

**NAME**

qsort – quicker sort

**SYNOPSIS**

(end+1 of data in r2)

(element width in r3)

**jsr pc,qsort**

**qsort(base, nel, width, compar)**

**char \*base;**

**int (\*compar)();**

**DESCRIPTION**

*Qsort* is an implementation of the quicker-sort algorithm. The assembly-language version is designed to sort equal length elements. Registers r1 and r2 delimit the region of core containing the array of byte strings to be sorted: r1 points to the start of the first string, r2 to the first location above the last string. Register r3 contains the length of each string. r2-r1 should be a multiple of r3. On return, r0, r1, r2, r3 are destroyed.

The routine compar (q.v.) is called to compare elements and may be replaced by the user.

The C version has somewhat different arguments and the user must supply a comparison routine. The first argument is to the base of the data; the second is the number of elements; the third is the width of an element in bytes; the last is the name of the comparison routine. It is called with two arguments which are pointers to the elements being compared. The routine must return a negative integer if the first element is to be considered less than the second, a positive integer if the second element is smaller than the first, and 0 if the elements are equal.

**SEE ALSO**

compar (III)

**BUGS**