Grade Up /: _ _ _ Sort Up

`/:` grades any argument, yielding a permutation vector; `(/:y){y` sorts `y` in ascending order. For example:
`n=: 3 1 4 2 1 3 3`
`]g=: /: n`
`1 4 3 0 5 6 2`
`g { n`
`1 1 2 3 3 3 4`
	`x/:y` is `(/:y){x`; i.e., `x` is sorted to an order specified by `y`. In particular, `y/:y` (or `/:~y`) sorts `y`. For example:
	`y=: 'popfly'`
	`y /: 3 1 4 1 5 9`
	`ofpply`
	`y /: y`
	`floppy`

Elements of `/:y` that select equal elements of `y` are in ascending order. If `y` is a table, `/:y` grades the base value of the rows, using a base larger than twice the magnitude of any of the elements. Higher ranks are treated as `,.y`, (as if its items were each ravelled).

If `y` is literal, `/:y` grades according to the collating sequence determined by the alphabet `a.`; another collating sequence `cs` can be imposed by grading `cs i. y`. For example:

```
   ]n=: 3 1 4 1 6,2 7 1 8 3,:6 1 8 0 3
3 1 4 1 6
2 7 1 8 3
6 1 8 0 3

   /: n
1 0 2

   Aa=: ' ',. a. {~ 65 97 +/ i. 26
   x=: words=: >;: 'When eras die'
   j=: <./Aa i."1 _ x
   x ; (x/:x) ; (x/:j) ; Aa
+----+----+----+--------------------------+
|When|When|die | ABCDEFGHIJKLMNOPQRSTUVWXYZ|
|eras|die |eras| abcdefghijklmnopqrstuvwxyz|
|die |eras|When|                          |
+----+----+----+--------------------------+
```

The three types: numeric or empty, literal, and boxed, are so ordered; within them, a lower rank precedes a higher, and a smaller shape precedes a larger. Complex arguments are ordered by real part, then by imaginary. Boxed arrays are ordered according to the opened elements.