1 SUMMARY

Given a set of intervals on the real line, this subroutine **finds the smallest set of disjoint intervals** whose union is the union of the original set, i.e. given intervals which overlap one another it will reduce the number of intervals by merging together overlapping intervals.

Specifically, given the m closed intervals (a_1, b_1), (a_2, b_2),., (a_m, b_m) ordered so that $a_i \le a_j$ for $i<j$, the subroutine merges these into $n \le m$ intervals (x_1, y_1), (x_2, y_2),., (x_n, y_n) such that $y_i < x_{i+1}$, i=1,2,...,n–1.

ATTRIBUTES — Version: 1.0.0. **Types:** `KC01A; KC01AD.` **Original date:** August 1964. **Origin:** D.Willis*, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

The single precision version

```
CALL KC01A(M,A,B,N,X,Y)
```

The double precision version

```
CALL KC01AD(M,A,B,N,X,Y)
```

`M` is an `INTEGER` variable set by the user to m the number of the original intervals.

`A,B` are `REAL` (`DOUBLE PRECISION` in the D version) arrays which must be set by the user to the original intervals (a_i, b_i), i=1,2,...,m. **Restrictions:** $a_i \le b_i$, i=1,2,...,m–1 and $a_i \le a_j$ for $i<j$.

`N` is an `INTEGER` variable which is set by the subroutine to n the number of disjoint intervals obtained after merging the original intervals.

`X,Y` are `REAL` (`DOUBLE PRECISION` in the D version) arrays in which the subroutine returns the intervals (x_i, y_i), i=1,2,...,n. If the original intervals are already disjoint `N=M` and the contents of `X` and `Y` will be the same as those of `A` and `B.` **N.B.** it is safe to use the array `A` for `X` and the array `B` for `Y` so that the new intervals will overwrite the old.

3 GENERAL INFORMATION

Use of Common: none.

Workspace: none.

Other subroutines: none.

Input/Output: none.

System dependence: none.