Magnitude | 0 0 0 Residue

`|y` ↔ `%:y*+y`. For example:
`   | 6 _6 3j4`
`6 6 5`
	The familiar use of residue is in determining the remainder on dividing a non-negative integer by a positive:
	`   3 | 0 1 2 3 4 5 6 7`
	`0 1 2 0 1 2 0 1`

The definition `y-x*<. y % x+0=x` extends the residue to a zero left argument, and to negative and fractional arguments. For example:

```
   over =: ({. ,.@; }.)@":@,
   by   =: ' '&;@,.@[ ,. ]
   x=: 3 2 1 0 _1 _2 _3 [ y=: 0 1 2 3 4 5 6 7 8
   x by y over x |/ y
+--+-----------------------+
|  |0  1  2  3  4  5 6  7  8|
+--+-----------------------+
| 3|0  1  2  0  1  2 0  1  2|
| 2|0  1  0  1  0  1 0  1  0|
| 1|0  0  0  0  0  0 0  0  0|
| 0|0  1  2  3  4  5 6  7  8|
|_1|0  0  0  0  0  0 0  0  0|
|_2|0 _1  0 _1  0 _1 0 _1  0|
|_3|0 _2 _1  0 _2 _1 0 _2 _1|
+--+-----------------------+
```

To produce a true zero for cases such as `(%3)|(2%3)` the residue is made tolerant as shown in the definition of `res` below:

```
   res=: f`g@.agenda"0
      agenda=: ([ = 0:) +. (<. = >.)@S
         S=: ] % [ + [ = 0:
      f=: ] - [ * <.@S [, g=: ] * [ = 0:
   0.1 res 2.5 3.64 2 _1.6
0 0.04 0 0
```

Continued

Magnitude | 0 0 0 Residue

Continued

```
   (,. ; res/~ ; |/~) a=: 2 -~ i.5
+--+------------+------------+
|_2| 0 _1 0 _1 0| 0 _1 0 _1 0|
|_1| 0  0 0  0 0| 0  0 0  0 0|
| 0|_2 _1 0  1 2|_2 _1 0  1 2|
| 1| 0  0 0  0 0| 0  0 0  0 0|
| 2| 0  1 0  1 0| 0  1 0  1 0|
+--+------------+------------+
```

The dyad `|` applies to complex numbers. Moreover, the fit conjunction may be applied to control the tolerance used. The dyad `m&|@^` on integer arguments is computed in a way that avoids large intermediate numbers. For example: `2 (1e6&|@^) 10^100x`