkommen, dass REMODEL mit einer OM-1 Fehlermeldung abbricht, da es das Problem nicht diagnostizieren kann, ehe seine Ausfuehrung beginnt. Sie sollten deshalb immer eine neuere Sicherungskopie (backup) Ihres Programms haben.
4. Die Numerierung oder Verschiebung kleiner Programmblöcke braucht proportional weniger Zeit als die Manipulation grösserer Blöcke oder ganzer Programme. Wenn Sie waehlen koennen, entscheiden Sie sich fuer kleine Blöcke, um Zeit zu sparen.
5. REMODEL nimmt an, dass das zu numerierende oder zu verschiebende Programm syntaktisch korrekt ist. Es liegt in Ihrer Verantwortung sicherzustellen, dass alle Bezuege auf Zeilennummern korrekt und angemessen definiert sind. REMODEL ignoriert alle Bezuege auf nicht definierte Zeilennummern.
6. Das TRS-80 System prueft die Validitaet der Programme, die mit CLOAD geladen werden, nicht angemessen. Deshalb kann das geladene Programm Fehler enthalten, die REMODEL veranlassen koennen, die Ausfuehrung abzubrechen oder unerwartete Ergebnisse zu erzeugen.
7. REMODEL kann auch NACH einem BASIC Programm geladen werden, ohne dieses zu zerstoeren. Starten Sie einfach mit Schritt (4) der Anweisungen zum Laden auf Seite 3. Es ist jedoch erforderlich, den oberen Speicherbereich schon beim Einschalten fuer REMODEL zu reservieren.

NAMENSERKLAERUNG: PROLOAD = PROgram - LOADer
(Programm-Lader)

EINFUEHRUNG

PROLOAD ist ein leistungsfahiges Dienstprogramm, mit dem der TRS-80 Benutzer BASIC Programmotheken effizient aufbauen, pflegen und nutzen kann. Mit seinen einmaligen Moeglichkeiten ist PROLOAD eine Ergaenzung des leistungsfahigen REMODEL.

PROLOAD ist ein Subsystem von REMODEL. Sie koennen damit:

- o jeden beliebigen Teil eines BASIC Programms laden, das mit CSAVE gespeichert wurde;
- o ein BASIC Programm im Speicher durch einen beliebigen Teil eines mit CSAVE auf Band gespeicherten BASIC Programms erweitern;
- o die hinzugefuegten Programmzeilen neu numerieren, um ihrem Programm-Design gerecht zu werden. Dabei werden Bezuege auf geaenderte Zeilennummern im hinzugefuegten Teil automatisch geaendert.
- o das entstandene Gesamtprogramm umstrukturieren, indem Sie die hinzugefuegten Programmzeilen an eine beliebige Stelle verschieben;
- o einen beliebigen Teil eines BASIC Programms in einem kompatiblen Format abspeichern, das mit CLOAD oder PROLOAD wieder geladen werden kann;
- o pruefen, ob ein abgespeichertes Band korrekt ist.

Mit PROLOAD haben Sie auf Ihrem TRS-80 folgende Vorteile:

- o Sie koennen Programmotheken aus BASIC Programmen aufbauen, die einzeln oder in Verbindung mit anderen Programmen geladen werden koennen. Sie brauchen haeufig benutzte Routinen nicht immer wieder neu zu schreiben und erhoehen damit Ihre Produktivitaet als Programmierer erheblich.
- o Sie koennen schnell funktionierende und gut organisierte Programme erstellen, weil das manuelle umnumerieren von Programmzeilen entfaellt.
- o Sie koennen eine Bibliothek von DATA Befehlen nutzen, die selektiv mit vorhandenen Programmen gespeichert oder eingelesen werden koennen. Dies ist eine viel einfachere und effizientere Methode der Datenspeicherung und -wiedergewinnung als die Benutzung von Daten-Baendern.

ARBEITSWEISE

PROLOAD ist ein Subsystem von REMODEL. Beide muessen gleichzeitig im Speicher sein, damit die PROLOAD Befehle genutzt werden koennen. PROLOAD/REMODEL befindet sich im oberen Speicherbereich, waehrend BASIC Programme, die modifiziert oder geladen werden sollen, normalerweise den unteren Speicherbereich belegen. PROLOAD und REMODEL Befehle koennen so oft wie noetig mit Standard BASIC TRS-80 Operationen wie RUN oder EDIT vermischt werden, ohne dass PROLOAD/REMODEL neu geladen werden muessten.

PROLOAD (einschliesslich REMODEL) belegt etwa 3K des Speichers. Sogar bei sehr grossen BASIC Programmen ist es moeglich, PROLOAD/REMODEL Befehle auszufuehren. Es kann jedoch vorkommen, dass BASIC Programme, die viel Speicherplatz fuer Daten-Tabellen, Variable oder Strings benoetigen, nicht mit RUN ausgefuehrt werden koennen. In diesem Fall ist es notwendig, das BASIC Programm nach der Bearbeitung mit PROLOAD/REMODEL abzuspeichern und danach zur Ausfuehrung ohne PROLOAD/REMODEL neu zu laden.

LADEN UND AUSFUEHREN

PROLOAD wird mit dem TRS-80 SYSTEM-Befehl geladen. Das PROLOAD-Band enthaelt das REMODEL-System in einem Format, das mit oder ohne PROLOAD geladen werden kann. Befolgen Sie bitte die folgenden Schritte, um PROLOAD/REMODEL zu laden und auszufuehren. Beachten Sie, dass alle Benutzer-Eingaben unterstrichen sind. Ziehen Sie das REMODEL-Manual zu Rate, wenn Sie nur REMODEL laden wollen.

1. Schalten Sie Ihr System ein. Bei Systemen ohne Expansion-Interface koennen Sie den Schalter auch durch die folgende Eingabe simulieren:

SYSTEM
?/0

2. Beantworten Sie die Frage nach der Speichergroesse:

MEMORY SIZE?mmmmm

Bitte entnehmen Sie den richtigen Wert 'mmmmm' fuer Ihr System dem Anhang A.

3. Bereiten Sie Ihren Rekorder mit dem PROLOAD-Band vor.
4. Geben Sie die folgenden Befehle ein:

| | |
|---------------|-------------------------------------|
| <u>SYSTEM</u> | (Sie geben 'SYSTEM' ein.) |
| <u>?REM</u> | (Sie geben 'REM' ein.) |
| <u>?/</u> | (Sie geben '/' nach dem Laden ein.) |
| <u>READY</u> | (Das System reagiert mit READY.) |
| <u>CLEAR</u> | (Sie geben 'CLEAR' ein.) |

5. Fuehren Sie erforderliche BASIC Befehle aus.
6. Rufen Sie PROLOAD/REMODEL so oft wie noetig auf mit
?USR(0)
7. Fuehren Sie PROLOAD/REMODEL Operationen durch, wie sie im naechsten Abschnitt dieses Manuals und im REMODEL-Manual beschrieben sind.
8. Wenn Sie PROLOAD beenden, kehren Sie in den READY-Zustand zurueck und koennen mit normalen BASIC-Operationen weitermachen. Durch Wiederholung des Schrittes (6) koennen Sie PROLOAD/REMODEL so oft wie noetig aufrufen.

Wenn Sie die Speichergroesse neu setzen wollen, um sie grossen Datenmengen anzupassen, koennen Sie Schritt (1) oben ausfuehren. Beachten Sie, dass dadurch das BASIC Programm und PROLOAD/REMODEL aus dem Speicher geloescht werden. Sie muessen also vorher mit CSAVE oder einer PROLOAD-Operation den Speicherinhalt abspeichern. Damit koennen Sie sogar sehr grosse BASIC Programme bearbeiten.

BEFEHLS-EINGABE

Die Befehlseingabe bei PROLOAD ist im wesentlichen identisch mit der im REMODEL-Manual beschriebenen Befehlseingabe. Die INPUT-Felder, die in der Tabelle (2) des REMODEL-Manuals beschrieben wurden, sind gueltig, wenn Sie REMODEL Befehle benutzen.

Die Tabelle (1) unten beschreibt die Benutzung der INPUT-Felder, wenn Sie Programmteile von Band in den Speicher einlesen. Die Tabelle (2) unten beschreibt die Benutzung der INPUT-Felder, wenn Sie Programmteile auf Band speichern.

Wenn Sie Daten in die erforderlichen Felder eingegeben haben, druecken Sie auf ENTER. Dann ueberprueft PROLOAD alle Felder, sucht nach Fehlern und setzt Leerwerte (defaults) und/oder tatsaechliche Zeilennummern ein. Diese Werte ersetzen dann die Werte, die Sie eingegeben haben.

Danach geht das System in den EXECUTE Zustand, wobei der Cursor auf der ersten Zeile der Anzeige steht:

```
REMODEL V4.0 - OK(Y/N)?_
```

Das System wartet dann auf Ihren Befehl fortzufahren (Y) oder zurueck in den INPUT-Zustand zu gehen (N). Die nachfolgende Operation haengt davon ab, ob ein Band eingelesen oder abgespeichert werden soll.

Wenn Sie Programme von Band einlesen wollen, sollten Sie den Kassettenrekorder entsprechend vorbereiten, ehe Sie mit 'Y' antworten. Das Band wird dann auf das Programm hin durchsucht, das Sie im MODE-Feld angegeben haben. Wenn ein Programm durchlauft, wird sein Name im MODE-Feld angezeigt. Wenn das richtige Programm gefunden ist, wird es auf den angegebenen Bereich hin durchsucht. Ein blinkender * in der oberen rechten Bildschirmcke zeigt einen korrekten Lesevorgang an.

Nachdem das Programm in den Speicher eingelesen worden ist, kann es notwendig sein, dass es unnummeriert und/oder verschoben wird. Dies wird automatisch wie folgt bestimmt:

- o Wenn Sie das NEW- und das BY-Feld leer gelassen haben, versucht REMODEL die vom Band eingelesenen Programmzeilen an der richtigen Stelle einzumischen, ohne umzunummerieren.
- o Wenn Sie NEW und/oder BY angegeben haben, werden die eingelesenen Programmzeilen unnummeriert und, wenn moeglich, an die entsprechende Stelle geschoben.
- o Wenn dies zu einer Ueberlagerung von Zeilennummern fuehren sollte (die Programmzeilen wuerden nicht in den explizit oder implizit angegebenen Bereich passen), dann werden die Zeilen von REMODEL automatisch unnummeriert und ans Ende des Programms geschoben.

Wenn der Ladevorgang erfolgreich abgeschlossen wird, kehrt REMODEL in den INPUT-Zustand zurueck und wartet auf weitere Befehle. Wenn die Programmzeilen nicht in den angegebenen Bereich passen, erscheint die Meldung:

REMODEL V4.0 - MOVE BAD - REDO

(MOVE BAD - REDO = Verschiebung falsch - wiederhole) In diesem Fall ist das Programm wohl korrekt geladen worden, steht aber am Ende des Programms, wo es nicht hin sollte. Sie sollten dann mit REMODEL-Befehlen die Programmzeilen wie erforderlich verschieben:

Wenn Sie ein Programm auf Band speichern wollen, sollten Sie den Kassettenrekorder entsprechend vorbereiten, ehe Sie mit 'Y' antworten. Der Teil des Programms, den Sie im START- und END-Feld definiert haben, wird dann abgespeichert. Sie koennen diesen Teil jederzeit wieder mit CLOAD oder PROLOAD einlesen.

Programme, die Sie mit PROLOAD abgespeichert haben, koennen Sie mit dem =p= Befehl verifizieren, wobei 'p' fuer den Programmnamen steht. Der Inhalt des Bandes wird mit dem Speicherbereich verglichen, der im START- und END-Feld definiert ist. Diese Werte muessen identisch mit den Werten sein, mit denen dieser Bereich auf Band gespeichert wurde. Wenn die Daten falsch abgespeichert wurden, erscheint die Meldung:

REMODEL V4.0 - BAD COMPARE

(BAD COMPARE = Vergleich zeigt Fehler) In jedem Fall kehrt REMODEL in den INPUT-Zustand zurueck und wartet auf weitere Befehle.

PROLOAD fuehrt zusaetzliche Fehlerpruefungen durch, wenn ein Programm von Band eingelesen wird. Wenn ein Fehler entdeckt wird, erscheint die Meldung:

REMODEL V4.0 - TEXT BAD - REDO

(TEXT BAD - REDO = Text falsch - wiederhole) Daten, die vor dem Fehler eingelesen wurden, werden geloescht, woei das existierende Proramm erhalten bleibt. Diese Fehlerpruefung kann jedoch nicht alle Probleme erkennen, so dass Sie immer vorsichtig sein muessen.

| FELD | BEDEUTUNG | LEERWERT(DEFAULT) |
|-------|---|---|
| START | Zeilennummer auf dem Band, mit der das Laden beginnen soll. Gibt es diese Nummer nicht, wird die naechst-groessere vorhandene Nummer benutzt. | 0 |
| END | Zeilennummer auf dem Band, mit der das Laden enden soll. Gibt es diese Nummer nicht, wird die naechst-kleinere vorhandene Nummer benutzt. | 65535 |
| NEW | Wahlfreie neue Nummer, die der ersten vom Band eingelesenen Programmzeile zugeordnet wird. | Entweder keine oder die letzte Zeilennummer des Programms + BY. |
| BY | Wahlfreie Zahl, die zu jeder vom Band eingelesenen Zeilennummer addiert wird. | Entweder keine oder 10. |
| MODE | Der Lademodus wird durch "p" definiert, wobei 'p' der Name des zu ladenden Programms ist. | Das erste Programm auf dem Band wird geladen. |

Tabelle (1). INPUT-Felder bei Lade-Operationen

| FELD | BEDEUTUNG | LEERWERT(DEFAULT) |
|-------|--|-----------------------------------|
| START | Zeilennummer, mit der das Speichern oder Verifizieren beginnen soll. Gibt es diese Nummer nicht, wird die naechstgrossere vorhandene Nummer benutzt. | Die erste Zeile des Programms. |
| END | Zeilennummer, mit der das Speichern oder Verifizieren enden soll. Gibt es diese Nummer nicht, wird die naechstkleinere vorhandene Nummer benutzt. | Die letzte Zeile des Programms. |
| NEW | Wird fuer Speichern und Verifizieren nicht benutzt. | Nicht benutzt. |
| BY | Wird fuer Speichern und Verifizieren nicht benutzt. | Nicht benutzt. |
| MODE | Speichern wird durch (p), Verifizieren durch =p= definiert, wobei 'p' der entsprechende Programmname ist. | () oder = = leerer Programmname. |

Tabelle (2). INPUT-Felder fuer Speichern und Verifizieren

BEISPIELE - KORREKTE EINGABEN

An den beiden folgenden trivialen Programmen werden korrekte PROLOAD Befehle demonstriert. Es wird angenommen, dass sie auf Band gespeichert sind und an der richtigen Stellen stehen, wenn sie gebraucht werden:

PROGRAMM "A"

```
10 PRINT 1: GOTO 30
15 PRINT 2: END
30 PRINT 3: GOTO 15
```

PROGRAMM "B"

```
25 PRINT 4: GOTO 35
30 PRINT 5: STOP
35 PRINT 6: GOTO 30
```

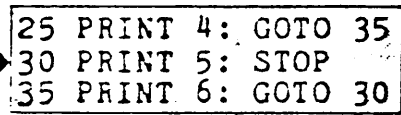
Ihre Eingabe ist unterstrichen. Die Bereiche, die eingelesen oder abgespeichert werden, sind schattiert.

1. WUNSCH: Lies Programm "A" in den Speicher ein.

Benutzer:START? END? NEW? BY? MODE?"A"
 Benutzt: START?0 END?65535 NEW? BY? MODE?"A"

SPEICHER VORHER

SPEICHER NACHHER



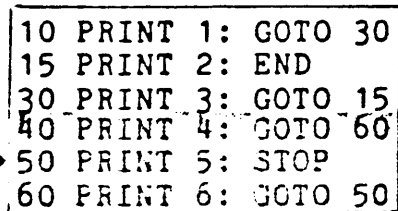
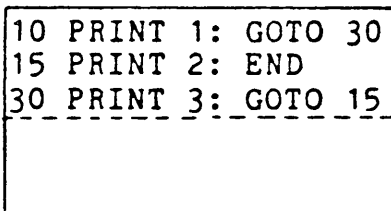
KOMMENTAR: Leerwerte (default values) wurden fuer das START- und END-Feld eingesetzt. Die Felder NEW und BY wurden nicht benutzt. Als Ergebnis wurden die Zeilen nicht umnummeriert.

2. WUNSCH: Lade Programm "B" nach Programm "A".

Benutzer:START? END? NEW?40 BY? MODE?"B"
 Benutzt: START?0 END?65535 NEW?40 BY?10 MODE?"B"

SPEICHER VORHER

SPEICHER NACHHER



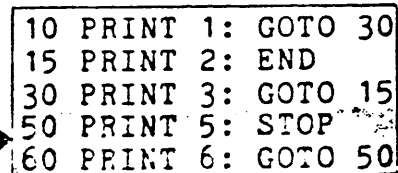
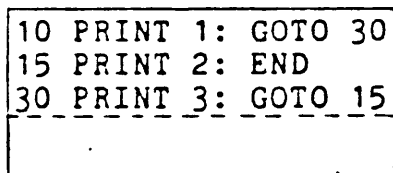
KOMMENTAR: Beachten Sie, dass der hinzugefuegte Teil unabhangig von den urspruenglichen Programmzeilen neu nummeriert wurde. Zeile 30 erscheint in beiden Programmen "A" und "B", wurde aber nur in Programm "B" umnummeriert.

3. WUNSCH: Lade Zeile 30 und 35 von Programm "B" ans Ende von Programm "A" und nummeriere neu mit 50 bzw. 60.

Benutzer:START?28 END?40 NEW?50 BY? MODE?"B"
 Benutzt: START?30 END?35 NEW?50 BY?10 MODE?"B"

SPEICHER VORHER

SPEICHER NACHHER



KOMMENTAR: Wie oben gezeigt wurden die genauen Werte nicht benoetigt.

4. WUNSCH: Lade Zeilen 30 und 35 von Programm "B" mit den neuen Nummern 20 und 25 zwischen die Zeilen 15 und 30 von Programm "A".

Benutzer: START?30 END?35 NEW?20 BY?5 MODE?"B"
 Benutzt: START?30 END?35 NEW?20 BY?5 MODE?"B"

SPEICHER VORHER

```
10 PRINT 1: GOTO 30
15 PRINT 2: END
30 PRINT 3: GOTO 15
```

SPEICHER NACHHER

```
10 PRINT 1: GOTO 30
15 PRINT 2: END
20 PRINT 5: STOP
25 PRINT 6: GOTO 20
30 PRINT 3: GOTO 15
```

KOMMENTAR: Diese Operation war erfolgreich, weil zwischen den Zeilen 15 und 30 genug Platz fuer die gewuenschten Zeilen war.

5. WUNSCH: Lade die Zeile 25 von Programm "B" zwischen die Zeilen 15 und 30 von Programm "A", ohne neu zu numerieren.

Benutzer: START?25 END?25 NEW? BY? MODE?"B"
 Benutzt: START?25 END?25 NEW? BY? MODE?"B"

SPEICHER VORHER

```
10 PRINT 1: GOTO 30
15 PRINT 2: END
30 PRINT 3: GOTO 15
```

SPEICHER NACHHER

```
10 PRINT 1: GOTO 30
15 PRINT 2: END
25 PRINT 4: GOTO 35
30 PRINT 3: GOTO 15
```

KOMMENTAR: Die Felder NEW und BY wurden nicht angegeben oder benutzt. Die eine eingelesene Zeile passt korrekt zwischen die angegebenen Zeilen von Programm "A". Sie als Benutzer sind verantwortlich, dass das Ergebnis uenschenswert ist.

6. WUNSCH: Speichere das Programm von Beispiel (5) oben von Zeile 15 bis 25 auf Band und nenne das Programm "C".

Benutzer: START?15 END?25 NEW? BY? MODE?(C)
 Benutzt: START?15 END?25 NEW? BY? MODE?(C)

SPEICHER VORHER

```
10 PRINT 1: GOTO 30
15 PRINT 2: END
25 PRINT 4: GOTO 35
30 PRINT 3: GOTO 15
```

SPEICHER NACHHER

```
15 PRINT 2: END
25 PRINT 4: GOTO 35
```

KOMMENTAR: Beachten Sie, dass die Felder NEW und BY nicht benutzt wurden.

7. WUNSCH: Verifiziere das Band, das im Beispiel (6) oben erzeugt wurde.

```
Benutzer:START?13   END?27   NEW?   BY?   MODE?=C=
Benutzt:  START?15   END?25   NEW?   BY?   MODE?=C=
```

KOMMENTAR: Unter der Annahme, dass der Speicherinhalt und das Band mit Programm "C" mit Beispiel (6) uebereinstimmen beendet PROLOAD die Pruefung ohne Fehlermeldung. Beachten Sie, dass der Bereich mit 13-27 angegeben wurde. Der tatsaechliche Bereich jedoch war 15-25, was richtig war.

BEISPIELE - FALSCH EINGABEN

Unten stehen Beispiele von PROLOAD Befehlen, die ungueltig sind. Es wird angenommen, dass sich Programm "A" im Speicher befindet und Programm "B" zum Einlesen vorbereitet ist.

1. PROBLEM: Sie geben ein nicht-numerisches Zeichen in einer der Felder START, END, NEW oder BY ein.

```
Benutzer:START?2R   END?   NEW?   BY?   MODE?"B"
```

Meldung: REMODEL V4.0 - DIGIT BAD - REDO

KOMMENTAR: (DIGIT BAD - REDO = falsche Zahl - wiederhole) Dieser und andere Fehler, die mit einer falschen Dateneingabe zu tun haben, sind im REMODEL-Manual beschrieben.

2. PROBLEM: Sie versuchen einen Bereich von Programmzeilen einzulesen, den es nicht gibt.

```
Benutzer:START?50   END?60   NEW?   BY?   MODE?"B"
```

Meldung: REMODEL V4.0 - RANGE BAD - REDO

KOMMENTAR: (RANGE BAD = Bereich falsch) Die Zeilen 50 und 60 gibt es in Programm "B" nicht. So entsteht der Fehler. Programm "A" bleibt intakt.

3. PROBLEM: Sie versuchen, Programm "B" in einen Bereich zu laden, in den es nicht hineinpasst.

```
Benutzer:START?   END?   NEW?   BY?   MODE?"B"
```

Meldung: REMODEL V4.0 - MOVE BAD - REDO

KOMMENTAR: (MOVE BAD = Verschiebung falsch) Durch die Leerwerte wird versucht, alle Zeilen von Programm "B" ohne Ummumerierung in den Speicher einzulesen. Da die Zeile 30 von Programm "B" die Zeile 30 von Programm "A" ueberlagern wuerde, erzwingt PROLOAD eine Ummumerierung. Dadurch wird Programm "B" ans Ende von Programm "A" geschoben:

| SPEICHER VORHER | SPEICHER NACHHER |
|---------------------|---------------------|
| 10 PRINT 1: GOTO 30 | 10 PRINT 1: GOTO 30 |
| 15 PRINT 2: END | 15 PRINT 2: END |
| 30 PRINT 3: GOTO 15 | 30 PRINT 3: GOTO 15 |
| | 40 PRINT 4: GOTO 60 |
| | 50 PRINT 5: STOP |
| | 60 PRINT 6: GOTO 50 |

4. PROBLEM: Sie versuchen den Speicher, der Programm "A" enthaelt, gegen das Programm "B" auf Band zu verifizieren.

Benutzer:START? END? NEW? BY? MODE?=B=

Meldung: REMODEL V4.0 - BAD COMPARE

KOMMENTAR: (BAD COMPARE = Vergleich schlecht) Der Fehler wuerde unmittelbar nach dem Lesen des Programmnamens entdeckt werden. Es wird Byte fuer Byte verifiziert, so dass sogar ein einziges falsches Zeichen entdeckt wird.

HILFREICHE HINWEISE UND VORSCHLÄGE ZU PROLOAD

1. Lesen Sie den entsprechenden Abschnitt im REMODEL-Manual.
2. REMODEL/PROLOAD benutzt den ganzen verfügbaren Speicher, wenn Programme geladen werden. Wenn Sie versuchen, zuviele Programmzeilen zu laden, erscheint folgende Meldung:

REMODEL V4.0 - MEMORY BAD - REDO

(MEMORY BAD = Speicher reicht nicht) In diesem Fall bleibt das existierende Programm intakt. Alle hinzugefügten Zeilen werden gelöscht.

3. Wenn PROLOAD ein BASIC Programm nicht vom Band einliest, liegt es normalerweise an einem fehlerhaften Band. Es kann jedoch möglich sein, das Programm zu CLOADen, da diese Funktion weniger Validitätsprüfungen durchführt als PROLOAD. Wenn das Ladeproblem gelöst ist, kann das Programm mit PROLOAD wie gewünscht bearbeitet werden.
4. Nur das erste Zeichen des MODE Befehls-Kodes und der entsprechende Programmname müssen im MODE-Feld angegeben werden. So sind z.B. folgende drei MODE-Felder gültig:

MODE?"A MODE?(A MODE?=A

PROLOAD setzt automatisch das folgende Zeichen ein.

5. Sie können auch leere Programmnamen benutzen. Beim Laden wird dann das jeweils nächste Programm geladen. Beim Speichern oder Verifizieren wird ein leerer Programmname benutzt. Folgende MODE-Felder sind also gültig:

MODE?" MODE?(MODE?=_

6. Um Zeit zu sparen, ist es am besten, PROLOAD Befehle zu verwenden, die keine Umnummerierung erfordern (d.h. die Felder NEW und BY werden leer gelassen). Sonst empfiehlt es sich, kleine Abschnitte zu laden, die unnummeriert werden müssen. Dies kann erreicht werden, indem grössere Programmteile, die nicht unnummeriert werden müssen, vor den kleineren Abschnitten geladen werden, die unnummeriert werden müssen.
7. Achten Sie darauf, gute Sicherheitskopien Ihrer Programme zu machen. REMODEL/PROLOAD hat sich als verlässlich erwiesen. Aber seine Leistungsfähigkeit und Benutzerfreundlichkeit verleiten leicht dazu, ein Programm bis zur Unkenntlichkeit zu manipulieren.

ANHANG A. - BENUTZUNG DER BANDKASSETTE

Software, die Sie von ELSOFT / RACET computes gekauft haben, sollte bei der gleichen Lautstaerkeeinstellung laden wie die Baender, die Sie selbst mit Ihrem Computer produzieren. Wir empfehlen, dass Sie den Lese-/Schreib-Kopf und die Andrucksrollen Ihres Kassettenrekorders regelmaessig mit kommerziell erhaeltlichen Kassettenreinigungs- und Entmagnetisierungsmitteln reinigen. Verschmutzte Koepfe koennen einen beträchtlichen Lautstaerkeverlust und unerwünschtes Rauschen verursachen.

Baender mit Maschinensprache, die mit dem SYSTEM-Befehl geladen werden, sind erheblich empfindlicher gegenueber der richtigen Lautstaerkeeinstellung als BASIC Programme, die mit CLOAD geladen werden. Moeglicherweise muessen Sie mehrere Lautstaerkeeinstellungen probieren, um die Baender mit Maschinensprache korrekt einzulesen.

Ein 'C' in der oberen rechten Ecke Ihres Bildschirms oder die Meldung 'BAD PARAMETERS' (falsche Parameter) zeigen fehlerhaftes Laden an. Der SYSTEM-Lader prueft jeden eingelesenen Block auf Datenvaliditaet. Dadurch wird sichergestellt, dass der Speicherinhalt korrekt ist, wenn waehrend des Ladens kein Fehler angezeigt wurde. Bei BASIC Programmen dagegen findet nur eine minimale Fehlerkontrolle statt. Als Ergebnis finden Sie oft einen anscheinend fehlerfreien Ladevorgang (keine Fehlermeldungen), doch das Programm im Speicher ist fehlerhaft. Sie koennen den Speicherinhalt nach dem Laden eines BASIC Programms mit dem LIST-Befehl ueberpruefen.

RACET Computes Programme sind zweimal auf derselben Bandseite aufgezeichnet. Obwohl dies etwas teurer ist als die zweiseitige Aufzeichnung auf einem etwas kuerzeren Band, haben Sie einen Vorteil davon. RACET Computes Programme werden in einer fuenffach verschraubten Kasette geliefert. Wenn das Band aus irgendeinem Grund reißen sollte, koennen Sie die Kasette oeffnen und das Band flicken, ohne dass die zweite Aufzeichnung beschaedigt wird.

Wenn Sie die Speichergroesse Ihres Ssystems vergroessern, koennen Sie die entsprechend groessere Version Ihres Programms gegen die reinen Bearbeitungskosten von DM 10,- von ELSOFT erhalten. Legen Sie Ihrer Bestellung die Rechnung bei und geben Sie den Programmnamen und die gewuenschte Version an.

| Programmname | Speicher | Version | MEMORY SIZE |
|-----------------|----------|---------|-------------|
| REMODEL | 16K | 3.0 | 30600 |
| | 32K | 3.1 | 46984 |
| | 48K | 3.2 | 63368 |
| REMODEL+PROLOAD | 16K | 4.0 | 29400 |
| | 32K | 4.1 | 45784 |
| | 48K | 4.2 | 62168 |

ANHANG B. - REMODEL/PROLOAD UNTER DOS

Die folgenden Schritte zeigen Ihnen, wie Sie REMODEL/PROLOAD unter DOS laden und benutzen koennen. Die notwendigen Eingaben sind unterstrichen. Die Werte 'aaaa', 'bbbb', 'cccc' und 'dddd' muessen der folgenden Tabelle entnommen werden:

| <u>SYSTEM</u> | <u>VERSION</u> | <u>aaaa</u> | <u>bbbb</u> | <u>cccc</u> | <u>dddd</u> |
|---------------|----------------|-------------|-------------|-------------|-------------|
| REMODEL | 16K | 7789 | 7FFF | 7789 | 30600 |
| | 32K | B789 | BFFF | B789 | 46984 |
| | 48K | F789 | FFFF | F789 | 63368 |
| REMODEL | 16K | 72D9 | 7FFF | 7789 | 29400 |
| + | 32K | B2D9 | BFFF | B789 | 45784 |
| PROLOAD | 48K | F2D9 | FFFF | F789 | 62168 |

A. LADEN UND AUSFUEHREN DES SYSTEMS VOM BAND

1. Rufen Sie TRSDOS auf.
2. BASIC (BASIC aufrufen.)
3. HOW MANY FILES? (Eingabe wie erfordert.)
4. MEMORY SIZE?dddd ('dddd' aus Tabelle oben.)
5. CMD"t" (Uhr ausschalten.)
6. SYSTEM (SYSTEM-Befehl ausfuehren.)
7. *?REM (System-Band laden.)
8. *?(BREAK) (BREAK-Taste druecken.)
9. DEFUSR=&Hcccc ('cccc' aus Tabelle oben.)
10. REMODEL/PROLOAD koennen jetzt benutzt werden. Rufen Sie das System ?USR(0) wie beschrieben auf.

B. UEBERTRAGUNG DER SYSTEME AUS DEM SPEICHER AUF DISKETTE

1. Laden Sie das System-Band wie oben beschrieben in den Speicher.
 2. CMD"S" (Rueckkehr zu TRSDOS.)
 3. DUMP REM/OBJ (START=X'aaaa',END=X'bbbb',TRA=X'cccc')
 4. DIR (Pruefen, ob REM/OBJ auf Diskette ist.)
- (Sie koennen auch einen anderen Namen als 'REM/OBJ' waehlen, etwa 'PRO/OBJ', wenn Sie mehrere Systeme gleichzeitig auf der Diskette haben.)

C. LADEN UND AUSFUEHREN VON DISKETTE

1. Rufen Sie TRSDOS auf.
2. LOAD REM/OBJ (Keine Anfuhrungszeichen.)
3. BASIC (BASIC aufrufen.)
4. HOW MANY FILES?_____ (Eingabe wie erfordert.)
5. MEMORY SIZE?dddddd ('dddddd' aus Tabelle oben.)
6. DEFUSR=&Hcccc ('cccc' aus Tabelle oben.)
7. CLEAR ('CLEAR' eintippen.)
8. REMODEL/PROLOAD koennen jetzt benutzt werden. Rufen Sie das System mit '?USR(0)' wie beschrieben auf.

Beachten Sie, dass REMODEL/PROLOAD so lange im geschuetzten Speicherbereich bleiben, wie der TRS-80 in DOS BASIC ist. Wenn Sie von DOS BASIC zu TRSDOS zurueckgehen, muessen Sie die Schritte (3-7) oben in (C) wiederholen, wenn Sie wiederum zu DOS BASIC gehen.

Benutzen Sie die normalen DOS Prozeduren, um Programme auf Diskette zu speichern oder von Diskette zu laden. Benutzen Sie PROLOAD zum Lesen oder Speichern von Programmen oder Programmteilen von bzw. auf BAND.

ANHANG C. - REMODEL/PROLOAD unter TRSDOS 2.2 und 2.3

Als TANDY von TRSDOS 2.1. auf TRSDOS 2.2. umruestete, hat die Firma willkuerlich entschieden, die oberen 51 Bytes des RAM zu benutzen, wenn von DOS zu BASIC gebootet wird. Dies steht in eklatantem Widerspruch zu den geschriebenen Worten im DOS/Manual, in dem versichert wird, dass der 'geschuetzte Speicherbereich' g e s c h u e t z t sei !!!

Was koennen Sie dagegen tun?

Statt REMODEL oder REMODEL+PROLOAD im DOS zu laden, gehen Sie ins BASIC und setzen Sie MEMORY SIZE wie gewohnt. Geben Sie dann ein: CMD"I","dateiname". Setzen Sie DEFUSR und benutzen Sie die Utility wie im Manual beschrieben.