



Von A bis Z

Alphabetisches Sortieren mit dem 6502

Eine Kartei ist eine Datensammlung in meist alphabetischer Anordnung. Eine solche Anordnung ist zwar bei einer Computerkartei [1] zum Finden eines Stichwortes nicht mehr von so ausschlaggebender Bedeutung wie bei einer auf Karten vorliegenden, sie kann aber nützlich sein, wenn man eine Kartei durchmustern oder ausdrucken lassen möchte [2]. Das folgende Programm ordnet eine Kartei, die z. B. mit Hilfe des Programms aus Heft 24/78 erstellt wurde, in alphabetischer Reihenfolge. Im Gegensatz zu ähnlichen Programmen in BASIC benötigt dieses Maschinenprogramm z. B. für einen 4-KByte-Block nur wenige Sekunden.

Arbeitsweise des Programms

Das Programm vergleicht, am Textanfang beginnend, aufeinanderfolgende Zeilen miteinander. Als Zeilen werden hier bis zu 256 Zeichen lange Strings verstanden, die mit CR-Zeichen abgeschlossen sind. Stößt es auf zwei Zeilen, die in der falschen Reihenfolge stehen, so tauscht es die beiden aus: Zeile 1 wird in einen Pufferspeicher geladen, Zeile 2 auf den Platz von Zeile 1 verschoben und Zeile 1 hinter die neue Zeile 1 gehängt. Als Abschluß der Austauschaktion wird ein Flag gesetzt. Nun wird die neue Zeile 2 mit der nächsten Zeile verglichen, gegebenenfalls ausgetauscht usw.

Das Verfahren wird wiederholt, bis der Rechner das Textende erkennt (hex 3 E). Hier wird das Austauschflag abgefragt. Wenn im ganzen Durchgang, vom

Anfang bis zum Ende des Textes, keine Austauschoperation vorgenommen wurde, ist das Programm beendet – alle Zeilen stehen in der richtigen Reihenfolge und es erfolgt ein Sprung zum KIM-Monitor¹⁾. Andernfalls springt das Programm an seinen Anfang zurück (0000).

Zeilenvergleich

Es werden nur Buchstaben und Ziffern zum Vergleich der Zeilen benützt, alle anderen Zeichen werden überlesen. Dazu dient ein Unterprogramm an der Adresse 0098. So wird beim Vergleich der Zeilen

PREIS...DM 4.70
PREIS...DM 42.00

die zweite Zeile an den Anfang gestellt, da für den Rechner die Zeilen bis zur 4 identisch sind, danach die 2 aber kleiner ist als die 7. Sind zwei unterschiedlich lange Zeilen bis zum Ende der kürzeren identisch, so wird die kürzere jeweils vorangestellt; Vertauscht würden z. B.

FUNKSCHAU-SONDERHEFT
FUNKSCHAU

Zeilen mit Ziffern am Anfang werden nach den Ziffern sortiert und den mit Buchstaben beginnenden Zeilen vorangestellt. Wie in [1] geht das Programm davon aus, daß der Text keine Kleinbuchstaben enthält; will man dies ändern, so ist der Inhalt der Zelle 00A6 auf 7B abzuändern.

¹⁾ Beim AIM-65 sollte man den Buffer bei 0170 beginnen lassen, wenn die Strings nicht länger als etwa 100 Zeichen sind.

Speicherbedarf

Das Programm liegt im Speicherbereich 0000 bis 00AF. In Seite Null werden die Zellen E0...E8 als Arbeitsspeicher benötigt. Außerdem ist ein Pufferspeicher erforderlich, der hier bei 0100 beginnt und dessen Umfang die maximale Zeilenlänge bestimmt (maximal 256 Zeichen). Das Programm prüft nicht nach, ob es den Pufferspeicher überschreibt, hier ist also Vorsicht geboten.

Zusätzlich sind noch zwei Speicherstellen mit der Anfangsadresse des Textbereiches belegt. Im Programm sind dies die Zero-Page-Zellen B0 und B1. Sie werden nur bei Adresse 0002 abgefragt und können bei Bedarf leicht durch Modifikation von Speicherzelle 0003 auf einen anderen Platz der Seite Null gelegt werden. Eine Änderung der Sortierichtung ist zu erreichen, indem man bei Adresse 004E den Befehl BCS (hex B0) durch BCC (hex 90) ersetzt.

Das Programm läuft ohne Änderung auf dem KIM-1. Auf dem AIM-65 läßt es sich vorteilhaft zusammen mit den Editor-Befehlen des Monitorprogramms verwenden. Es ist darauf zu achten, daß der erste Text nicht mit CR beginnt und der letzte Text mit CR (hex 0D) und „>“ (hex 3E) abgeschlossen ist. Die Textanfangsadresse in den Zellen 00B0 und 00B1 zeigt hier auf 0202, was dem Datensuche-Programm [1] entspricht.

Hans-Herbert Hagedorn

Literatur

- [1] KIM auf Datensuche. FUNKSCHAU 1978, Heft 24
- [2] Textausdruck mit Baudot-Fernschreiber. FUNKSCHAU 1979, Heft 16

```

0000 A2 02 LDX #02          | 002E A4 E3 LDY E3          | 005F A0 00 LDY #00          | 008B 85 E4 STA E4
0002 B5 AF LDA AF,X       | 0030 B1 E7 LDA (E7),Y     | 0061 B1 E7 LDA (E7),Y     | 008D A5 E8 LDA E8
0004 95 E4 STA E4,X       | 0032 C9 3E CMP #3E        | 0063 91 E5 STA (E5),Y     | 008F 85 E6 STA E6
0006 95 E6 STA E6,X       | 0034 D0 09 BNE 003F      | 0065 C9 0D CMP #0D        | 0091 A5 E7 LDA E7
0008 CA DEX              | 0036 C5 E4 CMP E4         | 0067 F0 04 BEQ 006D        | 0093 85 E5 STA E5
0009 D0 F7 BNE 0002       | 0038 F0 73 BEQ 00AD      | 0069 C8 00 INY            | 0095 4C 0B 00 JMP 000B
000B A0 00 LDY #00        | 003A 85 E4 STA E4         | 006A 4C 61 00 JMP 0061    | 0098 C8 00 INY
000D 84 E2 STY E2         | 003C 4C 00 00 JMP 0000    | 006D C8 00 INY            | 0099 C9 30 CMP #30
000F 84 E3 STY E3         | 003F C9 0D CMP #0D        | 006E 98 00 TYA            | 009B 90 0C BCC 00A9
0011 B1 E7 LDA (E7),Y     | 0041 F0 0D BEQ 0050      | 006F 18 00 CLC            | 009D C9 3A CMP #3A
0013 E6 E7 INC E7         | 0043 20 98 00 JSR 0098    | 0070 65 E5 ADC E5         | 009F 90 0A BCC 00AB
0015 D0 02 BNE 0019       | 0046 90 E8 BCC 0030      | 0072 85 E7 STA E7         | 00A1 C9 41 CMP #41
0017 E6 E8 INC E8         | 0048 84 E3 STY E3        | 0074 A5 E6 LDA E6         | 00A3 90 04 BCC 00A9
0019 C9 0D CMP #0D        | 004A C5 E1 CMP E1         | 0076 69 00 ADC #00        | 00A5 C9 5B CMP #5B
001B D0 F4 BNE 0011       | 004C F0 CF BEQ 001D      | 0078 85 E8 STA E8         | 00A7 90 02 BCC 00AB
001D A4 E2 LDY E2         | 004E B0 3D BCS 008D      | 007A A0 00 LDY #00        | 00A9 18 00 CLC
001F B1 E5 LDA (E5),Y     | 0050 A0 00 LDY #00        | 007C B9 00 01 LDA 0100,Y  | 00AA 60 00 RTS
0021 C9 0D CMP #0D        | 0052 B1 E5 LDA (E5),Y     | 007F 91 E7 STA (E7),Y     | 00AB 38 00 SEC
0023 F0 68 BEQ 008D       | 0054 99 00 01 STA 0100,Y | 0081 C9 0D CMP #0D        | 00AC 60 00 RTS
0025 20 98 00 JSR 0098    | 0057 C9 0D CMP #0D        | 0083 F0 04 BEQ 0089      |
0028 90 F5 BCC 001F       | 0059 F0 04 BEQ 005F      | 0085 C8 00 INY            |
002A 84 E2 STY E2         | 005B C8 00 INY            | 0086 4C 7C 00 JMP 007C    |
002C 85 E1 STA E1         | 005C 4C 52 00 JMP 0052    | 0089 A9 FF LDA #FF        |

```

6502-Programm zum Sortieren von Textblöcken nach dem Alphabet