Im Mathematikunterricht der Schule führt der Computer bisher ein Schattendasein, obwohl er aus Wirtschaft und Verwaltung nicht mehr wegzudenken ist. Das ist besonders für die Schüler der Sekundarstufe I (S I) beklagenswert, stellen sie doch den Hauptteil der Auszubildenden in allen Bereichen der Industrie und des öffentlichen Lebens, wo sie sehr bald mit einer Vielzahl von Anwendungen der EDV konfrontiert werden.

Gunter Alle

Computer in der Sekundarstufe I?

1 Problemstellung

Wenn die Schule ihren Bildungsauftrag, den jungen Menschen für unser heutiges Leben auszustatten, ernst nimmt, kommt die Sekundarstufe I (S I) nicht an der elektronischen Rechenmaschine vorbei; die von vielen Seiten geforderte „Entmystifizierung des Computers" muß in der S I angepackt werden.

Bisher konnte man sich bequem mit den horrenden Kosten für eine EDV-Anlage herausreden. Heute jedoch, wo Mikrocomputer unter DM 2000.– und programmierbare Taschenrechner (TR), die im Unterricht fast einen Kleincomputer ersetzen, unter DM 300.– zu haben sind, darf das Kostenproblem kein Argument mehr sein.

Die Schwierigkeiten liegen ganz woanders; sie sind **curricularer Art.** Einem 12- bis 16-jährigen Schüler mit bescheidenen Mathematikkenntnissen kann man nämlich keinen Informatikunterricht mit Erklärung der DIN-Symbole für **Flußdiagramme, Schaltalgebra, Konvergenzbetrachtungen, Top-down Methode** usw. anbieten. Unsere Forderung:

Der Computer (der programmierbare TR) darf nicht als Unterrichtsgegenstand, sondern muß als Hilfsmittel in den Mathematikunterricht der S I eingeführt werden, um damit eine Klasse von Aufgaben einfacher und wirksamer zu lösen. Der Schüler soll vor allem in der Mathematik weiterkommen; die dabei gewonnenen Einsichten in die Informatik nehmen wir als „wertvolle Nebenprodukte" dankbar zur Kenntnis.

In diesem Beitrag zeigen wir, wie eine für dieses Ziel besonders geeignet scheinende Klasse von Aufgaben, die Textgleichungen mit einer oder mehreren Variablen, linear oder quadratisch, im Unterricht der S I behandelt werden können.

Wir erklären bei den folgenden Beispielen, wie ein Schüler an Hand einer *Probiertabelle* die Struktur der Aufgabe aufspüren kann, wie er sich durch ma-

Curricular: Curriculum heißt so viel wie „Ablauf" (*Curriculum vitae:* Lebenslauf). Curricular ist ein daraus abgeleiteter Fachausdruck der Pädagogik und meint den stoffdidaktischen Ablauf eines Faches.

Flußdiagramm: Als Flußdiagramm bezeichnet man die graphische Darstellung eines logischen Ablaufs, also auch den Verlauf eines Programms.

Schaltalgebra: Von *Boole* wurde 1847 eine umfassende algebraische Beschreibung logischer Probleme entwickelt, die *Boolesche Algebra*. Sie wurde zunächst auf das von *Leibniz* 1700 in eine allgemeine Systematik eingebundene Dualsystem (Zweiersystem) einbezogen. Von *Shannon* wurden 1938 diese Erkenntnisse auf zweiwertige Zustände (Binärzustände) übertragen, woraus die Schaltalgebra entstand.

Konvergenzbetrachtungen: Konvergenz heißt Annäherung, Zusammenlaufen zweier Linien. Konvergenzbetrachtungen werden in der Mathematik angestellt, um eine Funktion daraufhin zu prüfen, ob sie gegen einen endlichen Wert strebt.

Top-down Methode: *Top* heißt „Spitze", *down* bedeutet „unten". Bei der Top-down Methode richtet sich also das didaktische Vorgehen von der Spitze nach unten.

nuelle Berechnungen der Lösung der Aufgabe annähert, und wie ihm schließlich der Rechner die langwierige Rechenarbeit abnimmt.

In einem zweiten Schritt setzen wir die Aufgabe in ein **Struktogramm** um. Der letzte Schritt zum fertigen Programm in BASIC und für den TI-58/59 ist mit minimalen Programmierkenntnissen zu meistern.

Noch eine Bemerkung zur Schreibweise der Programme:

Meines Erachtens fällt es Schülern wesentlich leichter, in horizontaler statt in vertikaler Richtung zu lesen; deshalb stehen in einer Programmzeile meistens mehrere Befehle. Der Mikrocomputer „Commodore 2001 – PET", in dessen BASIC-Version die Programme geschrieben sind, verlangt als Trennungszeichen zwischen den Befehlen lediglich den Doppelpunkt. Die Anordnung der Programme für den TI 58/59 entspricht den Zeilen der Struktogramme und der BASIC-Zeilen.

Wir erstellten die Programme mit didaktischer Absicht; sie sollten nicht möglichst elegant und effizient, sondern vielmehr für den Anfänger leicht durchschaubar sein.

2 Einführungsaufgabe

„Drei Brüder sind zusammen 170 Jahre alt. Der jüngste von ihnen ist drei Jahre jünger als der zweitälteste Bruder; der älteste ist fünf Jahre älter als der zweitälteste Bruder."

Wir fertigen eine „Probiertabelle" an, die nur eine Bedingung der Aufgabe, nämlich die Altersunterschiede, berücksichtigt. Die zweite Bedingung, die Summe der drei Altersangaben, setzen wir als *Zielwert* an das untere Ende der Summenspalte.

In die „Probiertabelle" (**Fig. 1**) läßt man beliebige Anfangswerte einsetzen, in unserem Beispiel wird das Alter des mittleren Bruders willkürlich auf 10 Jahre festgelegt; damit ist der jüngste Bruder 7 Jahre alt, der älteste aber 15 Jahre. Da die errechnete Summe 32 vom Zielwert 170 erheblich abweicht, nehmen wir als neues Alter eine größere Zahl für den mittleren Bruder an (s. **Fig. 1**), bestimmen wiederum das Alter der übrigen Brüder und vergleichen die Summe der Altersangaben mit dem Zielwert 170. Man sollte diese Prozedur nicht abkürzen; das langwierige Errechnen und Vergleichen motiviert den Rechnereinsatz. Nach einer Reihe von Durchgängen

A Ältester Bruder	M Mittlerer Bruder	J Jüngster Bruder	Summe
15	10	7	32
16	11	8	35
:	:	:	:
37	32	29	98
:	:	:	:
51	46	43	140
:	:	:	:
			→ 170

Fig. 1 Probiertabelle

```
INPUT STARTWERT M
SETZE ZIELWERT Z ⇐ 170
BERECHNE A ⇐ M + 5 : J ⇐ M − 3 ←
WENN A + M + J = Z DANN
ERHOEHE M UM 1 : RECHNE MIT NEUEM M WEITER
→ PRINT A M J
```

Fig. 2 Struktogramm

sollte die Struktur dieser Aufgabenstellung erkannt worden sein.

An dieser Stelle bringt man den Computer (Programmierbaren TR) ins Gespräch. Wie könnte man diese langwierige Rechnerei einem Computer übertragen?

- Einen Startwert für M eingeben
- A und J nach den Bedingungen der Aufgabe bestimmen
- Die Summe A + M + J mit 170 vergleichen
- Wenn die Summe mit dem Zielwert 170 übereinstimmt, soll der Computer die Altersangaben für A, M und J ausgeben
- Andernfalls soll der Rechner mit einem neuen Startwert wiederum A und J bestimmen
- Die Summe A + M + J mit 170 vergleichen
- . . . .

Das Struktogramm beschreibt diesen Sachverhalt mit Hilfe von Pfeilen (**Fig. 2**).

Struktogramm: Graphische Darstellungsmethode für Programmabläufe, bei der die Programmstruktur sichtbar werden soll.

```
10 INPUT „M=“; M
20 Z = 170
30 A = M + 5 : J = M – 3
40 IF A + M + J = Z THEN 60
50 M = M + 1 : GO TO 30
60 PRINT „A=“; A, „M=“; M, „J=“; J
```

Fig. 3 BASIC-Programm

```
[Startwert in] [STO] 00
170 [x⇄t]
[2nd] [Lbl] [A] [RCL] 00 [+] 5 [=] [STO] 01 [RCL] 00 [–] 3 [=] [STO] 02
[RCL] 00 [+] [RCL] 01 [+] [RCL] 02 [=] [2nd] [x = t] [B]
1 [SUM] 00 [A]
[2nd] [Lbl] [B] [RCL] 02 [R/S] [RCL] 00 [R/S] [RCL] 01 [R/S]
```

Fig. 4 Programm für den TI-58/59

Im Struktogramm und auch in den Rechnerprogrammen fügen wir nur eine Zeile, nämlich die Eingabe, bzw. die Zuweisung des Zielwertes 170 ein (**Fig. 3** und **Fig. 4**).

Auch einem Anfänger dürfte es keine unüberwindlichen Schwierigkeiten bereiten, ein Struktogramm in ein BASIC-Programm zu übersetzen. Wenn man gleichzeitig mit μ-Computern und programmierbaren TR arbeitet, wird man die unterschiedliche *Speicherorganisation* der Geräte näher erläutern müssen; ein Beispiel möge hier genügen:

A = M + 5 in BASIC bedeutet, daß die Speicherzelle mit dem Namen „A“ als neuen Speicherinhalt die Summe „Speicherinhalt M + 5“ zugewiesen bekommt. Der programmierbare TR hingegen „berechnet“ die Summe eines Speicherinhaltes + 5; dieses Ergebnis wird in einen anderen Speicher eingegeben. In unserem Beispiel: [RCL] 00 [+] 5 [=] [STO] 01.

Größere Schwierigkeiten wird der **bedingte Sprung** im Programm bereiten. Hier empfiehlt es sich, zunächst das t-**Register** und die **Programmadreßtasten** A, B, C ... E′ zu erklären. Mit dem Befehl [2nd] [x = t] vergleicht der TR, ob die Zahl in der Anzeige mit der durch [x⇄t] eingetasteten Zahl übereinstimmt. Trifft dies zu, so verzweigt der Rechner zur gewünschten Programmstelle, die wir mit einer Programmadreßtaste benannt haben.

Im andern Falle arbeitet der TR das Programm vom übernächsten Befehl an weiter ab. Auf den ersten Blick mag es dem Leser utopisch erscheinen, Schülern der S I das Arbeiten mit einem programmierbaren TR zuzumuten. Man sollte jedoch nicht vergessen, daß seit einigen Jahren nicht-programmierbare TR in den Schulen üblich sind, und daß für diese Geräte ja auch schon „Rechenablaufpläne“ erstellt wurden.

3 Aufgabe aus der Landwirtschaft

Wir kommen nun zu einer etwas komplizierteren Aufgabe aus der Landwirtschaft:

> „Ein Bauer besitzt 166 Stück Nutzvieh. Interessierten Kurgästen pflegt er auf die Frage nach der Anzahl seiner Tiere so zu antworten: ‚Der Schweinebestand ist um 9 kleiner als die vierfache Anzahl der Pferde und Rinder zusammen. Außerdem habe ich 6 mal soviel Kühe wie Pferde.‘ “

In **Fig. 5** sehen wir eine der Möglichkeiten, wie man (nach gründlichen Überlegungen und Diskussionen!) den mathematischen Inhalt der Aufgabe in eine „Probiertabelle“ umsetzt. **Fig. 6** zeigt das Struktogramm der Textaufgabe.

Auffällig ist die Strukturgleichheit zur Einführungsaufgabe, obwohl sie sich erheblich in Text, Zahlenangaben und mathematischen Relationen unterscheidet. In **Fig. 7** und **Fig. 8** geben wir die passenden Rechnerprogramme wieder.

Wir zeigen für diese Aufgabe noch eine weitere Möglichkeit, wie man von der „Probiertabelle“ aus zu einer unterschiedlichen Lösungstechnik kommen kann. Dazu untersuchen wir die Zahlbeziehungen der Tabelle **Fig. 9**.

Bedingter Sprung: Ein Programm läuft normalerweise Befehl für Befehl nacheinander ab. Oft ist es aber nötig, Programmteile zu überspringen. Soll dies in jedem Fall (ohne Bedingung) geschehen, spricht man von einem unbedingten Sprung. Beim bedingten Sprung ist die Beantwortung der Frage „Sprung ja oder nein“ von einer vorgebbaren Bedingung abhängig.

Register: Das sind einzelne Speicherstellen oder begrenzte Bereiche, die für einen bestimmten Zweck reserviert werden.

Programmadreßtaste: Spezielle Taste, mit deren Hilfe Markierungen (Marken, *Labels*) im Programm gesetzt werden können.

P Pferde	K Kühe p·6	S Schweine (P + K)·4 − 9	Summe P + K + S
1	6	19	26
2	12	47	61
·	·	·	·
·	·	·	·
·	·	·	·
			→ 166

Fig. 5 Probiertabelle

```
INPUT STARTWERT P
SETZE ZIELWERT Z ⇐ 166
BERECHNE K ⇐ P*6 : S ⇐ (P + K) * 4 − 9
WENN P + K + S = Z DANN
ERHOEHE P UM 1 : RECHNE MIT NEUEM P WEITER
→PRINT P K S
```

Fig. 6 Struktogramm

```
10 INPUT „P=“; P
20 Z = 166
30 K = P * 6 : S = (P + K) * 4 − 9
40 IF P + K + S = Z THEN 60
50 P = P + 1 : GO TO 30
60 PRINT „P=“; P, „K=“; K, „S=“; S
```

Fig. 7 BASIC-Programm

```
[Startwert in] [STO] 00
166 [x⇄t]
[2nd] [Lbl] [A] [RCL] 00 [X] 6 [=] [STO] 01 [(] [RCL] 00 [+] [RCL] 01 [)] [X] 4 [−] 9 [=] [STO] 02
[RCL] 00 [+] [RCL] 01 [+] [RCL] 02 [=] [2nd] [x = t] [B]
1 [SUM] 00 [A]
[2nd] [Lbl] [B] [RCL] 00 [R/S] [RCL] 01 [R/S] [RCL] 02 [R/S]
```

Fig. 8 Programm für den TI-58/59

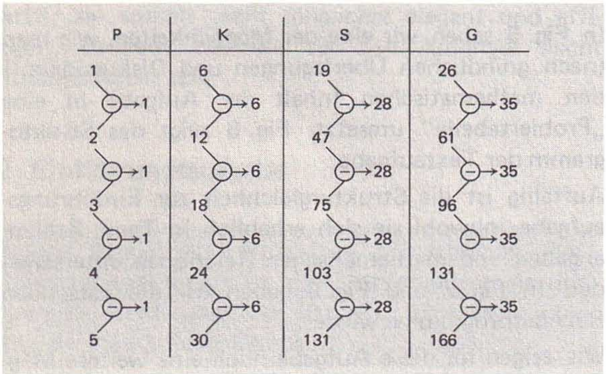

Fig. 9 Ausgefüllte Probiertabelle

Die Startzahl für die Gesamtzahl der Tiere ist 26: unter der Bedingung, daß der Bauer 1 Pferd besitzt, hat er insgesamt 26 Stück Vieh. Bei zwei Pferden steigt die Gesamtzahl um 35, die Zunahme um 1 Pferd entspricht also der Zunahme von 35 Tieren. Der funktionale Zusammenhang ist offensichtlich:

$$G = P \cdot 35 - 9$$

Wir setzen für G 166 ein und lösen die Gleichung nach P auf:

$$P = \frac{166 + 9}{35} \Rightarrow P = 5$$

Für die Schweine gilt: Wenn P um 1 zunimmt, dann steigt S um 28; wobei 19 der Startwert ist.

$$S = P \cdot 28 - 9 \Rightarrow S = 131 \text{ (P = 5 eingesetzt)}$$

Für die Anzahl der Kühe ist der Zusammenhang besonders einfach:

$$K = P \cdot 6 \Rightarrow K = 30 \text{ (P = 5 eingesetzt)}$$

Verzichten wir auf die weitere Auswertung dieser Gleichungen und ihre Umsetzung in Rechnerprogramme und wenden wir uns einer anspruchsvolleren Aufgabe zu.

4 Erweiterte Aufgabe

„Oma bringt ihren drei Enkeln Hannelore, Birgit und Steffen eine Riesentüte Bonbons mit. Hannelore entdeckt sie zuerst und nimmt ein Drittel der Bonbons weg. Birgit sieht die Bonbons später und nimmt sich ebenfalls ein Drittel davon. Zuletzt entdeckt Steffen die Tüte und zählt in dem Glauben, die Geschwister hätte noch nichts, für sich ebenfalls ein Drittel der Süßigkeiten ab. Es bleiben genau 24 Bonbons übrig.

a) Wieviel Bonbons waren in der Tüte?

b) Wieviel Bonbons nahm sich jedes Kind?“

Die Aufgabe wird lösbar, wenn man die Spalten der „Probiertabelle“ unterteilt. Wie bisher untersuchen wir nur Lösungen in der Menge der natürlichen Zahlen, deshalb kommen von vornherein nur durch 3 teilbare Zahlen in Betracht; aus der Tabelle ersieht man dann später, daß die Startzahlen sogar dreimal

Hannelore		Birgit		Steffen		Rest
sieht HS	nimmt HN = HS : 3	sieht BS = HS – HN	nimmt BN = BS : 3	sieht SS = BS – BN	nimmt SN = SS : 3	R = HS – HN – BN – SN
18	6	12	4	8	–	–
21	7	14	–	–	–	–
24	8	16	–	–	–	–
27	9	18	6	12	4	8
30	10	20	–	–	–	–
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Fig. 10 Probiertabelle

```
INPUT STARTWERT HS
SETZE ZIELWERT Z ⇐ 24
→BEREChNE HN ⇐ HS/3 : BS ⇐ HS – HN : BN ⇐ BS/3
          SS ⇐ BS – BN : SN ⇐ SS/3
           R ⇐ HS – HN – BN – SN
WENN R = Z DANN ──────────────────────────┐
ERHOEHE HS UM 3 : RECHNE MIT NEUEM HS WEITER │
PRINT HS HN BN SN ←───────────────────────┘
```

Fig. 11 Struktogramm

```
10  INPUT ,,HS=''; HS
20  Z = 24
30  HN = HS/3 : BS''HS – HN : BN = BS/3
40  SS = BS – BN : SN = SS/3
50  R = HS – HN – BN – SN
60  IF R = Z THEN 80
70  HS = HS + 3 : GO TO 30
80  PRINT ,,HS=''; HS, ,,HN=''; HN, ,,BN=''; BN, ,,SN=''; SN
```

Fig. 12 BASIC-Programm

durch 3 teilbar sein müssen, also zur Vielfachenmenge $\mathbb{V}_{27}$ gehören müssen, wenn der Rest ganzzahlig sein soll. Für den Rechner jedoch brauchen dergleichen Überlegungen nicht angestellt zu werden. **Fig. 10** zeigt die ,,Probiertabelle'' zur Aufgabe.

Im Struktogramm und im BASIC-Programm verwenden wir die folgenden Variablen:

HS: Hannelore sieht
HN: Hannelore nimmt
BS: Birgit sieht
BN: Birgit nimmt
SS: Steffen sieht
SN: Steffen nimmt
R: Rest

Das Struktogramm zu dieser Aufgabe ist umfangreicher geworden (**Fig. 11**), wir haben der besseren Lesbarkeit wegen die Rechenbefehle eingerückt. Um Verwechslungsmöglichkeiten und Verwirrungen auszuschalten, ist es angebracht, durchgehend ,,/'' als Divisionszeichen und den Doppelpunkt ,,:'' als Trennungszeichen zu benutzen. **Fig. 12** stellt das zugehörige BASIC-Programm dar.

In ,,Standard-BASIC'' ist bekanntlich für jede Zuweisung ,,LET'' vorgeschrieben; der Commodore 2001 verlangt diese Anweisung nicht. Bei unseren Programmen lassen wir ,,LET'' absichtlich weg, weil es dafür auf dem TI-58/59 keine Entsprechung gibt.

Das TI-58/59 Programm zu dieser Aufgabe (**Fig. 13**) ist in der S I bereits problematisch; vor der Programmierung empfiehlt es sich sehr, einen *Speicherbelegungsplan* zu erstellen; Verwechslungen können so vielleicht in Grenzen gehalten werden.

Man vergesse bei der Abarbeitung der Programme nicht, daß die Startzahl ein Vielfaches von 3 sein muß. In einer Klasse, wo möglichst viele Schüler mit dem Computer oder dem TR arbeiten sollen, bedeutet es eine Erleichterung, wenn man bei allen Programmen zum Programmanfang zurückverzweigt. Beim ,,PET'' fügt man einfach eine weitere Zeile mit dem **Statement** "GO TO 10" an, beim TI-58/59 braucht man nur zusätzlich die Taste ,,RST'' als letzten Programmbefehl aufzunehmen.

Statement: (Behauptung, Angabe). Eine der Bezeichnungen für ,,Programmbefehl''. Andere sind: Instruktion, Anweisung.

Iterationsmethode: Verbesserung der Lösung einer Gleichung durch wiederholtes Einsetzen aufeinanderfolgender Näherungswerte.

```
[Startwert für HS in] [STO] 00
24 [x⇄t]
[2nd] [Lbl] [A] [RCL] 00 [÷] 3 [=] [STO] 01 [+/-] [+] [RCL] 00 [=] [STO] 02 [÷] 3 [=] [STO] 03
[+/-] [+] [RCL] 02 [=] [STO] 04 [÷] 3 [=] [STO] 05
[RCL] 00 [-] [RCL] 01 [-] [RCL] 03 [-] [RCL] 05 [=] [2nd] [x=t] [B]
3 [SUM] 00 [A]
[2nd] [Lbl] [B] [RCL] 00 [R/S] [RCL] 01 [R/S] [RCL] 03 [R/S] [RCL] 05 [R/S]
```

Fig. 13 Programm für den TI-58/59 zur „erweiterten Aufgabe"

Alte Tagesleistung	Neue Tagesleistung Zu X 8 addieren	Anzahl der Tage (alt)	Anzahl der Tage (neu)	Differenz
X km	Y km	288 : X	288 : Y	T (a) – T (n)
2	10	144	–	–
4	12	72	24	48
8	16	36	18	18
12	20	24	–	–
⋮	⋮	⋮	⋮	⋮
				→ 3 Tage

Fig. 14 Probiertabelle

```
INPUT STARTWERT X KM
SETZE GESAMTSTRECKE GS ⇐ 288 KM : DIFFERENZ D ⇐ 3 TAGE
BERECHNE Y KM ⇐ X KM + 8 KM
WENN GS/X – GS/Y = D DANN
→ERHOEHE X UM 1 : RECHNE MIT NEUEM X WEITER
PRINT X KM Y KM GS/X TAGE GS/Y TAGE←
```

Fig. 15 Struktogramm

```
10 INPUT „X=“; X
20 GS = 288 : D = 3
30 Y = X + 8
40 IF GS/X – GS/Y = D THEN 60
50 X = X + 1 : GO TO 30
60 PRINT X„KM“, Y„KM“, GS/X„TAGE“, GS/Y„TAGE“
```

Fig. 16 BASIC-Programm

```
[Startwert für X in] [STO] 00
288 [STO] 02 3 [x⇄t]
[2nd] [Lbl] [A] [RCL] 00 [+] 8 [=] [STO] 01
[RCL] 02 [÷] [RCL] 00 [-] [RCL] 02 [÷] [RCL] 01 [=] [2nd] [x=t] [B]
1 [SUM] 00 [A]
[2nd] [Lbl] [B] [RCL] 00 [R/S] [RCL] 01 [R/S] [RCL] 02 [÷] [RCL] 00 [=] [R/S]
      [RCL] 02 [÷] [RCL] 01 [=] [R/S]
```

Fig. 17 Programm für den TI-58/59

Speicherbelegungsplan für den TI-58/59 (**Fig. 13**):

Hannelore sieht	(HS)	STO	00
Hannelore nimmt	(HN)	STO	01
Birgit sieht	(BS)	STO	02
Birgit nimmt	(BN)	STO	03
Steffen sieht	(SS)	STO	04
Steffen nimmt	(SN)	STO	05

5 Lösung quadratischer Gleichungen

Die **Iterationsmethode** zur Lösung von Gleichungen führt auch bei Aufgaben, die auf quadratische Gleichungen gründen, zum Erfolg. Der Anwender wird sich dessen allerdings kaum bewußt, weil an keiner Stelle Wurzeln vorkommen. Wie bei allen Aufgaben lassen wir den Rechner nur im Bereich der natürlichen Zahlen nach der Lösung suchen, bei den quadratischen Gleichungen müssen wir uns mit einer Lösung begnügen.

„Ein Forscherteam will für die Bewältigung einer 288 km langen Strecke jeden Tag die gleiche Marschleistung vollbringen. Am ersten Tag können jedoch bequem 8 km mehr als ursprünglich geplant, zurückgelegt werden; das ist auch an allen weiteren Tagen der Fall, so daß das Ziel drei Tage früher erreicht wird."

Wie in allen bisherigen Aufgaben ist die „Probiertabelle" der Schlüssel zur Lösung (**Fig. 14**). Nach unseren Erfahrungen ist dieses Hilfsmittel auch bei der „normalen" Bearbeitung von Textaufgaben dieser Art angebracht, um Schüler zum Erfolg zu führen.

In **Fig. 15** bis **Fig. 17** zeigen wir die aus der „Probiertabelle" abgeleiteten Programme, bzw. das Struktogramm.

Eier mit Normalgröße	Eier von Junghennen	Preis pro Normalei in	Preis pro J'hennenei in	Differenz P (N) – P (J) in
	Zu X 3 add.	DM	DM	DM
X	Y	5,50 : X	5,50 : Y	D
10	13	0,55	0,42	0,13
15	18	0,37	0,31	0,06
16	19	0,34	0,28	0,06
20	23	0,28	0,24	0,04
⋮	⋮	⋮	⋮	⋮
				→ 0,03

Fig. 18 Probiertabelle

```
INPUT STARTWERT X EIER
SETZE GESAMTPREIS GP ⇐ 5,50 DM : DIFFERENZ D ⇐ 0,03 DM
→ BERECHNE Y EIER ⇐ X EIER + 3 EIER
WENN GP/X – GP/Y = D DANN
ERHOEHE X UM 1 : RECHNE MIT NEUEM X WEITER
PRINT X NORMAL-EIER ZU JE GP/X DM ←
PRINT Y J'HENNENEIER ZU JE GP/Y DM
```

Fig. 19 Struktogramm

```
10  INPUT "X="; X
20  GP = 5.5 : D = 0.03
30  Y = X + 3
40  IF GP/X – GP/Y = D THEN 60
50  X = X + 1 : GO TO 30
60  PRINT X "N-EIER ZU JE"; GP/X "DM"
70  PRINT Y "J-EIER ZU JE"; GP/Y "DM"
```

Fig. 20 BASIC-Programm

Unsere letzte Aufgabe entspricht mathematisch der vorigen; neu daran ist, daß hier mit Dezimalzahlen gearbeitet wird. Die laufenden Variablen X und Y für die Anzahl von Eiern sind jedoch nach wie vor natürliche Zahlen. Die „Probiertabelle", das Struktogramm und die Rechnerprogramme entnehme der Leser den **Figuren 18** bis **21**.

Text der Aufgabe:

„Frau M. will für 5,50 DM Eier kaufen. Der Händler bietet ihr Junghenneneier an, die 3 Pfennig pro Stück billiger sind. Frau M. bekommt dadurch 3 Eier mehr als ursprünglich angenommen."

```
[Startwert X in] [STO] 00
5.5 [STO] 02 .03 [x⇄t]
[2nd] [Lbl] [A] [RCL] 00 [+] 3 [=] [STO] 01
[RCL] 02 [÷] [RCL] 00 [–] [RCL] 02 [÷] [RCL] 01 [=] [2nd] [x=t] [B]
1 [SUM] 00 [A]
[2nd] [Lbl] [B] [RCL] 00 [R/S] [RCL] 02 [÷] [RCL] 00 [=] [R/S]
                [RCL] 01 [R/S] [RCL] 02 [÷] [RCL] 01 [=] [R/S]
```

Fig. 21 Programm für den TI-58/59

Wenn der Leser die letzte Aufgabe mit Rechenmaschinen durchführt, wird er eine Überraschung erleben. Der kleine Taschenrechner TI-58 gibt nach wenigen Sekunden die korrekte Lösung 22 Eier zu je 0,25 DM und 25 Eier zu je 0,22 DM aus. Der μComputer hingegen rechnet und rechnet und rechnet ... Er findet die Lösung nicht! Es ist nun Ansichtssache, ob man mit Schülern tiefer in das Problem eindringen sollte, um den offensichtlichen Fehler zu beheben.

Ohne die Hilfe des Lehrers schafft ein Schüler der S I das **Debugging** nicht.

Die Lösung:

Zeile 40 GP/x – GP/y = D ist für den „PET" unerfüllbar. Er berechnet: $5.5/22 - 5.5/25 - .03 = 2{,}54658516 \cdot 10^{-11}$, wenn man D auf die linke Seite der Gleichung bringt; in keinem Fall jedoch das korrekte Ergebnis 0. Eine mögliche Programmänderung, die diesen Mißstand behebt, wäre, Zeile 40 folgendermaßen neu zu schreiben:

```
40 IF GP/X – GP/Y – D <= 1E-6 THEN 60
```

Jetzt gibt der Computer, wenn wir mit einer Startzahl ≤ 22 rechnen, in kürzester Zeit die richtige Lösung aus.

Debugging: Die wörtliche Übersetzung „Entlausung" sagt anschaulich, daß mit diesem Fachausdruck die Fehlersuche und Beseitigung gemeint ist.