

ATARI®

ATARI PROGRAMM

DXG 5730
Diskette

SUPERSORT

c 1983 Jegliche Rechte vorbehalten
ATARI ELEKTRONIK - Vertriebsges. mbH

ATARINSIDE



ATARI PROGRAMM

DXG 5730
Diskette

SUPERSORT

c 1983 Jegliche Rechte vorbehalten
ATARI ELEKTRONIK - Vertriebsges. mbH

ATARINSIDE

INHALTSVERZEICHNIS

1. EINLEITUNG

- 1.1. Programmbeschreibung
- 1.2. Hardware

2. BEDIENUNGSANLEITUNG

- 2.1. Vorbereitungen
- 2.2. Programmstart
- 2.3. BASIC-Programmentwurf
 - 2.3.1. Recordlänge
 - 2.3.2. Schlüsselfeldlänge
 - 2.3.3. M - Verteilungsmaß
 - 2.3.4. Startadressen
- 2.4. Parameterübergabe
- 2.5. Format des Sortier-Strings

3. PROGRAMMSTÖRUNGEN

4. TECHNISCHE INFORMATIONEN (für Fortgeschrittene)

5. ANWENDUNGSBEISPIEL

Dieses Programm ist nur für den persönlichen Gebrauch bestimmt. Die Anwendung von SUPERSORT in Programmen, die kommerziell genutzt werden, erfordert die Genehmigung von ATARI.

ATARINSIDE

einige Programmzeilen erweitert, damit SUPERSORT aufgerufen werden kann. Darin sollten enthalten sein:

1. Ein Unterprogrammaufruf mit verschiedenen Parametern, wie z.B. Record- und Schlüsselfeldlänge.
2. Ein langer String (Sortier-String), der alle Records enthält, die sortiert werden sollen.
3. Wahlweise ein Ausgabeformular für die sortierten Records.

Ein Beispiel für einen Programmaufruf wird im 5. Kapitel gegeben. Zeile 240 bis 270 enthalten die Parameter, mit denen SUPERSORT aufgerufen wird.

Wenn Ihre Vorbereitungen soweit abgeschlossen sind, können Sie SUPERSORT starten, wie im nächsten Kapitel beschrieben.

2.2. Programmstart

*** Für ATARI Privatcomputer

Wenn der ATARI Computer kein eingebautes BASIC hat, wird das BASIC-Steckmodul in den Modulschacht eingesteckt.

1. Schalten Sie Ihren Computer aus.
2. Schalten Sie die Diskettenstation ein.
3. Legen Sie die Programm-Diskette in das Laufwerk ein und Schließen Sie den Diskettenschacht.
4. Schalten Sie Ihren Computer ein. Das Programm wird nun in den Speicherbereich geladen und wird automatisch gestartet.

Es wird "unterhalb" des BASIC abgelegt, bei der Adresse \$1D5A (7514). Es ist nicht auflistbar. Das Vorhandensein kann jedoch mit Hilfe der FRE(0)-Funktion überprüft werden (es stehen Ihnen 865 Byte weniger zur Verfügung), oder mit Hilfe von PEEK(7514) - es sollte in dieser Speicherzelle der Wert 104 vorhanden sein.

Um die Daten zu sortieren, starten Sie Ihr BASIC-Programm wie gewöhnlich.

Um SUPERSORT aufrufen zu können, muß Ihr BASIC-Programm folgende 6 Parameter enthalten:

1. Recordlänge
2. Schlüsselfeldlänge
3. Verteilungsmaß
4. Startadresse von SUPERSORT
5. Startadresse des Sortier-Strings
6. Startadresse des letzten Records im Sortier-String

Die einzelnen Parameter werden in den folgenden Abschnitten genauer beschrieben, ergänzt mit einer Erklärung, wie diese im BASIC-Programm zu verwenden sind.

2.3.1. Recordlänge

Dieser Parameter bestimmt die feste Länge des Records (zur Erklärung: Ein Record ist ein Satz von Daten). Mit dem POKE-Befehl muß dieser Wert unter der Adresse \$600 (dezimal 1536) abgelegt werden. Alle Records müssen die selbe Länge aufweisen, ebenso wie die dazugehörigen Datenfelder. Benötigen einzelne Daten weniger Platz als im Datenfeld vorgesehen, so ist das Datenfeld mit Leerzeichen aufzufüllen. Ein Record enthält zwar eine Reihe von Datenfeldern, wird jedoch als eine Einheit betrachtet. Wird eine Adressenliste z.B. als Record aufgebaut, so enthält dieser Record Datenfelder für den Namen, den Wohnort, der Postleitzahl, den Straßennamen und der Hausnummer.

!----- Record 1 -----!	!----- Record 2 -----!
! Feld 1! Feld 2! Feld 3 ! !	! Feld 1! Feld 2! Feld 3! !
! (9) (15) (9) !	! ! !
!----!---- 31 Zeichen -----!	! ! !
! ! !	! ! !
Schlüsselfeld	Schlüsselfeld

Beachten Sie, daß das erste Feld jedes Records ein Schlüsselfeld 1, nachdem die Records sortiert werden. SUPERSORT sortiert nur nach diesem Schlüsselfeld. Das Schlüsselfeld kann maximal 185 Byte lang sein. In unserem Beispiel mit der Adressenliste kann zum Beispiel der Name in dem Schlüsselfeld stehen. SUPERSORT kann somit alle abgespeicherten Adressen alphabetisch nach dem Namen sortieren. Ist die Länge des Schlüsselfeldes mit 9 festgelegt worden, so ist bei Namen mit weniger als 9 Buchstaben das Feld mit Leerzeichen aufzufüllen.

2.3.2. Schlüsselwortlänge

Dieser Parameter spezifiziert, wie schon beschrieben, die Länge des ersten Feldes im Record. Legen Sie den Wert mit Hilfe des POKE-Befehls unter der Adresse \$601 (dezimal 1537) ab. Zeile 240 im Beispielprogramm zeigt diesen Weg. Wenn der gesamte Record aus nur einem Datenfeld besteht, so sind Recordlänge und Schlüsselwortlänge identisch.


```

SCHMIDT PARKSTRASSE HAMBURG MEIER SCHLOSSALLEE BERLIN
!      !      !      !      !      !      !
! Feld 1 ! Feld 2 ! Feld 3 ! Feld 1 ! Feld 2 ! Feld 3 !
!      !      !      !      !      !      !
!----- Record 1 -----!----- Record 2 -----!
!      !      !      !      !      !      !
!----- Sortier-String -----!

```

Die Namen "SCHMIDT" und "MEIER" stehen in den Schlüsselfeldern, nach denen SUPERSORT sortiert. Entspricht die Namenlänge nicht der Feldlänge, so ist mit Leerzeichen aufzufüllen (Zeile 100 und 160 im Beispielprogramm).

Nach Beendigung des Sortiervorganges stehen die Ergebnisse wieder im Sortier-String. Benötigen Sie jedoch auch den originalen Sortier-String, so ist dieser unter einem anderen Stringnamen vorher abzulegen.

3. PROGRAMMSTÖRUNGEN

SUPERSORT enthält keine Fehlerbehandlungsroutinen - daher können in Zusammenhang mit Ihrem BASIC-Programm nur die üblichen Fehlermeldungen auftreten, wie sie im BASIC Reference-Manual beschrieben sind. Auf die Eingabe falscher Werte kann Ihr Programm mit folgenden drei Ergebnissen reagieren:

1. SUPERSORT beläßt die Daten im unsortierten Zustand.
2. Die Daten werden durcheinander gewirbelt.
3. Das Programm "stürzt ab".

Um sich gegen solche Ereignisse zu schützen, empfehlen wir, vor dem Programmstart das Programm auf Diskette oder Cassette zu speichern. Ergeben sich dann irgendwelche Probleme, können Sie Ihr BASIC-Programm ohne Schwierigkeiten wieder in den Speicherbereich laden und die notwendigen Korrekturen vornehmen.

Durch Aufruf von DOS II werden in der Regel alle im Speicherbereich vorhandenen Daten gelöscht. Damit SUPERSORT und Ihr BASIC-Programm nicht verlorengehen, enthält die Programmdiskette einen MEM.SAV. Wie Ihnen sicherlich bekannt ist, werden mit dem MEM.SAV bestimmte Abschnitte des RAM-Bereichs beim Aufruf von DOS auf Diskette gesichert und danach wieder in den Speicher geladen. Einziger Nachteil ist der erhöhte Zeitaufwand bei dieser Arbeitsweise. Benutzen Sie keinen MEM.SAV, so beachten Sie, daß mit dem Aufruf von DOS II SUPERSORT gelöscht wird. Versuchen Sie in diesem Fall SUPERSORT aufzurufen, wird Ihr System "abstürzen". SUPERSORT ist dann, wie in 2.2. beschrieben, wieder zu laden.

Das Benutzen der Taste SYSTEM RESET hat keinen Einfluß auf SUPERSORT.

4. TECHNISCHE INFORMATIONEN (für Fortgeschrittene)

Die folgenden Hinweise sind nur für den erfahrenen Programmierer gedacht, der auch über das ASSEMBLER-Sprachmodul verfügt. Auf der Programmdiskette sind die beiden Quellprogramme für SUPERSORT (Filename: SUPER2.ASM) und für den AUTORUN.SYS (Filename: ATRSYS.ASM) vorhanden. Die ersten beiden Vorschläge "erkaufen" sich Ihren Vorteil mit einer Verlangsamung der Sortiergeschwindigkeit.

a. SUPERSORT kann so modifiziert werden, daß auch Records mit mehr als 185 Byte sortiert werden können. Die Variable TREC wird als Zeichenspeicher benutzt und belegt Speicherplatz auf "Seite 6". Da sie diesen Bereich mit anderen Variablen teilen muß, ist ihre nutzbare Länge auf 185 Byte begrenzt. Wird für die Variable TREC ein anderer Speicherbereich gewählt, kann die maximale Recordlänge erhöht werden.

b. Wie schon beschrieben, akzeptiert SUPERSORT nur Ein-Byte-Daten, selbst wenn TREC in einen anderen Bereich verlegt wird. Um zu erreichen, daß SUPERSORT auch Records mit Doppel-Byte-Informationen verarbeitet, muß die Variable RLENG (Recordlänge), die nur während der Addition benutzt wird, benutzt werden. Da Addition sowieso mit Doppel-Bytes durchgeführt werden, muß lediglich die höherwertige Byte-Addition von "ADC £0" in "ADC RLENG" umgeformt werden.

c. SUPERSORT benutzt keine Unterprogramme, da die Parameter-Übergabe immer sehr zeitraubend ist. Ist es notwendig, Speicherplatz einzusparen, so kann durch Verwendung von Unterprogrammen SUPERSORT um einen Faktor 1/3 bis 1/2 verkleinert werden.

Es lassen sich weitere 155 Bytes einsparen, indem ein Sortieralgorithmus entfernt wird (Label Q9 bis L21A). Sie können in diesem Fall nur den QUICKSORT-Teil von SUPERSORT benutzen - der Parameter Verteilungsmaß ist dann auf M=1 zu setzen.

5. ANWENDUNGSBEISPIEL

Auf den folgenden Seiten finden Sie ein Beispiel, wie SUPERSORT in BASIC-Programme integriert werden kann. Das abgebildete Programm befindet sich auch auf der Programmdiskette und kann mit RUN"D:SDRT.BSP" gestartet werden. Geben Sie als Recordlänge den Wert 30, als maximale Recordanzahl den Wert 8 und für die Schlüsselfeldlänge den Wert 6 ein. Geben Sie dann folgende 8 Records ein:

BBBB, AAABBB, ABDD, AAAA, AD, AACD, ABCD, ABC

Jede Eingabe beenden Sie bitte mit RETURN. Nach Eingabe

des achten Records wird automatisch sortiert, da die maximale Recordsanzahl erreicht wurde. Nach Ende des Sortiervorganges werden alle Records in sortierter und unsortierter Form aufgelistet.

Es erfolgt dann weiter noch eine zeilenweise Erklärung des Beispielprogramms

LISTING SORT.BSP

```

10 ? "/"
20 ? "RECORDLÄNGE EINGEBEN (1-185) ";; INPUT RLENG
30 ? "MAXIMALE RECORDANZAHL ";; INPUT NREC
40 ? "SCHLÜSSELFELD-LÄNGE EINGEBEN ";; INPUT KLENG
50 IF KLENG > RLENG THEN KLENG=RLENG
60 ? "/"
80 TABLE=(NREC+2)*RLENG
90 DIM A$(TABLE),B$(RLENG),BLANK$(RLENG)
100 FOR I=1 TO RLENG:BLANK$(I,I)=" ":NEXT I
110 A$=BLANK$
120 ? "EINGABE MIT RETURN BEENDEN"
130 ? "RECORD EINGEBEN";; INPUT B$
140 IF LEN(B$)=RLENG THEN 170
150 IF LEN(B$)=0 THEN 200
160 B$(LEN(B$)+1)=BLANK$
170 A$(LEN(A$)+1)=B$
180 IF LEN(A$)=TABLE-RLENG THEN 200
190 GOTO 130
200 A$(LEN(A$)+1)=BLANK$
210 ? "/"
220 ? "UNSORTIERT:"
230 GOSUB 310
240 POKE 1536, RLENG
245 POKE 1537, KLENG
250 POKE 1538, 9
260 START=ADR(A$):LAST=START+LEN(A$)-2*RLENG
270 Q=USR(7514,START,LAST)
280 ? "SORTIERT:"
290 GOSUB 310
300 END

310 FOR J=1 TO LEN(A$)/RLENG-1
320 ? A$(J*RLENG+1,J*RLENG+RLENG)
330 NEXT J
340 RETURN

```

ERKLÄRUNG SORT.BSP

```

10 Bildschirm löschen
20 Recordlänge in der Variable RLENG abspeichern
30 Maximale Anzahl der Records in NREC abspeichern
40 Länge des Schlüsselfeldes in KLENG abspeichern
50 Sicherstellen das KLENG nicht grösser als RLENG wird
60 Bildschirm löschen
80 Maximale Größe des Sortier-Strings bestimmen
   Anzahl der eingegebenen Records + 2 Records für Anfang
   und Ende, multipliziert mit der Recordlänge
90 Strings dimensionieren : A$ - Sortier-String,
   B$ - Eingabe für Records, BLANK$ - füllt die
   Datenfelder mit Leerzeichen auf
100 BLANK$ mit Leerzeichen füllen
110 Erster Record im Sortier-String ist ein leerer Record
120 Mit RETURN kann die Eingabe abgebrochen werden
130 Eingabe eines Records
140 Hat der Record maximale Länge, wird der Füllteil
   übersprungen
150 Abfrage, ob mit RETURN abgebrochen wurde
160 Eingegebenen Record mit Leerzeichen auffüllen
170 Record in den Sortier-String einfügen
180 Wenn die maximale Länge des Sortier-Strings erreicht
   ist (ohne den End-Record), wird die Eingabe abgebro-
   chen
190 Rücksprung, um nächsten Record einzulesen
200 Letzter Record im Sortier-String ist ein leerer Record
210 Bildschirm löschen
220 Meldung ausdrucken
230 Sprung zum Unterprogramm : Ausgabe formatieren
240 Recordlänge abspeichern
245 Schlüsselfeldlänge abspeichern
250 Verteilungsmaß M abspeichern
260 Startadressen berechnen
270 SUPERSORT aufrufen, abgespeichert unter der Adresse
   7514
280 Meldung ausdrucken
290 Sprung zum Unterprogramm : Ausgabe formatieren
300 Programmende

```

UNTERPROGRAMM

```

310 Anzahl der auszudruckenden Records festlegen
320 Einen Record ausdrucken
330 FOR-Schleife abschließen
340 Rücksprung zum Hauptprogramm

```


Wichtige Informationen

Lieber Computerfreund, lieber Kunde, lieber Händler!

Jeder, der sich einmal selbst damit beschäftigt hat, ein Computerprogramm zu fertigen, weiß, welche Arbeit und geistige Mühe aufgewendet werden muß, um eine Problemlösung zu finden und sie anwenderfreundlich zu programmieren. Die Erfüllung dieser Voraussetzungen erfordert viel Erfahrung und hohe finanzielle und zeitliche Investitionen. Das Ergebnis sind gute und erfolgreiche Computerprogramme, die von interessierten Anwendern nachgefragt werden und deshalb für den Händler verkäuflich sind.

Diese Tatsache machen sich einige dadurch zunutze, daß sie die mit hohen Vorforderungen geschaffenen erfolgreichen Programme der Firma Atari kopieren oder ihren Kunden die Möglichkeit anbieten, die gewünschten Programme auf Diskette zu überspielen. Sie meinen, damit ihren Kunden ein gutes und billiges Angebot zu machen. Die Kunden wissen jedoch meist nicht, daß sie lediglich ein vermeintlich gutes und billiges Angebot erhalten.

Abgesehen davon, daß das Angebot zur Überspielung von Programmen und das Anbieten und Verkaufen illegal kopierter Programme strafrechtlich verboten ist, weil es sich dabei um Verletzungen des Urheberrechtes (COMPUTERPROGRAMM PIRATERIE) handelt, die von Atari gegenüber jedermann ohne Ansehen der Person gerichtlich verfolgt wird, so ist auch die Annahme falsch, das Angebot sei günstig oder billig:

- Gestohlene Ware ist immer billig. Der Dieb hat keine Vorforderungen. Er eignet sich nur fremdes Eigentum an, für die der Käufer keine Gewährleistung erhält.
- Der Händler, der das Kopieren von Programmen anbietet, anstatt Originale zu verkaufen schmarotzt an fremder Leistung.
- Der interessierte Kunde wird bald keine guten Programme mehr kaufen können und illegale Programme wird der Handel bald auch nicht mehr anbieten können.

Letzteres deswegen, weil niemand mehr bereit und in der Lage sein wird, gute verkaufsfähige Programme zu entwickeln, wenn nicht die Möglichkeit besteht, die hohen Vorforderungen durch Verkäufe wieder zu verdienen. Die Piraten sind geistig weder in der Lage noch überhaupt bereit, sich der Mühe zu unterziehen, Programme zu entwickeln. Sie können und wollen nur durch Diebstahl fremder guter Leistung eine schnelle bequeme Mark verdienen.

Wer also Interesse daran hat, daß das Angebot an guten Computerprogrammen wächst, sollte die illegalen „billigen“ Angebote meiden und mit dazu beitragen, daß den Totengräbern der Computer-Programmentwicklung und damit des Computerhandels das Handwerk gelegt wird.

Wir danken für Ihr Verständnis und freuen uns über jeden Hinweis von Ihnen.

Atari Elektronikvertriebsges. mbH

ATARI®

ATARI-Elektronik Vertriebsgesellschaft mbH
Postfach 60 01 69 · Bebelallee 10 · 2000 Hamburg 60

Jegliche Rechte vorbehalten.
Vermietung, Verleih, Vervielfältigung
und öffentliche Aufführung verboten.

ATARINSIDE