```
     ∇ZZ←N IB W;Z;B;R;I
[1]    ⍎(0>A←N-1)/'→0, 0⍴ZZ←0 3⍴0'
[2]    ZZ←Z,[0]A IB W[0 1],1↑Z←B,R,I
       where B←W[2]-R←W[1]-I←×/W[0 2]
[3]  ∇
```

Direct definition:

```
where:α:0:ω

IB:Z,[0]A IBω[0 1],1↑Z←B,R,I
   where B←ω[2]-R←ω[1]-I←×/ω[0 2]:
              0>A←α-1:0 3⍴0
```

where B = current balance; R = amount going to reduce principal; I = amount going to pay interest.

If the principal is $20,000, the interest is 10 percent, and the monthly payment is $1,000, the function *IB* computes a table for 12 months (numbers are rounded):

```
    12 IB 10 1000 20000÷1200 1 1
19167   833   167
18326   840   160
17479   847   153
16625   854   146
15763   861   139
14895   869   131
14019   876   124
13136   883   117
12245   891   109
11347   898   102
10442   905    95
 9529   913    87
```

In J's pure functional form, define *i*, *r*, and *b* as three forks:

```
   i=. 2&{ * {.
   r=. 1&{ - i
   b=. 2&{ - r

ib=.((b,r,i)@],
      <:@[ ib (2&{.,b)@])`
      (0 3&$ @ 0:) @.(=0:)
```

To understand the structure of this function, condense it as follows:

```
ib=.(f@], <:@[ ib g@])`h
       @.(=0:)
```

Read it thus:

To the result of function *f* of the right argument, catenate the item (row) resulting from the function *ib* with a decremented left argument, and a right argument computed by function *g* from the previous right argument. Function *h* gives the identity element for catenation of rows to a table with 3 columns. Terminate when the left argument is zero.

Because calculation of interest payments on a declining balance builds a table, we must start with 0 rows and 3 columns. Zero, then, is not enough; any language is incomplete if it fails to include different kinds of emptiness.

The identity element for matrix multiplication is the appropriately named *identity matrix*, first recognized by Cayley: "A matrix is not altered by its composition, either as first or second component matrix, with the matrix unity."[17] In the following example, the recursive function *MP* raises a matrix (left argument) to an integer power (right argument), and consequently requires the identity matrix of the same shape as the matrix argument.

Ordinary APL:

```
     ∇Z←M MP N;I
[1]    ⍎(N=0)/'→0, 0⍴Z←I∘.=I←⍳1↑⍴M'
[2]    Z←M+.×M MP N-1
[3]  ∇
```

More succinctly in direct definition:

```
MP:α+.×αMPω-1: ω=0 :I∘.=I←⍳1↑⍴α

      M←3 3⍴⍳9
      M+.×M+.×M
180  234  288
558  720  882
936 1206 1476

      M MP 3
180  234  288
558  720  882
936 1206 1476
```

Zero seems to behave like the queen in chess; for is it not the most powerful piece on the board? Any number multiplied by zero is reduced to zero. But *emptiness* is more powerful still, because any number, including zero, is reduced to emptiness when multiplied by an empty vector. Emptiness is not, however, to be confused with *nothing*, which is the result of executing an empty vector. You cannot multiply a number by nothing—a *value error* results if you try. Shakespeare made the fool touch something profound in saying to the king without a throne: "Now thou art an O without a figure. I am better than thou art now; I am a Fool, thou art nothing."[57]