multiples

syntax: `multiples(data)`
`[cnts,vals] = multiples(data)`
`multiples(data, valuelist)`

purpose: Counts how many times each value appears in the data set. If `valuelist` is specified, MULTIPLES counts how many times each of the values in `valuelist` appears in the data set.

examples: `data = [1 2 3 2 3 3 4];`

In this short data set, value 1 occurs once, value 2 occurs twice, value 3 occurs three times, and value 4 occurs once.
≫ `multiples(data)` ⇒ *ans: 1 2 3 1*
Sometimes you want to know which value corresponds to the count returned by `multiples`. To get this information, use the syntax
≫ `[cnts,vals] = multiples(data)`
`cnts` will contain the vector of counts, [1 2 3 1], while `vals` tells which value corresponds to each count, `[1 2 3 4]`. To see the values and their corresponding counts printed side by side, type
≫ `[vals, cnts]` ⇒ *ans:*
1 1
2 2
3 3
4 1

To extract those values that occurred, say, more than once, use the indexing command
≫ `vals( cnts>=2 )` ⇒ *ans: 2 3*

If you have a complete listing of the possible values, you can give this information to MULTIPLES which will then count how many times each of the possible values appears. This is useful if one or more of the possible values does not actually appear in the data set; then the count will be 0 for these values.
≫ `multiples([1 1 1 1 1], [1 2])` ⇒ *ans: 5 0*
If the second argument `valuelist` is not given, MULTIPLES always orders the `cnts` so that the `vals` are in ascending order. Otherwise, the order is the same as `valuelist`.

See also: DEDUP