```
      '49927398716'          ⍝ char vector
49927398716
      ⍎'49927398716'         ⍝ execute (eval)
4.992739872E10
      ⍎¨'49927398716'        ⍝ execute each (map)
4 9 9 2 7 3 9 8 7 1 6
      {⍵}⍎¨'49927398716'     ⍝ function of ⍵
4 9 9 2 7 3 9 8 7 1 6
      {⌽⍵}⍎¨'49927398716'    ⍝ reverse of
6 1 7 8 9 3 7 2 9 9 4
      {⍴⍵}⍎¨'49927398716'    ⍝ shape of
11
      {(⍴⍵)⍴1 2}⍎¨'49927398716'    ⍝ 11 reshape of 1 2 ...
1 2 1 2 1 2 1 2 1 2 1
      {(⌽⍵)×(⍴⍵)⍴1 2}⍎¨'49927398716'    ⍝ product
6 2 7 16 9 6 7 4 9 18 4
     {0 10⊤(⌽⍵)×(⍴⍵)⍴1 2}⍎¨'49927398716'    ⍝ base-10 encode
0 0 0 1 0 0 0 0 0 1 0
6 2 7 6 9 6 7 4 9 8 4
      {,0 10⊤(⌽⍵)×(⍴⍵)⍴1 2}⍎¨'49927398716'  ⍝ unravelled matrix
0 0 0 1 0 0 0 0 0 1 0 6 2 7 6 9 6 7 4 9 8 4
      {+/,0 10⊤(⌽⍵)×(⍴⍵)⍴1 2}⍎¨'49927398716'  ⍝ sum (plus fold)
70
      {10|+/,0 10⊤(⌽⍵)×(⍴⍵)⍴1 2}⍎¨'49927398716'  ⍝ 10-residue
0
      {0=10|+/,0 10⊤(⌽⍵)×(⍴⍵)⍴1 2}⍎¨'49927398716'  ⍝ equals 0?
1
     {0=10|+/,0 10⊤(⌽⍵)×(⍴⍵)⍴1 2}∘(⍎¨)'49927398716'  ⍝ composition
1
      luhn ← {0=10|+/,0 10⊤(⌽⍵)×(⍴⍵)⍴1 2}∘(⍎¨)     ⍝ name function

      luhn '49927398716'     ⍝ application of named function
1
      ⍝ Chopping up the expression into more "mind-sized" pieces:

      luhn←{                        ⍝ Luhn checksum checker
          weights←(⍴⍵)⍴1 2          ⍝ 1 2 1 2 ...
          weighted←weights×⌽⍵        ⍝ weighted reversed numbers
          digits←0 10⊤weighted      ⍝ separated digits
          dsum←+/,digits            ⍝ sum of digits
          0=10|dsum                 ⍝ multiple of 10?
      }
```