# Insert m / u / _ _ _ Table

<table>
<tr>
<td>

u/y applies the dyad u between the items of y. Thus:

```
   m=: i. 3 2
   m;(+/m);(+/"1 m);(+/2 3 4)
+---+---+-----+-+
|0 1|6 9|1 5 9|9|
|2 3|   |     | |
|4 5|   |     | |
+---+---+-----+-+
```

m/y inserts successive verbs from the gerund m between items of y, extending m cyclically as required. Thus, +`*/i.6 is 0+1*2+3*4+5.

</td>
<td>

If x and y are numeric lists, then x */ y is their multiplication table. Thus:

```
   1 2 3 */ 4 5 6 7
 4  5  6  7
 8 10 12 14
12 15 18 21
```

In general, each cell of x is applied to the entire of y. Thus x u/ y is equivalent to x"(lu,_) y where lu is the left rank of u.

The case */ is called outer product in tensor analysis.

</td>
</tr>
</table>

If y has no items (that is, 0=#y), the result of u/y is the neutral or identity element of the function u. A neutral of a function u is a value e such that x u e ↔ x or e u x ↔ x, for every x in the domain (or some significant sub-domain such as boolean) of u. This definition of insertion over an argument having zero items extends partitioning identities of the form u/y ↔ (u/k{.y) u (u/k}.y) to the cases k e. 0,#y.

The identity function of u is a function ifu such that ifu y is u/y if 0=#y. The identity functions used are:

| Identity function | For |
|---|---|
| `$&0@}.@$` | `<  >  + - +. ~: \| (2 4 5 6 b.)` |
| `$&1@}.@$` | `= <: >: * % *. %: ^ ! (1 9 11 13 b.)` |
| `$&_@}.@$` | `<.` |
| `$&__@}.@$` | `>.` |
| `i.@(0&,)@(2&}.)@$` | `,` |
| `i.@(1&{.)@}.@$` | `C. {` |
| `=@i.@(1&{.)@}.@$` | `%. +/ . *` |
| `ifu@#` | `u/` |
| `$&(v^:_1 ifu$0)@}.@$` | `u&.v` |