

Schach dem C 64!

Eines der interessantesten Gebiete der Computertechnik ist sicherlich die Beschäftigung mit »künstlicher Intelligenz«. Um erste Erfahrungen auf diesem Gebiet zu sammeln, bietet sich ein Schachprogramm gut an. Hier ist eines zum Abtippen!

Man erkennt auch sofort, daß es an diesem Programm keine »künstliche Intelligenz« im eigentlichen Sinne gibt. Die »Intelligenz« beruht nur auf festen, mathematischen Formeln. Trotzdem ist es interessant, auf diesem Gebiet zu experimentieren. Das vorliegende Programm ist ein voll funktionsfähiges Schachprogramm, das über eine grafische Spielfeldanzeige verfügt und eine sehr kurze Rechenzeit von unter 10 Sekunden hat. Natürlich ist das Programm sehr spielstark. Der Sinn dieses Programms ist denn auch ein anderer. Dieses Programm soll dem Benutzer ein Grundstein für selbstgeschriebene Erweiterungen sein. Der interessierte Anwender ist aufgefordert, kreativ zu werden und eigene Ergänzungen zu installieren. So bietet sich zum Beispiel das Stellungsbewertungsprogramm besonders dazu an, erweitert und verfeinert zu werden. Auch die Grafik kann sicher verbessert werden.

So arbeitet das Programm

Der Programmablauf sieht wie folgt aus: Nachdem das Spielfeld initialisiert und ausgegeben ist, wird eine Eingabe eines weißen Zugs erwartet, zum Beispiel E2E4. Der Zug wird auf seine Richtigkeit überprüft und, falls er korrekt ist, ausgeführt.

Dann wird eine Schleife gestartet, die sämtliche Spielfelder auf schwarze Figuren hin untersucht. Wird eine schwarze Figur gefunden, so wird der Zuggenerator angesprochen. Hier werden die der Figur möglichen Züge ausgeführt und mittels der Stellungsbewertung beurteilt. Nachdem alle Figuren überprüft wurden, wird der schwarze Zug mit seiner Wertigkeit ausgegeben. Nach einer kurzen Warteschleife wird die aktuelle Stellung ausgegeben, und es kann der nächste weiße Zug ausgeführt werden.

Die verwendete Maschinenunterroutine liegt im Speicherbereich von 49408 (\$C100) bis 49864 (\$C2D0). Das Unterprogramm überprüft, ob der schwarze König von irgendeiner weißen Figur angegriffen ist. Falls dies der Fall ist, so wird der Wert 128 an Basic übergeben, andernfalls der Wert 0. Es ist nun möglich, mit dem MC-Programm zu überprüfen, ob eine beliebige schwarze Figur angegriffen ist. Dazu wird in Speicherstelle 49409 die Position der aktuellen Figur (im Bewertungsprogramm ZF) geschrieben und in die Speicherstelle 49414 der Wert der Figur (zum Beispiel 130 = Springer). Nach Aufruf des Unterprogramms müssen unbedingt (!!) wieder die Standardwerte gePOKEt werden (49409:100 / 49414:134). Wird dies nicht beachtet, so kann der Computer keine regelwidrigen Stellungen mehr erkennen und antwortet mit unlogischen Zügen. Das Maschinenprogramm ist somit ideal dazu geeignet, um in einem selbst ergänzten Stellungsbewertungs-Programm eingesetzt zu werden. Man sollte jedoch beachten, daß ein umfangreiches Bewertungsprogramm die Rechenzeit pro Zug nicht unerheblich steigert.

Um dem Computer das Schachspiel beizubringen, sind im wesentlichen drei Probleme zu lösen:

1. Die interne Darstellung von Spielfeld und Figuren.
2. Der Zuggenerator: Er muß die Figuren den Regeln entsprechend bewegen und eine Liste möglicher Züge generieren.
3. Die Stellungsbewertung. Von ihrer Genauigkeit hängt die Spielstärke des Programms ab.

Nachfolgend nun eine Anleitung zur Lösung der drei Teilprobleme.

Da der Computer nicht auf das Schachbrett sehen kann, muß man es intern in einer Computerform darstellen, mit der der Computer arbeiten kann. Es bietet sich an, jedem Feld des Schachbretts eine Speicherstelle im Computer zuzuordnen.

100 bis 520:	Dieses Programm dient zur Stellungsbewertung. Die neue Position wird aufgestellt (100) und die Position auf Rechtmäßigkeit geprüft (110; Überprüfung, ob König im Schach). Falls die Position nicht legal ist, wird die weitere Bewertung übergangen.
210 bis 220:	Hier wird überprüft, ob eine Figur auf ihrem Ausgangsfeld beziehungsweise ihrem vorgesehenen Zielfeld von einem Bauern angegriffen ist.
500 bis 520:	Falls der Wert der Stellung besser ist wie der BEST-MOVE bisher, so wird der überprüfte Zug als bester Zug eingetragen. Bei gleicher Wertigkeit entscheidet der Zufallsgenerator (510).
1000 bis 1090:	Hier wird das Spielfeld aufgebaut. Näheres siehe Zeichnung des Spielfeldes. Außerdem wird der USER-Vektor für das MC-Unterprogramm, das die Stellungen auf Schachgebote testet, gesetzt.
2000 bis 2100:	Das Schachbrett mit Randbezeichnungen und der aktuellen Figurenstellung wird ausgegeben. Außerdem wird der Positionswert (BW) auf ein definiertes Minimum gesetzt.
3000 bis 3130:	Hier wird ein weißer Zug eingelesen und, wenn er auf seine Richtigkeit geprüft wurde, ausgeführt. Die Legalitätsprüfung kann noch erweitert werden.
4000 bis 4040:	Diese Schleife sucht die Positionen der schwarzen Figuren (Wert größer 128) und verzweigt bei gefundener Figur in den Zuggenerator.

5000 bis 5090:	Es wird der schwarze Zug und seine Wertigkeit ausgegeben. Besonders bei Erweiterungen der Stellungsbewertung ist diese Kontrolle wichtig, um zu sehen, ob der Computer den Wert des Zugs richtig einschätzt. Nach zirka 2 Sekunden wird wieder zur Spielfeldanzeige gesprungen.
6000 bis 6800:	Dies ist das Kernstück des Programms: der Zuggenerator. Hier wird geprüft, welche Figur gefunden wurde, und die möglichen Züge werden ausgeführt. Dabei wird bei jeder Stellung das Bewertungs-Unterprogramm aufgerufen. Anschließend wird wieder zur Suchschleife (4000-) verzweigt.

Tabelle 1. Die wichtigsten Routinen

AA	: Wert der aktuellen Stellung (128 = ungültig)
AF	: Ausgangsfeld, das vom Zuggenerator übergeben wird
AB	: Ausgangsfeld des bisher besten gefundenen Zuges
BZ	: Zielfeld des bisher besten gefundenen Zuges
W	: Figurenwert für die Spielfeldausgabe: 1 = Bauer, 2 = Springer, 3 = Läufer, 4 = Turm, 5 = Dame, 6 = König. Schwarze Figuren sind im Wert 128 höher.
WZ	: Wert des Zielfeldes. Wird gebraucht, da das Bewertungsprogramm den Inhalt des Zielfeldes überschreibt.
X	: Aktuelles Figurenfeld mit schwarzer Figur
ZF	: Zielfeld; wird vom Zuggenerator übergeben.

Tabelle 2. Variablenübersicht

Den Inhalt der entsprechenden Speicherstellen stellen die Figuren dar, die sich auf dem Schachbrett befinden. Nun taucht aber ein zweites Problem auf: Wie kann der Computer auf einfache Weise überprüfen, wann sich eine Figur am Spielfeldrand befindet und nicht weiterziehen kann. Hier bietet es sich an, daß man um das eigentliche Spielfeld noch einen Rand installiert, der einen speziellen Zahlenwert enthält (Bild 1). In diesem Fall signalisiert der Zahlenwert 128 dem Rechner, daß dieses Feld nicht zum eigentlichen Spielfeld gehört und er es somit mit seinen Figuren nicht betreten darf. Damit hätten wir das Schachbrett in einer computergerechten Form intern dargestellt. Als letztes Problem müssen noch die verschiedenen Spielfiguren auf dem internen Brett codiert werden. Dazu ordnet man jeder Figur eine Zahl zu. Die weißen Figuren bekommen die Werte 1 bis 6 und die schwarzen Figuren die Werte 1 bis 6 plus 128. Somit ist eine einfache Unterscheidung zwischen weißen und schwarzen Figuren möglich: Weiße Figuren haben einen Wert kleiner 128 und schwarze Figuren einen Wert größer 128. Leere Felder werden mit Null initialisiert. Sehen wir uns jetzt im Listing an, wie die Probleme innerhalb des Programms gelöst werden (Zeilen 1030 bis 1080). Das Spielfeld liegt im Rechner ab Speicherstelle 49152. In den Zeilen 1030 bis 1050 werden zunächst der Spielfeldrand (128) und die leeren Felder (0) initialisiert (vergleiche auch Bild 1). In Zeile 1060 werden dann die weißen (1) und schwarzen Bauern (1 + 128 = 129) auf die zweite beziehungsweise siebte Reihe des Spielfeldes positioniert. Die beiden Zeilen 1070 und 1080 dienen dazu, die restlichen Figuren auf den Grundreihen zu positionieren.

Der Zuggenerator

Doch mit dem Spielbeginn taucht ein neues Problem auf: Wie bringe ich den Computer dazu, daß er die Figuren entsprechend den Regeln bewegt? Als Beispiel wollen wir dies mit den Turmzügen durchsprechen (Zeilen 6440 bis 6560). Als Grundlage dient hier das Spielfeld in Bild 1. Der Turm darf den Regeln nach nur senkrecht und waagerecht ziehen, und zwar so lange, bis er auf den Spielfeldrand oder eine andere Figur trifft. Das Ziehen des Turmes bedeutet somit auf dem Computerspielfeld, daß zum Ausgangsfeld des Turmes die Vielfachen von 1 oder 10 addiert beziehungsweise subtrahiert werden. Dieses Verfahren wird so lange wiederholt, bis der Turm auf ein Hindernis (Spielfeldrand oder andere Figur) trifft. Diese Zuggenerierung wird vom Programm wie folgt realisiert: Als Beispiel betrachten wir die Zuggenerierung senkrecht nach »oben«, also die Addition der Vielfachen von 10 zum Ausgangsfeld. Die anderen Zugrichtungen ergeben sich

analog zu der betrachteten. In Zeile 6440 wird zunächst ein Flag auf Null gesetzt (FF=0). Dieses Flag wird immer dann auf 1 gesetzt, wenn der Turm bei seinem Zug auf ein Hindernis trifft. Als nächstes wird eine FOR-NEXT-Schleife gestartet, die es ermöglicht, alle Felder in »+10-Richtung« zu überprüfen. Die Variable X symbolisiert dabei das momentane Standfeld des Turmes, und Z durchläuft alle möglichen Zielfelder. Zum Schluß wird in Zeile 6440 geprüft, ob auf dem vorgesehenen Zielfeld bereits eine eigene Figur steht oder ob das Feld nicht mehr zum eigentlichen Spielfeld gehört. In beiden Fällen wird das Flag FF auf 1 gesetzt. In Zeile 6450 wird zunächst geprüft, ob das Flag FF gesetzt ist. Ist dies der Fall, so ist das Zielfeld Z regelwidrig, und das folgende wird übersprungen. Da FF jedoch in der gesamten Schleife nicht zurückgesetzt wird, wird sie bis zu ihrem Ende durchlaufen, ohne daß weitere Stellungen ausgewertet werden. Nehmen wir jetzt aber an, daß FF noch Null ist und die Stellung somit legal. Dann wird in Zeile 6450 das erwogene Zielfeld ZF gleich Z gesetzt und die Stellungsbewertung ab Zeile 100 aufgerufen. Anschließend wird noch geprüft, ob auf Feld Z eine weiße Figur steht. Diese Figur darf zwar geschlagen werden, verhindert aber den weiteren Vormarsch des Turmes in diese Richtung. Somit wird das Flag FF in diesem Falle auf 1 gesetzt. Ich nehme an, daß das Prinzip der Zuggenerierung jetzt deutlich geworden ist. Es läßt sich relativ einfach auf die anderen Figuren übertragen, wenn man ihre speziellen Gangarten berücksichtigt.

Die Stellungsbewertung

Somit bleibt als letztes Problem nur noch die Bewertung der entstehenden Stellungen (Zeilen 100 bis 520). Wozu überhaupt eine Stellungsbewertung? Jeder, der schon einmal Schach gespielt hat, weiß, daß es nicht genügt, nur die Regeln zu beherrschen. Man muß zwischen guten und schlechten Zügen unterscheiden können, wenn man das Spiel gewinnen will. Diese Aufgabe hat die Stellungsbewertung.

Versuchen wir, uns die Funktionsweise mit Hilfe des Listings zu verdeutlichen. Zeile 100 führt zunächst einmal den erwogenen Zug auf dem Spielfeld aus. Dazu muß der Inhalt des Zielfeldes in der Variablen WZ gesichert werden, da er sonst verloren gehen würde. Anschließend wird der Zug ausgeführt. Zeile 110 gehört noch nicht zum eigentlichen Stellungsbewertungsprogramm. Hier wird überprüft, ob in der entstandenen Stellung der schwarze König im Schach steht und die Stellung somit illegal ist. Dies wird dadurch angezeigt, daß das Maschinensprache-Programm der Variablen AA den Wert 128 übergibt. Ist dies der Fall, so wird die Stellungsbewertung übersprungen und nach Zeile 520 verzweigt, wo der ausgeführte Zug zurückgenommen und in den Zuggenerator gesprungen wird. In Zeile 200 wird der Wert der Stellung (AA) zunächst um den ursprünglichen Wert des Zielfeldes erhöht. Hat auf dem Zielfeld eine weiße Figur gestanden, so ist diese geschlagen, und der Zug ist somit höherwertig, als wenn

Bild 1.
Die interne Spielfeld-darstellung

Das Spielfeld:		110	111	112	113	114	115	116	117	118	119
		128	128	128	128	128	128	128	128	128	128
		100	101	102	103	104	105	106	107	108	109
		128	128	128	128	128	128	128	128	128	128
8		90	91	92	93	94	95	96	97	98	99
		128	132	130	131	133	134	131	130	132	128
7		80	81	82	83	84	85	86	87	88	89
		128	129	129	129	129	129	129	129	129	128
6		70	71	72	73	74	75	76	77	78	79
		128	0	0	0	0	0	0	0	0	128
5		60	61	62	63	64	65	66	67	68	69
		128	0	0	0	0	0	0	0	0	128
4		50	51	52	53	54	55	56	57	58	59
		128	0	0	0	0	0	0	0	0	128
3		40	41	42	43	44	45	46	47	48	49
		128	0	0	0	0	0	0	0	0	128
2		30	31	32	33	34	35	36	37	38	39
		128	1	1	1	1	1	1	1	1	128
1		20	21	22	23	24	25	26	27	28	29
		128	4	2	3	5	6	3	2	4	128
		10	11	12	13	14	15	16	17	18	19
		128	128	128	128	128	128	128	128	128	128
		0	1	2	3	4	5	6	7	8	9
		128	128	128	128	128	128	128	128	128	128
		A	B	C	D	E	F	G	H		

- schwarzer Bauer: -10 beziehungsweise -9 oder -11 beim Schlagen
- schwarzer Springer: -21, -19, -12, -8, 8, 12, 19, 21
- schwarzer Turm: nx-10, nx10, nx-1, nx1
- schwarzer Läufer: nx-11, nx11, nx-9, nx9
- schwarze Dame: kombiniert Läufer- und Turmzüge
- schwarzer König: -11, -10, -9, -1, 1, 9, 10, 11

Bild 2.
Die verschiedenen Zugmöglichkeiten

Schwarz auf ein leeres Feld (WZ=0) gezogen hätte. In Zeile 210 wird geprüft, ob die Figur auf ihrem Ausgangsfeld angegriffen war. Dies ist der Fall, wenn der Wert der Felder X-9 oder X-11 gleich 1 ist, was bedeutet, daß die Figur auf ihrem Ausgangsfeld bedroht ist. Ein Wegziehen ist somit zu empfehlen, und der Wert AA wird um den Figurenwert erhöht. Das gleiche wird für das Zielfeld überprüft. Ist die Figur dort angegriffen, so wird ihr Wert von AA abgezogen, um zu zeigen, daß das Zielfeld ungünstig gewählt ist. Falls der berechnete Zug eine höhere Wertigkeit hat als der bisher beste Zug (BW), wird der soeben berechnete Zug als bester Zug eingeordnet, und die Ausgangs- und Zielkoordinaten (BA und BZ) werden ebenso wie die bisher beste Wertigkeit (BW) aktualisiert. Danach wird aus Zeile 500 nach 520 gesprungen, wo das Spielfeld zurückgesetzt und das Bewertungsprogramm verlassen wird. Falls der aktuelle Zug exakt die gleiche Wertigkeit hat wie der

bisher beste Zug, so entscheidet der Zufallsgenerator darüber, ob der neue Zug als bester eingelistet werden soll. Diese Einrichtung sorgt dafür, daß der Computer in einer bestimmten Stellung nicht immer die gleichen Züge macht, sondern ein Spiel variiert. Das Stellungsbewertungs-Programm ist für eigene erste Experimente besonders geeignet. Man kann zum Beispiel noch eigene Kriterien zur Stellungsbewertung einführen. Jedoch sollte man bedenken, daß das Bewertungsprogramm besonders zeitkritisch ist, da es pro Zugberechnung etwa 30- bis 50mal durchlaufen wird. Jede weitere Zeile macht sich also in der Rechenzeit des Computers bemerkbar. (Bernd Bettermann/ev)

Literaturhinweis: Computerschach von Ludek und Vas I. Kühnmond, Heyne-Buch Nr. 4704, Wilhelm Heyne Verlag, München, Taschenbuch, Preis: zirka 8,80 Mark. Dieses Buch bietet eine ausführliche und leicht verständliche Einführung in das Thema Computerschach. Außerdem beinhaltet es einige interessante Partien zwischen Mensch und Computer.

```

1 REM ***** SCHACHLADER C-64 ***** <055>
2 REM ** BY BERND BETTERMANN ** <181>
3 REM ** AMELIETHER STR.21 ** <068>
4 REM ** 3417 BODENFELDE ** <134>
5 REM ** TEL:05572/330 AB 18'00 ** <060>
6 REM ***** <137>
7 : <239>
10 IF PEEK(49408)<>162 THEN SYS 5517 <224>
20 GOTO 1000 <216>
70 : <046>
90 REM ----- STELLUNGSBEWERTUNG ----- <093>
90 : <066>
100 WZ=PEEK(ZF):POKE ZF,PEEK(X):POKE X,0 <086>
110 AA=USR(128):IF AA=128 THEN 520 <055>
120 : <096>
200 AA=AA+WZ <218>
210 IF PEEK(X-9)=1 OR PEEK(X-11)=1 THEN AA <175>
    =AA+PEEK(ZF)-128
220 IF PEEK(ZF-9)=1 OR PEEK(ZF-11)=1 THEN <238>
    AA=AA-PEEK(ZF)-128
400 : <122>
500 IF AA>BW THEN BW=AA:BA=X:BZ=ZF:GOTO 52 <141>
0
510 IF AA=BW AND RND(1)>.7 THEN BA=X:BZ=ZF <231>
520 POKE X,PEEK(ZF):POKE ZF,BA:RETURN <026>
1000 : <214>
1010 REM ----- VARIABLENDEFINITION ----- <100>
1020 : <234>
1030 FOR X=49152 TO 49171:POKE X,128:POKE <079>
    X+100,128:NEXT X
1040 FOR X=49172 TO 49251:POKE X,0:NEXT X <060>
1050 FOR X=49172 TO 49242 STEP 10:POKE X,1 <014>
    20:POKE X+9,128:NEXT X
1060 FOR X=49183 TO 49190:POKE X,1:POKE X+ <086>
    50,129:NEXT X
1070 RESTORE:FOR X=49173 TO 49180:READ A:P <079>
    OKE X,A:POKE X+70,A+128:NEXT X
1080 DATA 4,2,3,5,6,3,2,4 <114>
1090 POKE 785,0:POKE 786,193 <182>
2000 : <198>
2010 REM ----- SPIELFELD AUSGEBEN ----- <107>
2020 : <218>
2030 POKE 53200,0:POKE 53201,0:PRINT"CLR, <213>
    GREEN,DOWN,2SPACE"
2040 FOR Y=0 TO 1 STEP 1:PRINT CHR$(48+Y): <076>
    "R":Y=0 TO 8
2050 W=PEEK(49162+Y*10+X) <036>
2060 IF W=0 THEN PRINT"0":GOTO 2090 <243>
2070 IF W=128 THEN PRINT"RUSON":MID$( "BS <095>
    ZF=ASC(MID$(Z#,3,1)):"RVOFF":
2080 IF W<128 THEN PRINT MID$( "BSLTDK",W,1 <213>
    ):
2090 NEXT X:PRINT <220>
2100 NEXT Y:PRINT"(2SPACE)TTTTTTTT":PRINT" <088>
    ABCDEFGH(2DOWN)":BW=50
3000 : <182>
3010 REM ----- ZUGEBINGE WEISS ----- <163>
3020 : <202>
3030 INPUT" IHR ZUG :?# <222>
3040 IF LEN(Z#)>4 THEN PRINT"(UP)":GOTO 3 <144>
    030
3050 IF MID$(Z#,1,1)<"A"OR MID$(Z#,1,1)>"H <222>
    THEN 3030
3060 IF MID$(Z#,3,1)<"A"OR MID$(Z#,3,1)>"H <076>
    THEN 3030
3070 IF MID$(Z#,2,1)<"1"OR MID$(Z#,2,1)>"8 <016>
    THEN 3030
3080 IF MID$(Z#,4,1)<"1"OR MID$(Z#,4,1)>"8 <159>
    THEN 3030
3090 AF=ASC(LEFT$(Z#,1))-64+10*VAL(MID$(Z# <088>
    ",2,1))+49162
3100 ZF=ASC(MID$(Z#,3,1))-64+10*VAL(RIGHT$ <063>
    (Z#,1))+49162
3110 IF PEEK(AF)=0 OR PEEK(AF)>6 THEN 3030 <211>
3120 IF PEEK(ZF)=0 AND PEEK(ZF)<129 THEN 3 <176>
    030
3130 POKE ZF,PEEK(AF):POKE AF,0 <233>
4000 : <158>
4010 REM ----- SCHW. FIGURENPOS. HOLEN --- <206>
4020 : <186>
4030 FOR X=49173 TO 49250:IF PEEK(X)>128 T <205>
    HEN GOSUB 6000
4040 NEXT X <178>
5000 : <158>
5010 REM ----- SCHW. ZUG AUSFUHREN ----- <019>
5020 : <178>
5030 POKE BZ,PEEK(BA):POKE BA,0:PRINT"(DOW <081>
    N)WERT:";BW
5040 PRINT"(DOWN)ZUG: "; <011>
5050 BB=BA-49172 <217>
5055 IF BB>10 THEN BB=BB-10:GOTO 5055 <085>
5057 PRINT CHR$(BB+64): <085>
5060 PRINT CHR$(INT((BA-49172)/10)+49);" " <080>
    :
5070 BB=BZ-49172 <051>
5075 IF BB>10 THEN BB=BB-10:GOTO 5075 <087>
5077 PRINT CHR$(BB+64): <185>
5080 PRINT CHR$(INT((BZ-49172)/10)+49) <094>
5085 IF BW=50 THEN PRINT"(DOWN)ICH GEBE A

```

```

UF":END <085>
5090 FOR X=1 TO 1000:NEXT X:GOTO 2000 <091>
5100 : <250>
5110 END <030>
5120 : <016>
6000 IF PEEK(X)<>129 THEN 6100 <137>
6010 : <144>
6020 REM ----- BAUERNZUEGE ----- <026>
6030 : <046>
6040 IF X>49233 AND PEEK(X-10)=0 AND PEEK( <037>
    X-20)=0 THEN ZF=X-20:GOSUB 100
6050 IF PEEK(X-10)=0 THEN ZF=X-10:GOSUB 10 <123>
    0
6060 IF PEEK(X-9)=0 AND PEEK(X-9)<7 THEN Z <093>
    F=X-9:GOSUB 100
6070 IF PEEK(X-11)=0 AND PEEK(X-11)<7 THEN <033>
    ZF=X-11:GOSUB 100
6080 RETURN <040>
6100 IF PEEK(X)<>130 THEN 6200 <106>
6110 : <244>
6120 REM ----- SPRINGERZUEGE ----- <116>
6130 : <080>
6140 Z#="0B121921":FOR Z=1 TO 7 STEP 2 <158>
6150 Z#Z=VAL(MID$(Z#,2,2))
6160 IF PEEK(X+Z)<128 THEN ZF=X+Z:GOSUB <083>
    100
6170 IF PEEK(X-Z)<128 THEN ZF=X-Z:GOSUB <033>
    100
6180 NEXT Z:RETURN <115>
6200 IF PEEK(X)<>131 THEN 6400 <164>
6210 : <089>
6220 REM ----- LAEFERZUEGE ----- <090>
6230 : <209>
6240 FF=0:FOR Z=X+11 TO X+77 STEP 11:IF PE <110>
    EK(Z)>6 THEN FF=1
6250 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <074>
    )>0 THEN FF=1
6260 NEXT Z <128>
6270 FF=0:FOR Z=X-11 TO X-77 STEP-11:IF PE <128>
    EK(Z)>6 THEN FF=1
6280 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <221>
    )>0 THEN FF=1
6290 NEXT Z <170>
6300 FF=0:FOR Z=X+9 TO X+63 STEP+9:IF PEEK <158>
    (Z)>6 THEN FF=1
6310 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <125>
    )>0 THEN FF=1
6320 NEXT Z <200>
6330 FF=0:FOR Z=X-9 TO X-63 STEP-9:IF PEEK <188>
    (Z)>6 THEN FF=1
6340 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <043>
    )>0 THEN FF=1
6350 NEXT Z <230>
6360 RETURN <218>
6400 IF PEEK(X)<>132 THEN 6600 <066>
6410 : <171>
6420 REM ----- TURMZUEGE ----- <036>
6430 : <165>
6440 FF=0:FOR Z=X+10 TO X+70 STEP 10:IF PE <056>
    EK(Z)>6 THEN FF=1
6450 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <184>
    )>0 THEN FF=1
6460 NEXT Z <086>
6470 FF=0:FOR Z=X-10 TO X-70 STEP-10:IF PE <074>
    EK(Z)>6 THEN FF=1
6480 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <043>
    )>0 THEN FF=1
6490 NEXT Z <116>
6500 FF=0:FOR Z=X+1 TO X+7 STEP 1:IF PEEK( <104>
    Z)>6 THEN FF=1
6510 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <094>
    )>0 THEN FF=1
6520 NEXT Z <146>
6530 FF=0:FOR Z=X-1 TO X-7 STEP-1:IF PEEK( <134>
    Z)>6 THEN FF=1
6540 IF FF=0 THEN ZF=Z:GOSUB 100:IF PEEK(Z <166>
    )>0 THEN FF=1
6550 NEXT Z <176>
6560 RETURN <164>
6600 IF PEEK(X)<>135 THEN 6700 <012>
6610 : <221>
6620 REM ----- DAMENZUEGE ----- <236>
6630 : <126>
6640 GOSUB 6210:GOSUB 6410:RETURN <080>
6700 : <156>
6710 REM ----- KOENIGSZUEGE ----- <072>
6720 : <090>
6730 Z#="01091011":FOR Z=1 TO 7 STEP 2 <027>
6740 Z#Z=VAL(MID$(Z#,2,2))
6750 IF PEEK(X+Z)<7 THEN ZF=X+Z:GOSUB 10 <085>
    0
6760 IF PEEK(X-Z)<7 THEN ZF=X-Z:GOSUB 10 <236>
    0
6770 NEXT Z:RETURN <046>
6780 : <152>
6790 REM ----- ENDE ----- <245>
6800 : <186>

```

Listing 1.
Maschinroutinen
zum Schach-Programm.
Beachten Sie die Eingabe-
hinweise auf Seite 53

Listing 2. Schach — das Hauptprogramm