Boolean   m b. _ 0 0

If `f` is a dyadic boolean function and `d=: 0 1`, then `d f/ d` (or `f/~d`) is its complete table. For example the tables for or, nor, and, and not-and (followed by their ravels) appear as follows:

```
   (+./~ ; +:/~ ; *./~ ; *:/~) d=: 0 1
+---+---+---+---+
|0 1|1 0|0 0|1 1|
|1 1|0 0|0 1|1 0|
+---+---+---+---+

   ,&.> (+./~ ; +:/~ ; *./~ ; *:/~) d
+-------+-------+-------+-------+
|0 1 1 1|1 0 0 0|0 0 0 1|1 1 1 0|
+-------+-------+-------+-------+
```

If ordered by their ravels, each of the sixteen possible boolean dyads can be characterized by its index `k`; the phrase `k b.` produces the corresponding function. Moreover, negative indexing may be used. For example:

```
   (7 b./~;8 b./~;1 b./~;14 b./~;_2 b./~) d
+---+---+---+---+---+
|0 1|1 0|0 0|1 1|1 1|
|1 1|0 0|0 1|1 0|1 0|
+---+---+---+---+---+
```

The adverb `b.` also applies to array arguments. For example:

```
   (<"2) 2 0 1 |: 7 8 1 15 b./~ d
+---+---+---+---+
|0 1|1 0|0 0|1 1|
|1 1|0 0|0 1|1 1|
+---+---+---+---+
```

The monad (as in `m b. y`) is equivalent to a zero left argument (as in `0 m b. y`).

Basic Characteristics u b. _

`u b. y` gives the obverse of `u` if `y` is `_1`; its ranks if `y` is `0`; and its identity function if `y` is `1`.	

For example:

```
   ^ b. _1
^.

   ^ b. 0
0 0 0

   ^ b. 1
$&1@(}.@$)

   g=: +&2@(*&3@*:)
   ]y=: g 5
77

   g ^:_1 y
5

   g b. _1
%:@(%&3)@(-&2)

   %:@(%&3)@(-&2) y
5

   g b. 0
0 0 0
```