

G.O. Hamann J.-J. Eden

Mit **BASIC** ran  
ans **Schachprogramm**



für

**Commodore 64**

(und VC 20 mit 28 K RAM)

**DBV**

DEUTSCHER BETRIEBSWIRTE-VERLAG GMBH

G.O. Hamann · J.-J. Eden

Mit BASIC ran  
ans Schachprogramm  
für Commodore 64

M 296

# Mit BASIC ran ans Schachprogramm für Commodore 64

Prof. Dr. Günter O. Hamann · J.-J. Eden

**DBV** DEUTSCHER BETRIEBSWIRTE-VERLAG GMBH

CIP-Kurztitelaufnahme der Deutschen Bibliothek  
Hamann, Günter O., Eden, J.-J.  
Mit BASIC ran ans Schachprogramm für Commodore 64  
(und VC 20 mit 28 K RAM)  
Günter O. Hamann, J.-J. Eden:  
Deutscher Betriebswirte-Verlag GmbH, 1984. ISBN: 3-88640-019-0

© 1984 by Deutscher Betriebswirte Verlag GmbH, Gernsbach  
Satz: GOHMOSS  
Druck- und Bindearbeit: Clausen & Bosse, Leck  
ISBN: 3-88640-019-0

P

34/12994

R



# VORWORT

Um insbesondere dem Schachspiel-Anfänger jederzeit einen Partner zur Verfügung zu stellen, werden sowohl zahlreiche Schachcomputer als auch Schachprogramme angeboten. Sie gewähren jedoch keinen Einblick in die Ablauflogik, die vor allem für Besitzer eines Heimcomputers von Interesse ist. Diese Lücke soll das vorliegende Buch schließen. Es bietet

- das komplette Listing eines BASIC-Schachprogramms für den erfolgreichen Hobby-Computer Commodore 64 (und VC 20 mit 28 K RAM),
- ausführliche Erläuterungen zum Schachprogramm und zur Schachprogrammierung,
- Hinweise für Änderungsmöglichkeiten und Verbesserungsvorschläge.

Der Reiz dieses Buches liegt also weniger in der Möglichkeit, die Ausgaben für einen "spezialisierten" Schachcomputer oder ein teures Schachprogramm zu sparen. Der persönliche Gewinn ist vielmehr und vor allem darin zu sehen, daß das Schachspiel der Maschine seinen "black-box"-Charakter verliert, d. h. der Spieler durchschaut mit der Übernahme des

---

Programms die Strategie des Computers. Und wenn diese dem Anwender nicht behagt, kann er sie sogar verändern, weil BASIC eine Programmiersprache ist, die von fast allen Hobbyisten der Datenverarbeitung beherrscht wird.

Unser Dank gebührt Herrn Martin Janssen und Herrn Wolfgang Labenski für die freundliche Hilfe.

Im Juli 1984,

Günter O. Hamann  
Jan-Jürgen Eden

---

---

# INHALTSVERZEICHNIS

Seite

|   |            |
|---|------------|
| VORWORT .....   | Vorwort-01 |
| 1. Vor- und Nachteile der<br>Schachprogrammierung<br>in BASIC ..... | 1-01       |
| 2. Erläuterungen zum<br>BASIC-Schachprogramm .....                  | 2-01       |
| 2.1. Darstellung des<br>Schachbretts .....                          | 2-01       |
| 2.2. Abschnittsweise<br>Erläuterungen<br>zum Listing .....          | 2-07       |
| 3. Vollständige Liste des<br>BASIC-Schachprogramms .....            | 3-01       |
| 4. Verbesserungsvorschläge für<br>das BASIC-Schachprogramm .....    | 4-01       |
| 4.1. Blinken der<br>gezogenen Figur .....                           | 4-01       |

---

|  | Seite     |
|--|-----------|
| 4.2. Vorgabe einer Stellung .....                              | 4-03      |
| 4.3. Veränderung der<br>Stellungsbewertung .....               | 4-05      |
| 5. Überprüfung des<br>abgeschriebenen Programms .....          | 5-01      |
| 6. Notwendige Änderungen für<br>den VC 20 (mit 28 K RAM) ..... | 6-01      |
| LITERATURHINWEISE .....  | Anhang-01 |
| STICHWORTVERZEICHNIS .....                                     | Anhang-05 |

---



# 1. Vor- und Nachteile der Schachprogrammierung in BASIC

Die Realisierung eines Schachprogramms in einer Interpreter-Sprache wie BASIC ist in erster Linie deshalb problematisch, weil die Ausführungszeiten verhältnismäßig lang werden.

Diesem Nachteil stehen aber gewichtige Vorteile gegenüber:

- (1) Da die meisten Besitzer eines Heimcomputers die Programmiersprache BASIC beherrschen, können sie die Logik eines in BASIC verfaßten Schachprogramms nachvollziehen und damit verstehen.
  - (2) Das Nachvollziehen und Verstehen der vorgegebenen Logik eröffnet dann die Möglichkeit, das Programm
    - zu verändern,
    - zu verbessern,
    - zu erweitern und
    - zu korrigieren,
-

was im Interpreter-Betrieb der Programmiersprache BASIC besonders einfach ist.

Darüber hinaus läßt sich der anfänglich erwähnte Nachteil der verhältnismäßig langen Ausführungszeiten bei sehr vielen Rechnern dann aufheben, wenn - wie beim Commodore 64 - ein Compiler zur Verfügung steht, mit dessen Hilfe man das mit dem Interpreter erstellte und getestete BASIC-Programm in ein Maschinenprogramm übersetzen läßt, das wesentlich schneller arbeitet.

## 2. Erläuterungen zum BASIC-Schachprogramm

### 2.1. Darstellung des Schachbretts

Das als bekannt vorausgesetzte Schachbrett mit den Figuren in der Ausgangsstellung hat folgendes Aussehen:

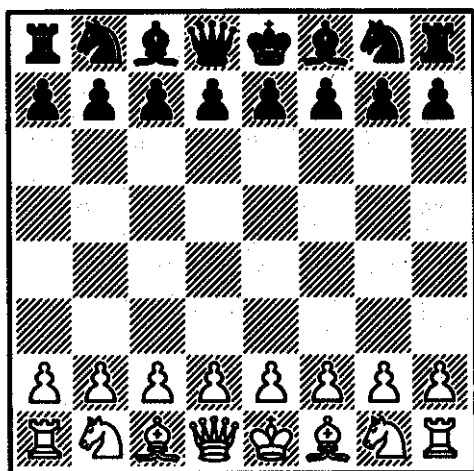


Abb. 1: Schachbrett mit Figuren in der Grundstellung

Das Schachbrett ist demnach so zu legen, daß das linke Eckfeld schwarz ist. Entsprechend einer internationalen Norm ist jedes Feld durch eine Buchstabe/Ziffer-Kombination bestimmt; beispielsweise hat das Feld unten links mit dem weißen Turm die Bezeichnung a1, das Feld mit dem weißen König die Bezeichnung e1, das Feld mit dem schwarzen König die Bezeichnung e8 und das Feld oben rechts mit dem schwarzen Turm die Bezeichnung h8.

Zusammenfassend ergeben sich also die folgenden Benennungen der einzelnen Schachbrettfelder:

|   |    |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|----|
| 8 | a8 | b8 | c8 | d8 | e8 | f8 | g8 | h8 |
| 7 | a7 | b7 | c7 | d7 | e7 | f7 | g7 | h7 |
| 6 | a6 | b6 | c6 | d6 | e6 | f6 | g6 | h6 |
| 5 | a5 | b5 | c5 | d5 | e5 | f5 | g5 | h5 |
| 4 | a4 | b4 | c4 | d4 | e4 | f4 | g4 | h4 |
| 3 | a3 | b3 | c3 | d3 | e3 | f3 | g3 | h3 |
| 2 | a2 | b2 | c2 | d2 | e2 | f2 | g2 | h2 |
| 1 | a1 | b1 | c1 | d1 | e1 | f1 | g1 | h1 |
|   | a  | b  | c  | d  | e  | f  | g  | h  |

Abb. 2: Schachbrett mit den üblichen Feldbenennungen

Die erläuterte Darstellung eines Schachbretts ließe sich in gleicher Weise auch in einem Computer-Programm verwenden. Es hat sich allerdings als zweckmäßig erwiesen, die folgende Form zu verwenden, bei der die Buchstaben durch Ziffern ersetzt werden:

|   |    |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|----|
| 8 | 18 | 28 | 38 | 48 | 58 | 68 | 78 | 88 |
| 7 | 17 | 27 | 37 | 47 | 57 | 67 | 77 | 87 |
| 6 | 16 | 26 | 36 | 46 | 56 | 66 | 76 | 86 |
| 5 | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 |
| 4 | 14 | 24 | 34 | 44 | 54 | 64 | 74 | 84 |
| 3 | 13 | 23 | 33 | 43 | 53 | 63 | 73 | 83 |
| 2 | 12 | 22 | 32 | 42 | 52 | 62 | 72 | 82 |
| 1 | 11 | 21 | 31 | 41 | 51 | 61 | 71 | 81 |
|   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |

**Abb. 3:** Schachbrett mit den computer-internen Feldbenennungen

In Anlehnung an unsere Beispiele auf Seite 2-02 hat dann das Feld unten links mit dem weißen Turm die

Bezeichnung 11, das Feld mit dem weißen König die Bezeichnung 51, das Feld mit dem schwarzen König die Bezeichnung 58 und das Feld mit dem schwarzen Turm oben rechts die Bezeichnung 88. (Für die Zugberechnung werden wir das soeben erläuterte Feld allerdings noch erweitern; vgl. hierzu Seite 2-13, Abb. 6: Erweitertes computer-internes Schachbrett!)

Abweichend von einer eingängigeren Abkürzung der Figuren, bei der beispielsweise d für Dame, k für König, t für Turm verwendet wird, stellt man in einem Programm auch die Figuren mittels Ziffern dar, nämlich

- |   |     |                            |
|---|-----|----------------------------|
| 1 | für | Bauer (in Grundposition),  |
| 2 | für | Bauer (bereits gezogen),   |
| 3 | für | Springer,                  |
| 4 | für | Läufer,                    |
| 5 | für | Turm (noch rochadefähig),  |
| 6 | für | Turm (bereits gezogen),    |
| 7 | für | Dame,                      |
| 8 | für | König (noch rochadefähig), |
| 9 | für | König (bereits gezogen).   |

In dem von uns realisierten Programm unterscheiden

---

wir die Figuren der beiden Parteien (weiße Figuren einerseits und schwarze andererseits) lediglich durch ein Vorzeichen, und zwar sind jene Ziffern, welche die weißen Figuren des Spielers repräsentieren, negativ; jene des Rechners (schwarz) hingegen ohne Vorzeichen.

|   |    |    |    |    |    |    |    |    |
|---|----|----|----|----|----|----|----|----|
| 8 | 5  | 3  | 4  | 7  | 8  | 4  | 3  | 5  |
| 7 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| 6 |    |    |    |    |    |    |    |    |
| 5 |    |    |    |    |    |    |    |    |
| 4 |    |    |    |    |    |    |    |    |
| 3 |    |    |    |    |    |    |    |    |
| 2 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 |
| 1 | -5 | -3 | -4 | -7 | -8 | -4 | -3 | -5 |
|   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |

Abb. 4: Schachbrett mit der computer-internen Grundstellung

Abweichend von der erläuterten internen Darstellung erfolgt die Wiedergabe auf dem Bildschirm mittels der bekannten Abkürzungen, wobei die Figuren des Rechners als Kleinbuchstaben (z. B. d für Dame, k für König, t für Turm) und jene des Spielers als Großbuchstaben (z. B. D für Dame, K für König, T für Turm) abgebildet werden.

|   | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 8 | t | s | l | d | k | l | s | t |
| 7 | b | b | b | b | b | b | b | b |
| 6 |   |   |   |   |   |   |   |   |
| 5 |   |   |   |   |   |   |   |   |
| 4 |   |   |   |   |   |   |   |   |
| 3 |   |   |   |   |   |   |   |   |
| 2 | B | B | B | B | B | B | B | B |
| 1 | T | S | L | D | K | L | S | T |

Ihr Zug: ♚- : --

Abb. 5: Schachbrett auf dem Bildschirm mit "Figuren" in der Grundstellung



## 2.2. Abschnittsweise Erläuterungen zum Listing

Bei unseren Ausführungen zum Listing orientieren wir uns an der Reihenfolge der Befehle. Diese Vorgehensweise bietet sich an, weil

- das Programm ohnehin systematisch aufgebaut ist und weil
- so das Nachvollziehen der angestellten Überlegungen für den Leser erleichtert wird.

### Programmausschnitt (1):

```
10 REM *****
20 REM *
30 REM * SCHACHPROGRAMM IN BASIC *
40 REM *
50 REM * FUER DEN RECHNER *
60 REM *
70 REM * COMMODORE 64 *
80 REM *
90 REM *****
100 REM
```

---

**Programmausschnitt (1) / Forts.:**

```
110 REM *****
120 REM *
130 REM *  BEDIENUNGSHINWEISE:
140 REM *  =====
150 REM *  DIES IST EIN BASIC-SCHACH-
160 REM *  PROGRAMM. ES WIRD NACH DEN
170 REM *  ALLGEMEINEN SCHACHREGELN
180 REM *  GESPIELT. WENN SIE Z.B. VON
190 REM *  C3 NACH E5 ZIEHEN WOLLEN,
200 REM *  DANN GEBEN SIE EINFACH
210 REM *  C3 : E5 +RETURN EIN. FALLS
220 REM *  SIE SICH VERTIPPT HABEN,
230 REM *  DRUECKEN SIE DIE DEL-TASTE.
240 REM *  DANN KOENNEN SIE DEN ZUG NEU*
250 REM *  EINGEBEN. FUER DIE KLEINE
260 REM *  ROCHADE MUSS NUR DIE O-TASTE*
270 REM *  +RETURN BETAETIGT, UND FUER
280 REM *  DIE GROSSE ROCHADE MUSS DIE
290 REM *  O-TASTE ZWEIMAL (+RETURN)
300 REM *  BETAETIGT WERDEN.
310 REM *
320 REM *****
330 REM
```

## **Erläuterungen zum nebenstehenden Programmausschnitt (1): (vgl. auch Seite 2-07!)**

Die REM-Befehle der Zeilen 10 - 330 dienen der übersichtlichen Gestaltung des Programms und enthalten Bedienungshinweise, so daß der Anwender diese im Bedarfsfalle nach einem LIST-Kommando nachlesen kann.

---

**Programmausschnitt (2):**

```
340 REM *****
350 REM ***** INITIALISIERUNG *****
360 REM *****
370 DIM FW(10) :DIM F(99) :DIM F$(18)
380 DIM BS(132):DIM SK(132):DIM ZK(132)
390 DIM FA(8)
400 REM
```

## Erläuterungen zum nebenstehenden Programmausschnitt (2):

Im Abschnitt "INITIALISIERUNG" schaffen wir verschiedene Speicherplätze, und zwar reservieren wir mit

- DIM FW(10)

Speicherplätze für die Figurenwerte.

(Bekanntlich haben die einzelnen Schachfiguren einen unterschiedlichen strategischen Wert für das Spiel. Der König ist in jedem Fall die wichtigste Figur, der Bauer hingegen hat die geringste Bedeutung). Mit

- DIM F(99)

erzeugen wir die einzelnen Felder des Schachbretts.

(Abweichend von unserer Darstellung auf Seite 2-03 reservieren wir nicht nur die Plätze für die eigentlichen Schachbrettfelder, sondern darüber hinaus 36 zusätzliche Felder, die das Schachbrett "einrahmen". Wir werden diese Felder benötigen, um sie mit sog. Scheinfiguren zu besetzen. Sie werden verhindern, daß der Computer

---

**Programmausschnitt (2):**

```
340 REM *****
350 REM ***** INITIALISIERUNG *****
360 REM *****
370 DIM FW(10) :DIM F(99) :DIM F$(18)
380 DIM BS(132):DIM SK(132):DIM ZK(132)
390 DIM FA(8)
400 REM
```

## Erläuterungen zum nebenstehenden Programmausschnitt (2) / Forts.:

bei der Berechnung seines jeweiligen Zuges mit der  
Figur das Schachbrett "verläßt".)

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 09 | 19 | 29 | 39 | 49 | 59 | 69 | 79 | 89 | 99 |
| 08 | 18 | 28 | 38 | 48 | 58 | 68 | 78 | 88 | 98 |
| 07 | 17 | 27 | 37 | 47 | 57 | 67 | 77 | 87 | 97 |
| 06 | 16 | 26 | 36 | 46 | 56 | 66 | 76 | 86 | 96 |
| 05 | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 | 95 |
| 04 | 14 | 24 | 34 | 44 | 54 | 64 | 74 | 84 | 94 |
| 03 | 13 | 23 | 33 | 43 | 53 | 63 | 73 | 83 | 93 |
| 02 | 12 | 22 | 32 | 42 | 52 | 62 | 72 | 82 | 92 |
| 01 | 11 | 21 | 31 | 41 | 51 | 61 | 71 | 81 | 91 |
| 00 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |

Abb. 6: Erweitertes computer-internes Schachbrett

**Programmausschnitt (2):**

```
340 REM *****  
350 REM ***** INITIALISIERUNG *****  
360 REM *****  
370 DIM FW(10) :DIM F(99) :DIM F$(18)  
380 DIM BS(132):DIM SK(132):DIM ZK(132)  
390 DIM FA(8)  
400 REM
```



## Erläuterungen zum nebenstehenden Programmausschnitt (2) / Forts.:

Mit

- DIM F\$(18)

erzeugen wir Speicherplätze, in denen die Druckzeichen der Figuren für den Bildschirm abzulegen sein werden. Die durch

- DIM BS(132) : DIM SK(132) : DIM ZK(132)

reservierten Variablen sollen dazu dienen, Bewertungen all jener Stellungen aufzunehmen, die sich aus den "gedachten" Zügen ergeben und deren

Startkoordinaten und

Zielkoordinaten zu speichern. Die Anweisung

- DIM FA(8)

schafft Plätze, welche Feldabstände aufnehmen werden. Wir verwenden diese, um die Zielkoordinaten einer Figur aus den Startkoordinaten zu berechnen.

---

**Programmausschnitt (3):**

```
410 REM ***** FARBZUWEISUNGEN *****
420 RF=0 : HF=0 : REM ** SCHWARZ
430 ZF$=CHR$(5) : REM ** WEISS
440 FH$=CHR$(154) : REM ** HELLBLAU
450 FD$=CHR$(31) : REM ** DUNKELBLAU
460 RO$=CHR$(18) : REM ** REVERSE ON
470 RF$=CHR$(146) : REM ** REVERSE OFF
480 REM
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (3):

In den Zeilen 420 - 450 weisen wir den Variablen

- RF (Rahmenfarbe),
- HF (Hintergrundfarbe),
- ZF\$ (Zeichenfarbe),
- FH\$ (Farbe für helle Felder),
- FD\$ (Farbe für dunkle Felder)

Farbwerte zu und erreichen damit, daß der Anwender sehr einfach eine andere Gestaltung des Bildschirmbildes herbeiführen kann.

Die Attribute für reversen (= inversen) Druck (Reverse On) und für Umschaltung auf Normalschrift (Reverse Off) werden in RO\$ und RF\$ abgelegt.

---

**Programmausschnitt (4):**

```
490 REM ***** FIGURENWERTE *****  
500 FOR A=0 TO 9 : READ FW(A) : NEXT  
510 DATA 3,10,10,30,30,50,50,110,999,999  
520 REM
```

## Erläuterungen zum nebenstehenden Programmausschnitt (4):

Den in Zeile 370 reservierten Plätzen für die Figuren-werte werden nun durch die Befehle der Zeilen 500 und 510 die folgenden "Gewichte" zugeordnet:

| <u>Variable</u> | <u>Wert</u> | <u>Feldbelegung</u>                     |
|-----------------|-------------|---|
| FW(0)           | 3           | 0 ( $\hat{=}$ leeres Feld)              |
| FW(1)           | 10          | 1 ( $\hat{=}$ Bauer in Grundposition)   |
| FW(2)           | 10          | 2 ( $\hat{=}$ Bauer, bereits gezogen)   |
| FW(3)           | 30          | 3 ( $\hat{=}$ Springer)                 |
| FW(4)           | 30          | 4 ( $\hat{=}$ Läufer)                   |
| FW(5)           | 50          | 5 ( $\hat{=}$ Turm, noch rochadefähig)  |
| FW(6)           | 50          | 6 ( $\hat{=}$ Turm, bereits gezogen)    |
| FW(7)           | 110         | 7 ( $\hat{=}$ Dame)                     |
| FW(8)           | 999         | 8 ( $\hat{=}$ König, noch rochadefähig) |
| FW(9)           | 999         | 9 ( $\hat{=}$ König, bereits gezogen)   |

---

**Programmausschnitt (5):**

```
530 REM ***** SCHEINFIGUREN *****
540 FOR A= 1 TO 9:F(A)=10:NEXT
550 FOR A=90 TO 98:F(A)=10:NEXT
560 FOR A= 0 TO 80 STEP 10:F(A)=10:NEXT
570 FOR A=19 TO 99 STEP 10:F(A)=10:NEXT
580 REM
590 REM ***** ECHTE FIGUREN *****
600 FOR A=18 TO 88 STEP 10
610 READ F(A) : NEXT A
620 FOR A=17 TO 87 STEP 10:F(A)= 1:NEXT
630 FOR A=12 TO 82 STEP 10:F(A)=-1:NEXT
640 FOR A=11 TO 81 STEP 10
650 READ F(A) : NEXT A
660 DATA 5, 3, 4, 7, 8, 4, 3, 5
670 DATA -5,-3,-4,-7,-8,-4,-3,-5
680 REM
```

## Erläuterungen zum nebenstehenden Programmausschnitt (5):


Zunächst weisen wir den Feldern des Schachbrett-rahmens (vgl. hierzu die Darstellung auf Seite 2-13!) sog. Scheinfiguren ( $\hat{=}$  10) zu, die weder gezogen noch geschlagen werden dürfen. Mit ihnen wollen wir später verhindern, daß der Computer bei der Berechnung seines jeweiligen Zuges das Schachbrett "verläßt" (vgl. Seite 2-11 ff.!).

Daran anschließend "stellen wir die Figuren", die durch die auf Seite 2-04 erwähnten Ziffern repräsentiert werden, in ihre Grundposition, wie wir sie bereits in der Abbildung 4 auf Seite 2-05 dargestellt haben. Es ist nicht notwendig, die unbesetzten Felder mit 0 zu belegen, da dies in BASIC der Ausgangswert aller numerischen Variablen ist.

---

## Programmausschnitt (6):


```
690 REM ***** DRUCKZEICHEN FIGUR *****
700 FOR A=0 TO 18
710 READ DF : F$(A)=CHR$(DF)
720 NEXT A
730 DATA 203, 203, 196, 212, 212 }
740 DATA 204, 211, 194, 194, 32 } -128 + 32
750 DATA      66,  66,  83,  76
760 DATA  84,  84,  68,  75,  75
770 REM
```





## Erläuterungen zum nebenstehenden Programmausschnitt (6):

Beim Commodore 64 ist jedes mögliche Binärmuster eines Bytes (Zeichenwert) doppelt belegt. Die nach F\$(0) ... F\$(18) zugewiesenen Werte entsprechen den Zeichen

K, K, D, T, T, L, S, B, B,  (Leerstelle)

b, b, s, l, t, t, d, k, k,

wenn im Kleinschriftmodus gearbeitet wird, den wir mit dem Befehl in Zeile 900 einschalten werden.

Eine Beziehung zwischen den intern als Ziffern dargestellten Figuren, nämlich


-9, -8, -7, -6, -5, -4, -3, -2, -1, 0 ( $\hat{=}$  leeres Feld),  
1, 2, 3, 4, 5, 6, 7, 8, 9,

einerseits und den obigen Druckzeichen andererseits läßt sich dadurch herstellen, daß man zu der als Ziffer repräsentierten Figur den Wert 9 addiert, um so den gewünschten Index für F\$(..) zu erhalten. Diese etwas umständliche Vorgehensweise ist deshalb erforderlich, weil negative Indizes nicht zulässig sind.

---

**Programmausschnitt (7):**

```
780 REM ***** FELDABSTAENDE *****  
790 FOR A=1 TO 8 : READ FA(A) : NEXT  
800 DATA 21, 19, 12, 8, 11, 9, 10, 1  
810 REM
```



## Erläuterungen zum nebenstehenden Programmausschnitt (7):

Den indizierten Variablen FA(1) ... FA(8) weisen wir an dieser Stelle Feldabstandswerte zu, die wir an einer späteren Stelle des Programms zur Ermittlung eines figurgemäßen Zuges benötigen. Die ersten vier Zahlen dienen der Berechnung der möglichen Springerzüge, die fünfte und sechste Zahl für jene des Läufers und die beiden letzten für jene des Turms. Darüber hinaus werden wir die vier letzten Zahlen ( $\hat{=}$  Läufer + Turm) zu Ermittlung der Königs- und Damenzüge verwenden. Für den Bauern erübrigt sich die Festlegung solcher Werte, da die Variationsbreite der Zugmöglichkeiten zu gering ist.


### Beispiel:

Wir unterstellen, der Springer des Rechners befände sich auf dem Feld 65 (vgl. Abb. 6 auf Seite 2-13!), was der Position f5 entspricht. Mit Hilfe der ersten vier Zahlen (21, 19, 12, 8) lassen sich nunmehr mögliche Springerzüge berechnen. Anders ausgedrückt: Ausgehend von der Startkoordinate 65 lassen sich denkbare Zielkoordinaten folgendermaßen ermitteln:

---

**Programmausschnitt (7):**

```
780 REM ***** FELDABSTAENDE *****  
790 FOR A=1 TO 8 : READ FA(A) : NEXT  
800 DATA 21, 19, 12, 8, 11, 9, 10, 1  
810 REM
```




## Erläuterungen zum nebenstehenden Programmausschnitt (7) / Forts.:

| <u>Startkoordinate</u> |   | <u>Feldabstand</u> |   | <u>Zielkoordinate</u> |
|------------------------|---|--------------------|---|-----------------------|
| 65                     | + | 21                 | ➔ | 86                    |
| 65                     | + | 19                 | ➔ | 84                    |
| 65                     | + | 12                 | ➔ | 77                    |
| 65                     | + | 8                  | ➔ | 73                    |
| 65                     | - | 21                 | ➔ | 44                    |
| 65                     | - | 19                 | ➔ | 46                    |
| 65                     | - | 12                 | ➔ | 53                    |
| 65                     | - | 8                  | ➔ | 57                    |

---

**Programmausschnitt (8):**

```
820 REM ***** GROSSBUCHSTABE *****
830 B$=CHR$(194) : I$=CHR$(201)
840 O$=CHR$(207) : P$=CHR$(208)
850 S$=CHR$(211) : Z$=CHR$(218)
860 REM
```



## **Erläuterungen zum nebenstehenden Programmausschnitt (8):**

Neben dem Schachbrett mit den "Figuren" wollen wir zusätzlich kurze erläuternde Texte (Beispiele: "Bitte warten", "Stellungsanalyse", "Ihr Zug", etc.) ausgeben, für die wir auch Großbuchstaben benötigen. Da das Programm im Kleinschriftmodus arbeitet, ist für die Wiedergabe der benötigten Großbuchstaben der "Umweg" über CHR\$(..) einzuschlagen.

---

## Programmausschnitt (9):

```

870 REM ***** SCHACHBRETT *****
880 POKE 53280, RF : POKE 53281, HF
890 PRINT CHR$(147);
900 PRINT CHR$(14) : REM KLEINSCHRIFT
910 FOR A=1 TO 4
920 PRINT TAB(8) " ";
930 FOR B=1 TO 2 : FOR C=1 TO 4
940 PRINT RO$; FH$; " "; FD$; " ";
950 NEXT C
960 PRINT:PRINT TAB(8) " ";
970 NEXT B
980 FOR B=1 TO 2 : FOR C=1 TO 4
990 PRINT RO$; FD$; " "; FH$; " ";
1000 NEXT C
1010 PRINT:PRINT TAB(8) " ";
1020 NEXT B,A
1030 PRINT CHR$(19);
1040 PRINT TAB(9) " "; ZF$;
1050 FOR A=193 TO 200
1060 PRINT CHR$(A); " ";
1070 NEXT A
1080 PRINT
1090 FOR A=8 TO 1 STEP -1
1100 PRINT:PRINT TAB(6) A : NEXT A
1110 REM *****
1120 REM *** INITIALISIERUNG/ENDE ***

```

COLOR  
CLS

WIDTH 40

LOCATE 1,1

(97 - 104)



## Erläuterungen zum nebenstehenden Programmausschnitt (9):

Mit den Befehlen in Zeile 880 legen wir die Rahmen- und Hintergrundfarbe fest. Da in Zeile 420 den Variablen RF und HF der Wert 0 (Null) zugewiesen wurde, erscheint der Rahmen und der Hintergrund schwarz.

Die folgende Anweisung (Zeile 890) löscht den Bildschirm und die Instruktion in Zeile 900 schaltet in den Kleinschriftmodus um.

Die Befehle von Zeile 910 - 1100 erzeugen das leere Schachbrett einschließlich der Randbeschriftung (Zeile 1030 - 1100), die der Felderbenennung dient.

Das Schachbrettmuster wird herbeigeführt durch Ausgabe von hellblauen/reversen und dunkelblauen/reversen Leerstellen.

---

## Programmausschnitt (10):

```

1130 REM *****
1140 REM ***** SPIELTEIL *****
1150 REM *****
1160 REM
1170 REM *****
1180 GOSUB 1470:REM ** STEINE SETZEN **
1190 GOSUB 1680:REM ** SPIELEREINGABE**
1200 GOSUB 3790:REM ** ZUGANGABE **
1210 GOSUB 1470:REM ** STEINE SETZEN **
1220 GOSUB 3960:REM ** COMPUTERZUG **
1230 IF MA=1 OR PA=1 THEN 1270 : REM **
1240 GOSUB 3790:REM ** ZUGANGABE **
1250 REM *****
1260 GOTO 1180
1270 GOSUB 7880
1280 IF PA <> 1 THEN 1340
1290 PRINT
1300 PRINT TAB(17) P$;
1310 PRINT "ATT !"
1320 PRINT TAB(17) "====="
1330 GOTO 1400
1340 PRINT TAB(14) S$;
1350 PRINT "CHACH MATT!"
1360 PRINT
1370 PRINT TAB(11) S$; "IE HABEN";
1380 PRINT " GEWONNEN"
1390 PRINT TAB(11)"===== ";
1400 END
1410 REM *****
1420 REM ***** SPIELTEIL/ENDE *****

```

## **Erläuterungen zum nebenstehenden Programmausschnitt (10):**

Die Zeilen 1130 - 1420 umfassen den Spielteil, in welchem die eigentliche Steuerung des Spieles von den Befehlen der Zeilen 1180 - 1260 ausgeht, und zwar dergestalt, daß für die verschiedenen Spielfunktionen Unterprogramme aufgerufen werden.

Wenn eine Patt- oder Mattsituation für den Computer eintritt, erfolgt die entsprechende Meldung durch die Anweisungen der Zeilen 1280 - 1390, wobei die Löschung des unteren Bildschirmteils durch das Unterprogramm ab Zeile 7880 erfolgt, das in Zeile 1270 aufgerufen wird.

Eine Patt- oder Mattstellung des Spielers erkennt man daran, daß der Rechner keine Zugeingabe mehr akzeptiert.

---

## Programmausschnitt (11):

```
1430 REM *****
1440 REM ***** UNTERPROGRAMME *****
1450 REM *****
1460 REM
1470 REM ***** STEINE SETZEN *****
1480 PRINT CHR$(19)
1490 FOR A=0 TO 6 STEP 2
1500 PRINT:PRINT TAB(8) " ";
1510 FOR B=18-A TO 88-A STEP 20
1520 PRINT RO$; FH$; " ";
1530 PRINT RO$; FH$; F$(F(B)+9); " ";
1540 PRINT RO$; FD$; " ";
1550 PRINT RO$; FD$; F$(F(B+10)+9); " ";
1560 NEXT B
1570 PRINT:PRINT:PRINT TAB(8) " ";
1580 FOR B=18-A TO 88-A STEP 20
1590 PRINT RO$; FD$; " ";
1600 PRINT RO$; FD$; F$(F(B-1)+9); " ";
1610 PRINT RO$; FH$; " ";
1620 PRINT RO$; FH$; F$(F(B+9)+9); " ";
1630 NEXT B
1640 PRINT
1650 NEXT A
1660 RETURN
```

## Erläuterungen zum nebenstehenden Programmausschnitt (11):

In das Schachbrett, das bereits mit den Befehlen der Zeilen 870 - 1120 erzeugt wurde, setzen wir nunmehr die einzelnen "Figuren" in Form der Druckzeichen (t, s, l, d, k, ... , T, S, L, D, K, ...), die wir in F\$(0) ... F\$(18) abgelegt hatten (vgl. Seite 2-22 f.). Die Ausgabe der die Figuren repräsentierenden Druckzeichen erfolgt im reversen Druck ( $\hat{=}$  RO\$; vgl. hierzu Seite 2-16 f.). Dadurch bleibt die Farbe des Schachbrettfeldes erhalten und die Zeichen erscheinen in Schwarz.

---

**Programmausschnitt (12):**

```
1670 REM *****
1680 REM ***** SPIELEREINGABE *****
1690 REM *****
1700 REM
1710 REM ***** ZUGEINGABE *****
1720 GOSUB 7880
1730 PRINT LOCATE
1740 PRINT TAB(12) I$;"HR "; Z$;"UG: ";
1750 PRINT RO$; " "; RF$;
1760 PRINT "- : -- ";
1770 FOR A=1 TO 8:PRINT CHR$(157);:NEXT
1780 GET IN$ : IF IN$ = "" THEN 1780
1790 IF IN$ = CHR$(20) THEN 1890
1800 IF IN$="O" AND F(51)=-8 THEN 2130
1810 IF IN$<"A" OR IN$>"H" THEN 1780
1820 PRINT CHR$(ASC(IN$)+128);
1830 PRINT RO$; " "; RF$; CHR$(157);
1840 GET IP$ : IF IP$ = "" THEN 1840
1850 IF IP$ = CHR$(20) THEN 1890
1860 IF IP$<"1" OR IP$>"8" THEN 1840
1870 SK = 10*(ASC(IN$)-64)+VAL(IP$)
1880 IF F(SK) < 0 THEN 1900
1890 PRINT CHR$(145) : GOTO 1740
1900 PRINT IP$; " : ";
```

## Erläuterungen zum nebenstehenden Programmausschnitt (12):

Durch Aufruf des ab Zeile 7880 beginnenden Unterprogramms löschen wir zunächst den unteren Teil des Bildschirms, um dann durch die Befehlszeilen 1740 - 1760 den Text

Ihr Zug: ♚- : --

auszugeben. Damit das vom Anwender eingegebene Zeichen auf der Stelle des Scheincursors (♚) gedruckt wird, setzen wir mit der Anweisung der Zeile 1770 die Druckposition auf den Scheincursor zurück. Die Eingabe des ersten Zeichens erfolgt in Zeile 1780. Daran anschließend untersucht das System, ob **DEL** gedrückt wurde (Zeile 1790), um ggf. mit diesem Programmabschnitt erneut zu beginnen. Zu diesem Zweck (wenn also **DEL** tatsächlich betätigt wurde) erfolgt ein Sprung nach Zeile 1890. Hier wird der Cursor an den Zeilenanfang zurückgesetzt (genauer: zunächst 1 Zeile nach oben und durch Zeilenvorschub an den Beginn der aktuellen Schreibzeile), um schließlich die Bearbeitung des Programms in Zeile 1740 wieder aufzunehmen.

Die Instruktion der Zeile 1800 untersucht, ob der Spieler eine Rochade wünscht. Sie wird mit dem

**Programmausschnitt (12):**

```
1670 REM *****
1680 REM ***** SPIELEREINGABE *****
1690 REM *****
1700 REM
1710 REM ***** ZUGEINGABE *****
1720 GOSUB 7880
1730 PRINT
1740 PRINT TAB(12) I$;"HR "; Z$;"UG: ";
1750 PRINT RO$; " "; RF$;
1760 PRINT "- : -- ";
1770 FOR A=1 TO 8:PRINT CHR$(157);:NEXT
1780 GET IN$ : IF IN$ = "" THEN 1780
1790 IF IN$ = CHR$(20) THEN 1890
1800 IF IN$="O" AND F(51)=-8 THEN 2130
1810 IF IN$<"A" OR IN$>"H" THEN 1780
1820 PRINT CHR$(ASC(IN$)+128);
1830 PRINT RO$; " "; RF$; CHR$(157);
1840 GET IP$ : IF IP$ = "" THEN 1840
1850 IF IP$ = CHR$(20) THEN 1890
1860 IF IP$<"1" OR IP$>"8" THEN 1840
1870 SK = 10*(ASC(IN$)-64)+VAL(IP$)
1880 IF F(SK) < 0 THEN 1900
1890 PRINT CHR$(145) : GOTO 1740
1900 PRINT IP$; " : ";
```

-32

29



## Erläuterungen zum nebenstehenden Programmausschnitt (12) / Forts.:

Buchstaben "O" in IN\$ angefordert und ist nur zulässig, wenn der König noch die Bezeichnung -8 (nicht -9!) hat. Falls rochiert werden soll und in F(51) der Wert -8 steht, setzt der Programmlauf in Zeile 2130 fort. Andernfalls erfolgt die Überprüfung (Zeile 1810), ob das eingegebene Zeichen eine Spalte des Schachbretts (A - H) spezifiziert. Eine unzulässige Angabe hat einen Rücksprung nach Zeile 1780 zur Konsequenz. Mit dem folgenden, etwas aufwendig wirkenden Befehl erreichen wir, daß der eingegebene Buchstabe in Großschrift ausgegeben wird, obwohl sich das Programm im Kleinschriftmodus befindet.

Die beschriebene Eingabeform ermöglicht die kontrollierte Abfrage von Zeichen und wird deshalb in analoger Weise für die Zeilenwerte verwendet (Zeile 1830 - 1860).

Die Anweisung in Zeile 1870 berechnet aus den Startkoordinaten der üblichen Schachnotation (Buchstabe/Ziffer) den Startkoordinatenwert (Index) für das computer-interne Feld F(Index). Nur wenn das ermittelte Feld F(Index) mit einer Figur des Spielers belegt ist, akzeptiert das System die angegebene Koordinate (Zeile 1880 - 1900). Nach dem Ausdruck des Doppelpunkts (Zeile 1900)

---

**Programmausschnitt (13):**

```
1910 PRINT RO$; " "; RF$; CHR$(157);
1920 GET IN$ : IF IN$ = "" THEN 1920
1930 IF IN$ = CHR$(20) THEN 1890
1940 IF IN$ <"A" OR IN$ >"H" THEN 1920
1950 PRINT CHR$(ASC(IN$)+128);
1960 PRINT RO$; " "; RF$; CHR$(157);
1970 GET IP$ : IF IP$ = "" THEN 1970
1980 IF IP$ = CHR$(20) THEN 1890
1990 IF IP$ <"1" OR IP$ >"8" THEN 1970
2000 PRINT IP$;
2010 PRINT RO$; " "; RF$; CHR$(157);
2020 GET E$ : IF E$ = "" THEN 2020
2030 IF E$ = CHR$(20) THEN 1890
2040 IF E$ <> CHR$(13) THEN 2020
2050 ZK = 10*(ASC(IN$)-64)+VAL(IP$)
2060 PRINT " ";
2070 IF F(ZK) >= 0 THEN 2690
2080 FOR A=1 TO 3:PRINT CHR$(157);:NEXT
2090 PRINT RO$; " "; RF$; "-";
2100 PRINT CHR$(157); CHR$(157);
2110 GOTO 1920
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (13):

erfolgt - in bereits erläuteter Weise - die Zielkoordinateneingabe und deren Überprüfung (Zeile 1910 - 2000).

Abweichend von der Prozedur bei der Eingabe der Startkoordinaten wird nach der Eingabe der Zielkoordinaten überprüft, ob `DEL` oder `RETURN` betätigt wurde, um ggf. zurückzuzweigen (erneute Eingabemöglichkeit) oder vor der Fortsetzung des Programmlaufs

- aus den Zielkoordinaten der üblichen Schachnotation (Buchstabe/Ziffer) den Zielkoordinatenwert (Index) für das computer-interne Feld  $F(\text{Index})$  zu berechnen (Zeile 2050) und
- zu überprüfen, ob der Inhalt des Feldes  $\geq 0$  (0 bedeutet leeres Feld,  $>0$  bedeutet Figur des Rechners) ist.

Falls der Inhalt des Zielkoordinatenfeldes  $< 0$  ist, werden die eingegebenen Zielkoordinaten gelöscht (Zeile 2080 - 2100), um dann mit der Anweisung in Zeile 2110 nach Zeile 1920 zurückzuspringen. Dort ist die Eingabe der Zielkoordinaten zu wiederholen.

---

**Programmausschnitt (13):**

```
1910 PRINT RO$; " "; RF$; CHR$(157);
1920 GET IN$ : IF IN$ = "" THEN 1920
1930 IF IN$ = CHR$(20) THEN 1890
1940 IF IN$ <"A" OR IN$ >"H" THEN 1920
1950 PRINT CHR$(ASC(IN$)+128);
1960 PRINT RO$; " "; RF$; CHR$(157);
1970 GET IP$ : IF IP$ = "" THEN 1970
1980 IF IP$ = CHR$(20) THEN 1890
1990 IF IP$ <"1" OR IP$ >"8" THEN 1970
2000 PRINT IP$;
2010 PRINT RO$; " "; RF$; CHR$(157);
2020 GET E$ : IF E$ = "" THEN 2020
2030 IF E$ = CHR$(20) THEN 1890
2040 IF E$ <> CHR$(13) THEN 2020
2050 ZK = 10*(ASC(IN$)-64)+VAL(IP$)
2060 PRINT " ";
2070 IF F(ZK) >= 0 THEN 2690
2080 FOR A=1 TO 3:PRINT CHR$(157);:NEXT
2090 PRINT RO$; " "; RF$; "-";
2100 PRINT CHR$(157); CHR$(157);
2110 GOTO 1920
```

---

## **Erläuterungen zum nebenstehenden Programmausschnitt (13) / Forts.:**

Falls die Zielkoordinaten den Ansprüchen genügen, wird der Programmlauf in Zeile 2690 fortgesetzt, wo eine Überprüfung dahingehend vorgenommen wird, ob der beabsichtigte Zug den Schach-Regeln entspricht.

---

**Programmausschnitt (14):**

```
2120 REM *****
2130 REM *** ROCHADE MIT KONTROLLE ***
2140 REM *****
2150 RO=1
2160 PRINT " "; O$; "-"; O$; "  ";
2170 FOR A=1 TO 3:PRINT CHR$(157);:NEXT
2180 PRINT RO$; " "; RF$; CHR$(157);
2190 GET IN$ : IF IN$ = "" THEN 2190
2200 IF IN$ = CHR$(20) THEN 1890
2210 IF IN$ = CHR$(13) THEN 2280
2220 IF IN$ <> "O" THEN 2190
2230 PRINT "-"; O$; : RO=-1
2240 PRINT RO$; " "; RF$; CHR$(157);
2250 GET E$ : IF E$ = "" THEN 2250
2260 IF E$ = CHR$(20) THEN 1890
2270 IF E$ <> CHR$(13) THEN 2250
2280 IF F(46+RO*35) <>-5 THEN 2600
2290 IF F(51+RO*10) <> 0 THEN 2600
2300 IF F(51+RO*20) <> 0 THEN 2600
2310 IF RO=-1 AND F(21)<>0 THEN 2600
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (14):

Wenn in Zeile 1800 festgestellt wurde, daß der Anwender eine Rochade wünscht und der König noch rochadefähig ist (vgl. unsere Ausführungen auf Seite 2-37 ff.), verzweigte das System zu dem hier zu erläuternden Programmausschnitt.

In ihm hat sich der Anwender für die kleine (O-O) oder die große Rochade (O-O-O) zu entscheiden, wodurch das Merkfeld RO den Wert 1 (kleine Rochade) oder -1 (große Rochade) erhält (Zeile 2150 - 2270). Es folgen verschiedene Überprüfungen, nämlich

- ob der zu rochierende Turm noch nicht bewegt wurde (Zeile 2280 / die analoge Überprüfung des Königs erfolgte bereits in Zeile 1800),
  - ob die Felder zwischen den beiden zu rochierenden Figuren (König / Turm) schon leer sind (Zeile 2290 - 2310) und
-

**Programmausschnitt (15):**

```
2320 PRINT " "; : B=51
2330 FOR A=B+1 TO 78
2340 IF F(A)=0 THEN NEXT
2350 IF F(A)=6 OR F(A)=7 THEN 2600
2360 IF F(B+1)=9 THEN 2600
2370 FOR A=B+11 TO 86 STEP 11
2380 IF F(A)=0 THEN NEXT
2390 IF F(A)=4 OR F(A)=7 THEN 2600
2400 IF F(B+11)=9 THEN 2600
2410 FOR A=B+10 TO 81 STEP 10
2420 IF F(A)=0 THEN NEXT
2430 IF F(A)=6 OR F(A)=7 THEN 2600
2440 IF F(B+10)=9 THEN 2600
2450 FOR A=B-10 TO 11 STEP -10
2460 IF F(A)=0 THEN NEXT
2470 IF F(A)=6 OR F(A)=7 THEN 2600
2480 IF F(B-10)=9 THEN 2600
2490 FOR A=B-9 TO 13 STEP -9
2500 IF F(A)=0 THEN NEXT
2510 IF F(A)=4 OR F(A)=7 THEN 2600
2520 IF F(B-9)=9 THEN 2600
2530 IF F(B-19)=3 THEN 2600
2540 IF F(B-8)=3 THEN 2600
2550 IF F(B+12)=3 THEN 2600
2560 IF F(B+21)=3 THEN 2600
2570 IF F(B-9)=2 OR F(B+11)=2 THEN 2600
2580 IF B=51+RO*20 THEN 2620
2590 B=B+RO*10 : GOTO 2330
2600 RO=0
2610 GOTO 1890 : REM ROCHADE UNMOEGLICH
```

---



## Erläuterungen zum nebenstehenden Programmausschnitt (15):

- ob die vom König zu überstreichenden Felder unbedroht sind (Zeile 2320 - 2590).

Es muß erwähnt werden, daß wir uns hier (und an später folgenden Stellen) dazu entschließen mußten, FOR ... NEXT-Konstruktionen zu verwenden, die aufgrund einer IF-Abfrage meistens nicht bis zum Schleifenendwert durchlaufen werden (Beispiele: Zeile 2330 - 2340 / Zeile 2370 - 2380 / Zeile 2410 - 2420 etc.) Diese nicht unproblematische Codierung erfolgt zum Zwecke der Erhöhung der Ablaufgeschwindigkeit (FOR ... NEXT ist schneller als eine hier eigentlich angezeigte Schleifenzählererhöhung mit LET).

Um die oben erwähnte Überprüfung, ob die vom König zu überstreichenden Felder unbedroht sind, durchführen zu können, setzen wir in Zeile 2320 die Variable B auf 51 ( $\hat{=}$  Position des Königs). Mit diesem Wert untersuchen wir zunächst, ob sich der König in unbedrohter Position befindet. Eine Rochade ist nämlich nicht zulässig, wenn der König im Schach steht (Zeile 2320 - 2570).

**Programmausschnitt (15):**

```
2320 PRINT " "; : B=51
2330 FOR A=B+1 TO 78
2340 IF F(A)=0 THEN NEXT
2350 IF F(A)=6 OR F(A)=7 THEN 2600
2360 IF F(B+1)=9 THEN 2600
2370 FOR A=B+11 TO 86 STEP 11
2380 IF F(A)=0 THEN NEXT
2390 IF F(A)=4 OR F(A)=7 THEN 2600
2400 IF F(B+11)=9 THEN 2600
2410 FOR A=B+10 TO 81 STEP 10
2420 IF F(A)=0 THEN NEXT
2430 IF F(A)=6 OR F(A)=7 THEN 2600
2440 IF F(B+10)=9 THEN 2600
2450 FOR A=B-10 TO 11 STEP -10
2460 IF F(A)=0 THEN NEXT
2470 IF F(A)=6 OR F(A)=7 THEN 2600
2480 IF F(B-10)=9 THEN 2600
2490 FOR A=B-9 TO 13 STEP -9
2500 IF F(A)=0 THEN NEXT
2510 IF F(A)=4 OR F(A)=7 THEN 2600
2520 IF F(B-9)=9 THEN 2600
2530 IF F(B-19)=3 THEN 2600
2540 IF F(B-8)=3 THEN 2600
2550 IF F(B+12)=3 THEN 2600
2560 IF F(B+21)=3 THEN 2600
2570 IF F(B-9)=2 OR F(B+11)=2 THEN 2600
2580 IF B=51+RO*20 THEN 2620
2590 B=B+RO*10 : GOTO 2330
2600 RO=0
2610 GOTO 1890 : REM ROCHADE UNMOEGlich
```

---

## **Erläuterungen zum nebenstehenden Programmausschnitt (15) / Forts.:**

Die Untersuchung der Position 61 / 71 (bei der kleinen Rochade) bzw. 41 / 31 (bei der großen Rochade) wird ermöglicht durch die Veränderung von B in Zeile 2590. Die Abfrage in Zeile 2580 stellt sicher, daß nur die Positionen 51 / 61 / 71 ( $\hat{=}$  kleine Rochade) bzw. 51 / 41 / 31 ( $\hat{=}$  große Rochade) untersucht werden. Im Falle der Bedrohungssituation für eines der aufgeführten Felder verzweigt das Programm nach Zeile 2600, um von dort über Zeile 2610 nach Zeile 1890 zurückzukehren.

---

**Programmausschnitt (16):**

```
2620 REM ***** ROCHADENAUSFUEHRUNG *****
2630 F(51)=0          : F(51+RO*20)=-9
2640 F(46+RO*35)=0   : F(51+RO*10)=-6
2650 GOTO 3720
2660 REM *****
2670 REM ***** ROCHADE/ENDE *****
```

---

## **Erläuterungen zum nebenstehenden Programmausschnitt (16):**

Wenn keine Umstände der Rochade entgegenstehen, erfolgt hier ihre interne Ausführung, und der Programmablauf wird in Zeile 3720 fortgesetzt (Ende des Unterprogramms SPIELEREINGABE mit automatischem Rücksprung nach Zeile 1200 des Spielteils).

---

**Programmausschnitt (17):**

```
2680 REM *****
2690 REM ***** EINGABEKONTROLLE *****
2700 REM *****
2710 ON ABS(F(SK)+1) GOTO 2780,2860,2920
,3080,3080,3240,3320,3320
2720 REM
2730 REM ***** BAUER 2-SCHRITT *****
2740 IF ZK-SK<>2 THEN 2780
2750 IF F(SK+1)=0 AND F(ZK)=0 THEN 3370
2760 GOTO 2080
2770 REM
2780 REM ***** BAUER ALLGEMEIN *****
2790 IF SK-ZK<>9 AND ZK-SK<>11 THEN 2820
2800 IF F(ZK)>0 THEN 3370
2810 GOTO 2080
2820 IF ZK-SK<>1 THEN 2080
2830 IF F(ZK)<>0 THEN 2080
2840 GOTO 3370
2850 REM
```

## Erläuterungen zum nebenstehenden Programmausschnitt (17):

Während in den vorangegangenen Programmausschnitten (14) - (16) die vom Anwender gewünschte Rochade auf Zulässigkeit überprüft und ggf. ausgeführt wurde, erfolgt ab Zeile 2680 die Kontrolle aller anderen jeweils geforderten Züge.

Mit Hilfe eines ON .. GOTO-Befehls in Zeile 2710 verzweigen wir - abhängig vom Inhalt des Startkoordinatenfeldes (also abhängig von der Feldfigur) - an 8 verschiedene Stellen. Nur wenn in  $F(SK) = -1$  ( $\hat{=}$  Bauer in Grundposition) steht, setzt der Programm-  
lauf mit den unmittelbar anschließenden Anweisungen fort, weil  $ABS(-1 + 1) = 0$  ist. Hier untersuchen wir die Zugangabe für einen Bauern in Grundposition: Falls in Zeile 2740 festgestellt wird, daß kein Doppelschritt gewünscht ist, erfolgt ein Sprung nach Zeile 2780. Andernfalls untersuchen wir in Zeile 2750, ob die beiden tangierten Felder leer sind. Ist diese Bedingung erfüllt, setzt der Programmlauf mit dem Abschnitt INTERNE ZUGAUSFUEHRUNG fort.

Mit Ausnahme des bereits untersuchten Doppelschritts eines Bauern werden alle anderen Bauernzüge ab Zeile 2780 analysiert, und zwar überprüfen wir das "Schrägschlagen" in Zeile 2790 - 2810 und das "Vorwärtsschreiten" in Zeile 2820 - 2840.

---

**Programmausschnitt (18):**

```
2860 REM ***** SPRINGER *****
2870 FOR A=1 TO 4
2880 IF ABS(ZK-SK)=FA(A) THEN 3370
2890 NEXT A
2900 GOTO 2080
2910 REM
2920 REM ***** LAEUFER/DAME *****
2930 DI=ABS(ZK-SK) : V=SGN(ZK-SK)
2940 IF DI/11=INT(DI/11) THEN 2970
2950 IF DI/ 9=INT(DI/ 9) THEN 3020
2960 GOTO 2080
2970 IF DI=11 THEN 3370
2980 FOR A=SK+11*V TO ZK-11*V STEP 11*V
2990 IF F(A)<>0 THEN 2080
3000 NEXT A
3010 GOTO 3370
3020 IF DI=9 THEN 3370
3030 FOR A=SK+9*V TO ZK-9*V STEP 9*V
3040 IF F(A)<>0 THEN 2080
3050 NEXT A
3060 GOTO 3370
3070 REM
```



## Erläuterungen zum nebenstehenden Programmausschnitt (18):

An dieser Stelle (Zeile 2860 - 2900) untersuchen wir die Zulässigkeit des Springer-Zuges. Der Absolutwert der Differenz aus Start- und Zielcoordinate muß mit einem jener Feldabstandswerte übereinstimmen, die wir in Zeile 790 u. a. für den Springer definierten (vgl. hierzu Seite 2-24 f.).

Die richtige "Behandlung" des Läufers überprüfen wir folgendermaßen:

Zeile 2930 - 2960: Der Absolutwert der Differenz aus Start- und Zielcoordinate muß durch 11 oder 9 teilbar sein (schräges Ziehen).

Zeile 2970 - 3060: Die Felder zwischen Start- und Zielcoordinate dürfen durch keine Figur besetzt sein.

---

**Programmausschnitt (19):**

```
3080 REM ***** TURM/DAME *****
3090 DI=ABS(ZK-SK) : V=SGN(ZK-SK)
3100 IF DI<8 THEN 3130
3110 IF DI/10=INT(DI/10) THEN 3180
3120 GOTO 2080
3130 IF DI=1 THEN 3370
3140 FOR A=SK+V TO ZK-V STEP V
3150 IF F(A)<>0 THEN 2080
3160 NEXT A
3170 GOTO 3370
3180 IF DI=10 THEN 3370
3190 FOR A=SK+10*V TO ZK-10*V STEP 10*V
3200 IF F(A)<>0 THEN 2080
3210 NEXT A
3220 GOTO 3370
3230 REM
3240 REM ***** DAME *****
3250 DI=ABS(ZK-SK) : V=SGN(ZK-SK)
3260 IF DI/11=INT(DI/11) THEN 2970
3270 IF DI/10=INT(DI/10) THEN 3180
3280 IF DI/ 9=INT(DI/ 9) THEN 3020
3290 IF DI<8 THEN 3130
3300 GOTO 2080
3310 REM
3320 REM ***** KOENIG *****
3330 DI=ABS(ZK-SK)
3340 IF DI=10 OR DI=11 THEN 3370
3350 IF DI<>1 AND DI<>9 THEN 2080
3360 REM
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (19):

Die zu einem Turm-Zug anzustellenden Überlegungen (Zeile 3080 - 3220) entsprechen weitgehend den bereits dargestellten zum Läufer. Der Unterschied besteht nur darin, daß der Absolutwert der Differenz aus Start- und Zielcoordinate durch 10 teilbar oder  $>8$  sein muß (gerades Ziehen).

Für die Dame (Zeile 3240 - 3300) nutzen wir ausschließlich die Untersuchungen zum Läufer oder Turm durch Rücksprung an die entsprechende Stelle.

Der Zug eines Königs (Zeile 3240 - 3300) muß lediglich folgenden Ansprüchen genügen: Der Absolutwert der Differenz aus Start- und Zielcoordinate darf nur 1, 9, 10 oder 11 betragen.

---

**Programmausschnitt (20):**

```
3370 REM *** INTERNE ZUGAUSFUEHRUNG ***
3380 REM
3390 REM * KONTROLLE KOENIG IN SCHACH **
3400 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
3410 FOR PK=11 TO 88
3420 IF F(PK)=-8 OR F(PK)=-9 THEN 3440
3430 NEXT PK
3440 B=-1
3450 MW=50+B*39
3460 FOR A=PK+B TO MW STEP B
3470 IF F(A)=0 THEN NEXT
3480 IF F(A)>=5 AND F(A)<=7 THEN 3730
3490 IF F(PK+B)=8 OR F(PK+B)=9 THEN 3730
3500 FOR A=PK+B*11 TO MW STEP B*11
3510 IF F(A)=0 THEN NEXT
3520 IF F(A)=4 OR F(A)=7 THEN 3730
3530 IF F(PK+B*11)=8 OR F(PK+B*11)=9 THEN
N 3730
3540 FOR A=PK+B*10 TO MW STEP B*10
3550 IF F(A)=0 THEN NEXT
3560 IF F(A)>=5 AND F(A)<=7 THEN 3730
3570 IF F(PK+B*10)=8 OR F(PK+B*10)=9 THEN
N 3730
3580 FOR A=PK+B*9 TO MW STEP B*9
3590 IF F(A)=0 THEN NEXT
3600 IF F(A)=4 OR F(A)=7 THEN 3730
3610 IF F(PK+B*9)=8 OR F(PK+B*9)=9 THEN
3730
3620 FOR A=1 TO 4
3630 S=PK+FA(A)*B
3640 IF S<11 OR S>88 THEN 3660
3650 IF F(S)=3 THEN 3730
3660 NEXT A
3670 IF B=-1 THEN B=1 : GOTO 3450
```

```

3680 IF F(PK-9)=1 OR F(PK+11)=1 THEN 3730
3690 IF F(PK-9)=2 OR F(PK+11)=2 THEN 3730
3700 IF F(ZK)=-2 AND ZK-INT(ZK/10)*10=8
THEN F(ZK)=-7
3710 IF F(ZK)=-1 OR F(ZK)=-5 OR F(ZK)=-8
THEN F(ZK)=F(ZK)-1
3720 RETURN
3730 F(SK)=F(ZK) : F(ZK)=FZ : GOTO 2080
3740 REM *****
3750 REM *** EINGABEKONTROLLE/ENDE ***
3760 REM *****
3770 REM *** SPIELEREINGABE/ENDE *****

```

## Erläuterungen zum nebenstehenden / obenstehenden Programm- ausschnitt (20):

Nachdem festgestellt worden ist, daß der geforderte Zug den jeweiligen Anforderungen genügt, gelangt er hier - zunächst nur intern - zur Ausführung.<sup>1)</sup> Dabei wird vorher die Figur des Zielkoordinatenfeldes in FZ abgelegt (Zeile 3400). Es folgt die Ermittlung der Koordinate des spielereigenen Königs (Zeile 3410 - 3430), um zu überprüfen, ob er durch den - zunächst nur vorläufigen - Zug ins Schach gerät (Zeile 3440 - 3690). Ist dies der Fall, dann wird der Zug rückgängig gemacht (Zeile 3730). Wenn der König nicht bedroht ist, gilt der Zug als akzeptiert. Außerdem wird ggf. die Zahl geändert, welche die soeben gezogene Figur repräsentierte (Zeile 3710), nämlich beim Bauern ggf. von -1 zu -2, beim Turm ggf. von -5 zu -6 und beim König ggf. von -8 zu -9 (vgl. hierzu die Ausführungen auf Seite 2-04!). Und wenn ein Bauer die gegnerische Grundlinie erreicht hat, erfolgt seine Umwandlung in eine Dame (Zeile 3700).

<sup>1)</sup> Dies gilt nicht für die Rochade, die in Zeile 2620 - 2640 zur Ausführung gelangt (vgl. Seite 2-50 f.).

**Programmausschnitt (21):**

```
3780 REM *****
3790 REM ***** ZUGANGABE *****
3800 REM *****
3810 PRINT CHR$(19); TAB(33) "";
3820 FOR A=1 TO 9:PRINT CHR$(17); :NEXT
3830 IF RO=0 THEN 3880
3840 IF RO=1 THEN PRINT " ";
3850 IF RO=-1 THEN PRINT O$; "-";
3860 PRINT O$; "-"; O$; " "
3870 RO=0 : GOTO 3920
3880 PRINT CHR$(64192+INT(SK/10));
3890 PRINT RIGHT$(STR$(SK),1); ":";
3900 PRINT CHR$(64192+INT(ZK/10));2
3910 PRINT RIGHT$(STR$(ZK),1)
3920 RETURN
3930 REM *****
3940 REM ***** ZUGANGABE/ENDE *****
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (21):

Nachdem das System die Zugeingabe des Spielers (bzw. des Computers) akzeptiert hat, wird aus dem SPIEL-TEIL (vgl. hierzu Seite 2-32 f.!) in Zeile 1200 (bzw. 1240) das Unterprogramm ZUGANGABE aufgerufen. Es hat den alleinigen Zweck, das Spiel auf dem Bildschirm dadurch übersichtlicher zu gestalten, daß rechts neben dem Schachbrett der zuletzt getätigte Zug erscheint.

Für den weniger erfahrenen Programmierer könnten vor allem die Befehlszeilen 3880 - 3910 schwer verständlich sein. Sie dienen dazu, aus dem numerischen Inhalt von SK und ZK die Zeichen der Schachnotation zu ermitteln.

### Beispiel:

Der numerische Inhalt der Startkoordinate sei 52 ( $\hat{=}$  E2). Nun muß zunächst aus der Ziffer 5 der Buchstabe E generiert werden. Wir ermitteln deshalb (Zeile 3880) den Ganzzahlwert der ersten Ziffer durch  $\text{INT}(\text{SK}/10)$ , was 5 ergibt; wir erhöhen diesen Wert um die Konstante 192 mit dem Ergebnis, daß wir den ASCII-Wert für den Großbuchstaben E erhalten, um diesen

**Programmausschnitt (21):**

```
3780 REM *****
3790 REM ***** ZUGANGABE *****
3800 REM *****
3810 PRINT CHR$(19); TAB(33) "";
3820 FOR A=1 TO 9:PRINT CHR$(17); :NEXT
3830 IF RO=0 THEN 3880
3840 IF RO=1 THEN PRINT " ";
3850 IF RO=-1 THEN PRINT O$; "-";
3860 PRINT O$; "-"; O$; " "
3870 RO=0 : GOTO 3920
3880 PRINT CHR$(192+INT(SK/10));
3890 PRINT RIGHT$(STR$(SK),1); ":";
3900 PRINT CHR$(192+INT(ZK/10));
3910 PRINT RIGHT$(STR$(ZK),1)
3920 RETURN
3930 REM *****
3940 REM ***** ZUGANGABE/ENDE *****
```

---



## Erläuterungen zum nebenstehenden Programmausschnitt (21) / Forts.:

dann mittels der vordefinierten Funktion CHR\$(..) zu erzeugen; durch den vorangestellten PRINT-Befehl wird der Buchstabe ausgegeben.

In Zeile 3890 wollen wir die zweite Ziffer in SK (52) abtrennen, um sie nach dem E (ohne Zwischenraum!) auf dem Bildschirm erscheinen zu lassen. Zu diesem Zweck wandeln wir SK durch STR\$(SK) in einen String um und trennen hiervon rechts mittels RIGHT\$(..,1) ein Zeichen (2) ab, das durch das vorangestellte PRINT ausgegeben wird. Nach dem Schreiben des Doppelpunkts (auch in Zeile 3890) wiederholt sich in den Zeilen 3900 und 3910 der Vorgang für die Zielkoordinate.

---

**Programmausschnitt (22):**

```
3950 REM *****
3960 REM ***** COMPUTERZUG *****
3970 REM *****
3980 GOSUB 7880
3990 PRINT TAB(14) B$;"ITTE WARTEN"
4000 PRINT
4010 PRINT TAB(12) S$;"TELLUNGSANALYSE"
4020 PRINT
4030 PRINT TAB(13) "80 BIS ZUM ";
4040 PRINT Z$;"UG"; CHR$(145)
4050 ZN=0 : BZ=1
4060 IF F(58)=8 THEN GOSUB 6140
4070 SK=11
4080 PRINT TAB(13) "";
4090 PRINT RIGHT$(STR$(91-SK),2);
4100 PRINT CHR$(145)
4110 IF F(SK)<=0 OR F(SK)=10 THEN 4130
4120 GOSUB 5410
4130 IF SK<88 THEN SK=SK+1 : GOTO 4080
4140 FOR A=1 TO ZN-1 : FOR B=A+1 TO ZN
4150 IF BS(A)>=BS(B) THEN 4190
4160 BS(0)=BS(A):BS(A)=BS(B):BS(B)=BS(0)
4170 SK(0)=SK(A):SK(A)=SK(B):SK(B)=SK(0)
4180 ZK(0)=ZK(A):ZK(A)=ZK(B):ZK(B)=ZK(0)
4190 NEXT B,A
4200 PRINT TAB(14) "2"; CHR$(145)
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (22):

Nachdem in Zeile 1220 des Spielteils das Unterprogramm COMPUTERZUG aufgerufen worden ist, wird in Zeile 3980 in die ab Zeile 7880 beginnende Subroutine verzweigt, mit der wir den unteren Teil des Bildschirms löschen, um dann durch die Befehlszeilen 3990 - 4040 den Text

Bitte warten  
Stellungsanalyse  
80 bis zum Zug

auszugeben. Diese Nachricht erscheint, damit der Spieler durch die von 80 bis 0 ( $\hat{=}$  Zugausführung) abnehmende Zahl überblicken kann, wie weit der Rechner mit seinen "Überlegungen" fortgeschritten ist.

In Zeile 4050 setzen wir dann die Variable ZN ( $\hat{=}$  Zugnummer) auf 0, mit deren Hilfe wir an späterer Stelle jedem möglichen Computerzug eine Nummer zuordnen. Sie soll als Index dazu dienen, den taktischen Wert, die Startkoordinate und die Zielkoordinate des jeweiligen Computerzugs zu speichern. Die Variable BZ ( $\hat{=}$  Bester Zug) mit dem Anfangswert von 1 wird dazu dienen, den bestmöglichen Zug des

**Programmausschnitt (22):**

```
3950 REM *****
3960 REM ***** COMPUTERZUG *****
3970 REM *****
3980 GOSUB 7880
3990 PRINT TAB(14) B$;"ITTE WARTEN"
4000 PRINT
4010 PRINT TAB(12) S$;"TELLUNGSANALYSE"
4020 PRINT
4030 PRINT TAB(13) "80 BIS ZUM ";
4040 PRINT Z$;"UG"; CHR$(145)
4050 ZN=0 : BZ=1
4060 IF F(58)=8 THEN GOSUB 6140
4070 SK=11
4080 PRINT TAB(13) "";
4090 PRINT RIGHT$(STR$(91-SK),2);
4100 PRINT CHR$(145)
4110 IF F(SK)<=0 OR F(SK)=10 THEN 4130
4120 GOSUB 5410
4130 IF SK<88 THEN SK=SK+1 : GOTO 4080
4140 FOR A=1 TO ZN-1 : FOR B=A+1 TO ZN
4150 IF BS(A)>=BS(B) THEN 4190
4160 BS(0)=BS(A):BS(A)=BS(B):BS(B)=BS(0)
4170 SK(0)=SK(A):SK(A)=SK(B):SK(B)=SK(0)
4180 ZK(0)=ZK(A):ZK(A)=ZK(B):ZK(B)=ZK(0)
4190 NEXT B,A
4200 PRINT TAB(14) "2"; CHR$(145)
```

## Erläuterungen zum nebenstehenden Programmausschnitt (22) / Forts.:

Rechners zu ermitteln.

Wenn die Abfrage in Zeile 4060 ergibt, daß der König noch rochadefähig ist, erfolgt der Aufruf einer Subroutine (ab Zeile 6140), in der die weiteren Voraussetzungen für eine Rochade untersucht werden.

Anschließend verwenden wir SK (Startkoordinate) mit dem Anfangswert 11 ( $\hat{=}$  A1) als Zählvariable einer Schleifenkonstruktion, die dazu dient, computer-eigene Figuren zu suchen (Zeile 4070 - 4130). Damit der Spieler den Fortschritt des Suchlaufs verfolgen kann, geben wir - beginnend mit 80 (vgl. Seite 2-65!) - eine Zahl aus, die in Abhängigkeit von einem steigenden Wert der Variablen SK abnimmt (Zeile 4080 - 4100).

Wenn der Rechner eine "ihm gehörende" Figur gefunden hat (Zeile 4110), erfolgt in Zeile 4120 der Aufruf einer Subroutine. In ihr werden für die gefundene Figur alle möglichen Züge erzeugt, bewertet (durch Aufruf einer "Sub-Subroutine") und gespeichert.

---

**Programmausschnitt (22):**

```
3950 REM *****
3960 REM ***** COMPUTERZUG *****
3970 REM *****
3980 GOSUB 7880
3990 PRINT TAB(14) B$;"ITTE WARTEN"
4000 PRINT
4010 PRINT TAB(12) S$;"TELLUNGSANALYSE"
4020 PRINT
4030 PRINT TAB(13) "80 BIS ZUM ";
4040 PRINT Z$;"UG"; CHR$(145)
4050 ZN=0 : BZ=1
4060 IF F(58)=8 THEN GOSUB 6140
4070 SK=11
4080 PRINT TAB(13) "";
4090 PRINT RIGHT$(STR$(91-SK),2);
4100 PRINT CHR$(145)
4110 IF F(SK)<=0 OR F(SK)=10 THEN 4130
4120 GOSUB 5410
4130 IF SK<88 THEN SK=SK+1 : GOTO 4080
4140 FOR A=1 TO ZN-1 : FOR B=A+1 TO ZN
4150 IF BS(A)>=BS(B) THEN 4190
4160 BS(0)=BS(A):BS(A)=BS(B):BS(B)=BS(0)
4170 SK(0)=SK(A):SK(A)=SK(B):SK(B)=SK(0)
4180 ZK(0)=ZK(A):ZK(A)=ZK(B):ZK(B)=ZK(0)
4190 NEXT B,A
4200 PRINT TAB(14) "2"; CHR$(145)
```

## Erläuterungen zum nebenstehenden Programmausschnitt (22) / Forts.:

Der auf Seite 2-67 geschilderte Funktionsablauf wird so lange wiederholt, bis SK den Wert 88 ( $\hat{=}$  H8) angenommen hat (Zeile 4130): Dann ist jeder mögliche Zug des Rechners mit seiner Wertung gespeichert.

Im Anschluß daran werden die in SK(..) und ZK(..) gespeicherten Züge entsprechend ihrer in BS(..) (Bewertung der Stellung) aufgezeichneten Bewertungen sortiert<sup>1)</sup> (Zeile 4140 - 4190), und zwar gelangt der Zug mit der besten Wertung nach SK(1) und ZK(1), derjenige mit der zweitbesten Wertung nach SK(2) und ZK(2) etc.

In der letzten Befehlszeile unseres Abschnitts wird die Ziffer "2" ausgegeben, um damit dem "Eingeweihten" zu signalisieren, daß die Sortierung der bewerteten Züge abgeschlossen ist:

Bitte warten  
Stellungsanalyse  
2 bis zum Zug

---

1) Vgl. hierzu die Erläuterungen bei Hamann, G. O.: Lerne BASIC mit dem Commodore 64 / VC 20, (Deutscher Betriebswirte-Verlag GmbH, ISBN: 3-88640-016-6), Gernsbach 1984, Seite 21-17 ff.

---

**Programmausschnitt (23):**

```
4210 IF BZ<ZN+1 THEN 4240
4220 IF MA=1 THEN PA=1 : GOTO 4320
4230 MA=1 : BZ=1
4240 SK=SK(BZ) : ZK=ZK(BZ)
4250 IF ZK>1 THEN 4300
4260 RO=ZK
4270 ZK=58+RO*10
4280 F(58)=0
4290 F(58+RO*20)=9
4300 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
4310 REM
```

---



## Erläuterungen zum nebenstehenden Programmausschnitt (23):

Von den in "abnehmender Qualität" sortierten Zügen können wir nicht ohne weiteres den in SK(1) und ZK(1) gespeicherten ausführen lassen, weil die Bewertung zunächst in sehr abstrakter Weise geschah, d. h. es wurde beispielsweise noch nicht untersucht, ob eine Ausführung des "besten Zugs" dazu führt, daß der "König des Rechners" ins Schach gerät und - wenn der König nicht ins Schach gerät - ob die gezogene Figur bedroht wäre etc. Diese (und andere) Überprüfungen sind also nachzuholen.

Wir beginnen in Zeile 4210 mit der Abfrage, ob der Inhalt der Variablen BZ (Bester Zug) - sie wurde in Zeile 4050 auf 1 gesetzt - kleiner ist als ZN + 1, wobei ZN hier die Anzahl der ermittelten Züge enthält. Ist die Bedingung erfüllt (was in den allermeisten Fällen der Überprüfung der Fall sein wird), erfolgt ein Sprung nach Zeile 4240, wo wir SK den Inhalt von SK(BZ) und ZK jenen von ZK(BZ) zuweisen, um jetzt den Zug (Zeile 4300) - zunächst nur intern - auszuführen. Wenn allerdings ZK den Wert -1 oder 1 hat - d. h. die Bedingung (Zeile 4250) ZK > 1 ist nicht erfüllt - soll statt der "normalen" Zugausführung in Zeile 4300 eine große (ZK = -1)

---

**Programmausschnitt (23):**

```
4210 IF BZ<ZN+1 THEN 4240
4220 IF MA=1 THEN PA=1 : GOTO 4320
4230 MA=1 : BZ=1
4240 SK=SK(BZ) : ZK=ZK(BZ)
4250 IF ZK>1 THEN 4300
4260 RO=ZK
4270 ZK=58+RO*10
4280 F(58)=0
4290 F(58+RO*20)=9
4300 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
4310 REM
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (23) / Forts.:

bzw. kleine Rochade ( $ZK = 1$ ) zur Ausführung gelangen (Zeile 4260 - 4300).

Es folgen nun die schon angedeuteten Überprüfungen (zunächst also bei einem Wert von  $BZ = 1$ ). Wir wollen einmal unterstellen, daß die weiteren Untersuchungen für jenen Zug, der vorläufig als bester ermittelt worden war ( $BZ = 1$ ), zu seiner Verwerfung führe, weil z. B. der König dadurch ins Schach geräte. Dann würde der ursprünglich zweitbeste Zug ( $BZ = 2$ ) analysiert werden; bei seiner Verwerfung folgte eine Kontrolle für  $BZ = 3$ ; (ggf.) etc.

Wir unterstellen jetzt ferner (den sehr seltenen Fall), daß alle ermittelten Züge im nachhinein verworfen werden mußten. Die Bedingung in Zeile 4210 wäre demnach nicht mehr erfüllt, also  $BZ = ZN + 1$ . Dann wird erstmals überprüft, ob  $MA = 1$  ist (Zeile 4220), wobei  $MA$  (Matt-Vermutung) die Funktion eines "Merkers" erfüllt. Wie jede numerische Variable hat  $MA$  (hier zunächst) den Anfangswert 0. Die Bedingung ist somit nicht erfüllt, und in Zeile 4230 wird  $MA$  deshalb auf 1 gesetzt, um damit "festzuhalten", daß in der ersten "Überprüfungsrunde" kein gespeicherter Zug

---

**Programmausschnitt (23):**

```
4210 IF BZ<ZN+1 THEN 4240
4220 IF MA=1 THEN PA=1 : GOTO 4320
4230 MA=1 : BZ=1
4240 SK=SK(BZ) : ZK=ZK(BZ)
4250 IF ZK>1 THEN 4300
4260 RO=ZK
4270 ZK=58+RO*10
4280 F(58)=0
4290 F(58+RO*20)=9
4300 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
4310 REM
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (23) / Forts.:

den Ansprüchen genügen konnte. Zusätzlich erhält BZ wieder den Anfangswert 1, um damit eine neue "Überprüfungsrunde" zu beginnen, in der nur noch nach einem Zug gesucht wird, bei dem die gezogene Figur zwar gefährdet ist, der König aber nicht ins Schach gerät. Wir unterstellen weiter (vor allem zum Zwecke der Erläuterung des Programms), daß auch in dieser Runde alle Züge verworfen werden müssen, weil der König immer ins Schach gerät. Dadurch ist schließlich die Bedingung in Zeile 4210 abermals nicht mehr erfüllt, also  $BZ = ZN + 1$ . Die Abfrage in Zeile 4220 führt nun dazu, daß der Variablen PA (Patt-Vermutung), die immer noch den Anfangswert 0 hat, eine 1 zugewiesen wird. Es folgt ein Sprung nach Zeile 4320 - es wird also kein Zug mehr ausgeführt - und ausschließlich die Untersuchung, ob der König im Schach steht. Sollte dies der Fall sein, dann ist der Rechner schachmatt, andernfalls besteht eine Patt-Situation.

Wenn allerdings in den ersten beiden "Überprüfungsrunden" ein geeigneter Zug gefunden wird, so gilt dieser als gewählt, und es erfolgt dessen Ausgabe auf dem Bildschirm.

**Programmausschnitt (24):**

```
4320 REM * KONTROLLE KOENIG IN SCHACH **
4330 FOR PK=11 TO 88
4340 IF F(PK)=8 OR F(PK)=9 THEN 4360
4350 NEXT PK
4360 B=-1
4370 MW=50+B*39
4380 FOR A=PK+B TO MW STEP B
4390 IF F(A)=0 THEN NEXT
4400 IF F(A)>=-7 AND F(A)<=-5 THEN 5320
4410 IF F(PK+B)=-8 OR F(PK+B)=-9 THEN 53
20
4420 FOR A=PK+B*11 TO MW STEP B*11
4430 IF F(A)=0 THEN NEXT
4440 IF F(A)=-4 OR F(A)=-7 THEN 5320
4450 IF F(PK+B*11)=-8 OR F(PK+B*11)=-9 T
HEN 5320
4460 FOR A=PK+B*10 TO MW STEP B*10
4470 IF F(A)=0 THEN NEXT
4480 IF F(A)>=-7 AND F(A)<=-5 THEN 5320
4490 IF F(PK+B*10)=-8 OR F(PK+B*10)=-9 T
HEN 5320
4500 FOR A=PK+B*9 TO MW STEP B*9
4510 IF F(A)=0 THEN NEXT
4520 IF F(A)=-4 OR F(A)=-7 THEN 5320
4530 IF F(PK+B*9)=-8 OR F(PK+B*9)=-9 THE
N 5320
4540 FOR A=1 TO 4
4550 S=PK+FA(A)*B
4560 IF S<11 OR S>88 THEN 4580
4570 IF F(S)=-3 THEN 5320
4580 NEXT A
4590 IF B=-1 THEN B=1 : GOTO 4370
```

```
4600 IF F(PK+9)=-1 OR F(PK-11)=-1 THEN 5
320
4610 IF F(PK+9)=-2 OR F(PK-11)=-2 THEN 5
320
4620 PRINT TAB(14) "1"; CHR$(145)
4630 IF PA=1 THEN 5300
4640 IF MA=1 THEN MA=0 : GOTO 5260
4650 IF F(ZK)=7 THEN 4680
4660 IF FW(F(ZK))<=FW(-FZ) THEN 5260
```

## **Erläuterungen zum nebenstehenden / obenstehenden Programm- ausschnitt (24):**

Die in den Erläuterungen zum Programmausschnitt (23) mehrmals erwähnten Überprüfungen bedürfen jetzt nur noch eines sehr knappen Kommentars:

Zunächst ermitteln wir die Position des Königs (Zeile 4330 - 4350), um anschließend zu untersuchen, ob dieser sich im Schach befindet (Zeile 4360 - 4610). Nur wenn dies der Fall ist, setzt der Programmablauf in Zeile 5320 fort, wo der Zug zurückgesetzt wird, um dann mit einem um 1 erhöhten BZ in Zeile 4210 (vgl. Seite 2-71!) fortzufahren.

---

## Erläuterungen zum Programmausschnitt (24) auf Seite 2-76 f. / Forts.:

Ist der König hingegen nicht im Schach, folgen weitere Untersuchungen, die in den folgenden Abschnitten darzustellen sein werden.

Die Ausgabe der Ziffer "1" (Zeile 4620) signalisiert dem "Eingeweihten", daß die Zugüberprüfung stattfindet:

```
        Bitte warten  
Stellungsanalyse  
        1 bis zum Zug
```

Wenn die Bedingung  $PA = 1$  in Zeile 4630 erfüllt ist, besteht eine Patt-Situation, mit der Folge eines Rücksprungs zum SPIELTEIL und der Ausgabe von

```
        Patt !  
        =====
```

Nur wenn in der zweiten "Überprüfungsrunde" festgestellt wird, daß der König nicht im Schach steht, ist die Bedingung in Zeile 4640 erfüllt, und der gerade untersuchte Zug gilt als ausgeführt.

---



In der ersten "Überprüfungsrunde" wird hingegen verglichen (Zeile 4660), ob der Wert der gezogenen Figur kleiner/gleich ist wie der Wert jener Figur, die vorher auf dem Zielfeld stand. Bei Bejahung wird der Zug akzeptiert. Ein Damenabtausch, der aus spielstrategischen Gründen unerwünscht ist, wird allerdings durch die Abfrage in Zeile 4650 vermieden.

---

**Programmausschnitt (25):**

```
4670 REM *****
4680 REM **** DECKUNG VON FIGUREN ****
4690 REM *****
4700 REM
4710 REM ***** FIGUR BEDROHT ??? *****
4720 B=-1
4730 MW=50+B*39
4740 FOR A=ZK+B TO MW STEP B
4750 IF F(A)=0 THEN NEXT
4760 IF F(A)>=-7 AND F(A)<=-5 THEN 4990
4770 IF F(ZK+B)=-8 OR F(ZK+B)=-9 THEN 49
90
4780 FOR A=ZK+B*11 TO MW STEP B*11
4790 IF F(A)=0 THEN NEXT
4800 IF F(A)=-4 OR F(A)=-7 THEN 4990
4810 IF F(ZK+B*11)=-8 OR F(ZK+B*11)=-9 T
HEN 4990
4820 FOR A=ZK+B*10 TO MW STEP B*10
4830 IF F(A)=0 THEN NEXT
4840 IF F(A)>=-7 AND F(A)<=-5 THEN 4990
4850 IF F(ZK+B*10)=-8 OR F(ZK+B*10)=-9 T
HEN 4990
4860 FOR A=ZK+B*9 TO MW STEP B*9
4870 IF F(A)=0 THEN NEXT
4880 IF F(A)=-4 OR F(A)=-7 THEN 4990
4890 IF F(ZK+B*9)=-8 OR F(ZK+B*9)=-9 THE
N 4990
4900 FOR A=1 TO 4
4910 S=ZK+FA(A)*B
4920 IF S<11 OR S>88 THEN 4940
4930 IF F(S)=-3 THEN 4990
4940 NEXT A
4950 IF B=-1 THEN B=1 : GOTO 4730
```

```
4960 IF F(ZK+9)=-1 OR F(ZK-11)=-1 THEN 4  
990  
4970 IF F(ZK+9)=-2 OR F(ZK-11)=-2 THEN 4  
990  
4980 GOTO 5260  
4990 IF F(ZK)=7 THEN 5320
```

## **Erläuterungen zum nebenstehenden / obenstehenden Programm- ausschnitt (25):**

In diesem Programmteil überprüfen wir, ob die gezogene Figur geschlagen werden könnte. Ist dies der Fall, erfolgt ein Sprung nach Zeile 4990, um nur für die Dame auszuschließen, daß sie verloren gehen könnte, weil dies aus spielstrategischen Gründen unerwünscht ist. Für andere gezogene Figuren würde eine Bedrohungssituation akzeptiert werden, falls für die jeweilige Figur Deckung bestünde.

Wenn die gezogene Figur hingegen nicht bedroht ist, gilt der Zug.

---

**Programmausschnitt (26):**

```
5000 REM
5010 REM ***** FIGUR GEDECKT ??? *****
5020 B=-1
5030 MW=50+B*39
5040 FOR A=ZK+B TO MW STEP B
5050 IF F(A)=0 THEN NEXT
5060 IF F(A)>=5 AND F(A)<=7 THEN 5260
5070 FOR A=ZK+B*11 TO MW STEP B*11
5080 IF F(A)=0 THEN NEXT
5090 IF F(A)=4 OR F(A)=7 THEN 5260
5100 FOR A=ZK+B*10 TO MW STEP B*10
5110 IF F(A)=0 THEN NEXT
5120 IF F(A)>=5 AND F(A)<=7 THEN 5260
5130 FOR A=ZK+B*9 TO MW STEP B*9
5140 IF F(A)=0 THEN NEXT
5150 IF F(A)=4 OR F(A)=7 THEN 5260
5160 FOR A=1 TO 4
5170 S=ZK+FA(A)*B
5180 IF S<11 OR S>88 THEN 5200
5190 IF F(S)=3 THEN 5260
5200 NEXT A
5210 IF B=-1 THEN B=1 : GOTO 5030
5220 IF F(ZK-9)=1 OR F(ZK+11)=1 THEN 5260
5230 IF F(ZK-9)=2 OR F(ZK+11)=2 THEN 5260
5240 GOTO 5320
5250 REM *****
5260 REM ** DECKUNG VON FIGUREN/ENDE **
5270 REM *****
```

## **Erläuterungen zum nebenstehenden Programmausschnitt (26):**

In diesem Abschnitt wird untersucht, ob eine (vorläufig) gezogene Figur, die sich in einer Bedrohungssituation befindet, gedeckt ist. Falls Deckung besteht, ist der Zug akzeptiert. Andernfalls setzt der Programmlauf mit einem um 1 erhöhten Wert für BZ die Zugsuche fort.

---

**Programmausschnitt (27):**

```
5280 IF F(ZK)=1 OR F(ZK)=5 OR F(ZK)=8 TH
EN F(ZK)=F(ZK)+1
5290 IF F(ZK)=2 AND ZK-INT(ZK/10)*10=1 T
HEN F(ZK)=7
5300 PRINT TAB(14) "0"
5310 RETURN
5320 IF PA=1 THEN PA=0 : GOTO 5300
5330 F(SK)=F(ZK) : F(ZK)=FZ : BZ=BZ+1
5340 IF RO=0 THEN 4210
5350 F(58)=8 : F(58+RO*20)=0
5360 RO=0
5370 GOTO 4210
5380 REM *****
5390 REM ***** COMPUTERZUG/ENDE *****
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (27):

Nachdem ein Zug akzeptiert worden ist, muß u. U. die ~~die~~ Zahl geändert werden, welche die soeben gezogene Figur repräsentierte (Zeile 5280), nämlich beim Bauern ggf. von 1 zu 2, beim Turm ggf. von 5 zu 6 und beim König ggf. von 8 zu 9 (vgl. hierzu die Ausführungen auf Seite 2-04!). Außerdem wird ein Bauer des Rechners, der die Grundlinie des Spielers erreicht hat, automatisch in eine Dame umgewandelt (Zeile 5290). Dann geben wir eine "0" aus (Zeile 5300), um dem "Eingeweihten" zu signalisieren, daß der Rechner seinen Zug gefunden hat:

```
        Bitte warten  
    Stellungsanalyse  
        0 bis zum Zug
```

Da der Zug des Rechners nunmehr endgültig abgeschlossen ist, erfolgt eine Rückkehr zum SPIELTEIL (Zeile 5310).

Wenn aber ein Zug in den schon erläuterten Programmausschnitten verworfen worden war, erfolgte ein Sprung nach Zeile 5320 dieses Abschnitts, damit der Zug rückgängig gemacht wird (Zeile 5330 - 5370). Allerdings überprüfen wir vorher, ob  $PA = 1$  ist (Zeile 5320), um auf diese Weise ggf. festzustellen, daß der Rechner schachmatt ist.

---

**Programmausschnitt (28):**

```
5400 REM *****
5410 REM ***** ZUGERZEUGER *****
5420 REM *****
5430 ON F(SK)-1 GOTO 5510,5710,5810,5920
,5920,5810,6030,6030
5440 REM
5450 REM ***** BAUER 2-SCHRITT *****
5460 IF F(SK-2)<>0 THEN 5510
5470 IF F(SK-1)<>0 THEN 5510
5480 ZK=SK-2
5490 F(SK)=2 : GOSUB 6590 : F(SK)=1
5500 REM
5510 REM ***** BAUER ALLGEMEIN *****
5520 FS=F(SK) : F(SK)=2
5530 IF F(SK-11)>=0 THEN 5580
5540 ZK=SK-11
5550 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5560 GOSUB 6590
5570 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5580 IF F(SK-1)<>0 THEN 5630
5590 ZK=SK-1
5600 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5610 GOSUB 6590
5620 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5630 IF F(SK+9)>=0 THEN 5680
5640 ZK=SK+9
5650 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5660 GOSUB 6590
5670 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5680 F(SK)=FS
5690 GOTO 6100
5700 REM
```



## Erläuterungen zum nebenstehenden Programmausschnitt (28):

Wir zitieren hier die Ausführungen von Seite 2-67:

"Wenn der Rechner eine ihm gehörende Figur gefunden hat ..., erfolgt ... der Aufruf einer Subroutine. In ihr werden für die gefundene Figur alle möglichen Züge erzeugt, bewertet (durch Aufruf einer "Sub-Subroutine") und gespeichert."

Die oben angesprochene Subroutine beginnt mit dem Programmausschnitt (28).

Zu Beginn (Zeile 5430) verzweigt das Programm, abhängig von der zu ziehenden Figur, an verschiedene Stellen, wo die figurespezifischen Züge erzeugt werden sollen, die wir nun gem. der vorgesehenen Reihenfolge erläutern wollen.

Zuerst wird überprüft, ob mit dem Bauern überhaupt ein Doppelschritt möglich ist (Zeile 5460 - 5470). Wenn das Zielfeld (Zeile 5460) und jenes zwischen Start- und Zielfeld (Zeile 5470) leer sind, berechnen wir in Zeile 5480 die Zielkoordinate, um anschließend die den Bauern in Grundposition repräsentierende Ziffer von 1 in 2 umzuwandeln. Mit dem Aufruf der ab Zeile

---

**Programmausschnitt (28):**

```
5400 REM *****
5410 REM ***** ZUGERZEUGER *****
5420 REM *****
5430 ON F(SK)-1 GOTO 5510,5710,5810,5920
,5920,5810,6030,6030
5440 REM
5450 REM ***** BAUER 2-SCHRITT *****
5460 IF F(SK-2)<>0 THEN 5510
5470 IF F(SK-1)<>0 THEN 5510
5480 ZK=SK-2
5490 F(SK)=2 : GOSUB 6590 : F(SK)=1
5500 REM
5510 REM ***** BAUER ALLGEMEIN *****
5520 FS=F(SK) : F(SK)=2
5530 IF F(SK-11)>=0 THEN 5580
5540 ZK=SK-11
5550 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5560 GOSUB 6590
5570 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5580 IF F(SK-1)<>0 THEN 5630
5590 ZK=SK-1
5600 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5610 GOSUB 6590
5620 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5630 IF F(SK+9)>=0 THEN 5680
5640 ZK=SK+9
5650 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5660 GOSUB 6590
5670 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5680 F(SK)=FS
5690 GOTO 6100
5700 REM
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (28) / Forts.:

6590 beginnenden "Sub-Subroutine" geschieht dort die Durchführung, Bewertung und Rückgängigmachung des Bauernzugs, so daß im Anschluß an die Rückkehr nach Zeile 5490 die den Bauern darstellende Ziffer wieder in 1 geändert werden muß. Unabhängig davon, ob eine Bewertung des Doppelschritts erfolgte, ist immer auch der Einzelschritt und das Schrägschlagen einer Analyse zu unterziehen (Zeile 5510 - 5690); außerdem ist ggf. der die Grundlinie des Spielers erreichende Bauer in eine Dame zu verwandeln (Zeilen 5550, 5600, 5650), diesen Zug zu bewerten ("Sub-Subroutine" ab Zeile 6590) und anschließend wieder rückgängig zu machen (Zeilen 5570, 5620, 5670).

---

**Programmausschnitt (29):**

```
5710 REM ***** SPRINGER *****
5720 FOR B1=-1 TO 1 STEP 2
5730 FOR A1=1 TO 4
5740 ZK=SK+B1*FA(A1)
5750 IF ZK<0 THEN ZK=ZK+100
5760 IF ZK>99 THEN ZK=ZK-100
5770 IF F(ZK)<=0 THEN GOSUB 6590
5780 NEXT A1,B1
5790 GOTO 6100
5800 REM
5810 REM ***** LAEUFER/DAME *****
5820 FOR B1=-1 TO 1 STEP 2
5830 FOR C1=5 TO 6
5840 FOR A1=SK+B1*FA(C1) TO 50+B1*39 STE
P B1*FA(C1)
5850 IF F(A1)<=0 THEN 5870
5860 A1=50+B1*39 : GOTO 5890
5870 ZK=A1 : GOSUB 6590
5880 IF F(A1)<0 THEN 5860
5890 NEXT A1,C1,B1
5900 IF F(SK)<>7 THEN 6100
5910 REM
5920 REM ***** TURM/DAME *****
5930 FOR B1=-1 TO 1 STEP 2
5940 FOR C1=7 TO 8
5950 FOR A1=SK+B1*FA(C1) TO 50+B1*39 STE
P B1*FA(C1)
5960 IF F(A1)<=0 THEN 5980
5970 A1=50+B1*39 : GOTO 6000
5980 ZK=A1 : GOSUB 6590
5990 IF F(A1)<0 THEN 5970
6000 NEXT A1,C1,B1
6010 GOTO 6100
6020 REM
```

```
6030 REM ***** KOENIG *****
6040 FOR B1=-1 TO 1 STEP 2
6050 FOR A1=5 TO 8
6060 ZK=SK+B1*FA(A1)
6070 IF F(ZK)>0 THEN 6090
6080 GOSUB 6590
6090 NEXT A1,B1
6100 RETURN
6110 REM *****
6120 REM ***** ZUGERZEUGER/ENDE *****
```

## Erläuterungen zum nebenstehenden / obenstehenden Programm- ausschnitt (29):

In den Zeilen 5710 - 5790 werden alle für einen Springer möglichen Züge erzeugt. Jetzt erweist es sich als besonders nützlich, daß wir hierfür in Ausschnitt (7) die Feldabstände in FA(..) definierten, weil so die möglichen Zielkoordinaten des Springers leicht zu berechnen sind (Zeile 5740). Dabei ist durch die Kontrollen der Zeilen 5750 - 5760 sichergestellt, daß die Figur das "interne Brett" nicht "verlassen" kann. Es wird jeder Springer-Zug einer Bewertung unterzogen, bei dem das Zielfeld nicht mit einer eigenen Figur belegt ist (Zeile 5770).

Die obigen Aussagen zum Springer gelten in ähnlicher Weise für den Läufer, den Turm (damit auch für die Dame) und für den König. Die Unterschiede resultieren in erster Linie aus den anderen Feldabständen.

---

**Programmausschnitt (30):**

```
6130 REM *****
6140 REM *** ROCHADE MIT KONTROLLE ***
6150 REM *****
6160 RO=-1 : B=58
6170 IF F(53+RO*35)<>5 THEN 6530
6180 IF F(58+RO*10)<>0 THEN 6530
6190 IF F(58+RO*20)<>0 THEN 6530
6200 IF RO=-1 AND F(28)<>0 THEN 6530
6210 FOR A=B-1 TO 31 STEP -1
6220 IF F(A)=0 THEN NEXT
6230 IF F(A)=-6 OR F(A)=-7 THEN 6530
6240 IF F(B-1)=-9 THEN 6530
6250 FOR A=B-11 TO (13)STEP -11
6260 IF F(A)=0 THEN NEXT
6270 IF F(A)=-4 OR F(A)=-7 THEN 6530
6280 IF F(B-11)=-9 THEN 6530
6290 FOR A=B+10 TO 88 STEP 10
6300 IF F(A)=0 THEN NEXT
6310 IF F(A)=-6 OR F(A)=-7 THEN 6530
6320 IF F(B+10)=-9 THEN 6530
6330 FOR A=B-10 TO 18 STEP -10
6340 IF F(A)=0 THEN NEXT
6350 IF F(A)=-6 OR F(A)=-7 THEN 6530
6360 IF F(B-10)=-9 THEN 6530
6370 FOR A=B+9 TO 87 STEP 9
6380 IF F(A)=0 THEN NEXT
6390 IF F(A)=-4 OR F(A)=-7 THEN 6530
6400 IF F(B+ 9)=-9 THEN 6530
6410 IF F(B+19)=-3 THEN 6530
6420 IF F(B+ 8)=-3 THEN 6530
6430 IF F(B-12)=-3 THEN 6530
6440 IF F(B-21)=-3 THEN 6530
6450 IF F(B+9)=-2 OR F(B-11)=-2 THEN 6530
6460 IF B=58+RO*20 THEN 6480
```

```
6470 B=B+RO*10 : GOTO 6210
6480 F(58)=0 : F(58+RO*20)=9
6490 SK=53+RO*35 : ZK=58+RO*10
6500 GOSUB 6590
6510 F(58)=8 : F(58+RO*20)=0
6520 ZK(ZN)=RO : BS(ZN)=BS(ZN)+100
6530 IF RO=-1 THEN RO=1 : GOTO 6170
6540 RO=0
6550 RETURN
6560 REM *****
6570 REM ***** ROCHADE/ENDE *****
```

## Erläuterungen zum nebenstehenden / obenstehenden Programm- ausschnitt (30):

Zu Beginn des Abschnitts COMPUTERZUG (vgl. Programmausschnitt (22) auf Seite 2-66 f.) galt die allererste Untersuchung der Frage, ob der König noch rochadefähig ist (Zeile 4060). Im Bejahungsfall wurde in den hier zu erläuternden Abschnitt verzweigt. In ihm finden verschiedene Überprüfungen statt, und zwar zuerst für die große (RO = -1; Zeile 6160) und dann für die kleine Rochade (RO = 1; Zeile 6530), nämlich

- ob der zu rochierende Turm noch nicht bewegt wurde (Zeile 6170 / die analoge Überprüfung des Königs erfolgte bereits in Zeile 4060),

## Erläuterungen zum Programmausschnitt (30) auf Seite 2-92 f. / Forts.:

- ob die Felder zwischen den beiden zu rochierenden Figuren (König / Turm) schon leer sind (Zeile 6180 - 6200) und
- ob die vom König zu überstreichenden Felder unbedroht sind (Zeile 6210 - 6470).

Um die oben erwähnte Überprüfung, ob die vom König zu überstreichenden Felder unbedroht sind, durchführen zu können, setzen wir in Zeile 6160 die Variable B auf 58 ( $\hat{=}$  Position des Königs). Mit diesem Wert untersuchen wir zunächst, ob sich der König in unbedrohter Position befindet. Eine Rochade ist nämlich nicht zulässig, wenn der König im Schach steht (Zeile 6210 - 6450).

Die Untersuchung der Position 68/78 (bei der kleinen Rochade) bzw. 48/38 (bei der großen Rochade) wird ermöglicht durch die Veränderung von B in Zeile 6470. Die Abfrage in Zeile 6460 stellt sicher, daß nur die Positionen 58/68/78 ( $\hat{=}$  kleine Rochade) bzw. 58/48/38 ( $\hat{=}$  große Rochade) überprüft werden. Im Falle der Bedrohungssituation für eines der aufgeführten Felder verzweigt das Programm nach Zeile 6530, um dann mit den Analysen für die kleine

---



Rochade zu beginnen. Jede festgestellte Möglichkeit einer Rochade führt zu ihrer - zunächst nur internen - Ausführung und Bewertung (Zeile 6480 - 6510). Während wir bei einem "normalen" Zug in ZK(..) die Zielkoordinate speichern, wird dort bei einer kleinen Rochade der Wert 1 und bei einer großen Rochade der Wert -1 abgelegt (Zeile 6520). Darüber hinaus erhöhen wir aus spielstrategischen Gründen die Bewertung einer Rochade um zusätzliche 100 Punkte (Zeile 6520).

---

**Programmausschnitt (31):**

```
6580 REM *****
6590 REM ***** WERTUNG *****
6600 REM *****
6610 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
6620 ZN=ZN+1 : SW=0
6630 FOR P=11 TO 88
6640 IF F(P)=0 OR F(P)=10 THEN 7810
6650 IF F(P)>0 THEN 7240
6660 REM *****
6670 REM ***** WERTUNG/GEGNER *****
6680 REM *****
6690 SW=SW-3*FW(-F(P))
6700 ON -F(P)-1 GOTO 6720,6810,6900,7010
,7010,6900,7120,7120
6710 REM
6720 REM ***** BAUER ALLGEMEIN *****
6730 IF F(P+1)=0 THEN SW=SW-3
6740 IF F(P-9)>0 THEN SW=SW-FW(F(P-9))
6750 IF F(P+11)>0 THEN SW=SW-FW(F(P+11))
6760 IF F(P)=-2 THEN 7810
6770 REM ***** BAUER 2-SCHRITT *****
6780 IF F(P+1)<>0 THEN 6800
6790 IF F(P+2)=0 THEN SW=SW-3
6800 GOTO 7810
```

## Erläuterungen zum nebenstehenden Programmausschnitt (31):

Neben der Subroutine ZUGERZEUGER (ab Zeile 5400) gehört die "Sub-Subroutine" WERTUNG (ab Zeile 6580) zu einem der wichtigsten Teile des Programms. Hier wird jeder Zug, der im Abschnitt ZUGERZEUGER berechnet wurde, zunächst - nur vorläufig - ausgeführt (Zeile 6610). Daraufhin erhöhen wir die Variable ZN (Zugnummer) um 1 und setzen SW (Stellungwert) auf 0 (Zeile 6620). Es beginnt dann eine sehr umfangreiche FOR .. NEXT-Konstruktion (Zeile 6630), die bis zu Zeile 7810 reicht. In ihr werden mit Hilfe der Zählvariablen P (Position) alle Felder des Spielfelds analysiert. Ist das untersuchte Feld  $F(P)$  leer ( $= 0$ ) oder im Rahmen mit einer Scheinfigur ( $= 10$ ) belegt (wie z. B.  $F(19)$ ; vgl. die Abbildung 6 auf Seite 2-13!), erfolgt unverzüglich eine Verzweigung (Zeile 6640) zur NEXT-Anweisung in Zeile 7810, so daß dann das nächste Feld analysiert werden kann.

Wurde hingegen ein Feld mit einer Figur gefunden (Bedingung in Zeile 6640 nicht erfüllt!), untersuchen wir, ob es sich um eine solche des Spielers ( $F(P) < 0$ ) oder des Rechners ( $F(P) > 0$ ) handelt, um abhängig davon die Verarbeitung mit den unmittelbar folgenden Instruktionen ( $\hat{=}$  Figur des Spielers) oder mit jenen ab Zeile 7240 ( $\hat{=}$  Figur des Rechners) fortzusetzen.

---

**Programmausschnitt (31):**

```
6580 REM *****
6590 REM ***** WERTUNG *****
6600 REM *****
6610 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
6620 ZN=ZN+1 : SW=0
6630 FOR P=11 TO 88
6640 IF F(P)=0 OR F(P)=10 THEN 7810
6650 IF F(P)>0 THEN 7240
6660 REM *****
6670 REM ***** WERTUNG/GEGNER *****
6680 REM *****
6690 SW=SW-3*FW(-F(P))
6700 ON -F(P)-1 GOTO 6720,6810,6900,7010
,7010,6900,7120,7120
6710 REM
6720 REM ***** BAUER ALLGEMEIN *****
6730 IF F(P+1)=0 THEN SW=SW-3
6740 IF F(P-9)>0 THEN SW=SW-FW(F(P-9))
6750 IF F(P+11)>0 THEN SW=SW-FW(F(P+11))
6760 IF F(P)=-2 THEN 7810
6770 REM ***** BAUER 2-SCHRITT *****
6780 IF F(P+1)<>0 THEN 6800
6790 IF F(P+2)=0 THEN SW=SW-3
6800 GOTO 7810
```

---

## Erläuterungen zum nebenstehenden Programmausschnitt (31) / Forts.:

Für den Zweck unserer Erläuterungen unterstellen wir, daß es sich bei der gefundenen Figur um eine solche des Spielers handelt. Um den Wert dieser Figur für diese zu analysierende Stellung zu ermitteln, indizieren wir FW (Figurenwert) mit dem Inhalt des Feldes  $F(P)$  und fügen ein negatives Vorzeichen hinzu ( $-F(P)$ ), weil wir statt einer negativen Zahl eine positive als Index benötigen. Wir erhalten so den einfachen Figurenwert, den wir mit 3 multiplizieren:  $3*FW(-F(P))$ . Der Multiplikator 3 dient dem Zweck, eine auf dem Brett existierende Figur wesentlich höher zu bewerten als eine solche, die geschlagen werden könnte. Selbstverständlich führt ein anderer Multiplikator zu einer anderen Spieleigenschaft des Programms. Wir subtrahieren die mit dem Ausdruck  $3*FW(-F(P))$  ermittelten Punkte von SW (Zeile 6690), während die Punkte für die Rechner-Figuren zu addieren sein werden.

Neben der Vergabe von Punkten für die bloße Existenz einer Figur sind jetzt außerdem für den taktischen Wert jeder Figur (Wieviel Felder beherrscht die untersuchte Figur? Welche Figuren könnten von der untersuchten Figur geschlagen werden?) Punkte zu vergeben. Zu diesem Zweck verzweigt das Programm (Zeile 6700),

---

**Programmausschnitt (31):**

```
6580 REM *****
6590 REM ***** WERTUNG *****
6600 REM *****
6610 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
6620 ZN=ZN+1 : SW=0
6630 FOR P=11 TO 88
6640 IF F(P)=0 OR F(P)=10 THEN 7810
6650 IF F(P)>0 THEN 7240
6660 REM *****
6670 REM ***** WERTUNG/GEGNER *****
6680 REM *****
6690 SW=SW-3*FW(-F(P))
6700 ON -F(P)-1 GOTO 6720,6810,6900,7010
,7010,6900,7120,7120
6710 REM
6720 REM ***** BAUER ALLGEMEIN *****
6730 IF F(P+1)=0 THEN SW=SW-3
6740 IF F(P-9)>0 THEN SW=SW-FW(F(P-9))
6750 IF F(P+11)>0 THEN SW=SW-FW(F(P+11))
6760 IF F(P)=-2 THEN 7810
6770 REM ***** BAUER 2-SCHRITT *****
6780 IF F(P+1)<>0 THEN 6800
6790 IF F(P+2)=0 THEN SW=SW-3
6800 GOTO 7810
```

## Erläuterungen zum nebenstehenden Programmausschnitt (31) / Forts.:

abhängig von der Figurenart, an verschiedene Stellen,  
um dort

- für jedes leere, von der jeweils analysierten Figur besetzbare Feld 3 Punkte von SW abziehen und
  - für jede schlagbare Figur den (1-fachen) Figurenwert von SW zu subtrahieren.
-

**Programmausschnitt (32):**

```
6810 REM ***** SPRINGER *****
6820 A=-1
6830 FOR B=1 TO 4
6840 SP=P+A*FA(B)
6850 IF SP<11 OR SP>88 THEN 6870
6860 IF F(SP)>=0 THEN SW=SW-FW(F(SP))
6870 NEXT B
6880 IF A=-1 THEN A=1 : GOTO 6830
6890 GOTO 7810
6900 REM ***** LAEUFER/DAME *****
6910 B=-1
6920 FOR C=5 TO 6
6930 A=P+B*FA(C)
6940 IF F(A)>=0 THEN SW=SW-FW(F(A))
6950 IF F(A)<>0 THEN 6980
6960 A=A+B*FA(C)
6970 GOTO 6940
6980 NEXT C
6990 IF B=-1 THEN B=1 : GOTO 6920
7000 IF F(P)<>-7 THEN 7810
7010 REM ***** TURM/DAME *****
7020 B=-1
7030 FOR C=7 TO 8
7040 A=P+B*FA(C)
7050 IF F(A)>=0 THEN SW=SW-FW(F(A))
7060 IF F(A)<>0 THEN 7090
7070 A=A+B*FA(C)
7080 GOTO 7050
7090 NEXT C
7100 IF B=-1 THEN B=1 : GOTO 7030
7110 GOTO 7810
```



```
7120 REM ***** KOENIG *****
7130 B=-1
7140 FOR C=5 TO 8
7150 SP=F(P+B*FA(C))
7160 IF SP<0 AND RO=0 THEN 7180
7170 SW=SW-FW(ABS(SP))
7180 NEXT C
7190 IF B=-1 THEN B=1 : GOTO 7140
7200 GOTO 7810
7210 REM *****
7220 REM **** WERTUNG/GEGNER/ENDE ****
```

## Erläuterungen zum nebenstehenden / obenstehenden Programm- ausschnitt (32):

Während wir beim Bauern jedes von ihm besetzbare Feld in einer eigenen Programmzeile bewerten (vgl. Seite 2-100, Zeile 6730 - 6790), erfolgt die Durchführung dieser Programmfunktion für die anderen Figuren ausnahmslos innerhalb von FOR .. NEXT- und IF .. THEN-Schleifen. Hierbei ist allerdings wieder zu beachten, daß die Analyse für die Dame durch die kombinierte Verwendung der Routinen für den Läufer und für den Turm geschieht. Die Bewertung der jeweiligen Figur ist abgeschlossen mit einem Sprung zur NEXT-Anweisung in Zeile 7810, durch welche die Suche nach einer weiteren Figur veranlaßt wird, und zwar so lange, bis das letzte Feld (88) erreicht ist.

**Programmausschnitt (33):**

```
7230 REM *****
7240 REM ***** WERTUNG/COMPUTER *****
7250 REM *****
7260 SW=SW+3*FW(F(P))
7270 ON F(P)-1 GOTO 7290,7390,7480,7590
,7590,7480,7700,7700
7280 REM
7290 REM ***** BAUER ALLGEMEIN *****
7300 IF F(ZK)<=2 THEN SW=SW+3
7310 IF F(P-1)=0 THEN SW=SW+3
7320 IF F(P+9)<0 THEN SW=SW+FW(-F(P+9))
7330 IF F(P-11)<0 THEN SW=SW+FW(-F(P-11))
7340 IF F(P)=2 THEN 7810
7350 REM ***** BAUER 2-SCHRITT *****
7360 IF F(P-1)<>0 THEN 7380
7370 IF F(P-2)=0 THEN SW=SW+3
7380 GOTO 7810
7390 REM ***** SPRINGER *****
7400 A=-1
7410 FOR B=1 TO 4
7420 SP=P+A*FA(B)
7430 IF SP<11 OR SP>88 THEN 7450
7440 IF F(SP)<=0 THEN SW=SW+FW(-F(SP))
7450 NEXT B
7460 IF A=-1 THEN A=1 : GOTO 7410
7470 GOTO 7810
7480 REM ***** LAEUFER/DAME *****
7490 B=-1
7500 FOR C=5 TO 6
7510 A=P+B*FA(C)
7520 IF F(A)<=0 THEN SW=SW+FW(-F(A))
7530 IF F(A)<>0 THEN 7560
7540 A=A+B*FA(C)
7550 GOTO 7520
7560 NEXT C
```

```
7570 IF B=-1 THEN B=1 : GOTO 7500
7580 IF F(P)<>7 THEN 7810
7590 REM ***** TURM/DAME *****
7600 B=-1
7610 FOR C=7 TO 8
7620 A=P+B*FA(C)
7630 IF F(A)<=0 THEN SW=SW+FW(-F(A))
7640 IF F(A)<>0 THEN 7670
7650 A=A+B*FA(C)
7660 GOTO 7630
7670 NEXT C
7680 IF B=-1 THEN B=1 : GOTO 7610
7690 GOTO 7810
7700 REM ***** KOENIG *****
7710 B=-1
7720 FOR C=5 TO 8
7730 SP=F(P+B*FA(C))
7740 IF SP>0 AND RO=0 THEN 7760
7750 SW=SW+FW(ABS(SP))
7760 NEXT C
7770 IF B=-1 THEN B=1 : GOTO 7720
7780 REM *****
7790 REM *** WERTUNG/COMPUTER/ENDE ***
7800 REM *****
7810 NEXT P
```

## Erläuterungen zum nebenstehenden / obenstehenden Programm- ausschnitt (33):

In diesem Abschnitt WERTUNG/COMPUTER werden nahezu die gleichen Prozeduren ausgeführt wie im vorangegangenen (WERTUNG/GEGNER). Der wesentliche Unterschied besteht darin, daß die ermittelten Punkte zu SW addiert werden.

---

**Programmausschnitt (34):**

```
7820 F(SK)=F(ZK) : F(ZK)=FZ
7830 BS(ZN)=SW : SK(ZN)=SK : ZK(ZN)=ZK
7840 RETURN
7850 REM *****
7860 REM ***** WERTUNG/ENDE *****
7870 REM *****
7880 REM ***** 5 ZEILEN LOESCHEN *****
7890 REM *****
7900 PRINT CHR$(19); ZF$ Loc 1,1
7910 FOR A=1 TO 18 : PRINT : NEXT
7920 FOR A=1 TO 5
7930 PRINT TAB(12) " "
7940 NEXT A 30
7950 FOR A=1 TO 5:PRINT CHR$(145);:NEXT
7960 RETURN
7970 REM *****
7980 REM *** 5 ZEILEN LOESCHEN/ENDE ***
7990 REM *****
8000 REM ***** UNTERPROGRAMME/ENDE *****
8010 REM *****
```

## **Erläuterungen zum nebenstehenden Programmausschnitt (34):**

Der in Zeile 6610 nur vorläufig und für interne Untersuchungszwecke ausgeführte Zug wird nunmehr rückgängig gemacht (Zeile 7820). Anschließend speichern wir den ermittelten Stellungswert (Zeile 7830), der sich aus der Differenz zwischen dem Stellungswert des Gegners und jenem des Computers ergeben hat. (Ein positiver Wert von SW bedeutet, daß die damit jeweils charakterisierte Stellung des Rechners besser ist als jene des Gegners; ein negativer Wert von SW kennzeichnet hingegen eine Überlegenheit beim Gegner.)

Außerdem wird in derselben Programmzeile die Start- und Zielcoordinate des bewerteten Zuges gespeichert. Danach erfolgt mit einer RETURN-Anweisung ein Rücksprung zum Abschnitt ZUGERZEUGER.

Mit den Befehlen der Zeilen 7900 - 7940 löschen wir die erläuternden Ausgaben in den unteren fünf Bildschirmzeilen und positionieren den Cursor für die nächste Textausgabe zu Beginn der 20. Zeile.

---



### 3. Vollständige Liste des BASIC-Schachprogramms

```
10 REM *****
20 REM *
30 REM * SCHACHPROGRAMM IN BASIC *
40 REM *
50 REM * FUER DEN RECHNER *
60 REM *
70 REM * COMMODORE 64 *
80 REM *
90 REM *****
100 REM
110 REM *****
120 REM *
130 REM * BEDIENUNGSHINWEISE: *
140 REM * ===== *
150 REM * DIES IST EIN BASIC-SCHACH- *
160 REM * PROGRAMM. ES WIRD NACH DEN *
170 REM * ALLGEMEINEN SCHACHREGELN *
180 REM * GESPIELT. WENN SIE Z.B. VON *
190 REM * C3 NACH E5 ZIEHEN WOLLEN, *
200 REM * DANN GEBEN SIE EINFACH *
210 REM * C3 : E5 +RETURN EIN. FALLS *
```

---

```

220 REM * SIE SICH VERTIPPT HABEN,      *
230 REM * DRUECKEN SIE DIE DEL-TASTE.  *
240 REM * DANN KOENNEN SIE DEN ZUG NEU*
250 REM * EINGEBEN. FUER DIE KLEINE   *
260 REM * ROCHADE MUSS NUR DIE O-TASTE*
270 REM * +RETURN BETAETIGT, UND FUER *
280 REM * DIE GROSSE ROCHADE MUSS DIE *
290 REM * O-TASTE ZWEIMAL (+RETURN)   *
300 REM * BETAETIGT WERDEN.           *
310 REM *                               *
320 REM *****
330 REM
340 REM *****
350 REM ***** INITIALISIERUNG *****
360 REM *****
370 DIM FW(10) :DIM F(99) :DIM F$(18)
380 DIM BS(132):DIM SK(132):DIM ZK(132)
390 DIM FA(8)
400 REM
410 REM ***** FARBZUWEISUNGEN *****
420 RF=0 : HF=0 : REM ** SCHWARZ
430 ZF$=CHR$(5) : REM ** WEISS
440 FH$=CHR$(154) : REM ** HELLBLAU
450 FD$=CHR$(31) : REM ** DUNKELBLAU

```



```
460 RO$=CHR$(18) : REM ** REVERSE ON
470 RF$=CHR$(146) : REM ** REVERSE OFF
480 REM
490 REM ***** FIGURENWERTE *****
500 FOR A=0 TO 9 : READ FW(A) : NEXT
510 DATA 3,10,10,30,30,50,50,110,999,999
520 REM
530 REM ***** SCHEINFIGUREN *****
540 FOR A= 1 TO 9:F(A)=10:NEXT
550 FOR A=90 TO 98:F(A)=10:NEXT
560 FOR A= 0 TO 80 STEP 10:F(A)=10:NEXT
570 FOR A=19 TO 99 STEP 10:F(A)=10:NEXT
580 REM
590 REM ***** ECHETE FIGUREN *****
600 FOR A=18 TO 88 STEP 10
610 READ F(A) : NEXT A
620 FOR A=17 TO 87 STEP 10:F(A)= 1:NEXT
630 FOR A=12 TO 82 STEP 10:F(A)=-1:NEXT
640 FOR A=11 TO 81 STEP 10
650 READ F(A) : NEXT A
660 DATA 5, 3, 4, 7, 8, 4, 3, 5
670 DATA -5,-3,-4,-7,-8,-4,-3,-5
680 REM
```

---

```
690 REM ***** DRUCKZEICHEN FIGUR *****
700 FOR A=0 TO 18
710 READ DF : F$(A)=CHR$(DF)
720 NEXT A
730 DATA 203, 203, 196, 212, 212
740 DATA 204, 211, 194, 194, 32
750 DATA      66, 66, 83, 76
760 DATA 84, 84, 68, 75, 75
770 REM
780 REM ***** FELDABSTAENDE *****
790 FOR A=1 TO 8 : READ FA(A) : NEXT
800 DATA 21, 19, 12, 8, 11, 9, 10, 1
810 REM
820 REM ***** GROSSBUCHSTABE *****
830 B$=CHR$(194) : I$=CHR$(201)
840 O$=CHR$(207) : P$=CHR$(208)
850 S$=CHR$(211) : Z$=CHR$(218)
860 REM
870 REM ***** SCHACHBRETT *****
880 POKE 53280, RF : POKE 53281, HF
890 PRINT CHR$(147);
900 PRINT CHR$(14) : REM KLEINSCHRIFT
910 FOR A=1 TO 4
920 PRINT TAB(8) " ";
```

```
930 FOR B=1 TO 2 : FOR C=1 TO 4
940 PRINT RO$; FH$; "  "; FD$; "  ";
950 NEXT C
960 PRINT:PRINT TAB(8) " ";
970 NEXT B
980 FOR B=1 TO 2 : FOR C=1 TO 4
990 PRINT RO$; FD$; "  "; FH$; "  ";
1000 NEXT C
1010 PRINT:PRINT TAB(8) " ";
1020 NEXT B,A
1030 PRINT CHR$(19);
1040 PRINT TAB(9) " "; ZF$;
1050 FOR A=193 TO 200
1060 PRINT CHR$(A); "  ";
1070 NEXT A
1080 PRINT
1090 FOR A=8 TO 1 STEP -1
1100 PRINT:PRINT TAB(6) A : NEXT A
1110 REM *****
1120 REM **** INITIALISIERUNG/ENDE ****
1130 REM *****
1140 REM ***** SPIELTEIL *****
1150 REM *****
1160 REM
```

---

```
1170 REM *****
1180 GOSUB 1470:REM ** STEINE SETZEN **
1190 GOSUB 1680:REM ** SPIELEREINGABE**
1200 GOSUB 3790:REM ** ZUGANGABE **
1210 GOSUB 1470:REM ** STEINE SETZEN **
1220 GOSUB 3960:REM ** COMPUTERZUG **
1230 IF MA=1 OR PA=1 THEN 1270 : REM **
1240 GOSUB 3790:REM ** ZUGANGABE **
1250 REM *****
1260 GOTO 1180
1270 GOSUB 7880
1280 IF PA <> 1 THEN 1340
1290 PRINT
1300 PRINT TAB(17) P$;
1310 PRINT "ATT !"
1320 PRINT TAB(17) "====="
1330 GOTO 1400
1340 PRINT TAB(14) S$;
1350 PRINT "CHACH MATT!"
1360 PRINT
1370 PRINT TAB(11) S$; "IE HABEN";
1380 PRINT " GEWONNEN"
1390 PRINT TAB(11)"===== ";
1400 END
```

```
1410 REM *****
1420 REM ***** SPIELTEIL/ENDE *****
1430 REM *****
1440 REM ***** UNTERPROGRAMME *****
1450 REM *****
1460 REM
1470 REM ***** STEINE SETZEN *****
1480 PRINT CHR$(19)
1490 FOR A=0 TO 6 STEP 2
1500 PRINT:PRINT TAB(8) "";
1510 FOR B=18-A TO 88-A STEP 20
1520 PRINT RO$; FH$; " ";
1530 PRINT RO$; FH$; F$(F(B)+9); " ";
1540 PRINT RO$; FD$; " ";
1550 PRINT RO$; FD$; F$(F(B+10)+9); " ";
1560 NEXT B
1570 PRINT:PRINT:PRINT TAB(8) "";
1580 FOR B=18-A TO 88-A STEP 20
1590 PRINT RO$; FD$; " ";
1600 PRINT RO$; FD$; F$(F(B-1)+9); " ";
1610 PRINT RO$; FH$; " ";
1620 PRINT RO$; FH$; F$(F(B+9)+9); " ";
1630 NEXT B
1640 PRINT
```

---

```
1650 NEXT A
1660 RETURN
1670 REM *****
1680 REM ***** SPIELEREINGABE *****
1690 REM *****
1700 REM
1710 REM ***** ZUGEINGABE *****
1720 GOSUB 7880
1730 PRINT
1740 PRINT TAB(12) I$;"HR "; Z$;"UG: ";
1750 PRINT RO$; " "; RF$;
1760 PRINT "- : -- ";
1770 FOR A=1 TO 8:PRINT CHR$(157);:NEXT
1780 GET IN$ : IF IN$ = "" THEN 1780
1790 IF IN$ = CHR$(20) THEN 1890
1800 IF IN$="O" AND F(51)=-8 THEN 2130
1810 IF IN$<"A" OR IN$>"H" THEN 1780
1820 PRINT CHR$(ASC(IN$)+128);
1830 PRINT RO$; " "; RF$; CHR$(157);
1840 GET IP$ : IF IP$ = "" THEN 1840
1850 IF IP$ = CHR$(20) THEN 1890
1860 IF IP$<"1" OR IP$>"8" THEN 1840
1870 SK = 10*(ASC(IN$)-64)+VAL(IP$)
1880 IF F(SK) < 0 THEN 1900
```

```
1890 PRINT CHR$(145) : GOTO 1740
1900 PRINT IP$; " : ";
1910 PRINT RO$; " "; RF$; CHR$(157);
1920 GET IN$ : IF IN$ = "" THEN 1920
1930 IF IN$ = CHR$(20) THEN 1890
1940 IF IN$ <"A" OR IN$ >"H" THEN 1920
1950 PRINT CHR$(ASC(IN$)+128);
1960 PRINT RO$; " "; RF$; CHR$(157);
1970 GET IP$ : IF IP$ = "" THEN 1970
1980 IF IP$ = CHR$(20) THEN 1890
1990 IF IP$ <"1" OR IP$ >"8" THEN 1970
2000 PRINT IP$;
2010 PRINT RO$; " "; RF$; CHR$(157);
2020 GET E$ : IF E$ = "" THEN 2020
2030 IF E$ = CHR$(20) THEN 1890
2040 IF E$ <> CHR$(13) THEN 2020
2050 ZK = 10*(ASC(IN$)-64)+VAL(IP$)
2060 PRINT " ";
2070 IF F(ZK) >= 0 THEN 2690
2080 FOR A=1 TO 3:PRINT CHR$(157);:NEXT
2090 PRINT RO$; " "; RF$; "-";
2100 PRINT CHR$(157); CHR$(157);
2110 GOTO 1920
```

---

```
2120 REM *****
2130 REM *** ROCHADE MIT KONTROLLE ***
2140 REM *****
2150 RO=1
2160 PRINT " "; O$; "-"; O$; " ";
2170 FOR A=1 TO 3:PRINT CHR$(157);:NEXT
2180 PRINT RO$; " "; RF$; CHR$(157);
2190 GET IN$ : IF IN$ = "" THEN 2190
2200 IF IN$ = CHR$(20) THEN 1890
2210 IF IN$ = CHR$(13) THEN 2280
2220 IF IN$ <> "0" THEN 2190
2230 PRINT "-"; O$; : RO=-1
2240 PRINT RO$; " "; RF$; CHR$(157);
2250 GET E$ : IF E$ = "" THEN 2250
2260 IF E$ = CHR$(20) THEN 1890
2270 IF E$ <> CHR$(13) THEN 2250
2280 IF F(46+RO*35) <> -5 THEN 2600
2290 IF F(51+RO*10) <> 0 THEN 2600
2300 IF F(51+RO*20) <> 0 THEN 2600
2310 IF RO=-1 AND F(21)<>0 THEN 2600
2320 PRINT " "; : B=51
2330 FOR A=B+1 TO 78
2340 IF F(A)=0 THEN NEXT
2350 IF F(A)=6 OR F(A)=7 THEN 2600
```

---



```
2360 IF F(B+1)=9 THEN 2600
2370 FOR A=B+11 TO 86 STEP 11
2380 IF F(A)=0 THEN NEXT
2390 IF F(A)=4 OR F(A)=7 THEN 2600
2400 IF F(B+11)=9 THEN 2600
2410 FOR A=B+10 TO 81 STEP 10
2420 IF F(A)=0 THEN NEXT
2430 IF F(A)=6 OR F(A)=7 THEN 2600
2440 IF F(B+10)=9 THEN 2600
2450 FOR A=B-10 TO 11 STEP -10
2460 IF F(A)=0 THEN NEXT
2470 IF F(A)=6 OR F(A)=7 THEN 2600
2480 IF F(B-10)=9 THEN 2600
2490 FOR A=B-9 TO 13 STEP -9
2500 IF F(A)=0 THEN NEXT
2510 IF F(A)=4 OR F(A)=7 THEN 2600
2520 IF F(B- 9)=9 THEN 2600
2530 IF F(B-19)=3 THEN 2600
2540 IF F(B- 8)=3 THEN 2600
2550 IF F(B+12)=3 THEN 2600
2560 IF F(B+21)=3 THEN 2600
2570 IF F(B-9)=2 OR F(B+11)=2 THEN 2600
2580 IF B=51+RO*20 THEN 2620
2590 B=B+RO*10 : GOTO 2330
2600 RO=0
```

---

```
2610 GOTO 1890 : REM ROCHADE UNMOEGLICH
2620 REM ***** ROCHADENAUSFUEHRUNG *****
2630 F(51)=0 : F(51+RO*20)=-9
2640 F(46+RO*35)=0 : F(51+RO*10)=-6
2650 GOTO 3720
2660 REM *****
2670 REM ***** ROCHADE/ENDE *****
2680 REM *****
2690 REM ***** EINGABEKONTROLLE *****
2700 REM *****
2710 ON ABS(F(SK)+1) GOTO 2780,2860,2920
,3080,3080,3240,3320,3320
2720 REM
2730 REM ***** BAUER 2-SCHRITT *****
2740 IF ZK-SK<>2 THEN 2780
2750 IF F(SK+1)=0 AND F(ZK)=0 THEN 3370
2760 GOTO 2080
2770 REM
2780 REM ***** BAUER ALLGEMEIN *****
2790 IF SK-ZK<>9 AND ZK-SK<>11 THEN 2820
2800 IF F(ZK)>0 THEN 3370
2810 GOTO 2080
2820 IF ZK-SK<>1 THEN 2080
2830 IF F(ZK)<>0 THEN 2080
2840 GOTO 3370
```

```
2850 REM
2860 REM ***** SPRINGER *****
2870 FOR A=1 TO 4
2880 IF ABS(ZK-SK)=FA(A) THEN 3370
2890 NEXT A
2900 GOTO 2080
2910 REM
2920 REM ***** LAEUFER/DAME *****
2930 DI=ABS(ZK-SK) : V=SGN(ZK-SK)
2940 IF DI/11=INT(DI/11) THEN 2970
2950 IF DI/ 9=INT(DI/ 9) THEN 3020
2960 GOTO 2080
2970 IF DI=11 THEN 3370
2980 FOR A=SK+11*V TO ZK-11*V STEP 11*V
2990 IF F(A)<>0 THEN 2080
3000 NEXT A
3010 GOTO 3370
3020 IF DI=9 THEN 3370
3030 FOR A=SK+9*V TO ZK-9*V STEP 9*V
3040 IF F(A)<>0 THEN 2080
3050 NEXT A
3060 GOTO 3370
3070 REM
```

---

```
3080 REM ***** TURM/DAME *****
3090 DI=ABS(ZK-SK) : V=SGN(ZK-SK)
3100 IF DI<8 THEN 3130
3110 IF DI/10=INT(DI/10) THEN 3180
3120 GOTO 2080
3130 IF DI=1 THEN 3370
3140 FOR A=SK+V TO ZK-V STEP V
3150 IF F(A)<>0 THEN 2080
3160 NEXT A
3170 GOTO 3370
3180 IF DI=10 THEN 3370
3190 FOR A=SK+10*V TO ZK-10*V STEP 10*V
3200 IF F(A)<>0 THEN 2080
3210 NEXT A
3220 GOTO 3370
3230 REM
3240 REM ***** DAME *****
3250 DI=ABS(ZK-SK) : V=SGN(ZK-SK)
3260 IF DI/11=INT(DI/11) THEN 2970
3270 IF DI/10=INT(DI/10) THEN 3180
3280 IF DI/ 9=INT(DI/ 9) THEN 3020
3290 IF DI<8 THEN 3130
3300 GOTO 2080
3310 REM
```

---

```
3320 REM ***** KOENIG *****
3330 DI=ABS(ZK-SK)
3340 IF DI=10 OR DI=11 THEN 3370
3350 IF DI<>1 AND DI<>9 THEN 2080
3360 REM
3370 REM *** INTERNE ZUGAUSFUEHRUNG ***
3380 REM
3390 REM * KONTROLLE KOENIG IN SCHACH **
3400 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
3410 FOR PK=11 TO 88
3420 IF F(PK)=-8 OR F(PK)=-9 THEN 3440
3430 NEXT PK
3440 B=-1
3450 MW=50+B*39
3460 FOR A=PK+B TO MW STEP B
3470 IF F(A)=0 THEN NEXT
3480 IF F(A)>=5 AND F(A)<=7 THEN 3730
3490 IF F(PK+B)=8 OR F(PK+B)=9 THEN 3730
3500 FOR A=PK+B*11 TO MW STEP B*11
3510 IF F(A)=0 THEN NEXT
3520 IF F(A)=4 OR F(A)=7 THEN 3730
3530 IF F(PK+B*11)=8 OR F(PK+B*11)=9 THE
N 3730
```

---

```
3540 FOR A=PK+B*10 TO MW STEP B*10
3550 IF F(A)=0 THEN NEXT
3560 IF F(A)>=5 AND F(A)<=7 THEN 3730
3570 IF F(PK+B*10)=8 OR F(PK+B*10)=9 THE
N 3730
3580 FOR A=PK+B*9 TO MW STEP B*9
3590 IF F(A)=0 THEN NEXT
3600 IF F(A)=4 OR F(A)=7 THEN 3730
3610 IF F(PK+B*9)=8 OR F(PK+B*9)=9 THEN
3730
3620 FOR A=1 TO 4
3630 S=PK+FA(A)*B
3640 IF S<11 OR S>88 THEN 3660
3650 IF F(S)=3 THEN 3730
3660 NEXT A
3670 IF B=-1 THEN B=1 : GOTO 3450
3680 IF F(PK-9)=1 OR F(PK+11)=1 THEN 3730
3690 IF F(PK-9)=2 OR F(PK+11)=2 THEN 3730
3700 IF F(ZK)=-2 AND ZK-INT(ZK/10)*10=8
THEN F(ZK)=-7
3710 IF F(ZK)=-1 OR F(ZK)=-5 OR F(ZK)=-8
THEN F(ZK)=F(ZK)-1
3720 RETURN
```

---

```
3730 F(SK)=F(ZK) : F(ZK)=FZ : GOTO 2080
3740 REM *****
3750 REM *** EINGABEKONTROLLE/ENDE ***
3760 REM *****
3770 REM **** SPIELEREINGABE/ENDE ****
3780 REM *****
3790 REM ***** ZUGANGABE *****
3800 REM *****
3810 PRINT CHR$(19); TAB(33) "";
3820 FOR A=1 TO 9:PRINT CHR$(17); :NEXT
3830 IF RO=0 THEN 3880
3840 IF RO=1 THEN PRINT " ";
3850 IF RO=-1 THEN PRINT O$; "-";
3860 PRINT O$; "-"; O$; " "
3870 RO=0 : GOTO 3920
3880 PRINT CHR$(192+INT(SK/10));
3890 PRINT RIGHT$(STR$(SK),1); ":";
3900 PRINT CHR$(192+INT(ZK/10));
3910 PRINT RIGHT$(STR$(ZK),1)
3920 RETURN
3930 REM *****
3940 REM ***** ZUGANGABE/ENDE *****
3950 REM *****
```

---

```
3960 REM ***** COMPUTERZUG *****
3970 REM *****
3980 GOSUB 7880
3990 PRINT TAB(14) B$;"ITTE WARTEN"
4000 PRINT
4010 PRINT TAB(12) S$;"TELLUNGSANALYSE"
4020 PRINT
4030 PRINT TAB(13) "80 BIS ZUM ";
4040 PRINT Z$;"UG"; CHR$(145)
4050 ZN=0 : BZ=1
4060 IF F(58)=8 THEN GOSUB 6140
4070 SK=11
4080 PRINT TAB(13) "";
4090 PRINT RIGHT$(STR$(91-SK),2);
4100 PRINT CHR$(145)
4110 IF F(SK)<=0 OR F(SK)=10 THEN 4130
4120 GOSUB 5410
4130 IF SK<88 THEN SK=SK+1 : GOTO 4080
4140 FOR A=1 TO ZN-1 : FOR B=A+1 TO ZN
4150 IF BS(A)>=BS(B) THEN 4190
4160 BS(0)=BS(A):BS(A)=BS(B):BS(B)=BS(0)
4170 SK(0)=SK(A):SK(A)=SK(B):SK(B)=SK(0)
4180 ZK(0)=ZK(A):ZK(A)=ZK(B):ZK(B)=ZK(0)
4190 NEXT B,A
```

---



```
4200 PRINT TAB(14) "2"; CHR$(145)
4210 IF BZ<ZN+1 THEN 4240
4220 IF MA=1 THEN PA=1 : GOTO 4320
4230 MA=1 : BZ=1
4240 SK=SK(BZ) : ZK=ZK(BZ)
4250 IF ZK>1 THEN 4300
4260 RO=ZK
4270 ZK=58+RO*10
4280 F(58)=0
4290 F(58+RO*20)=9
4300 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
4310 REM
4320 REM * KONTROLLE KOENIG IN SCHACH **
4330 FOR PK=11 TO 88
4340 IF F(PK)=8 OR F(PK)=9 THEN 4360
4350 NEXT PK
4360 B=-1
4370 MW=50+B*39
4380 FOR A=PK+B TO MW STEP B
4390 IF F(A)=0 THEN NEXT
4400 IF F(A)>=-7 AND F(A)<=-5 THEN 5320
4410 IF F(PK+B)=-8 OR F(PK+B)=-9 THEN 53
20
```

---

```
4420 FOR A=PK+B*11 TO MW STEP B*11
4430 IF F(A)=0 THEN NEXT
4440 IF F(A)=-4 OR F(A)=-7 THEN 5320
4450 IF F(PK+B*11)=-8 OR F(PK+B*11)=-9 T
HEN 5320
4460 FOR A=PK+B*10 TO MW STEP B*10
4470 IF F(A)=0 THEN NEXT
4480 IF F(A)>=-7 AND F(A)<=-5 THEN 5320
4490 IF F(PK+B*10)=-8 OR F(PK+B*10)=-9 T
HEN 5320
4500 FOR A=PK+B*9 TO MW STEP B*9
4510 IF F(A)=0 THEN NEXT
4520 IF F(A)=-4 OR F(A)=-7 THEN 5320
4530 IF F(PK+B*9)=-8 OR F(PK+B*9)=-9 THE
N 5320
4540 FOR A=1 TO 4
4550 S=PK+FA(A)*B
4560 IF S<11 OR S>88 THEN 4580
4570 IF F(S)=-3 THEN 5320
4580 NEXT A
4590 IF B=-1 THEN B=1 : GOTO 4370
4600 IF F(PK+9)=-1 OR F(PK-11)=-1 THEN 5
320
```

---

```
4610 IF F(PK+9)=-2 OR F(PK-11)=-2 THEN 5
320
4620 PRINT TAB(14) "1"; CHR$(145)
4630 IF PA=1 THEN 5300
4640 IF MA=1 THEN MA=0 : GOTO 5260
4650 IF F(ZK)=7 THEN 4680
4660 IF FW(F(ZK))<=FW(-FZ) THEN 5260
4670 REM *****
4680 REM **** DECKUNG VON FIGUREN ****
4690 REM *****
4700 REM
4710 REM ***** FIGUR BEDROHT ??? *****
4720 B=-1
4730 MW=50+B*39
4740 FOR A=ZK+B TO MW STEP B
4750 IF F(A)=0 THEN NEXT
4760 IF F(A)>=-7 AND F(A)<=-5 THEN 4990
4770 IF F(ZK+B)=-8 OR F(ZK+B)=-9 THEN 49
90
4780 FOR A=ZK+B*11 TO MW STEP B*11
4790 IF F(A)=0 THEN NEXT
4800 IF F(A)=-4 OR F(A)=-7 THEN 4990
4810 IF F(ZK+B*11)=-8 OR F(ZK+B*11)=-9 T
HEN 4990
```

---

```
4820 FOR A=ZK+B*10 TO MW STEP B*10
4830 IF F(A)=0 THEN NEXT
4840 IF F(A)>=-7 AND F(A)<=-5 THEN 4990
4850 IF F(ZK+B*10)=-8 OR F(ZK+B*10)=-9 T
HEN 4990
4860 FOR A=ZK+B*9 TO MW STEP B*9
4870 IF F(A)=0 THEN NEXT
4880 IF F(A)=-4 OR F(A)=-7 THEN 4990
4890 IF F(ZK+B*9)=-8 OR F(ZK+B*9)=-9 THE
N 4990
4900 FOR A=1 TO 4
4910 S=ZK+FA(A)*B
4920 IF S<11 OR S>88 THEN 4940
4930 IF F(S)=-3 THEN 4990
4940 NEXT A
4950 IF B=-1 THEN B=1 : GOTO 4730
4960 IF F(ZK+9)=-1 OR F(ZK-11)=-1 THEN 4
990
4970 IF F(ZK+9)=-2 OR F(ZK-11)=-2 THEN 4
990
4980 GOTO 5260
4990 IF F(ZK)=7 THEN 5320
5000 REM
```

---

```
5010 REM ***** FIGUR GEDECKT ??? *****
5020 B=-1
5030 MW=50+B*39
5040 FOR A=ZK+B TO MW STEP B
5050 IF F(A)=0 THEN NEXT
5060 IF F(A)>=5 AND F(A)<=7 THEN 5260
5070 FOR A=ZK+B*11 TO MW STEP B*11
5080 IF F(A)=0 THEN NEXT
5090 IF F(A)=4 OR F(A)=7 THEN 5260
5100 FOR A=ZK+B*10 TO MW STEP B*10
5110 IF F(A)=0 THEN NEXT
5120 IF F(A)>=5 AND F(A)<=7 THEN 5260
5130 FOR A=ZK+B*9 TO MW STEP B*9
5140 IF F(A)=0 THEN NEXT
5150 IF F(A)=4 OR F(A)=7 THEN 5260
5160 FOR A=1 TO 4
5170 S=ZK+FA(A)*B
5180 IF S<11 OR S>88 THEN 5200
5190 IF F(S)=3 THEN 5260
5200 NEXT A
5210 IF B=-1 THEN B=1 : GOTO 5030
5220 IF F(ZK-9)=1 OR F(ZK+11)=1 THEN 5260
0
5230 IF F(ZK-9)=2 OR F(ZK+11)=2 THEN 5260
0
```

---

```
5240 GOTO 5320
5250 REM *****
5260 REM ** DECKUNG VON FIGUREN/ENDE **
5270 REM *****
5280 IF F(ZK)=1 OR F(ZK)=5 OR F(ZK)=8 TH
EN F(ZK)=F(ZK)+1
5290 IF F(ZK)=2 AND ZK-INT(ZK/10)*10=1 T
HEN F(ZK)=7
5300 PRINT TAB(14) "0"
5310 RETURN
5320 IF PA=1 THEN PA=0 : GOTO 5300
5330 F(SK)=F(ZK) : F(ZK)=FZ : BZ=BZ+1
5340 IF RO=0 THEN 4210
5350 F(58)=8 : F(58+RO*20)=0
5360 RO=0
5370 GOTO 4210
5380 REM *****
5390 REM ***** COMPUTERZUG/ENDE *****
5400 REM *****
5410 REM ***** ZUGERZEUGER *****
5420 REM *****
5430 ON F(SK)-1 GOTO 5510,5710,5810,5920
,5920,5810,6030,6030
5440 REM
```

```
5450 REM ***** BAUER 2-SCHRITT *****
5460 IF F(SK-2)<>0 THEN 5510
5470 IF F(SK-1)<>0 THEN 5510
5480 ZK=SK-2
5490 F(SK)=2 : GOSUB 6590 : F(SK)=1
5500 REM
5510 REM ***** BAUER ALLGEMEIN *****
5520 FS=F(SK) : F(SK)=2
5530 IF F(SK-11)>=0 THEN 5580
5540 ZK=SK-11
5550 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5560 GOSUB 6590
5570 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5580 IF F(SK-1)<>0 THEN 5630
5590 ZK=SK-1
5600 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5610 GOSUB 6590
5620 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5630 IF F(SK+9)>=0 THEN 5680
5640 ZK=SK+9
5650 IF SK-INT(SK/10)*10=2 THEN F(SK)=7
5660 GOSUB 6590
5670 IF SK-INT(SK/10)*10=2 THEN F(SK)=2
5680 F(SK)=FS
5690 GOTO 6100
```

```
5700 REM
5710 REM ***** SPRINGER *****
5720 FOR B1=-1 TO 1 STEP 2
5730 FOR A1=1 TO 4
5740 ZK=SK+B1*FA(A1)
5750 IF ZK<0 THEN ZK=ZK+100
5760 IF ZK>99 THEN ZK=ZK-100
5770 IF F(ZK)<=0 THEN GOSUB 6590
5780 NEXT A1,B1
5790 GOTO 6100
5800 REM
5810 REM ***** LAEUFER/DAME *****
5820 FOR B1=-1 TO 1 STEP 2
5830 FOR C1=5 TO 6
5840 FOR A1=SK+B1*FA(C1) TO 50+B1*39 STEP
P B1*FA(C1)
5850 IF F(A1)<=0 THEN 5870
5860 A1=50+B1*39 : GOTO 5890
5870 ZK=A1 : GOSUB 6590
5880 IF F(A1)<0 THEN 5860
5890 NEXT A1,C1,B1
5900 IF F(SK)<>7 THEN 6100
5910 REM
```



```
5920 REM ***** TURM/DAME *****
5930 FOR B1=-1 TO 1 STEP 2
5940 FOR C1=7 TO 8
5950 FOR A1=SK+B1*FA(C1) TO 50+B1*39 STE
P B1*FA(C1)
5960 IF F(A1)<=0 THEN 5980
5970 A1=50+B1*39 : GOTO 6000
5980 ZK=A1 : GOSUB 6590
5990 IF F(A1)<0 THEN 5970
6000 NEXT A1,C1,B1
6010 GOTO 6100
6020 REM
6030 REM ***** KOENIG *****
6040 FOR B1=-1 TO 1 STEP 2
6050 FOR A1=5 TO 8
6060 ZK=SK+B1*FA(A1)
6070 IF F(ZK)>0 THEN 6090
6080 GOSUB 6590
6090 NEXT A1,B1
6100 RETURN
6110 REM *****
6120 REM ***** ZUGERZEUGER/ENDE *****
6130 REM *****
```

```
6140 REM *** ROCHADE MIT KONTROLLE ****
6150 REM *****
6160 RO=-1 : B=58
6170 IF F(53+RO*35)<>5 THEN 6530
6180 IF F(58+RO*10)<>0 THEN 6530
6190 IF F(58+RO*20)<>0 THEN 6530
6200 IF RO=-1 AND F(28)<>0 THEN 6530
6210 FOR A=B-1 TO 31 STEP -1
6220 IF F(A)=0 THEN NEXT
6230 IF F(A)=-6 OR F(A)=-7 THEN 6530
6240 IF F(B-1)=-9 THEN 6530
6250 FOR A=B-11 TO 13 STEP -11
6260 IF F(A)=0 THEN NEXT
6270 IF F(A)=-4 OR F(A)=-7 THEN 6530
6280 IF F(B-11)=-9 THEN 6530
6290 FOR A=B+10 TO 88 STEP 10
6300 IF F(A)=0 THEN NEXT
6310 IF F(A)=-6 OR F(A)=-7 THEN 6530
6320 IF F(B+10)=-9 THEN 6530
6330 FOR A=B-10 TO 18 STEP -10
6340 IF F(A)=0 THEN NEXT
6350 IF F(A)=-6 OR F(A)=-7 THEN 6530
6360 IF F(B-10)=-9 THEN 6530
```

---

```
6370 FOR A=B+9 TO 87 STEP 9
6380 IF F(A)=0 THEN NEXT
6390 IF F(A)=-4 OR F(A)=-7 THEN 6530
6400 IF F(B+ 9)=-9 THEN 6530
6410 IF F(B+19)=-3 THEN 6530
6420 IF F(B+ 8)=-3 THEN 6530
6430 IF F(B-12)=-3 THEN 6530
6440 IF F(B-21)=-3 THEN 6530
6450 IF F(B+9)=-2 OR F(B-11)=-2 THEN 6530
6460 IF B=58+RO*20 THEN 6480
6470 B=B+RO*10 : GOTO 6210
6480 F(58)=0 : F(58+RO*20)=9
6490 SK=53+RO*35 : ZK=58+RO*10
6500 GOSUB 6590
6510 F(58)=8 : F(58+RO*20)=0
6520 ZK(ZN)=RO : BS(ZN)=BS(ZN)+100
6530 IF RO=-1 THEN RO=1 : GOTO 6170
6540 RO=0
6550 RETURN
6560 REM *****
6570 REM ***** ROCHADE/ENDE *****
6580 REM *****
```

---

```
6590 REM ***** WERTUNG *****
6600 REM *****
6610 FZ=F(ZK) : F(ZK)=F(SK) : F(SK)=0
6620 ZN=ZN+1 : SW=0
6630 FOR P=11 TO 88
6640 IF F(P)=0 OR F(P)=10 THEN 7810
6650 IF F(P)>0 THEN 7240
6660 REM *****
6670 REM ***** WERTUNG/GEGNER *****
6680 REM *****
6690 SW=SW-3*FW(-F(P))
6700 ON -F(P)-1 GOTO 6720,6810,6900,7010
,7010,6900,7120,7120
6710 REM
6720 REM ***** BAUER ALLGEMEIN *****
6730 IF F(P+1)=0 THEN SW=SW-3
6740 IF F(P-9)>0 THEN SW=SW-FW(F(P-9))
6750 IF F(P+11)>0 THEN SW=SW-FW(F(P+11))
6760 IF F(P)=-2 THEN 7810
6770 REM ***** BAUER 2-SCHRITT *****
6780 IF F(P+1)<>0 THEN 6800
6790 IF F(P+2)=0 THEN SW=SW-3
6800 GOTO 7810
```

```
6810 REM ***** SPRINGER *****
6820 A=-1
6830 FOR B=1 TO 4
6840 SP=P+A*FA(B)
6850 IF SP<11 OR SP>88 THEN 6870
6860 IF F(SP)>=0 THEN SW=SW-FW(F(SP))
6870 NEXT B
6880 IF A=-1 THEN A=1 : GOTO 6830
6890 GOTO 7810
6900 REM ***** LAEUFER/DAME *****
6910 B=-1
6920 FOR C=5 TO 6
6930 A=P+B*FA(C)
6940 IF F(A)>=0 THEN SW=SW-FW(F(A))
6950 IF F(A)<>0 THEN 6980
6960 A=A+B*FA(C)
6970 GOTO 6940
6980 NEXT C
6990 IF B=-1 THEN B=1 : GOTO 6920
7000 IF F(P)<>-7 THEN 7810
7010 REM ***** TURM/DAME *****
7020 B=-1
7030 FOR C=7 TO 8
7040 A=P+B*FA(C)
```

---

```
7050 IF F(A)>=0 THEN SW=SW-FW(F(A))
7060 IF F(A)<>0 THEN 7090
7070 A=A+B*FA(C)
7080 GOTO 7050
7090 NEXT C
7100 IF B=-1 THEN B=1 : GOTO 7030
7110 GOTO 7810
7120 REM ***** KOENIG *****
7130 B=-1
7140 FOR C=5 TO 8
7150 SP=F(P+B*FA(C))
7160 IF SP<0 AND RO=0 THEN 7180
7170 SW=SW-FW(ABS(SP))
7180 NEXT C
7190 IF B=-1 THEN B=1 : GOTO 7140
7200 GOTO 7810
7210 REM *****
7220 REM **** WERTUNG/GEGNER/ENDE ****
7230 REM *****
7240 REM ***** WERTUNG/COMPUTER *****
7250 REM *****
7260 SW=SW+3*FW(F(P))
7270 ON F(P)-1 GOTO 7290,7390,7480,7590
,7590,7480,7700,7700
```

---

```
7280 REM
7290 REM ***** BAUER ALLGEMEIN *****
7300 IF F(ZK)<=2 THEN SW=SW+3
7310 IF F(P-1)=0 THEN SW=SW+3
7320 IF F(P+9)<0 THEN SW=SW+FW(-F(P+9))
7330 IF F(P-11)<0 THEN SW=SW+FW(-F(P-11))
7340 IF F(P)=2 THEN 7810
7350 REM ***** BAUER 2-SCHRITT *****
7360 IF F(P-1)<>0 THEN 7380
7370 IF F(P-2)=0 THEN SW=SW+3
7380 GOTO 7810
7390 REM ***** SPRINGER *****
7400 A=-1
7410 FOR B=1 TO 4
7420 SP=P+A*FA(B)
7430 IF SP<11 OR SP>88 THEN 7450
7440 IF F(SP)<=0 THEN SW=SW+FW(-F(SP))
7450 NEXT B
7460 IF A=-1 THEN A=1 : GOTO 7410
7470 GOTO 7810
7480 REM ***** LAEUFER/DAME *****
7490 B=-1
7500 FOR C=5 TO 6
7510 A=P+B*FA(C)
```

---

```
7520 IF F(A)<=0 THEN SW=SW+FW(-F(A))
7530 IF F(A)<>0 THEN 7560
7540 A=A+B*FA(C)
7550 GOTO 7520
7560 NEXT C
7570 IF B=-1 THEN B=1 : GOTO 7500
7580 IF F(P)<>7 THEN 7810
7590 REM ***** TURM/DAME *****
7600 B=-1
7610 FOR C=7 TO 8
7620 A=P+B*FA(C)
7630 IF F(A)<=0 THEN SW=SW+FW(-F(A))
7640 IF F(A)<>0 THEN 7670
7650 A=A+B*FA(C)
7660 GOTO 7630
7670 NEXT C
7680 IF B=-1 THEN B=1 : GOTO 7610
7690 GOTO 7810
7700 REM ***** KOENIG *****
7710 B=-1
7720 FOR C=5 TO 8
7730 SP=F(P+B*FA(C))
7740 IF SP>0 AND RO=0 THEN 7760
7750 SW=SW+FW(ABS(SP))
7760 NEXT C
```



```
7770 IF B=-1 THEN B=1 : GOTO 7720
7780 REM *****
7790 REM *** WERTUNG/COMPUTER/ENDE ***
7800 REM *****
7810 NEXT P
7820 F(SK)=F(ZK) : F(ZK)=FZ
7830 BS(ZN)=SW : SK(ZN)=SK : ZK(ZN)=ZK
7840 RETURN
7850 REM *****
7860 REM ***** WERTUNG/ENDE *****
7870 REM *****
7880 REM ***** 5 ZEILEN LOESCHEN *****
7890 REM *****
7900 PRINT CHR$(19); ZF$
7910 FOR A=1 TO 18 : PRINT : NEXT
7920 FOR A=1 TO 5
7930 PRINT TAB(12) " "
7940 NEXT A
7950 FOR A=1 TO 5:PRINT CHR$(145);:NEXT
7960 RETURN
7970 REM *****
7980 REM *** 5 ZEILEN LOESCHEN/ENDE ***
7990 REM *****
8000 REM ***** UNTERPROGRAMME/ENDE *****
8010 REM *****
```



## 4. Verbesserungsvorschläge für das BASIC-Schachprogramm

Für den eingeweihten Programmierer gibt es zahlreiche Möglichkeiten, das BASIC-Schachprogramm entsprechend den eigenen Vorstellungen zu verändern und zu verbessern. Hier sollen drei Beispiele als Anregung dienen:

### 4.1. Blinken der gezogenen Figur

Die Durchführung der Stellungsänderung einer Figur geschieht auf dem Bildschirm so schnell, daß man bisweilen Mühe hat, den Positionswechsel zu erkennen. Das folgende Unterprogramm läßt die bewegte Figur sowohl auf dem alten als auch auf dem neuen Feld mehrmals aufblinken, so daß man die eingetretene Änderung nicht übersehen kann.

**Programmausschnitt:** (nur für C 64 !)

```
1475 IF SK > 0 THEN GOSUB 8500
```

---

```
8500 REM *****
8510 REM ** GEZOGENE FIGUR BLINKEND **
8520 REM *****
8530 PRINT CHR$(19);
8540 FOR A=1 TO 2*(9-SK+10*INT(SK/10))
8550 PRINT : NEXT
8560 PRINT TAB(INT(SK/10)*3+6) " ";
8570 FOR A=1 TO 5
8580 PRINT F$(F(ZK)+9); CHR$(157);
8590 FOR B=1 TO 100 : NEXT
8600 PRINT " "; CHR$(157);
8610 FOR B=1 TO 100 : NEXT
8620 NEXT A
8630 PRINT CHR$(19);
8640 FOR A=1 TO 2*(9-ZK+10*INT(ZK/10))
8650 PRINT : NEXT
8660 PRINT TAB(INT(ZK/10)*3+6) " ";
8670 FOR A=1 TO 5
8680 PRINT " "; CHR$(157);
8690 FOR B=1 TO 100 : NEXT
8700 PRINT F$(F(ZK)+9); CHR$(157);
8710 FOR B=1 TO 100 : NEXT
8720 NEXT A
8730 RETURN
```

---

## 4.2. Vorgabe einer Stellung

Oft möchte der Anwender wissen, wie der Rechner auf eine bestimmte Stellung reagiert oder auch nur ein abgebrochenes Spiel fortsetzen. Mit Hilfe der folgenden Programmergänzung kann man Stellungen vorgeben. Zuerst wird die interne Feldnummer (vgl. S. 2-03 f.) und anschließend die interne Ziffer der Figur (vgl. S. 2-04 f.) für das bestimmte Feld eingegeben. Da diese Ergänzung keine fehlerhaften Eingaben abfängt (was allerdings ohne großen Aufwand leicht zu ändern wäre!), ist darauf zu achten, daß beide Könige gesetzt werden und keine Figur auf dem internen Rahmen (vgl. Seite 2-11 ff.) plaziert wird.

### Programmausschnitt:

```

595 GOSUB 9000 : IF SV=1 THEN 690
.
.
9000 REM *****
9010 REM ***** STELLUNGSVORGABE *****
9020 REM *****
9030 POKE 53280,254 : POKE 53281,246
9040 PRINT CHR$(142);CHR$(147);CHR$(5)
          COLOR 7

```

```
9050 PRINT " SOLL EINE STELLUNG"
9060 PRINT " VORGEGEBEN WERDEN ? (J/N)"
9070 GET IN$ : IF IN$ = "" THEN 9070
9080 IF IN$ = "N" THEN 9200
9090 IF IN$ <> "J" THEN 9070
9100 FOR A=1 TO 16 : READ X : NEXT
9110 SV = 1
9120 PRINT:PRINT " ENDE MIT 0"
9130 PRINT
9140 INPUT " FELDNR."; F1
9150 IF F1 = 0 THEN 9200
9160 INPUT " FIGURENNR."; F2
9170 IF F2 = 0 THEN 9200
9180 F(F1) = F2
9190 GOTO 9130
9200 RETURN
```

---

### 4.3. Veränderung der Stellungsbewertung

Eine Veränderung der Stellungsbewertung geschieht in erster Linie durch Zuordnung anderer Figurenwerte in den Befehlszeilen 500 und 510. Verringert man beispielsweise den Wert der Dame von 110 auf 60, so hat dies zur Folge, daß der Rechner eher bereit ist, im Rahmen eines Zuges seine Dame zu opfern.

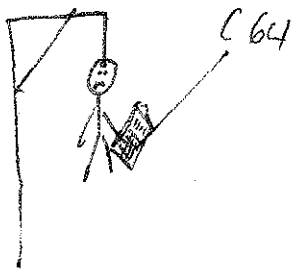
Außerdem ist es möglich, das Gewicht der leeren Felder, die besetzt werden können, zu verändern. Würde man die Zahl 3 in den Zeilen 510, 6730, 6790, 7300, 7310 und 7370 z. B. auf 15 heraufsetzen, so wäre ein leeres Feld im Rahmen der Wertung höher gewichtet als ein Bauer, der geschlagen werden kann. Dies hätte zur Folge, daß das Spiel weniger aggressiv geführt wird.

---

*[The body of the page contains extremely faint, illegible text, likely bleed-through from the reverse side of the document. The text is too light to transcribe accurately.]*



Eigene Verbesserungen (1):



Eigene Verbesserungen (2):

Eigene Verbesserungen (3):

Eigene Verbesserungen (4):

---

Eigene Verbesserungen (5):

---

Eigene Verbesserungen (6):

---

Eigene Verbesserungen (7):

Eigene Verbesserungen (8):



## 5. Überprüfung des abgeschriebenen Programms

- (1) Nachdem man das BASIC-Schachprogramm eingegeben hat, sollte es sorgfältig überprüft werden, da nur bei vollkommener Fehlerfreiheit der Befehlszeilen ein einwandfreier Programmlauf gewährleistet ist. Wir empfehlen daher, eine Abschreibekontrolle mit einer zweiten Person durchzuführen: Die erste liest das Programm vom Bildschirm vor, und die zweite kontrolliert es im Buch (nicht umgekehrt!).
- (2) Darüber hinaus ist eine Überprüfung der Wertung des Computers sehr einfach möglich:
  - (a) Starten des Programms mit RUN,
  - (b) Eingabe des Zuges E2 : E4,
  - (c) nachdem der Computer seinen Gegenzug ausgeführt hat, ist die RUN/STOP-Taste zu betätigen,
  - (d) anschließend folgende Befehlszeile im direkten Modus eingeben:  

```
for a = 1 to zn : print bs(a); : next
```

(e) folgende Zahlen müssen erscheinen:

24 24 18 15 0 -3 -3 -3 -3 -3 -6 -7  
-9 -9 -10 -16 -17 -24 -30 -57

Wenn andere Zahlen ausgegeben werden, so muß ein Fehler im Abschnitt COMPUTERZUG vorliegen. Sind es mehr oder weniger als 20 Zahlen, so ist der Fehler im ZUGERZEUGER zu suchen. Werden hingegen 20 Zahlen mit anderen Werten ausgegeben, so hat man den Fehler im Abschnitt WERTUNG zu suchen. Jene Fehler, die sich gegenseitig aufheben, sind mit der geschilderten Vorgehensweise allerdings nicht zu lokalisieren.

Hinweis:

Falls das Programm nicht ordnungsgemäß läuft, so kann dies nicht auf einen Satzfehler zurückgeführt werden, da alle Programmzeilen per LIST-Kommando auf einem an den Commodore 64 angeschlossenen Drucker ausgedruckt und so direkt als Druckvorlage verwendet worden sind.

## 6. Notwendige Änderungen für den VC 20 (mit 28 K RAM)

Da der Befehlssatz des Commodore 64 mit dem des VC 20 weitgehend übereinstimmt, ist es ohne große Schwierigkeiten möglich, das Schachspiel auch auf dem VC 20 zu programmieren. Zu diesem Zweck muß lediglich die Bildschirmausgabe geändert werden, weil der VC 20 mit einer anderen Zeilenbreite arbeitet.

### Zu ändernde bzw. zu ergänzende Programmzeilen:

```
420 BF=8           : REM ** SCHWARZ
440 FH$=CHR$(159) : REM ** HELLBLAU
880 POKE 36879, BF
920 PRINT TAB(3) " ";
940 PRINT RO$; FH$; " "; FD$; " ";
960 PRINT:PRINT TAB(3) " ";
990 PRINT RO$; FD$; " "; FH$; " ";
1010 PRINT:PRINT TAB(3) " ";
1040 PRINT TAB(4) " "; ZF$;
1060 PRINT CHR$(A); " ";
1100 PRINT:PRINT TAB(1) A : NEXT A
1500 PRINT:PRINT TAB(3) " ";
1530 PRINT RO$; FH$; F$(F(B)+9);
```

```
1550 PRINT RO$; FD$; F$(F(B+10)+9);
1570 PRINT:PRINT:PRINT TAB(3) " ";
1600 PRINT RO$; FD$; F$(F(B-1)+9);
1620 PRINT RO$; FH$; F$(F(B+9)+9);
1740 PRINT TAB(3) I$;"HR "; Z$;"UG: ";
3810 PRINT CHR$(19)
3820 FOR A=1 TO 7:PRINT:NEXT
3840 IF RO=1 THEN PRINT TAB(20) " "
3850 IF RO=-1 THEN PRINT TAB(20) O$ : PRI
NT TAB(20) I$
3860 PRINT TAB(20) O$
3861 PRINT TAB(20) I$
3862 PRINT TAB(20) O$
3863 PRINT TAB(20) " "
3880 PRINT TAB(20) CHR$(192+INT(SK/10))
3890 PRINT TAB(20) RIGHT$(STR$(SK),1)
3895 PRINT TAB(20) " : "
3900 PRINT TAB(20) CHR$(192+INT(ZK/10))
3910 PRINT TAB(20) RIGHT$(STR$(ZK),1)
3990 PRINT TAB(5) B$;"ITTE WARTEN"
4010 PRINT TAB(3) S$;"TELLUNGSANALYSE"
4030 PRINT TAB(4) "80 BIS ZUM ";
```

---

```
4080 PRINT TAB(4) " ";
4200 PRINT TAB(5) "2"; CHR$(145)
4620 PRINT TAB(5) "1"; CHR$(145)
5300 PRINT TAB(5) "0";
7910 FOR A=1 TO 17 : PRINT : NEXT
7930 PRINT " "
7945 PRINT " ";
7950 FOR A=1 TO 5:PRINT CHR$(145);:NEXT
7955 PRINT
```

---



## LITERATURHINWEISE

Die folgenden drei Bücher behandeln in umfassender Weise das Thema "Schach und Computer":

Ketterling, Hans-Peter/Schwenkel, Frieder/Weiner, Ossi:  
"Schach dem Computer", Goldmann-Taschenbuch 10861,  
München 1983, ISBN: 3-442-10861-6

Pachman, Ludek/Kühnmund, Vas I.:  
"COMPUTERSCHACH - Intelligenzduelle zwischen  
Mensch und Denkmaschine", Heyne-Buch 4704,  
München 1980, ISBN: 3-453-41365-2

Opfermann, H. C.:  
"Spielen mit dem Schachcomputer", Econ Verlag,  
Düsseldorf und Wien 1980, ISBN: 3-430-17307-8

Alle drei aufgeführten Werke enthalten weitere Literaturhinweise; wir zitieren hier jene aus dem Buch von Ketterling, H.-P./Schwenkel, F./Weiner, O.:

Das historisch erste Fragment eines Schachprogramms wurde 1945 von dem deutschen Computer-Pionier Konrad Zuse verfaßt. Veröffentlicht wurde es allerdings erst in 1972 in

K. Zuse. Der Plankalkül. Bericht der Gesellschaft für Mathematik und Datenverarbeitung Nr. 63, Bonn 1972, S. 235-285.

---

Komplette, lauffähige Computerprogramme zur Erzeugung legaler Züge sind u. a. beschrieben in:

A. G. Bell. How to program a computer to play legal chess. Computer Journal, Band 13 (1970), S. 208-219.

A. G. Bell. Games playing with Computers. Verlag George Allen & Unwin, London 1972, S. 74-100.

Programme zur Lösung von Mattproblemen gibt es schon seit dem Anfang der 50er Jahre.

Komplette Programme wurden u. a. beschrieben von

A. G. Bell (siehe oben).

G. Veenker. Ein Programm zur Lösung von Schachaufgaben. Elektronische Rechenanlagen, Band 7 (1965), Heft 1, S. 25-29.

J. R. Manning. White to move and mate in N moves. Computer Journal, Band 14 (1971), S. 208-213.

Die Grundprinzipien der heutigen Schachprogramme beschrieb als erster

C. E. Shannon. A chess-playing machine. Scientific American, Band 182 (1950), S. 48-51.

Von Shannon stammen die Minimax-Untersuchung von Spielbäumen nach der A-Strategie (erschöpfende Vorausrechnung bis zu einer festen Tiefe) und der B-Strategie (Vorselektion der Züge, variable Vorausrechnung), die in heutigen Programmen meist kombiniert werden, sowie Vorschläge zu einer Bewertungsfunktion. Das erste brauchbar spielende Schachprogramm stammt von R. D. Greenblatt, D. E. Eastlake und S. D. Crocker. The Greenblatt chess program. AFIPS Conference Proceedings Band 31 (1967), S. 801-810.

Damit beginnt die Ära der modernen Großrechner-Schachprogramme, gekennzeichnet durch die großen internationalen Wettbewerbe.

---



Sie ist u. a. beschrieben in

M. Newborn. Computer Chess. Verlag Academic Press, New York 1975.

D. Levy. Chess and Computers. Verlag Batsford, London 1976.

A. G. Bell. The Machine Plays Chess? Verlag Pergamon, London 1978.

Den gegenwärtigen Stand des Computerschachs ersieht man am besten aus den Sammelbänden

M. R. B. Clarke (Herausgeber). Advances in Computer Chess. Band 1 (1977), Band 2 (1980), Verlag Edinburgh University Press.

P. W. Frey (Herausgeber). Chess Skill in Man and Machine. Springer-Verlag, Heidelberg 1977.

Vollständige partiienspielende Programme sind abgedruckt in:

P. W. Frey und L. R. Atkin. Creating a chess player. Part 1, 2, 3, 4; in: BYTE, Band 3 (1978), Heft 10, S. 182-191; Heft 11, S. 162-181; Heft 12, S. 140-157; Band 4 (1979), Heft 1, S. 126-145.

D. und K. Spracklen. SARGON. A Computer Chess Program. Verlag Hayden, Rochelle Park 1978.

Typische Lehrbücher der Schachstrategie (auch in neueren Auflagen erschienen) sind

M. Euwe. Urteil und Plan im Schach. Verlag de Gruyter, Berlin 1956.

L. Pachman. Moderne Schachstrategie, Band 1, 2, 3. Sportverlag, Berlin 1958.

Vom wissenschaftlichen Standpunkt aus ragen die folgenden Werke heraus, die daher auch für die Schachprogrammierung am meisten

---

versprechen:

H. Kmoch. Die Kunst der Bauernführung. Engelhardt, Berlin 1956.

A. de Groot. Thought and Choice in Chess, Mouton, Den Haag 1965.

---

# STICHWORTVERZEICHNIS

- Abkürzung der Figuren 2-04  
Ablauflogik Vorw.-01  
Abschreibekontrolle 5-01  
Absolutwert 2-55, 2-57  
Änderungen für VC 20 (mit  
28 K RAM) 6-01  
Anfangswert 2-65, 2-75  
ASCII-Wert 2-61  
Ausführung 2-95
- Bauer 2-53, 2-59, 2-85,  
2-87, 2-89, 2-103  
BAUER ALLGEMEIN 2-52,  
2-86, 2-96, 2-98,  
2-100, 2-104, 3-12,  
3-25, 3-30, 3-33  
Bauer, bereits gezogen  
2-19, 2-04  
Bauer des Rechners 2-85  
Bauer in Grundposition  
2-04, 2-53, 2-87  
Bauernzug 2-53, 2-89  
BAUER 2-SCHRITT 2-52,  
2-86, 2-96, 2-98,  
2-100, 2-104, 3-12,  
3-25, 3-30, 3-33  
Bedienungshinweis 2-08 f.  
BEDIENUNGSHINWEISE 2-08,  
3-01
- Bedrohungssituation 2-49,  
2-81, 2-83, 2-94  
Benennung des Schachbrett-  
feldes 2-02  
Bewertung 2-69, 2-89,  
2-91, 2-95  
Bewertung der Stellung  
2-15  
Bewertung des Doppel-  
schritts 2-89  
Bewertung einer Rochade  
2-95  
Bildschirm 2-06  
Bildschirmteil, unterer  
2-33  
Binärmuster 2-23  
black-box Vorw.-01  
Blinken der gezogenen  
Figur 4-01  
Brett, internes 2-91  
Buchstabe/Ziffer-Kombi-  
nation 2-02
- CHR\$(..) 2-63  
Compiler 1-02  
COMPUTERZUG 2-32,  
2-64 ff., 2-93, 3-06,  
3-18  
Computerzug 2-65  
Cursor 2-37

## Anhang-06

---

- DAME 2-56, 3-14  
Dame 2-04, 2-19, 2-25,  
2-57, 2-59, 2-85, 2-89,  
2-91, 2-103  
Damenabtausch 2-79  
Deckung 2-81, 2-83  
DECKUNG VON FIGUREN 2-80,  
3-21  
DEL 2-37, 2-41  
Doppelschritt 2-53, 2-87  
Druck, inverser 2-17  
Druck, reverser 2-17,  
2-35  
Druckzeichen 2-23, 2-35  
Druckzeichen der Figur  
2-15  
DRUCKZEICHEN FIGUR 2-22,  
3-04  
Durchführung 2-89
- ECHTE FIGUREN 2-20, 3-03  
Einfachschritt 2-89  
EINGABEKONTROLLE 2-52,  
3-12  
Existenz einer Figur 2-99
- Farbe des Schachbrett-  
feldes 2-35  
Farbe für dunkles Feld  
2-17  
Farbe für helles Feld  
2-17  
Farbwert 2-17
- FARBZUWEISUNGEN 2-16 f.,  
3-02  
FELDBASTAENDE 2-24, 2-26,  
3-04  
Feldabstand 2-15, 2-27,  
2-91  
Feldabstandswert 2-25,  
2-55  
Feldbelegung 2-19  
Feldbenennung 2-02 ff.  
Feldbenennung, computer-  
interne 2-03 f.  
Feld, besetzbares 2-103  
Feld, computer-internes  
2-39, 2-41  
Feld des Schachbretts  
2-11  
Feld des Spielfelds 2-97  
Felderbenennung 2-31  
Feldfigur 2-53  
Feld, leeres 2-19, 2-97  
Feld, zusätzliches 2-11  
Feld, zu Überstreichendes  
2-47, 2-94  
Figur 2-23, 2-35  
Figur, Ausgabe der 2-35  
FIGUR BEDROHT ??? 2-80,  
3-21  
Figur, computereigene 2-67  
Figur des Rechners  
2-05 f., 2-97  
Figur des Spielers  
2-05 f., 2-39, 2-97  
Figurenart 2-101  
Figurenwert 2-11, 2-19  
FIGURENWERTE 2-18 f.,  
3-03
-

- Figurenwert, einfacher  
2-99
- FIGUR GEDECKT ??? 2-82,  
3-23
- Figur, gefundene 2-99
- Figur, gezogene 2-79,  
2-81, 2-85
- Figur, schwarze 2-05
- Figur, weiÙe 2-05
- Figur, zu ziehende 2-87
- GROSSBUCHSTABE 2-28,  
3-04
- GroÙbuchstabe 2-06, 2-61
- GroÙschrift 2-39
- Grundlinie des Spielers  
2-85, 2-89
- Grundlinie, gegnerische  
2-59
- Grund, spielstrategischer  
2-79, 2-81, 2-95
- Grundstellung 2-01
- Grundstellung, computer-  
interne 2-05
- Hintergrund 2-31
- Hintergrundfarbe 2-17,  
2-31
- Index, negativer 2-23
- INITIALISIERUNG 2-10 ff.,  
3-02
- INTERNE ZUGAUSFUEHRUNG  
2-53, 2-58, 3-15
- Interpreter-Sprache 1-01
- Kleinbuchstabe 2-06
- Kleinschriftmodus 2-23,  
2-31, 2-39
- KOENIG 2-56, 2-91, 2-103,  
2-105, 3-15, 3-27,  
3-32, 3-34
- König 2-25, 2-39, 2-47,  
2-57, 2-59, 2-73, 2-75,  
2-77, 2-85, 2-91,  
2-93 f.
- König, bereits gezogen  
2-04, 2-19
- König im Schach 2-94
- König, noch rochadefähig  
2-04, 2-19, 2-67, 2-93
- König, spielereigener 2-59
- König, zu rochierender  
2-45
- KONTROLLE KOENIG IN SCHACH  
2-76, 3-15
- Läufer 2-04, 2-19, 2-25,  
2-55, 2-57, 2-91, 2-103
- LAEUFER/DAME 2-54, 2-90,  
2-102, 2-104, 3-13,  
3-26, 3-31, 3-33
- Leerstelle 2-31
- Listing 2-07 ff.
- Maschinenprogramm 1-02
- Mattsituation 2-33
- Matt-Vermutung 2-73
- Mattstellung 2-33
- Merker 2-73
- Multiplikator 2-99

## Anhang-08

---

- Norm, internationale 2-02
- Pattsituation 2-33, 2-75  
Pattstellung 2-33  
Patt-Vermutung 2-75  
Position, unbedrohte  
2-47, 2-94  
Programmlauf, einwand-  
freier 5-01
- Qualität, abnehmende 2-71
- Rahmen 2-31  
Rahmenfarbe 2-17, 2-31  
Randbeschriftung 2-31  
Rechner-Figur 2-99  
RETURN 2-41  
RIGHT\$(..) 2-63  
Rochade 2-37, 2-39, 2-45,  
2-47, 2-51, 2-53, 2-59,  
2-67  
Rochade, große 2-08,  
2-45, 2-49, 2-71 ff.,  
2-94 f.  
Rochade, kleine 2-08,  
2-45, 2-49, 2-71 ff.,  
2-94 f.  
ROCHADE MIT KONTROLLE  
2-44, 2-92, 3-10, 3-28  
ROCHADENAUSFUEHRUNG 2-50  
Rückgängigmachung 2-89
- Schach 2-47, 2-59  
2-73, 2-75, 2-77 f.
- SCHACHBRETT 2-30, 3-04  
Schachbrett 2-01 ff.,  
2-35  
Schachbrett, computer-  
internes 2-04  
Schachbrett, erweitertes  
computer-internes 2-13  
Schachbrett, leeres 2-31  
Schachbrettmuster 2-31  
Schachbrettrahmen 2-21  
Schachcomputer Vorw.-01  
schachmatt 2-75, 2-85  
Schachnotation 2-61  
Schachnotation, übliche  
2-39, 2-41  
Schachprogramm Vorw.-01  
Schach-Regel 2-43  
Scheincursor 2-37  
SCHEINFIGUREN 2-20, 3-03  
Scheinfligur 2-11, 2-21,  
2-97  
Schrägschlagen 2-53, 2-89  
Setzfehler 5-02  
Sortierung 2-69  
Spiel, abgebrochenes 4-03  
Spieleigenschaft des  
Programms 2-99  
Spieler 2-99  
SPIELEREINGABE 2-32,  
2-36, 2-38, 3-06, 3-08  
Spielfunktion 2-33  
SPIELTEIL 2-32, 2-85,  
3-05  
Spielteil 2-33  
SPRINGER 2-54, 2-90,  
2-102, 2-104, 3-13,  
3-26, 3-31, 3-33
-

- Springer 2-04, 2-19,  
2-25, 2-55, 2-91  
Springer-Zug 2-91  
Startfeld 2-87  
Startkoordinate 2-15,  
2-27, 2-39, 2-41, 2-55,  
2-57, 2-61, 2-65, 2-67,  
2-107  
Startkoordinatenfeld 2-53  
Startkoordinatenwert 2-39  
STEINE SETZEN 2-32, 2-43,  
3-06 f.  
Stellungswert 2-97  
Stellungswert des  
Computers 2-107  
Stellungswert des Gegners  
2-107  
Stellungswert, ermittelter  
2-107  
Stellung, zu analysierende  
2-99  
Steuerung des Spiels 2-33  
String 2-63  
STR\$(..) 2-63  
Suchlauf 2-67  
  
Text, erläuternder 2-29  
Turm 2-25, 2-57, 2-59,  
2-85, 2-91, 2-94, 2-103  
Turm, bereits gezogen  
2-04, 2-19  
TURM/DAME 2-56, 2-90,  
2-102, 2-105, 3-14,  
3-27, 3-31, 3-34  
Turm, noch rochadefähig  
2-04, 2-19  
  
Turm-Zug 2-57  
Turm, zu rochierender  
2-45, 2-93  
  
Überprüfung des abge-  
schriebenen Programms  
5-01  
Überprüfungsrunde, erste  
2-73, 2-79  
Überprüfungsrunde, neue  
2-75  
Umschaltung auf Normal-  
schrift 2-17  
Umwandlung 2-59  
UNTERPROGRAMME 2-34, 3-07  
Untersuchungszweck,  
interner 2-107  
  
VC 20 6-01  
Veränderung der Stellungs-  
bewertung 4-05  
Verbesserungsvorschlag  
4-01  
Vorgabe einer Stellung  
4-03  
Vorwärtsschreiten 2-53  
Vorzeichen 2-05  
  
Wert der gezogenen Figur  
2-79  
Wert, taktischer 2-65,  
2-99  
WERTUNG 2-96 ff., 3-30

## Anhang-10

---

- Wertung, beste 2-69  
WERTUNG/COMPUTER 2-104 f.  
WERTUNG/GEGNER 2-96 ff.,  
2-105, 3-30  
Wertung, zweitbeste 2-69
- Zeichenfarbe 2-17  
Zeichenwert 2-23  
Zeichen, gerades 2-57  
Zeichen, schräges 2-55  
Zielfeld 2-87, 2-91  
Zielkoordinate 2-15,  
2-27, 2-41, 2-43, 2-55,  
2-57, 2-63, 2-65, 2-87,  
2-95, 2-107  
Zielkoordinate des  
Springers 2-91  
Zielkoordinateneingabe  
2-41  
Zielkoordinatenfeld 2-59  
Zielkoordinatenwert 2-41  
ZUGANGABE 2-32, 2-60,  
3-06, 3-17
- Zug, ausgeführter 2-107  
Zugberechnung 2-04  
Zug, bestmöglicher 2-65  
Zug des Rechners 2-85  
ZUGEINGABE 2-36, 2-38,  
3-08  
Zugeingabe 2-33, 2-61  
ZUGERZEUGER 2-86, 2-97,  
2-107, 3-24  
Zug, figurespezifischer  
2-87  
Zug, figurgemäßer 2-25  
Zug, gedachter 2-15  
Zug, gespeicherter 2-73  
Zug, normaler 2-95  
Zugnummer 2-65, 2-97  
Zugsuche 2-83  
Zugüberprüfung 2-78  
Zug, vorläufiger 2-59  
Zug, zuletzt getätigter  
2-61  
Zug, zweitbesten 2-73  
5 ZEILEN LOESCHEN 2-106,  
3-35
-



Günter O. Hamann

**Lerne  
BASIC  
mit dem 64  
Commodore VC 20**

Programmierte Unterweisung

**B** DEUTSCHER BETRIEBSWIRTE-VERLAG GMBH**Günter O. Hamann****Lerne BASIC  
mit dem Commodore <sup>64</sup>  
VC 20**

– Programmierte Unterweisung –

512 Seiten, 1. Auflage 1984,  
Snolin broschiert

DM 32,80

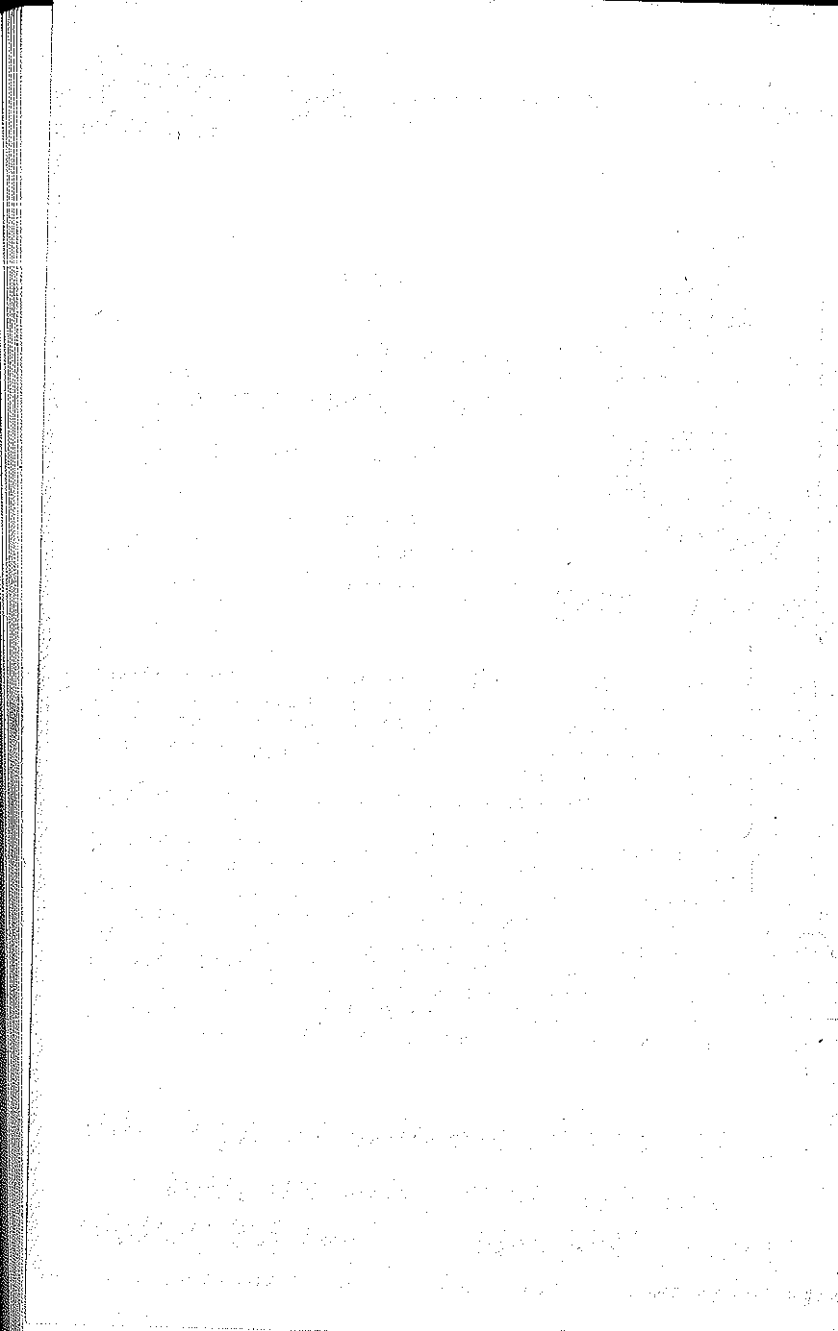
ISBN 3-88640-016-6

Der Commodore 64 und der VC 20 haben in erheblichem Maße dazu beigetragen, daß sich heute auch der „Normalverbraucher“ einen Computer leisten kann. Das Buch „Lerne BASIC mit dem Commodore 64/VC 20“ beschreibt die Programmiersprache dieser beiden Rechner. – Nach dem Durcharbeiten aller Lektionen wird der Leser in der Lage sein, selbständig

- BASIC-Programme für den Commodore 64, den VC 20 und die übrigen Commodore-Rechner zu erstellen,
- mit Hilfe der Handbücher und Systemunterlagen verschiedener Hersteller sich in kurzer Zeit in die BASIC-Versionen anderer Rechner einzuarbeiten.

Der zu vermittelnde Stoff ist in kleine, überschaubare Lerneinheiten (LE) unterteilt. Jeder LE folgt eine Prüfeinheit (PE) mit Übungsaufgaben, Ergänzungs- und/oder Auswahltests, die sowohl der Erfolgskontrolle als auch der Festigung des Lernstoffs dienen. Nach einer PE folgt immer die Lösung (LÖS). Außerdem enthält das Buch zahlreiche ausführlich erläuterte Übungs- und Spielprogramme (z.B. „Lottozahlen“, „Deutschlandlied“, „Zahlenraten“, „Training des Kopfrechnens“, „Verkauf von Manhattan durch die Indianer“, „Druckaufbereitung von Zahlen“ etc.).

**Deutscher Betriebswirte-Verlag GmbH****7562 Gernsbach 1, Bleichstr. 20-22****Telefon (07224) 3091****Telex 789 15 dbv-d**



Günter O. Hamann

Lerne  
**BASIC**  
mit dem  
**VolksComputer VC20**  
Programmierte Unterweisung

**B** DEUTSCHER BETRIEBSWIRTE-VERLAG GMBH

Günter O. Hamann

Lerne **BASIC**  
mit dem  
**VolksComputer VC20**

Programmierte Unterweisung

5. Auflage 1984

488 Seiten, fester Snolineinband

ISBN: 3-88640-017-4 DM 29,80

Die vorliegende erweiterte und verbesserte 5. Auflage des Buches „Lerne BASIC mit dem VolksComputer VC 20“ beschreibt das BASIC (von Microsoft) eines bekannten Hobby-Rechners.

Der zu vermittelnde Stoff ist – wie man es von dem Autor bereits kennt – in kleine, überschaubare Lerneinheiten (LE) unterteilt. Jeder LE folgt eine Prüfeinheit (PE) mit Übungsaufgaben, Ergänzungs- und/oder Auswahltests, die sowohl der Erfolgskontrolle als auch der Festigung des Lernstoffes dienen. Nach einer PE folgt immer die Lösung (LÖS).

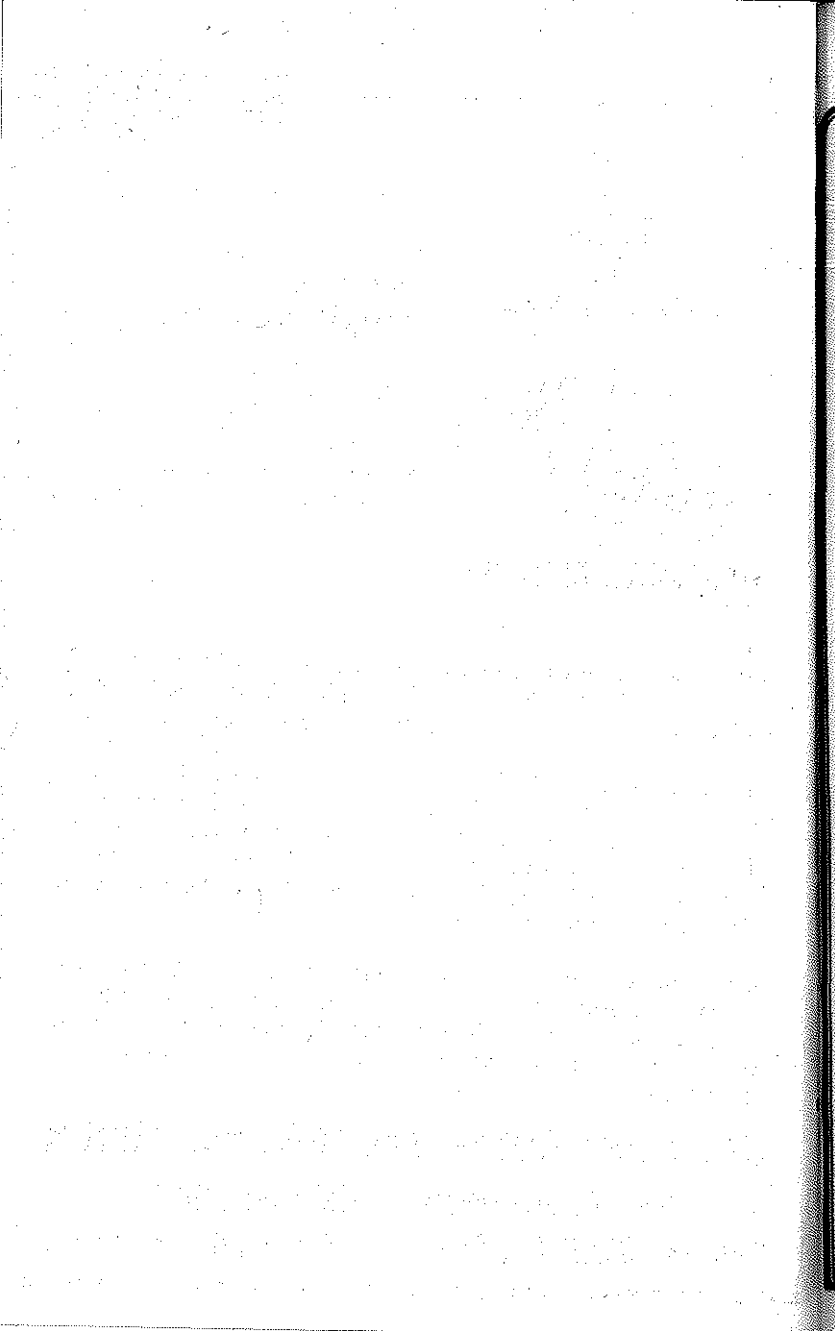
Außerdem enthält das Buch zahlreiche ausführlich erläuterte **Übungs- und Spielprogramme** (z.B. „Lottozahlen“, „Deutschlandlied“, „Zahlenraten“, „Training des Kopfrechnens“, „Verkauf von Manhattan durch die Indianer“, „Druckaufbereitung von Zahlen“ etc.)

**Deutscher Betriebswirte-Verlag GmbH**

7562 Gernsbach 1, Bleichstr. 20-22

Telefon (07224) 3091

Telex 789 15 dbv-d



Günter O. Hamann

**D**atenverarbeitung mit  
**BASIC**

Programmierte Unterweisung

*5., erweiterte und verbesserte Auflage***B** DEUTSCHER BETRIEBSWIRTE-VERLAG GMBH

Günter O. Hamann

**Datenverarbeitung  
mit BASIC**

Programmierte Unterweisung

5., erweiterte und verbesserte  
Auflage, 1984  
458 Seiten, Snotin broschiert

DM 32,80

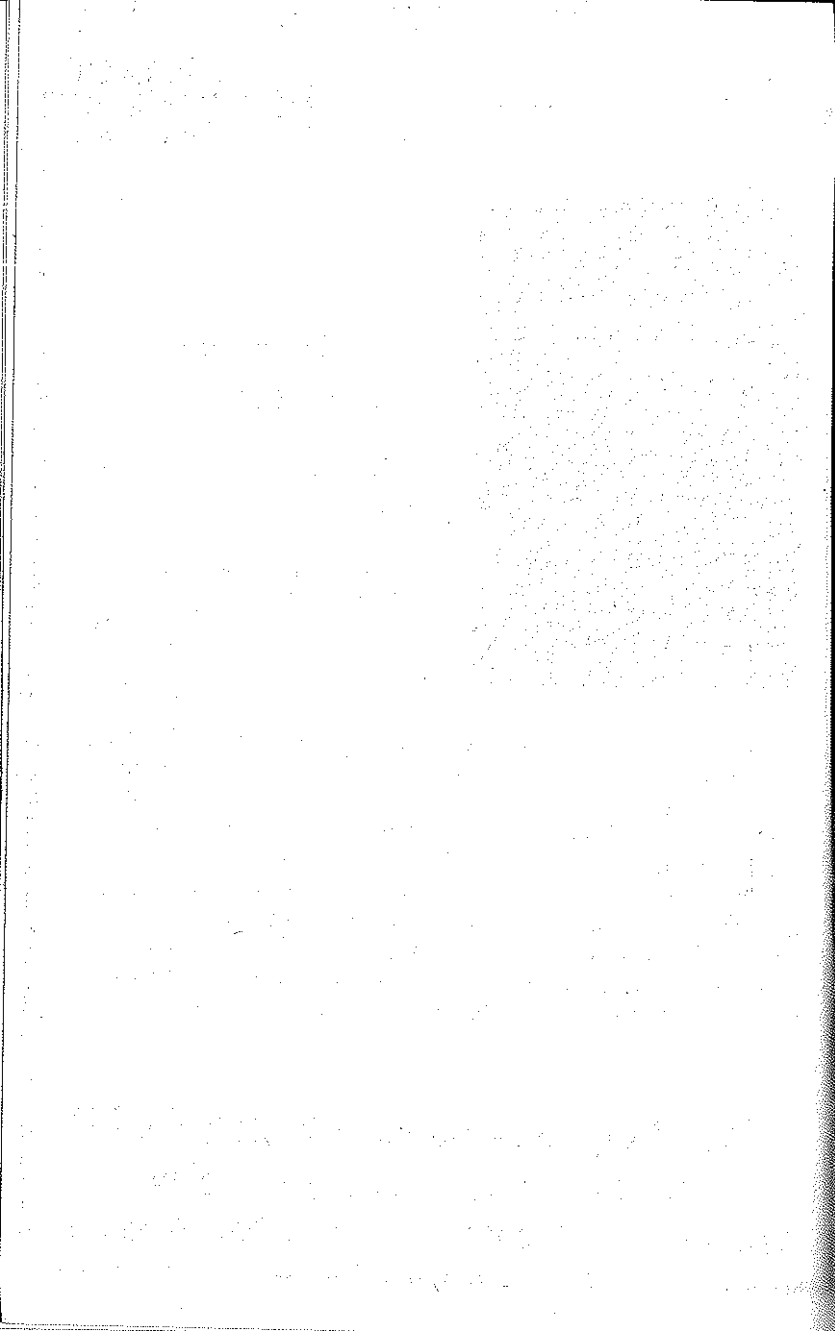
Die vorliegende verbesserte und erweiterte 5. Auflage des Buches „Datenverarbeitung mit BASIC“ beschreibt eine verbreitete BASIC-Version (Microsoft BASIC-80/Disk und BASIC-86/Disk), die für kommerziellen Einsatz in Unternehmungen konzipiert ist. Der zu vermittelnde Stoff ist – wie man es von dem Autor bereits kennt – in kleine, überschaubare Lerneinheiten (LE) unterteilt. Jeder LE folgt eine Prüfeinheit (PE) mit Übungsaufgaben, Ergänzungs- und/oder Auswahltests, die sowohl der Erfolgskontrolle als auch der Festigung des Lernstoffes dienen. Nach einer PE folgt immer die Lösung (LÖS).

**Deutscher Betriebswirte-Verlag GmbH**

7562 Gernsbach 1, Bleichstr. 20-22

Telefon (07224) 3091

Telex 789 15 dbv-d



Günter O. Hamann

**Logik der  
Programmierung**

Programmierte Unterweisung

**DBV** 

Günter O. Hamann

**Logik der  
Programmierung**

Programmierte Unterweisung

Strukturierte und Normierte  
Programmierung  
mit Programmablaufplänen und  
Struktogrammen2., erweiterte und verbesserte Auflage,  
(Juli) 1983  
ISBN 3-88640-012-3  
448 Seiten

DM 44,—

Ohne fundierte Kenntnisse der Programmierlogik (= Lehre über die Erstellung von Struktogrammen und Programmablaufplänen) ist es kaum möglich, komplexe Computer-Programme zu schreiben.

Das vorliegende Werk ist ein Lehrbuch und als Programmierte Unterweisung (PU) abgefaßt. Es vermittelt in einfacher und besonders verständlicher Form die notwendigen Grundkenntnisse der Programmierlogik. Darüber hinaus wird ein Verfahren der „Normierten Programmierung“ dargestellt, mit dessen Hilfe komplexe kommerzielle Problemstellungen leicht in ein Struktogramm oder einen Programmablaufplan umgesetzt werden können.

Der zu vermittelnde Stoff ist in kleine, überschaubare Lerneinheiten (LE) unterteilt. Jeder LE folgt eine Prüfeinheit (PE), der sich immer eine Lösung (LÖS) anschließt.

**Deutscher Betriebswirte-Verlag GmbH****7562 Gernsbach 1, Bleichstr. 20-22****Telefon (07224) 3091****Telex 789 15 dbv-d**