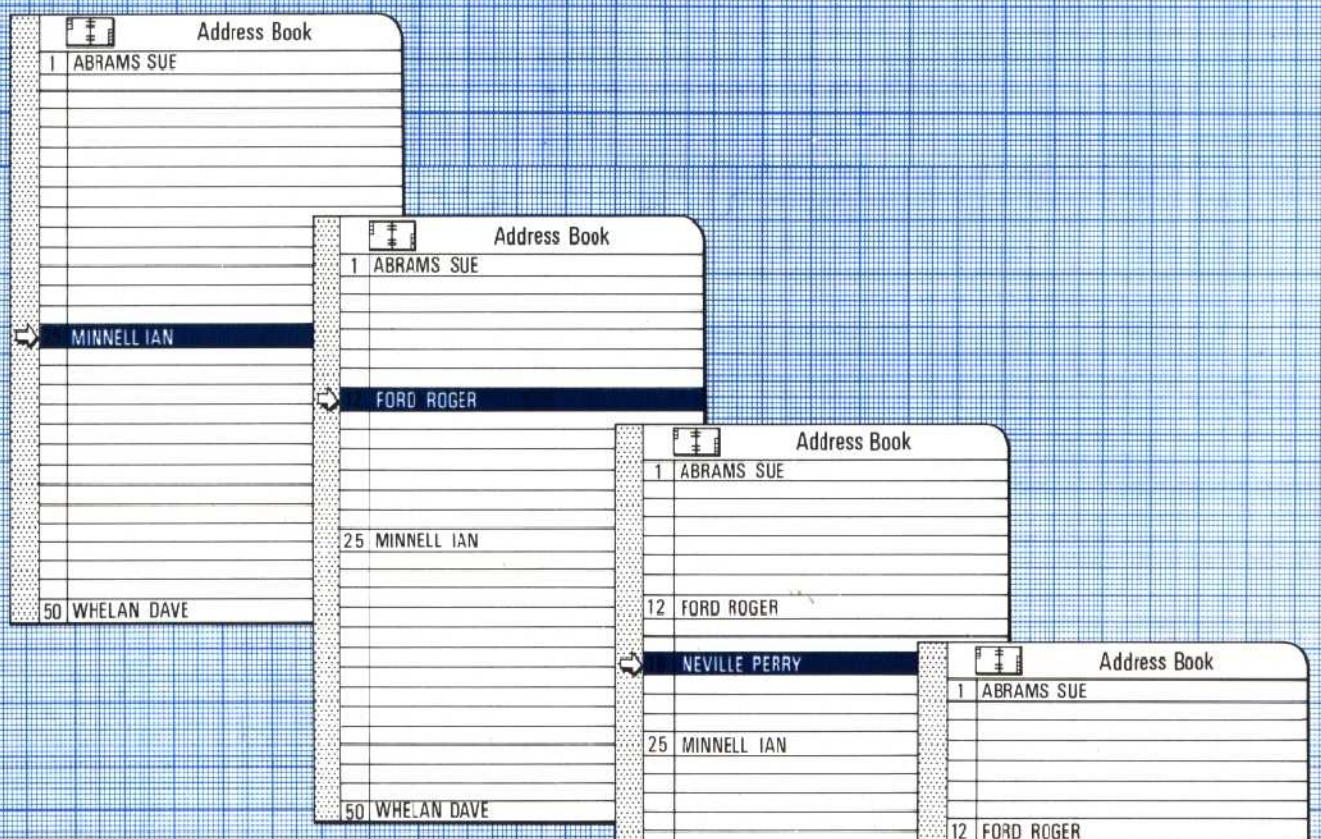
As we have already noted, however, a linear search is not efficient compared with a binary search if the data is already sorted. The search routine can ensure that the records are sorted by starting with an IF RMOD = 1 THEN GOSUB *SRTREC*. The program knows that the lowest element in the array to be searched will be MODFLD$(1) and the highest will be MODFLD$(SIZE – 1). To conduct the search, we will need three variables: BTM for the bottom of the array (MODFLD$(1) at the beginning); TOP for the top of the array (MODFLD$(SIZE – 1) at the beginning); and MID for the value corresponding to the middle element.

Using the dictionary analogy, we can assume that BTM = ARRAY(1) and TOP = ARRAY(SIZE – 1). In other words, the array we have to consider for the search starts with the 'smallest' element and ends with the 'largest' element. We can therefore LET BTM = 1 and LET TOP = SIZE – 1 (remember that SIZE is always one larger than the number of records currently in the address book).

Suppose that there are 21 valid entries in the address book. SIZE will have a value of 22. BTM will have a value of 1. TOP will have a value of 21. The value of MID, the position of the middle element, can be derived in BASIC from INT((BTM + TOP)/2). If the BTM value is 1, and the TOP value is 21, the MID value will be 11.

To conduct a binary search, we first assume that the whole file is valid and find the mid point INT((BTM + TOP)/2) inside a loop that is terminated either if the target is found or if there is no match. Then we check to see if the search key (SCHKEY$) happens to be equal to the MID value of the array. If the MID value of the array is too small, we know that ARRAY(MID) is the lowest part of the array we

Address Book Search



Binary Search

If an address book file contains 50 entries, sorted alphabetically from Abrams to Whelan, then the position to start looking for a particular entry (here we are searching for JONES PETER) will be halfway between the two extremes. This is the principle on which the binary search is constructed.

Position 25 contains MINNELL IAN, which coming after JONES PETER, indicates that the second half of the file can now be ignored in the search.

Dissecting the remaining list would give us position 12, which turns out to be FORD ROGER, indicating that JONES PETER must lie between elements 13 and 24. So the next search position will be 18 (NEVILLE PERRY), followed by 21. Here we hit our target name, JONES PETER.

KEVIN JONES