ROYAL McBEE CORPORATION

ELECTRONIC COMPUTER DEPT.

SORTING ROUTINE

FUNCTION:

To sort into ascending numerical order, a list of N consecutive words (numbers) in memory. This sorting routine requires no extra reserved space; the list is sorted without writing a second list of N words. The sort can be made on any part of the word, i.e. bits 7 - 11 and 24 - 29.

INPUT:

The addresses of the first (L_i) and last (max) words of the list and a mask specifying the bits on which the sort is to be made.

CALLING SEQUENCE:

LOCATION	ORDER	ADDRESS
α - 1	B	L [L_i]
α	R	(Lo + 3)
α + 1	U	Lo
α + 2	[Mask]*	
α + 3	XZ	L max of list
α + 4	etc.	

,0000001

*01W000WJ means the sort will be made on bits 7 thru 11 and 24 thru 29 inclusive. 7WWWWWWQ means the sort will be made on bits 1 thru 30 inclusive. No provision has been made for handling negative numbers. The mask may not contain a "1" at 0.

OUTPUT:

The ordered list in locations L_i thru L max.

TIME:

Time is dependent on the order of the unsorted list. Listed below are a few averages:

32 word list	38 seconds
64 word list	2 minutes
128 word list	9 minutes

SHORTAGE:

96 words (1-1/2 tracks). Temporary storage: Sectors 54,55,62, 63 of track 63.

EXIT:
The routine exits to α +4.

SORTING ROUTINE

N = No. of words in List.

Flowchart:

- $0 \rightarrow i$
- Is $L_i > L_{i+1}$ both masked — YES → Store L_i in L_{i+1}
- NO → $i + 1 \rightarrow i$ → is $i = N$
  - NO → (loop back to "Is $L_i > L_{i+1}$")
  - YES → EXIT
- Store L_i in L_{i+1} → $i - 1 \rightarrow j$ → is $j < 0$
  - YES → Store L_{i+1} in L_0
  - NO → Is $L_j > L_{i+1}$ both masked
    - YES → Store L_j in L_{j+1} → $j - 1 \rightarrow j$ → (back to "is $j < 0$")
    - NO → Store L_{i+1} in L_{j+1}