

6

Index

This program lets you enter items in any order and then displays them in alphabetical order. In short, it produces an index. The program can help you keep a list of your tape collection, a list of members of your club, or a list of names and addresses of your friends. Authors will find it very helpful when indexing a book, and you can also use it for producing catalogues, glossaries, and any other kind of list in which the items are to be arranged in alphabetic order.

The indexed list may be saved to tape or disk, and added to at some future date. There is no need to enter all the items at one session. If you have a printer attached to your Electron, it is easy to modify the program to display the whole or a part of the list, as explained later. The list can contain up to two hundred items, provided that your entries are not excessively long. The program has a special feature to allow you to see how much memory is left.

Using the program

The first question the computer asks is 'Load from tape?'. If you wish to look at or add to a list that you have already saved on tape, key 'Y'. If you are starting a new list, key 'N', and you will be taken directly to the part of the program which deals with entering items (see later).

On keying 'Y' you are asked 'Name of file?'. Key in the file name that you used for the list you want to load, then press RETURN. You will now be able to fast-forward or rewind the tape to locate the required part of the tape. Then press PLAY on the recorder. After several seconds - or up to five minutes if the list is nearly full - the screen clears and the query 'Entry?' is displayed.

Making Entries

The query 'Entry?' is asking you to type in the key-word or keywords for the entry. This is the word (or words) which are to be sorted alphabetically.

At the bottom right corner of the screen, you will see 'Entryno:'. This is the number to be given to the next entry to be put on the list. If you are starting a new list, the entry number is 1. If you have just loaded in a list from tape, this number is one greater than the number of items already on the list. If the number is close to 200, the list is almost full. The computer will 'beep' at you as you make entries 195 to 200 to warn you of this. Below the entry no. is the statement 'Memory ='. This tells you how many bytes of memory are available for holding entries. If you are starting a new list, this number is quite large (about 15000). As you enter more and more items, the number decreases. Sometimes it decreases a little, sometimes a lot, while at other times it stays the same. This is because of the way the data is being stored by the computer. The general trend is downward and the computer beeps at each entry when the number falls below 200. You may be able to squeeze in a few short entries at that stage, but do not try to type in long ones. If in doubt, save your existing entries before trying to cram in a few more.

Before continuing with a description of the program, the meaning of 'alphabetical order' needs to be clarified. The program works according to several rules:

(1) It sorts on a letter-by-letter basis, not on a word-by-word basis. Here are items sorted in the two different ways:

<i>Letter-by-letter</i>	<i>Word-by-word</i>
abacus	A.C. mains
A.C. mains	A.C. theory
Acorn computers	A.M. radio
A.C. theory	abacus
A.M. radio	Acorn computers
Arcade games	arc-back
arc-back	Arcade games

Letter-by letter sorting ignores all punctuation and spaces. Word-by-word sorting sorts by the first word (or the first single letter) of an item, ignoring subsequent words or letters. Then it sorts the second

words (or second single letters), and so on. Parts of hyphenated words are usually taken to be separate words.

(2) All punctuation and spaces are ignored in sorting.

(3) Whether letters are capitals or lower-case letters makes no difference to the sorting.

(4) Names that begin with ' St.' are sorted as if they begin with ' Saint' .

(5) Names that begin with ' M'', Mc' or ' MC' are sorted as if they begin with ' Mac' .

Note that although the program may ignore punctuation, or may sort ' Mc's ' Mac'all entries are displayed exactly as you type them in. Probably you will wish to key in entries in upper-case (capitals) and lower-case, just as on a typewriter. If so, press SHIFT and CAP LK FUNC, so that the yellow lamp goes out. Your entries will then be in lower-case, and you obtain capitals by pressing the SHIFT key.

It is important that only the part of the item that is to be used for alphabetic sorting is keyed in as the ' Entry'. When this has been typed in, press RETURN. Lower down on the screen you will then see: the word ' Reference'. This is asking you to key in the remainder of the item, if any. It is here that you key in page numbers, if you are indexing a book, for example. You can key in a single page number, a span of several pages (e.g. ' 123—7') or a cross-reference (e.g. (see Computers, Acorn)). If you are preparing an address book, the address and telephone number can be typed in as ' Reference'. The only point to remember is that the program will not hold many entries if you type a lot into ' Reference' .

As soon as you have completed the reference section, press RETURN. The screen clears and you are asked for the next ' Entry'. If you are indexing a book, you will sometimes need to add further page numbers to an existing entry. This is done by typing in the entry again. Make sure it is spelt exactly as before, with the same punctuation and spaces, if any. The new page number may then be typed in as the ' Reference'. The computer adds this new page number to those already listed under this item. You will see that the ' Entry' at the bottom of the screen does not increase after such an operation. No new entry has been created. You are merely adding a page number to the existing entry. Doing this requires very little additional memory, perhaps none at all.

Other commands

The ' Entry?message offers you the opportunity to make the program do other things beside accept entries. Instead of typing an entry, key in one of the following:

(1) = The computer displays the list of entries, with their references. If you type ' =alone, you will be shown the whole list. This is displayed a screenful at a time; to see the next screenful, press SHIFT. If you type ' =followed by one or more letters, you will see just that part of the list which contains words beginning with that letter (or letters). For example, typing ' =D(no space between ' =and ' D'gives you all entries beginning with ' Dòr ' d'arranged in alphabetical order. Typing ' =ELEwould give you all words beginning with these three letters, such as ' Electron'',electricityand ' elephant'If there are no words beginning with these letters, or when the computer has displayed all that there are, you see the message "Space Bar to continue: ' R'to review". Pressing the space-bar takes you back to ' Entry?'while pressing ' Rlets you see the whole or the specified part of the list again.

(2) | (key SHIFT and the ' \right-arrow' key.) This is used for deleting items from the list. If you type ' |on its own, you will be able to look through the whole list, as when you type ' =If you type ' | followed by one or more letters, you will be shown only those words beginning with the letters you have typed. Each item is displayed separately and you are asked to key ' DIf you want it to be deleted, or ' RIf you want it to be retained (nor deleted). Take care, for an item once deleted cannot be recovered; you will have to type it in again! Deleting an item does not free the memory it occupied. Deleted items are still included in the maximum of 200 items that the program allows.

When all the items called for have been displayed the message ' Deletionscompleted' appears. After a few seconds delay, you are taken back for the next entry.

(3) { (key CTRL and the ' £{ arrow' key.) This takes you to the routine for saving the list on tape. If you have previously loaded the list from tape, the computer now reminds you of the file name you used. It then asks you to key in the file name under which the list is to

be saved. Probably you will choose to key in the same name, but you can use a different name if you wish. After you have keyed in the name, the message ' RECORDthen RETURN' appears. Rewind the tape, if necessary, so that it is ready to receive the recording. Press RECORD and PLAY on the recorder, then press RETURN on the Electron. Saving takes up to five minutes if you have a full list. When saving is complete, the message ' Backup' appears. If you want to take a backup copy, key ' Y' and repeat the procedure described above. If not, key ' N' and you will see the message ' Index program finished' .

Keying in

Note the space in ' St.' used to identify a saint's name. Take particular care with typing lines 560 to 600 and 700 to 820. The ON ERROR command is in force in these sections of the program. If you make a typing error you will not get the usual error messages to tell you what you have done wrong. If you find that the displaying of the list or the deletion routine always seems to end too quickly it could be that there are errors in the typing of these lines.

Program design

- 20-60 Initialising arrays and variables.
- 70-110 Load from tape?
- 120-200 Loading from tape.
- 210-250 ' Entry?' display.
- 260-310 Inputting the entry.
- 320-340 To various routines.
- 350 Rejecting entry when memory is full.
- 360 Inputting reference.
- 370 To PROCconvert.
- 380-400 Dealing with ' Mac' s and ' St. ' s.
- 410-520 Sorting by calculating key pointers.
- 530-650 Displaying the list.
- 660-840 Deleting routine.
- 850-1000 Saving to tape.
- 1010-1080 PROCconvert to convert an entry into a string of capital letters with all spaces and punctuation omitted, ready for sorting.
- 1090-1120 PROCpause to produce a pause.

Points of interest

It was mentioned under INFO FILE that large amounts of memory can be wasted if strings are frequently moved around in memory. In INFO FILE this waste was eliminated by filling all the fields of each record to their maximum extent at the start of the program. This was practicable in that program because the number of characters required in each field had previously been specified by the user.

This approach is not suitable for INDEX. An entry in an index may be very short, perhaps only a single letter. Yet other entries may require twenty or more characters. It is wasteful of memory to fill each entry with, say, twenty characters when many may need much less than this. The solution is not to move the entries around in memory. But how can we avoid doing this when an index requires that the entries be sorted into alphabetical order? How can we avoid rearranging the entries every time a new entry is made?

INDEX does not attempt to move the entries around. Each entry goes into a new location in the array E\$ () as it is keyed in, and stays there permanently. At the same time the converted ' allcapitals' version, used in sorting goes to the corresponding location in F\$(). The reference goes into R\$(). Thus, a given entry always has the same location number in all three arrays. The other array used in this program is K%(), which holds what are called the key pointers. These are numbers which ' point to' locations in F\$() (or E\$() and R\$ ()). The computer finds its way to the first item in alphabetical order by using the *head-of-list pointer*, H%. If H% has the value 25, for example, it means that the alphabetic first entry is number 25, which is to be found at E\$(25), and F\$(25), with its reference at R\$(25). When asked to display the whole list, the computer first prints what it finds at E\$(25) and R\$(25). Then it looks at the number stored in K%(25). This tells it where the next entry in alphabetic order is to be found. K%(25) might hold the value 12, for example, indicating that the second item on the list is at E\$(12) and R\$(12). Then, at K%(12) it finds the pointer to the third item, and so on to the end of the list. When it reaches the last item in the list, it finds the value -1 in K%(), indicating that it has reached the end. The computer does not allow such expressions as K%(-1), and at this stage the ON ERROR condition causes the computer to go to the end of the routine.

Using this system, it is necessary only to alter a few values in K%()

whenever an item is to be added to or deleted from the list. Changing values in a numeric array requires no additional memory. This technique saves much time too. If we were to keep entries in E\$, F\$() in alphabetical order we would need, on average, to shift half of them to new locations every time a new entry was made. Changing the few relevant key pointers is very much quicker. A list of items such as this is sometimes referred to as a *linked list*.

The program

```

10 REM ** INDEX **
20 MODE 6
30 DIM E$(201),F$(201),K%(200),R$(200)
)
40 H%=1:K%(1)=-1:PC%=1:N%=1:X$=""
50 VDU 14
60 CLS
70 REPEAT
80 PRINT TAB(0,10)"Load from tape? (Y
/N) ";
90 Y$=GET$
100 UNTIL INSTR("YNyn",Y$)
110 IF INSTR("Nn",Y$) THEN 210
120 INPUT'"Name of file? "X$
130 X=OPENIN X$
140 INPUT#X,N%
150 INPUT#X,H%
160 FOR J=1 TO N%:INPUT#X,E$(J):NEXT
170 FOR J=1 TO N%:INPUT#X,F$(J):NEXT
180 FOR J=1 TO N%:INPUT#X,K%(J):NEXT
190 FOR J=1 TO N%:INPUT#X,R$(J):NEXT
200 CLOSE#X
210 CLS:IF N%=201 THEN PRINT'"TAB(25)
** Index FULL";CHR$(7)
220 PRINT TAB(24,21)"Entry no: ";N%
230 PRINT TAB(24,22);"Memory = ";HIMEM
-?2-256*?3
240 IF N%>194 OR HIMEM-?2-256*?3<200 T
HEN PRINT CHR$(7)
250 PRINT TAB(0,3)"Entry ?"
260 VDU 31,0,8:E$(N%)=""
270 REPEAT

```

86 Practical Programs for the Electron

```
280 Y$=GET$:IF Y$=CHR$(13) THEN 310
290 IF Y$=CHR$(127) THEN E$(N%)=LEFT$(
E$(N%),LEN(E$(N%))-1):PRINT Y$;:GOTO 310
300 PRINT Y$;:E$(N%)=E$(N%)+Y$
310 UNTIL Y$=CHR$(13)
320 IF LEFT$(E$(N%),1)=" " THEN 530
330 IF LEFT$(E$(N%),1)="|" THEN 660
340 IF E$(N%)="{ " THEN 850
350 IF N%>200 OR HIMEM-?2-256*?3<100TH
EN 210
360 INPUT TAB(0,12)"Reference? "R$
370 PROCconvert
380 T$=LEFT$(E$(N%),2):IF T$="M' " THEN
F$(N%)="MAC"+MID$(F$(N%),2)
390 IF T$="MC " OR T$="Mc " THEN F$(N%)="
MAC"+MID$(F$(N%),3)
400 IF LEFT$(E$(N%),4)="St. " THEN F$(N
%)="SAINT"+MID$(F$(N%),3)
410 IF N%=1 THEN 510
420 L%=0:C%=H%
430 D$=F$(C%):QC%=PC%:PC%=C%:C%=K%(C%)
440 IF C%=-1 AND F$(N%)>D$ THEN K%(N%)
=-1:K%(PC%)=N%:L%=1:GOTO 460
450 IF F$(N%)>D$ THEN 430
460 IF L%=1 THEN 510
470 IF F$(N%)=D$ THEN R$(PC%)=R$(PC%)+
", "+R$:GOTO 210
480 IF PC%=H% THEN H%=N%:K%(N%)=PC%:GO
TO 510
490 K%(QC%)=N%
500 K%(N%)=PC%
510 R$(N%)=R$:N%=N%+1
520 GOTO 210
530 CLS:VDU 31,0,3
540 IF H%=-1 THEN PRINT'"'"Index empty
":PROCpause:GOTO 210
550 PROCconvert
560 ON ERROR GOTO 600
570 C%=H%
580 IF F$(C%)<F$(N%) THEN C%=K%(C%):GO
TO 580
590 I=INSTR(F$(C%),F$(N%)):IF I=1 THEN
PRINT E$(C%),R$(C%):C%=K%(C%):GOTO 590
600 ON ERROR OFF
```



```

610 PRINT'" Space Bar to continue: 'R
' to review"
620 REPEAT:Y$=GET$
630 UNTIL Y$=" " OR Y$="R"
640 IF Y$="R" THEN CLS:VDU 31,0,3:GOTO
560
650 GOTO 210
660 CLS:VDU 31,0,3
670 IF H%=-1 THEN PRINT'"Index empty
":GOTO 830
680 PRINT'"D = DELETE : R = RETAIN"
:VDU 31,0,8
690 PROCconvert
700 ON ERROR GOTO 820
710 C%=H%
720 IF F$(C%)<F$(N%) THEN QC%=PC%:PC%=
C%:C%=K%(C%):GOTO 720
730 I=INSTR(F$(C%),F$(N%))
740 IF I=1 THEN PRINT E$(C%),R$(C%);"
";QC%=PC%:PC%=C%:C%=K%(C%)
750 IF I=0 THEN 820
760 REPEAT:Y$=GET$:UNTIL INSTR("DRdr",
Y$)
770 PRINT Y$
780 IF INSTR("Rr",Y$) THEN 730
790 IF PC%=H% THEN H%=C%:GOTO 730
800 IF PC%<>H% THEN K%(QC%)=K%(PC%):PC
%=QC%
810 GOTO 730
820 ON ERROR OFF:PRINT'"Deletions comp
leted"
830 PROCpause
840 GOTO 210
850 CLS:IF X$<>" THEN PRINT'"This wa
s saved as file '";X$;"'"
860 INPUT'"What file name? "X$:X$=LE
FT$(X$,7)
870 X=OPENOUT X$
880 PRINT#X,N%
890 PRINT#X,H%
900 FOR J=1 TO N%:PRINT#X,E$(J):NEXT
910 FOR J=1 TO N%:PRINT#X,F$(J):NEXT
920 FOR J=1 TO N%:PRINT#X,K$(J):NEXT
930 FOR J=1 TO N%:PRINT#X,R$(J):NEXT

```

```

940 CLOSE#X
950 PRINT'"Backup? (Y/N) " ;
960 REPEAT:Y$=GET$:UNTIL INSTR("YNyn",
Y$)
970 IF INSTR("Yy",Y$) THEN 850
980 VDU 15
990 PRINT'"Index program finished"
1000 END
1010 DEF PROCconvert
1020 F$(N%)=""
1030 FOR J=1 TO LEN(E$(N%))
1040 M$=MID$(E$(N%),J,1):M=ASC(M$)
1050 IF M>96 THEN M=M-32
1060 IF M>64 AND M<91 THEN F$(N%)=F$(N%
)+CHR$(M)
1070 NEXT
1080 ENDPROC
1090 DEF PROCpause
1100 TIME=0
1110 REPEAT:UNTIL TIME>500
1120 ENDPROC

```

Variations

The program has been used on the BBC Micro, in colijunction with a printer. At the time of writing, no printer interface is available for the Electron, but it seems that when interfaces are available there will be no difficulty in adapting this program to print out any display at the same time as it appears on the screen. All that needs to be done is to add a printer-enabling command (e.g. VDU 2) to line 550, and a disabling command (e.g. VDU 3) on line 610. Consult the manual for your printer or printer interface to find out exactly what commands to use.