

1 SUMMARY

To sort an **array of numbers** into **descending order** maintaining an **index array** to preserve a record of the original order.

The 'Quicksort' algorithm is used, see, C.A.R. Hoare, 'Quicksort', Computer Journal, April 1962.

ATTRIBUTES — **Version:** 1.0.0. (12th July 2004) **Types:** Real (single, double), Integer. **Original date:** April 1980. **Origin:** C.Birch*, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

Sorting single precision numbers

```
CALL KB08A (ARRAY, N, INDEX)
```

Sorting double precision numbers

```
CALL KB08AD (ARRAY, N, INDEX)
```

Sorting integer numbers

```
CALL KB08AI (ARRAY, N, INDEX)
```

ARRAY is an **array** containing the numbers to be sorted and the user must put these in $ARRAY(I)$, $I=1, N$. On return from the subroutine they will have been sorted into descending order. **ARRAY** should be of the Fortran type corresponding to the name of the sorting subroutine being used.

N is an **INTEGER** variable and must be set by the user to the number of numbers in the array.

INDEX is an **INTEGER** array of length at least N , which will be set by the subroutine to the original (unsorted) order of the numbers in **ARRAY**, so that reference may be made to both orderings of the array. On return from the subroutine the I^{th} element of the index array, $J=INDEX(I)$, gives the position in the original ordering of the number now in $ARRAY(I)$, i.e. the sorting process has moved the number which was originally in $ARRAY(J)$ to $ARRAY(I)$.

3 GENERAL INFORMATION

Use of Common: none.

Workspace: private integer workspace of length 100, which limits the size of **ARRAY** to $2^{50} \approx 10^{15}$.

Other subroutines: none.

Input/Output: prints an error message if $n < 1$.

Restrictions: $n \geq 1$.

4 METHOD

The 'Quicksort' method is used, see C.A.R. Hoare, 'Quicksort', Computer Journal, April 1962.

5 EXAMPLE OF USE

This example program uses KB08AD to sort the N numbers in the array ARRAY into descending order while preserving the original ordering in the array INDEX.

```
PROGRAM MAIN
INTEGER N, I
PARAMETER ( N = 10 )
INTEGER INDEX( N )
DOUBLE PRECISION ARRAY( N )
DATA ARRAY / 1.0D0, 5.0D0, 7.0D0, 0.0D0, 4.0D0,
*           6.0D0, 2.0D0, 3.0D0, 9.0D0, 8.0D0 /
CALL KB08AD( ARRAY, N, INDEX )
WRITE( 6, "( ' reordered array = ', /,
*         ' index array ', /, ( I6, F6.2 ) )" )
* ( INDEX( I ), ARRAY( I ), I = 1, N )
STOP
END
```

This produces the following output:

```
reordered array =
index array
 9  9.00
10  8.00
 3  7.00
 6  6.00
 2  5.00
 5  4.00
 8  3.00
 7  2.00
 1  1.00
 4  0.00
```