Listing 1

```
«081E» 10 MEMORY &89FF:LOAD"sort.rsx":CALL &8A0
       0
«026B» 20 MODE 1:INK 2,24,0
«0E1A» 30 PRINT:PRINT" Schneider 64K Microcompu
       ter  (v1)":PRINT
«0F46» 40 PRINT" ©1984 Amstrad Consumer Electro
       nics plc"
«0DD7» 50 PRINT"              and Locomotive Softw
       are Ltd.":PRINT
«0CB9» 60 PRINT" BASIC 1.0";:PEN 2:PRINT" with
       SORT-PACK 1.0":PRINT:PEN 1
«00FD» 70 NEW
```

Listing 2

```
«068C» 100 '************************************
«056A» 110 '*                                  *
«06E8» 120 '*   S O R T - P A C K              *
«057E» 130 '*                                  *
«0B90» 140 '*   eine RSX-Befehlssammlung       *
«0B7F» 150 '*   zum Sortieren eindimen-        *
«0B62» 160 '*   sionaler Stringfelder          *
«05A6» 170 '*                                  *
«06A3» 180 '*   von                            *
«05BA» 190 '*                                  *
«095E» 200 '*   Thomas Naumann                 *
«0A41» 210 '*   Bluecherstrasse 13             *
«0733» 220 '*   2300 Kiel 1                    *
«05E2» 230 '*                                  *
«069B» 240 '*   30. 7. 1987                    *
«05F6» 250 '*                                  *
«062D» 260 '************************************
«00D7» 270 '
«00E1» 280 '
«0BEB» 290 DATA 01,0A,8A,21,24,8A,CD,D1,BC,C9,2
       8,8A,C3,55,8A,C3,079E
«0BF5» 300 DATA C2,8A,C3,87,8D,C3,89,8B,C3,29,8
       C,C3,54,8D,C3,61,093A
«0B84» 310 DATA 8D,C3,47,8E,00,00,00,00,53,57,4
       1,D0,53,4F,52,D4,05A8
«0BC6» 320 DATA 53,4F,52,54,49,CE,41,43,43,4F,4
       D,50,41,4E,D9,43,05BD
«0B96» 330 DATA 4F,4D,50,41,52,C5,50,41,52,41,C
       D,45,58,54,52,C1,0639
«0C21» 340 DATA 4D,4F,56,C5,00,FE,03,C2,77,8A,D
       D,7E,00,B7,C8,47,079C
«0C33» 350 DATA DD,6E,02,DD,66,03,DD,5E,04,DD,5
       6,05,7E,F5,1A,77,070E
«0BD5» 360 DATA F1,12,23,13,10,F6,C9,21,7D,8A,C
       3,B8,8A,50,61,72,0758
```

```
«0BAB» 370 DATA 61,6D,65,74,65,72,66,65,68,6C,6
       5,72,0D,0A,07,00,0512
«0BF6» 380 DATA 21,96,8A,C3,B8,8A,57,65,72,74,6
       6,65,68,6C,65,72,075E
«0C1E» 390 DATA 0D,0A,07,00,21,AA,8A,C3,B8,8A,4
       4,69,73,6B,66,65,05CE
«0C4E» 400 DATA 68,6C,65,72,0D,0A,07,00,7E,B7,C
       8,CD,5A,BB,23,C3,068E
«0C6E» 410 DATA B8,8A,FE,02,C2,77,8A,DD,5E,00,D
       D,56,01,ED,53,42,07F6
«0C70» 420 DATA 8B,DD,6E,02,DD,66,03,22,40,8B,B
       7,CB,1A,CB,1B,01,068E
«0C1B» 430 DATA 00,00,69,60,E5,D5,5D,54,19,19,E
       D,5B,40,8B,19,22,05B4
«0CC5» 440 DATA 3E,8B,D1,E1,E5,D5,E5,DD,E1,19,E
       5,FD,E1,5D,54,19,0A7E
«0CC3» 450 DATA 19,ED,5B,40,8B,19,ED,5B,3E,8B,C
       D,54,8C,DA,1E,8B,0786
«0CBC» 460 DATA CD,44,8B,D1,E1,B7,ED,52,D2,E4,8
       A,C3,20,8B,D1,E1,0AA4
«0C6B» 470 DATA 2A,42,8B,B7,ED,52,2B,B7,ED,42,3
       8,06,28,04,03,C3,062E
«0CE2» 480 DATA E2,8A,3E,01,BB,DA,DA,8A,3D,BA,C
       2,DA,8A,C9,00,00,088A
«0CA7» 490 DATA 00,00,00,00,F5,C5,E5,D5,FD,E5,0
       6,03,CD,6C,8A,FD,081F
«0CFB» 500 DATA 21,C6,8B,D1,D5,FD,7E,02,B7,CA,8
       2,8B,FD,6E,00,FD,098B
«0CBB» 510 DATA 66,01,47,19,10,FD,E5,DD,E5,D1,F
       D,6E,00,FD,66,01,081B
«0BED» 520 DATA 47,19,10,FD,47,D1,CD,6C,8A,FD,2
       3,FD,23,FD,23,C3,086B
«0C25» 530 DATA 53,8B,FD,E1,D1,E1,C1,F1,C9,CB,4
       7,C2,77,8A,CB,3F,0AC8
«0C2E» 540 DATA 47,FD,21,C6,8B,B7,CA,BB,8B,DD,7
       E,00,FD,77,02,DD,092B
«0C03» 550 DATA 7E,02,FD,77,00,DD,7E,03,FD,77,0
       1,DD,23,DD,23,DD,07A4
«0C1D» 560 DATA 23,DD,23,FD,23,FD,23,FD,23,10,D
       E,AF,FD,77,00,FD,0891
«0B23» 570 DATA 77,01,FD,77,02,C9,00,00,00,00,0
       0,00,00,00,00,00,02B7
«0AAD» 580 DATA 00,00,00,00,00,00,00,00,00,00,0
       0,00,00,00,00,00,0000
«0AB7» 590 DATA 00,00,00,00,00,00,00,00,00,00,0
       0,00,00,00,00,00,0000
«0AC1» 600 DATA 00,00,00,00,00,00,00,00,00,00,0
       0,00,00,00,00,00,0000
«0ACB» 610 DATA 00,00,00,00,00,00,00,00,00,00,0
       0,00,00,00,00,00,0000
«0AD5» 620 DATA 00,00,00,00,00,00,00,00,00,00,0
       0,00,00,00,00,00,0000
«0BA6» 630 DATA 00,00,00,00,00,00,00,00,00,FE,0
       3,C2,77,8A,DD,6E,040F
«0C11» 640 DATA 00,DD,66,01,22,52,8C,DD,5E,02,D
       D,56,03,DD,6E,04,0606
«0C36» 650 DATA DD,66,05,CD,54,8C,2A,52,8C,38,0
       3,AF,77,C9,3E,01,0666
«0C8A» 660 DATA 77,C9,00,00,F5,C5,D5,E5,DD,E5,F
       D,E5,EB,7E,23,4E,0A32
«0C55» 670 DATA 23,46,EB,5F,1C,C5,FD,E1,56,14,2
       3,4E,23,46,C5,DD,0758
«0C5B» 680 DATA E1,0E,00,CB,41,28,06,CB,81,7C,C
       3,92,8C,DD,7E,00,072D
«0CAB» 690 DATA DD,23,15,CA,C1,8C,45,CD,D5,8C,D
       2,91,8C,CB,C1,7C,0996
«0CD8» 700 DATA 65,68,CD,3A,8D,47,CB,49,CA,A1,8
       C,CB,89,7D,C3,B8,08FF
«0D01» 710 DATA 8C,FD,7E,00,FD,23,1D,CA,CB,8C,D
       5,EB,CD,D5,8C,D2,0A25
«0D04» 720 DATA B6,8C,CB,C9,7C,5D,EB,D1,CD,3A,8
       D,B8,CA,73,8C,38,09B8
«0D0C» 730 DATA 0A,FD,E1,DD,E1,E1,D1,C1,F1,B7,C
       9,FD,E1,DD,E1,E1,0D07
«0CE3» 740 DATA D1,C1,F1,37,C9,FE,41,38,0E,FE,5
       B,DA,36,8D,FE,61,095D
«0C9D» 750 DATA 38,05,FE,7B,DA,36,8D,21,80,8D,B
       E,20,07,26,41,2E,05FB
«0C82» 760 DATA 65,C3,38,8D,23,BE,20,07,26,4F,2
       E,65,C3,38,8D,23,05A8
«0B67» 770 DATA BE,20,07,26,55,2E,65,C3,38,8D,2
       3,BE,20,07,26,61,050A
«0BA9» 780 DATA 2E,65,C3,38,8D,23,BE,20,07,26,6
       F,2E,65,C3,38,8D,05D3
«0B91» 790 DATA 23,BE,20,07,26,75,2E,65,C3,38,8
       D,23,BE,20,07,26,04EC
```

```
«0BEB» 800 DATA 73,2E,73,C3,38,8D,B7,C9,37,C9,E
       5,F5,21,60,8D,AF,08B3
«0C19» 810 DATA BE,20,0E,F1,FE,61,38,0A,FE,7B,3
       0,06,CB,AF,C3,52,07BC
«0C11» 820 DATA 8D,F1,E1,C9,FE,01,C2,77,8A,DD,7
       E,00,32,60,8D,C9,092D
«0C01» 830 DATA 00,FE,01,C2,77,8A,DD,6E,00,DD,6
       6,01,7E,FE,07,C2,0796
«0BD1» 840 DATA 90,8A,23,5E,23,56,21,80,8D,EB,0
       1,07,00,ED,B0,C9,069B
«0C64» 850 DATA 7B,5C,7D,5B,7C,5D,7E,FE,04,C2,7
       7,8A,DD,6E,00,DD,07F3
«0BF7» 860 DATA 66,01,22,52,8C,DD,6E,02,DD,66,0
       3,22,3E,8B,DD,6E,0630
«0BF2» 870 DATA 04,DD,66,05,22,42,8B,E5,DD,6E,0
       6,DD,66,07,22,40,061D
«0BE4» 880 DATA 8B,E1,01,00,00,B7,ED,42,CA,33,8
       E,2B,EB,21,00,00,0615
«0CB1» 890 DATA E5,D5,EB,B7,ED,52,CB,7C,D1,E1,C
       2,F6,8D,E5,D5,19,0BAC
«0C58» 900 DATA B7,CB,1C,CB,1D,4D,44,09,09,EB,2
       A,40,8B,19,EB,2A,0637
«0C56» 910 DATA 3E,8B,CD,54,8C,D1,E1,D2,F0,8D,0
       3,69,60,C3,C0,8D,0953
«0C1B» 920 DATA 0B,59,50,C3,C0,8D,E5,EB,2A,52,8
       C,73,23,72,2A,42,0710
«0C5F» 930 DATA 8B,B7,ED,52,CA,1F,8E,5D,54,19,1
       9,4D,44,2A,42,8B,0663
«0C43» 940 DATA EB,2A,40,8B,19,19,19,2B,5D,54,1,
       3,13,13,ED,B8,D1,05B6
«0C4F» 950 DATA 6B,62,19,19,EB,2A,40,8B,19,EB,2
       A,3E,8B,01,03,00,04DA
«0C7B» 960 DATA ED,B0,C9,2A,40,8B,EB,2A,3E,8B,0
       1,03,00,ED,B0,2A,0704
«0C9B» 970 DATA 52,8C,AF,77,23,77,C9,FE,03,C2,7
       7,8A,DD,4E,00,DD,0833
«0C63» 980 DATA 46,01,21,00,00,B7,ED,42,C8,DD,5
       E,02,DD,56,03,DD,0666
«0C91» 990 DATA 6E,04,DD,66,05,E5,B7,ED,52,E1,3
       8,03,ED,B0,C9,09,0820
«0C48» 1000 DATA 2B,EB,09,2B,EB,ED,B8,C9,00,00,
       00,00,00,00,00,00,04A3
«0823» 1010 zeile=1:adr=35328
«06CB» 1020 FOR loop1%=1 TO 72
«095C» 1030 summe=0:FOR loop2%=1 TO 16
«10A8» 1040 READ byte$:byte=VAL("&"+byte$):POKE
        adr,byte
«129D» 1050 summe=summe+byte:adr=adr+1:NEXT loo
       p2%
«10F0» 1060 READ pruefsum$:pruefsum=VAL("&"+pru
       efsum$)
«1442» 1070 IF pruefsum<>summe THEN PRINT"Fehle
       r in Zeile :";zeile:END
«0ADB» 1080 zeile=zeile+1:NEXT loop1%
«07AC» 1090 SAVE"sort.rsx",b,35328 ,1145
```

Listing 3

```
«0CBC» 10 'Falls SORT-PACK nicht geladen ist...
«1190» 20 IF PEEK(&8A00)<>1 AND PEEK(&8A01)<>10
        THEN MEMORY &89FF:LOAD"sort.rsx":CALL &
       8A00
«07E8» 30 MODE 2:PRINT"|MOVE-Beispiel":PRINT
«1BB4» 40 DIM d$(500):PRINT"Zuerst baue ich mir
        501 Stringvariablen d$(0) bis d$(500) -
        Bitte warten"
«0149» 50 ZONE 7
«093B» 60 FOR i=0 TO 500:FOR j=1 TO 5
«045E» 70 d$(i)="     "
«0AFB» 80 MID$(d$(i),j,1)=CHR$(65+RND*25)
«0110» 90 NEXT
«134F» 100 IF i MOD 100=0 THEN IF i<>0 THEN PRI
       NT i;"Strings geschafft."
«0124» 110 NEXT
«0DF9» 120 PRINT CHR$(7);"O. k. - Tastendruck!"
       :WHILE INKEY$="":WEND
«0F50» 130 PRINT"Jetzt vernichte ich d$(34) mit
        BASIC - JETZT"
«1141» 140 FOR i=34 TO 499:MID$(d$(i),1,5)=d$(i
       +1):NEXT:PRINT"FERTIG!"
```

```
«07C8» 150 d$(500)="Bloedsinn"
«1339» 160 PRINT CHR$(7);"Tastendruck fuer das
       |MOVE-Demo!":WHILE INKEY$="":WEND
«0E11» 170 PRINT"Und jetzt dasselbe mit |MOVE -
        JETZT!"
«0BDC» 180 |MOVE,@d$(35),@d$(34),466*3:PRINT"FE
       RTIG!"
«2965» 190 PRINT:PRINT"Ueberzeugt? ... Und nun
       warten wir auf das Ende der Garbage Coll
       ection...":PRINT".... (<CTRL><SHIFT><ESC
       > ist schneller) ...."
«0166» 200 END
```

Listing 4

```
«0CBC» 10 'Falls SORT-PACK nicht geladen ist...
«1190» 20 IF PEEK(&8A00)<>1 AND PEEK(&8A01)<>10
        THEN MEMORY &89FF:LOAD"sort.rsx":CALL &
       8A00
«232C» 30 a$="Ich bin a$":b$="Ich bin b$":PRINT
       "a$ = ";a$:PRINT"b$ = ";b$:PRINT:PRINT"|
       SWAP,@a$,@b$,5":PRINT:|SWAP,@a$,@b$,3:PR
       INT"a$ = ";a$:PRINT"b$ = ";b$
```

Listing 5

```
«0CBC» 10 'Falls SORT-PACK nicht geladen ist...
«1190» 20 IF PEEK(&8A00)<>1 AND PEEK(&8A01)<>10
        THEN MEMORY &89FF:LOAD"sort.rsx":CALL &
       8A00
«100C» 30 MODE 2:PRINT"Beispiel zu |EXTRA wie i
       n der Anleitung"
«0D95» 40 sonder$="][\{|¦¯":|EXTRA,@sonder$
```

Listing 6

```
«0CBC» 10 'Falls SORT-PACK nicht geladen ist...
«1190» 20 IF PEEK(&8A00)<>1 AND PEEK(&8A01)<>10
        THEN MEMORY &89FF:LOAD"sort.rsx":CALL &
       8A00
«1676» 30 DEFINT a-z:result=0:a$="Ich bin klein
       er":b$="Wir sind groesser"
«1FC1» 40 PRINT"Ich vergleiche ";CHR$(34);a$;CH
       R$(34);" und ";CHR$(34);b$;CHR$(34):|COM
       PARE,@a$,@b$,@result:PRINT result
«1FCB» 50 PRINT"Ich vergleiche ";CHR$(34);b$;CH
       R$(34);" und ";CHR$(34);a$;CHR$(34):|COM
       PARE,@b$,@a$,@result:PRINT result
```

Listing 7

```
«00EF» 10 MODE 2
«3C3F» 20 IF PEEK(&8A00)=1 AND PEEK(&8A01)=10 T
       HEN PRINT"Wenn dies der erste Durchlauf
       des Beispiels ist, sind die Umlaute noch
        nicht    initialisiert worden. Geben Si
       e GOSUB 10000 und dann RUN 30 ein!":END
«145E» 30 IF PEEK(&8A00)<>1 AND PEEK(&8A01)<>10
        THEN SYMBOL AFTER 256:MEMORY &89FF:LOAD
       "sort.rsx":CALL &8A00:GOSUB 9990
«031B» 40 DEFINT a-z:ZONE 26
«4A18» 50 PRINT"Jetzt bereite ich - abweichend
       vom Anleitungstext - 21 Stringvariablen
       vor!":PRINT"Ich werde dabei einige Varia
       blen mitschleifen (ACCOMPANY)":PRINT:PRI
       NT"Beachten Sie, wie die deutschen Umlau
       te einsortiert werden!":PRINT
«08DB» 60 DIM a$(20),b(20),c(20),d$(20),e!(20)
«0462» 70 FOR i=0 TO 20
«0453» 80 FOR j=1 TO 5
«1781» 90 a$(i)=a$(i)+CHR$(65+RND*62):d$(i)=d$(
       i)+CHR$(65+RND*62)
«0236» 100 NEXT j
«1424» 110 b(i)=INT(RND*30):c(i)=INT(RND*30):e!
       (i)=RND*30
«0249» 120 NEXT i
«0C24» 130 |ACCOMPANY,@b(0),2,@c(0),2,@d$(0),3,
       @e!(0),5
```

```
«1E45» 140 FOR i=0 TO 20:PRINT CHR$(24);a$(i);C
       HR$(24);:PRINT USING" ## ## ";b(i);c(i);
       :PRINT d$(i);:PRINT USING" ##.#";e!(i);:
       PRINT,:NEXT
«0736» 150 PRINT"*** SORTIEREN BEGINNT"
«1008» 160 |SORT,@a$(0),21:REM Von 0 bis 20 sin
       d es 21 Elemente
«07C3» 170 PRINT"*** SORTIEREN HOERT AUF"
«1E6D» 180 FOR i=0 TO 20:PRINT CHR$(24);a$(i);C
       HR$(24);:PRINT USING" ## ## ";b(i);c(i);
       :PRINT d$(i);:PRINT USING" ##.#";e!(i);:
       PRINT,:NEXT
«183A» 190 INPUT"Nochmal: <J>/andere Taste: Abb
       ruch: ",e$:IF LOWER$(e$)="j"THEN MODE 2:
       RUN 30 ELSE END
«0959» 9990 'Deutscher Zeichensatz
«205A» 10000 SYMBOL AFTER 91:SYMBOL 91,66,24,60
       ,102,102,126,102,0:SYMBOL 92,68,56,108,1
       98,198,108,56,0:SYMBOL 93,102,0,102,102,
       102,102,60,0:SYMBOL 123,108,0,120,12,124
       ,204,118,0:SYMBOL 124,0,102,0,60,102,102
       ,60,0
«167E» 10010 SYMBOL 125,0,102,0,102,102,102,62,
       0:SYMBOL 126,60,102,102,108,102,102,108,
       96:KEY DEF 19,1,125,93:KEY DEF 17,1,123,
       91:KEY DEF 24,1,94,126
«099F» 10020 SYMBOL 255,255,129,129,129,129,129
       ,129,255
```

```
«1F5E» 10030 KEY DEF 55,1,&76,&56,185:KEY DEF 9
       ,1,186,186,189:KEY DEF 54,1,&62,&42,187:
       KEY DEF 69,1,&61,&41,188:KEY DEF 60,1,&7
       3,&53,190:KEY DEF 50,1,&72,&52,191:KEY D
       EF 58,1,&65,&45,202
«0609» 10040 SYMBOL 152,0,0,0,51,122,252,72,0
«0138» 10050 RETURN
```

Listing 8

```
«0CBC» 10 'Falls SORT-PACK nicht geladen ist...
«1190» 20 IF PEEK(&8A00)<>1 AND PEEK(&8A01)<>10
        THEN MEMORY &89FF:LOAD"sort.rsx":CALL &
       8A00
«0226» 30 DIM a$(4)
«1866» 40 a$(0)="erster":a$(1)="ich bin Nr. zwe
       i":a$(2)="ich drei":a$(3)="vierter"
«0EE4» 50 stelle%=0:b$="Mal gucken, wo ich land
       e"
«131F» 60 FOR i=0 TO 4:PRINT"a$(";i;") = ";a$(i
       ):NEXT:PRINT"**** Einsortieren ****"
«09FF» 70 |SORTIN,@a$(0),4,@b$,@stelle%
«0B8E» 80 FOR i=0 TO 4:PRINT"a$(";i;") = ";a$(i
       ):NEXT
«0783» 90 PRINT"Stelle="; stelle%
```

Listing 9

```
«00EF» 10 MODE 2
«447C» 20 IF PEEK(&8A00)=1 AND PEEK(&8A01)=10 T
       HEN PRINT"Dieses Demo soll zeigen, wie d
       ie Startzeile aussehen kann. Damit es ue
       berzeugt, muessen Sie den Computer zurue
       cksetzen, ehe Sie dieses Beispiel ausfue
       hren.":END
«1C19» 30 SYMBOL AFTER 256:OPENOUT"d":MEMORY HI
       MEM-1:CLOSEOUT:MEMORY &89FF:LOAD"sort.rs
       x":CALL &8A00:SYMBOL AFTER 90:'oder irge
       ndein anderer Wert
«0DCA» 40 PRINT"Die Startzeile koennte heissen:
       ":PRINT
«3830» 50 PRINT"10 SYMBOL AFTER 256:OPENOUT"+CH
       R$(34)+"d"+CHR$(34)+":MEMORY HIMEM-1:CLO
       SEOUT:MEMORY &89FF:LOAD"+CHR$(34)+"sort.
       rsx"+CHR$(34)+":CALL &8A00:SYMBOL AFTER
       90:'oder irgendein anderer Wert"
«2909» 60 PRINT:PRINT"Outputbuffer und ein beli
       ebig festlegbarer Bereich fuer selbstdef
       inierte Zeichenist bereitgestellt":PRINT
```

Listing 10

```
«09F1» 10 MODE 2:PRINT"Memory-Demo":PRINT"-----
       -------":PRINT
«0DE4» 20 PRINT"Himem ohne Outputbuffer: &";HEX
       $(HIMEM)
«079E» 30 PRINT:SYMBOL AFTER 256:OPENOUT"d":MEM
       ORY HIMEM-1:CLOSEOUT
«0D97» 40 PRINT"Himem mit Outputbuffer: &";HEX$
       (HIMEM)
«029B» 50 PRINT:SYMBOL AFTER 0
«12FD» 60 PRINT"Himem mit Outputbuffer und SYMB
       OL AFTER 0: &";HEX$(HIMEM)
«1360» 70 PRINT:MEMORY HIMEM-&480:PRINT"minus &
       480 Bytes fuer SORT-PACK: &";HEX$(HIMEM)
«10A5» 80 PRINT:PRINT"Daher wurde SORT-PACK ab
       &8A00 assembliert":PRINT
```