Subarray Functions - Definitions

- **subvector(*v*,*a*,*b*):** extracts the elements of vector ***v*** lying between indices ***a*** and ***b*** inclusive.
  If ***a*** is negative then subvector indexes from the end of the vector, et sim for ***b***. Note: if ***a*** > ***b*** then subvector selects elements of ***v*** from ***a*** *back* to ***b*** thus reversing the order of the corresponding elements in ***v***.
  Returns 0 ; if ***b*** > last(***v***) returns ***v***(***a***..last(***v***)).
  Returns 0 if ***v*** is not a Mathcad vector (that is, a one-column array)

$$\text{subvector}(v,a,b) := \left| \begin{array}{l} \text{return } 0 \text{ if } \text{cols}(v) \ne 1 \\ \text{return } 0 \text{ if } (a < -\text{rows}(v) \lor a > \text{last}(v)) \lor (b < -\text{rows}(v) \lor b > \text{last}(v)) \\ \text{submatrix}(v, \text{if}(a < 0, \text{rows}(v) + a, a), \text{if}(b < 0, \text{rows}(v) + b, b), 0, 0) \end{array} \right.$$

$$\text{subvector}(v,a,b) := \left| \begin{array}{l} \text{return } 0 \text{ if } \text{cols}(v) \ne 1 \\ \text{return } 0 \text{ if } a < -\text{rows}(v) \lor a > \text{last}(v) \\ \text{return } 0 \text{ if } b < -\text{rows}(v) \lor b > \text{last}(v) \\ a \leftarrow \text{rows}(v) + a \text{ if } a < 0 \\ b \leftarrow \text{rows}(v) + b \text{ if } b < 0 \\ \text{submatrix}(v,a,b,0,0) \end{array} \right.$$

The function to the left expands the if funtions into if statements that handle each index separately ... more readable but not as compact.

- **submatrix(M,ir,jr,ic,jc):** extracts the elements of matrix *M* lying between rows *ir* and *jr* and columns *ic* and *jc* inclusive.
  If *ir* is negative then submatrix indexes from the last row, et sim for *jr, ic* and *jc.*

$$\text{submatrix}(M,ir,jr,ic,jc) := \left| \begin{array}{l} \text{return } 0 \text{ if } ir < -\text{rows}(M) \lor ir \ge \text{rows}(M) \\ \text{return } 0 \text{ if } jr < -\text{rows}(M) \lor jr \ge \text{rows}(M) \\ \text{return } 0 \text{ if } ic < -\text{cols}(M) \lor ic \ge \text{cols}(M) \\ \text{return } 0 \text{ if } jc < -\text{cols}(M) \lor jc \ge \text{cols}(M) \\ ir \leftarrow \text{rows}(M) + ir \text{ if } ir < 0 \\ jr \leftarrow \text{rows}(M) + jr \text{ if } jr < 0 \\ ic \leftarrow \text{cols}(M) + ic \text{ if } ic < 0 \\ jc \leftarrow \text{cols}(M) + jc \text{ if } jc < 0 \\ \text{submatrix}(M,ir,jr,ic,jc) \end{array} \right.$$

- **subrows(*M*,*a*,*b*):** extracts rows ***a*** to *b* from matrix *M.*

$$\text{subrows}(M,a,b) := \text{submatrix}(M,a,b,0,\text{cols}(M) - 1)$$

- **subcols(M,a,b):** extracts columns *a* to ***b*** from matrix ***M***.

$$\text{subcols}(M,a,b) := \text{submatrix}(M,0,\text{rows}(M) - 1,a,b)$$

- **firstrows(*A*,*n*):** extracts (at most) the first ***n*** rows of array ***A***.

$$\text{firstrows}(A,n) := \left| \begin{array}{l} \text{submatrix}(A,0,\min(n,\text{rows}(A)) - 1,0,\text{cols}(A) - 1) \text{ if } n \\ \text{return } 0 \text{ otherwise} \end{array} \right.$$

- **lastrows(*A*,*n*):** extracts (at most) the last ***n*** rows of array ***A***.

$$\text{lastrows}(A,n) := \left| \begin{array}{l} \text{submatrix}[A,\text{rows}(A) - (\min(n,\text{rows}(A))),\text{rows}(A) - 1,0,\text{cols}(A) - 1] \text{ if } n \\ \text{return } 0 \text{ otherwise} \end{array} \right.$$

- **firstcols(*A*,*n*):** extracts (at most) the first ***n*** columns of array ***A***.

$$\text{firstcols}(A,n) := \left| \begin{array}{l} \text{submatrix}(A,0,\text{rows}(A) - 1,0,\min(n,\text{cols}(A)) - 1) \text{ if } n \\ (\text{return } 0) \text{ otherwise} \end{array} \right.$$

- **lastcols(*A*,*n*):** extracts (at most) the last ***n*** columns of array ***A***.

```
lastcols(A,n) := | submatrix[A,0,rows(A) − 1,cols(A) − (min(n,cols(A))),cols(A) − 1]  if n
                 | return 0  otherwise
```

- **rowends(*a*,*n*):** extracts (at most) the first ***n*** rows and the last ***n*** rows of array ***a***, separating them with a line of dashes if ***n*** is positive; if ***n*** is negative then there is no separating line. Returns ***a*** if 2***n*** covers all of the rows.

```
rowends(a,n) := | m ← floor(rows(a)/2)
                | return a  if |n| > 2·m
                | n ← −3  if n = 0
                | if n ≥ 0
                |   | sep ← 0
                |   | for k ∈ 0..cols(a) − 1  if 2·n < rows(a)
                |   |   sep_{0,k} ← ""
                | n ← −n  otherwise
                | n ← min(m,n)
                | rs ← submatrix(a,0,n − 1,0,cols(a) − 1)
                | re ← submatrix(a,rows(a) − n,rows(a) − 1,0,cols(a) − 1)
                | stack(rs,sep,re)  if rows(sep)
                | stack(rs,re)  otherwise
```

- **colends(*a*,*n*):** extracts (at most) the first ***n*** columns and the last ***n*** columns of array ***a***, separating them with a line of dashes if ***n*** is positive; if ***n*** is negative then there is no separating line. Returns ***a*** if 2***n*** covers all of the columns.

```
colends(a,n) := | m ← floor(cols(a)/2)
                | return a  if |n| > 2·m
                | n ← −3  if n = 0
                | if n ≥ 0
                |   | sep ← 0
                |   | for k ∈ 0..rows(a) − 1  if 2·n < cols(a)
                |   |   sep_k ← ""
                | n ← −n  otherwise
                | n ← min(m,n)
                | rs ← submatrix(a,0,rows(a) − 1,0,n − 1)
                | re ← submatrix(a,0,rows(a) − 1,cols(a) − n,cols(a) − 1)
                | augment(rs,sep,re)  if cols(sep)
                | augment(rs,re)  otherwise
```

- **matends(*a*,*m*,*n*):** extracts (at most) the first and last ***m*** rows, and the first and last ***n*** columns of array ***a***, separating the rows and/or column with a line of dashes if either ***m*** or ***n*** is negative. returns all the rows and/or columns of ***a*** if 2***m*** and/or 2***n*** covers all of the rows.

```
matends(a,m,n) := rowends(colends(a,m),n)
```

Usage Examples and Test Harnesses

Test Data Generation Functions

- **fillinteger(*nrows*,*ncols*):** creates an ***nrows*** x ***ncols*** matrix, filled with successive integers, starting from 0 and increasing down each column.

```
fillinteger(m,n) := | return 0 if ¬(m ∧ n)
                    | s ← −1
                    | for j ∈ 0..n − 1
                    |   for i ∈ 0..m − 1
                    |     a[i,j] ← (s ← s + 1)
                    | a
```

$\text{fillint}(m,n) := \text{fillinteger}(m,n)$

$\text{fillint1}(m,n) := \text{fillinteger}(m,n) + 1$

$\text{matint}(n) := \text{fillinteger}(n,n) + 1$ $\quad$ $\text{vecint}(n) := \text{fillinteger}(n,1) + 1$

$\text{matint0}(n) := \text{fillinteger}(n,n)$ $\quad$ $\text{vecint0}(n) := \text{fillinteger}(n,1)$

$\text{matintx}(n,x) := \text{fillinteger}(n,n) + x$ $\quad$ $\text{vecintx}(n,x) := \text{fillinteger}(n,1) + x$

- **transpose(*A*):** functional version of the array transpose operator. Defining transpose as a function allows application of the vectorize operator to its arguments, which can prove useful for viewing a nested vector.
- Note: Unlike the built-in transpose operator, tranpose will return ***A*** if ***A*** is not an array rather than raising an error.

```
transpose(A) := | A^T if IsArray(A)
                | A otherwise
```

Sub-array Function Examples and Test Harnesses

- **subvector(*v*,*a*,*b*):** extracts the elements of vector ***v*** lying between indices ***a*** and ***b*** inclusive.
  If ***a*** is negative then subvector indexes from the end of the vector, et sim for ***b***. Note: if ***a*** > ***b*** then subvector selects elements of ***v*** from ***a*** *back* to ***b*** thus reversing the order of the corresponding elements in ***v***.
  Returns 0 ; if ***b*** > last(***v***) returns ***v***(***a***..last(***v***)).
  Returns 0 if ***v*** is not a Mathcad vector (that is, a one-column array)

$\text{subvector}(\text{vecint}(5),-1,5) = 0$ $\quad$ $\text{subvector}(\text{vecint}(5),-1,-1)^T = (5)$

$\text{subvector}(\text{vecint}(5),0,5) = 0$ $\quad$ $\text{subvector}(\text{vecint}(5),-1,-2)^T = (5\ \ 4)$

$\text{subvector}(\text{vecint}(5),-1,4)^T = (5)$ $\quad$ $\text{subvector}(\text{vecint}(5),-7,1) = 0$

$\text{subvector}(\text{vecint}(5),0,4)^T = (1\ \ 2\ \ 3\ \ 4\ \ 5)$ $\quad$ $\text{subvector}(\text{vecint}(5),-7,1) = 0$

```
s := | n ← 4
     | v ← vecint(n)
     | for a ∈ −n − 1..n
     |   | (s[a+n+3,0] ← a   s[a+n+3,1] ← mod(a,n))
     |   | for b ∈ −n − 1..n
     |   |   s[a+n+3,b+n+3] ← subvector(v,a,b)
     | for b ∈ −n − 1..n
     |   (s[0,b+n+3] ← b   s[1,b+n+3] ← mod(b,n))
     | (s[0,0] ← "a\b"   s[1,1] ← "mod")
     | s
```

program to generate, and apply subvector to, all combinations of a and b indices in range -(rows(v)+1) .. rows(v). Allows visual verification of subvector behaviour.

$\overrightarrow{\text{transpose}(s)} =$

"a\b"	0	−5	−4	−3	−2	−1	0	1	2
0	"mod"	−1	0	−3	−2	−1	0	1	2
−5	−1	0	0	0	0	0	0	0	0
−4	0	0	(1)	(1 2)	(1 2 3)	(1 2 3 4)	(1)	(1 2)	(1 2
−3	−3	0	(2 1)	(2)	(2 3)	(2 3 4)	(2 1)	(2)	(2 3
−2	−2	0	(3 2 1)	(3 2)	(3)	(3 4)	(3 2 1)	(3 2)	(3)
−1	−1	0	(4 3 2 1)	(4 3 2)	(4 3)	(4)	(4 3 2 1)	(4 3 2)	(4 3
0	0	0	(1)	(1 2)	(1 2 3)	(1 2 3 4)	(1)	(1 2)	(1 2
1	1	0	(2 1)	(2)	(2 3)	(2 3 4)	(2 1)	(2)	(2 3
2	2	0	(3 2 1)	(3 2)	(3)	(3 4)	(3 2 1)	(3 2)	(3)
3	3	0	(4 3 2 1)	(4 3 2)	(4 3)	(4)	(4 3 2 1)	(4 3 2)	(4 3
4	0	0	0	0	0	0	0	0	0

- **submatrix(M,ir,jr,ic,jc):** extracts the elements of matrix *M* lying between rows *ir* and *jr* and columns *ic* and *jc* inclusive. If *ir* is negative then submatrix indexes from the last row, et sim for *jr, ic* and *jc.*

$$\text{matint}(5) = \begin{pmatrix} 1 & 6 & 11 & 16 & 21 \\ 2 & 7 & 12 & 17 & 22 \\ 3 & 8 & 13 & 18 & 23 \\ 4 & 9 & 14 & 19 & 24 \\ 5 & 10 & 15 & 20 & 25 \end{pmatrix}$$

$$\text{submatrix}(\text{matint}(5), 0, -1, 0, -1) = \begin{pmatrix} 1 & 6 & 11 & 16 & 21 \\ 2 & 7 & 12 & 17 & 22 \\ 3 & 8 & 13 & 18 & 23 \\ 4 & 9 & 14 & 19 & 24 \\ 5 & 10 & 15 & 20 & 25 \end{pmatrix}$$

$$\text{submatrix}(\text{matint}(5), -7, 1, 0, -1) = 0$$

$$\text{submatrix}(\text{matint}(5), 1, 3, 1, 3) = \begin{pmatrix} 7 & 12 & 17 \\ 8 & 13 & 18 \\ 9 & 14 & 19 \end{pmatrix}$$

$$\text{submatrix}(\text{matint}(5), -1, -3, -1, -3) = \begin{pmatrix} 25 & 20 & 15 \\ 24 & 19 & 14 \\ 23 & 18 & 13 \end{pmatrix}$$

$$\text{submatrix}(\text{matint}(5), -3, -1, -3, -1) = \begin{pmatrix} 13 & 18 & 23 \\ 14 & 19 & 24 \\ 15 & 20 & 25 \end{pmatrix}$$

```
sr := | n ← 4
      | M ← matint(n)
      | for  a ∈ −n − 1 .. n
      |   | ( sr[a+n+3,0] ← a   sr[a+n+3,1] ← mod(a,n) )
      |   | for  b ∈ −n − 1 .. n
      |   |   sr[a+n+3,b+n+3] ← submatrix(M,a,b,0,cols(M) − 1)
      | for  b ∈ −n − 1 .. n
      |   ( sr[0,b+n+3] ← b   sr[1,b+n+3] ← mod(b,n) )
      | ( sr[0,0] ← "a\b"   sr[1,1] ← "mod" )
      | sr
```

program to generate, and apply submatrix to, all combinations of ir and jr indices in range -(rows(v)+1) .. rows(v). Allows visual verification of submatrix row extract behaviour.

sr =

"a\b"	0	−5	−4	−3	−2	−1	0	1
0	"mod"	−1	0	−3	−2	−1	0	1
−5	−1	0	0	0	0	0	0	(
−4	0	0	$\begin{pmatrix}1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\\2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\\2 & 6 & 10 & 14\\3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\\2 & 6 & 10 & 14\\3 & 7 & 11 & 15\\4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}1 & 5\\2 & 6\end{pmatrix}$
−3	−3	0	$\begin{pmatrix}2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\\3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\\3 & 7 & 11 & 15\\4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}2 & 6\end{pmatrix}$
−2	−2	0	$\begin{pmatrix}3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\\2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\\4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}3 & 7\\2 & 6\end{pmatrix}$
−1	−1	0	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\\2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}4 & 8\\3 & 7\\2 & 6\end{pmatrix}$
0	0	0	$\begin{pmatrix}1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\\2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\\2 & 6 & 10 & 14\\3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\\2 & 6 & 10 & 14\\3 & 7 & 11 & 15\\4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}1 & 5\\2 & 6\end{pmatrix}$
1	1	0	$\begin{pmatrix}2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\\3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\\3 & 7 & 11 & 15\\4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}2 & 6\end{pmatrix}$
2	2	0	$\begin{pmatrix}3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\\2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\\4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}3 & 7\\2 & 6\end{pmatrix}$
3	3	0	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\\2 & 6 & 10 & 14\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\end{pmatrix}$	$\begin{pmatrix}4 & 8 & 12 & 16\\3 & 7 & 11 & 15\\2 & 6 & 10 & 14\\1 & 5 & 9 & 13\end{pmatrix}$	$\begin{pmatrix}4 & 8\\3 & 7\\2 & 6\end{pmatrix}$
4	0	0	0	0	0	0	0	(

```
sc := | n ← 4
      | M ← matint(n)
      | for  a ∈ −n − 1 .. n
      |   | (sc[a+n+3,0] ← a   sc[a+n+3,1] ← mod(a,n))
      |   | for  b ∈ −n − 1 .. n
      |   |   sc[a+n+3,b+n+3] ← submatrix(M,0,rows(M) − 1,a,b)
      | for  b ∈ −n − 1 .. n
      |   (sc[0,b+n+3] ← b   sc[1,b+n+3] ← mod(b,n))
      | (sc[0,0] ← "a\b"   sc[1,1] ← "mod")
      | sc
```

program to generate, and apply submatrix to, all combinations of ic and jc indices in range -(cols(v)+1) .. cols(v). Allows visual verification of submatrix column extract behaviour.

sc =

"a\b"	0	−5	−4	−3	−2	−1	0	1
0	"mod"	−1	0	−3	−2	−1	0	1
−5	−1	0	0	0	0	0	0	0
−4	0	0	$\begin{pmatrix}1\\2\\3\\4\end{pmatrix}$	$\begin{pmatrix}1&5\\2&6\\3&7\\4&8\end{pmatrix}$	$\begin{pmatrix}1&5&9\\2&6&10\\3&7&11\\4&8&12\end{pmatrix}$	$\begin{pmatrix}1&5&9&13\\2&6&10&14\\3&7&11&15\\4&8&12&16\end{pmatrix}$	$\begin{pmatrix}1\\2\\3\\4\end{pmatrix}$	$\begin{pmatrix}1&5\\2&6\\3&7\\4&8\end{pmatrix}$
−3	−3	0	$\begin{pmatrix}5&1\\6&2\\7&3\\8&4\end{pmatrix}$	$\begin{pmatrix}5\\6\\7\\8\end{pmatrix}$	$\begin{pmatrix}5&9\\6&10\\7&11\\8&12\end{pmatrix}$	$\begin{pmatrix}5&9&13\\6&10&14\\7&11&15\\8&12&16\end{pmatrix}$	$\begin{pmatrix}5&1\\6&2\\7&3\\8&4\end{pmatrix}$	$\begin{pmatrix}5\\6\\7\\8\end{pmatrix}$
−2	−2	0	$\begin{pmatrix}9&5&1\\10&6&2\\11&7&3\\12&8&4\end{pmatrix}$	$\begin{pmatrix}9&5\\10&6\\11&7\\12&8\end{pmatrix}$	$\begin{pmatrix}9\\10\\11\\12\end{pmatrix}$	$\begin{pmatrix}9&13\\10&14\\11&15\\12&16\end{pmatrix}$	$\begin{pmatrix}9&5&1\\10&6&2\\11&7&3\\12&8&4\end{pmatrix}$	$\begin{pmatrix}9&5\\10&6\\11&7\\12&8\end{pmatrix}$
−1	−1	0	$\begin{pmatrix}13&9&5&1\\14&10&6&2\\15&11&7&3\\16&12&8&4\end{pmatrix}$	$\begin{pmatrix}13&9&5\\14&10&6\\15&11&7\\16&12&8\end{pmatrix}$	$\begin{pmatrix}13&9\\14&10\\15&11\\16&12\end{pmatrix}$	$\begin{pmatrix}13\\14\\15\\16\end{pmatrix}$	$\begin{pmatrix}13&9&5&1\\14&10&6&2\\15&11&7&3\\16&12&8&4\end{pmatrix}$	$\begin{pmatrix}13&9&5\\14&10&6\\15&11&7\\16&12&8\end{pmatrix}$
0	0	0	$\begin{pmatrix}1\\2\\3\\4\end{pmatrix}$	$\begin{pmatrix}1&5\\2&6\\3&7\\4&8\end{pmatrix}$	$\begin{pmatrix}1&5&9\\2&6&10\\3&7&11\\4&8&12\end{pmatrix}$	$\begin{pmatrix}1&5&9&13\\2&6&10&14\\3&7&11&15\\4&8&12&16\end{pmatrix}$	$\begin{pmatrix}1\\2\\3\\4\end{pmatrix}$	$\begin{pmatrix}1&5\\2&6\\3&7\\4&8\end{pmatrix}$
1	1	0	$\begin{pmatrix}5&1\\6&2\\7&3\\8&4\end{pmatrix}$	$\begin{pmatrix}5\\6\\7\\8\end{pmatrix}$	$\begin{pmatrix}5&9\\6&10\\7&11\\8&12\end{pmatrix}$	$\begin{pmatrix}5&9&13\\6&10&14\\7&11&15\\8&12&16\end{pmatrix}$	$\begin{pmatrix}5&1\\6&2\\7&3\\8&4\end{pmatrix}$	$\begin{pmatrix}5\\6\\7\\8\end{pmatrix}$
2	2	0	$\begin{pmatrix}9&5&1\\10&6&2\\11&7&3\\12&8&4\end{pmatrix}$	$\begin{pmatrix}9&5\\10&6\\11&7\\12&8\end{pmatrix}$	$\begin{pmatrix}9\\10\\11\\12\end{pmatrix}$	$\begin{pmatrix}9&13\\10&14\\11&15\\12&16\end{pmatrix}$	$\begin{pmatrix}9&5&1\\10&6&2\\11&7&3\\12&8&4\end{pmatrix}$	$\begin{pmatrix}9&5\\10&6\\11&7\\12&8\end{pmatrix}$
			$(13\ \ 9\ \ 5\ \ 1 \ldots$	$(13\ \ 9\ \ 5 \ldots$	$(13\ \ 9 \ldots$	$(13 \ldots$	$(13\ \ 9\ \ 5\ \ 1 \ldots$	$(13\ \ 9\ \ 5 \ldots$

$$
\begin{matrix}
3 & 3 & 0 & \begin{pmatrix} 14 & 10 & 6 & 2 \\ 15 & 11 & 7 & 3 \\ 16 & 12 & 8 & 4 \end{pmatrix} & \begin{pmatrix} 14 & 10 & 6 \\ 15 & 11 & 7 \\ 16 & 12 & 8 \end{pmatrix} & \begin{pmatrix} 14 & 10 \\ 15 & 11 \\ 16 & 12 \end{pmatrix} & \begin{pmatrix} 14 \\ 15 \\ 16 \end{pmatrix} & \begin{pmatrix} 14 & 10 & 6 & 2 \\ 15 & 11 & 7 & 3 \\ 16 & 12 & 8 & 4 \end{pmatrix} & \begin{pmatrix} 14 & 10 & 6 \\ 15 & 11 & 7 \\ 16 & 12 & 8 \end{pmatrix} \\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\Bigg]
$$

```
sm := | n ← 4
      | M ← matint(n)
      | for  a ∈ −n − 1 .. n
      |   | (sm[a+n+3, 0 ← a   sm[a+n+3, 1 ← mod(a, n))
      |   | for  b ∈ −n − 1 .. n
      |   |   sm[a+n+3, b+n+3 ← submatrix(M, a, b, a, b)
      | for  b ∈ −n − 1 .. n
      |   (sm[0, b+n+3 ← b   sm[1, b+n+3 ← mod(b, n))
      | (sm[0, 0 ← "a\b"   sm[1, 1 ← "mod")
      | sm
```

program to generate, and apply submatrix to, all combinations of i and j indices in range -(rows(v)+1) .. rows(v) and -(cols(v)+1) .. cols(v) respectively. Allows visual verification of square submatrix extract behaviour.

$$
sm = \begin{bmatrix}
\text{"a\\b"} & 0 & -5 & -4 & -3 & -2 & -1 & 0 & 1 \\
0 & \text{"mod"} & -1 & 0 & -3 & -2 & -1 & 0 & 1 \\
-5 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-4 & 0 & 0 & (1) & \begin{pmatrix} 1 & 5 \\ 2 & 6 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & (1) & \begin{pmatrix} 1 & 5 \\ 2 & 6 \end{pmatrix} \\
-3 & -3 & 0 & \begin{pmatrix} 6 & 2 \\ 5 & 1 \end{pmatrix} & (6) & \begin{pmatrix} 6 & 10 \\ 7 & 11 \end{pmatrix} & \begin{pmatrix} 6 & 10 & 14 \\ 7 & 11 & 15 \\ 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 6 & 2 \\ 5 & 1 \end{pmatrix} & (6) \\
-2 & -2 & 0 & \begin{pmatrix} 11 & 7 & 3 \\ 10 & 6 & 2 \\ 9 & 5 & 1 \end{pmatrix} & \begin{pmatrix} 11 & 7 \\ 10 & 6 \end{pmatrix} & (11) & \begin{pmatrix} 11 & 15 \\ 12 & 16 \end{pmatrix} & \begin{pmatrix} 11 & 7 & 3 \\ 10 & 6 & 2 \\ 9 & 5 & 1 \end{pmatrix} & \begin{pmatrix} 11 & 7 \\ 10 & 6 \end{pmatrix} \\
-1 & -1 & 0 & \begin{pmatrix} 16 & 12 & 8 & 4 \\ 15 & 11 & 7 & 3 \\ 14 & 10 & 6 & 2 \\ 13 & 9 & 5 & 1 \end{pmatrix} & \begin{pmatrix} 16 & 12 & 8 \\ 15 & 11 & 7 \\ 14 & 10 & 6 \end{pmatrix} & \begin{pmatrix} 16 & 12 \\ 15 & 11 \end{pmatrix} & (16) & \begin{pmatrix} 16 & 12 & 8 & 4 \\ 15 & 11 & 7 & 3 \\ 14 & 10 & 6 & 2 \\ 13 & 9 & 5 & 1 \end{pmatrix} & \begin{pmatrix} 16 & 12 & 8 \\ 15 & 11 & 7 \\ 14 & 10 & 6 \end{pmatrix} \\
0 & 0 & 0 & (1) & \begin{pmatrix} 1 & 5 \\ 2 & 6 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & (1) & \begin{pmatrix} 1 & 5 \\ 2 & 6 \end{pmatrix} \\
1 & 1 & 0 & \begin{pmatrix} 6 & 2 \\ 5 & 1 \end{pmatrix} & (6) & \begin{pmatrix} 6 & 10 \\ 7 & 11 \end{pmatrix} & \begin{pmatrix} 6 & 10 & 14 \\ 7 & 11 & 15 \\ 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 6 & 2 \\ 5 & 1 \end{pmatrix} & (6) \\
2 & 2 & 0 & \begin{pmatrix} 11 & 7 & 3 \\ 10 & 6 & 2 \\ 9 & 5 & 1 \end{pmatrix} & \begin{pmatrix} 11 & 7 \\ 10 & 6 \end{pmatrix} & (11) & \begin{pmatrix} 11 & 15 \\ 12 & 16 \end{pmatrix} & \begin{pmatrix} 11 & 7 & 3 \\ 10 & 6 & 2 \\ 9 & 5 & 1 \end{pmatrix} & \begin{pmatrix} 11 & 7 \\ 10 & 6 \end{pmatrix}
\end{bmatrix}
$$

3	3	0	$\begin{pmatrix} 16 & 12 & 8 & 4 \\ 15 & 11 & 7 & 3 \\ 14 & 10 & 6 & 2 \\ 13 & 9 & 5 & 1 \end{pmatrix}$	$\begin{pmatrix} 16 & 12 & 8 \\ 15 & 11 & 7 \\ 14 & 10 & 6 \end{pmatrix}$	$\begin{pmatrix} 16 & 12 \\ 15 & 11 \end{pmatrix}$	(16)	$\begin{pmatrix} 16 & 12 & 8 & 4 \\ 15 & 11 & 7 & 3 \\ 14 & 10 & 6 & 2 \\ 13 & 9 & 5 & 1 \end{pmatrix}$	$\begin{pmatrix} 16 & 12 & 8 \\ 15 & 11 & 7 \\ 14 & 10 & 6 \end{pmatrix}$
4	0	0	0	0	0	0	0	0

- **subrows(*M*,*a*,*b*):** extracts rows ***a*** to *b* from matrix *M.*

$$\text{subrows}(\text{matint}(5),1,3) = \begin{pmatrix} 2 & 7 & 12 & 17 & 22 \\ 3 & 8 & 13 & 18 & 23 \\ 4 & 9 & 14 & 19 & 24 \end{pmatrix} \qquad \text{subrows}(\text{fillint1}(5,6),2,4) = \begin{pmatrix} 3 & 8 & 13 & 18 & 23 & 28 \\ 4 & 9 & 14 & 19 & 24 & 29 \\ 5 & 10 & 15 & 20 & 25 & 30 \end{pmatrix}$$

- **subcols(M,a,b):** extracts columns *a* to ***b*** from matrix ***M***.

$$\text{subcols}(\text{matint}(5),1,3) = \begin{pmatrix} 6 & 11 & 16 \\ 7 & 12 & 17 \\ 8 & 13 & 18 \\ 9 & 14 & 19 \\ 10 & 15 & 20 \end{pmatrix} \qquad \text{subcols}(\text{fillint1}(5,6),2,4) = \begin{pmatrix} 11 & 16 & 21 \\ 12 & 17 & 22 \\ 13 & 18 & 23 \\ 14 & 19 & 24 \\ 15 & 20 & 25 \end{pmatrix}$$

- **firstrows(*A*,*n*):** extracts (at most) the first ***n*** rows of array ***A***.

$$\text{firstrows}(\text{fillint1}(5,5),0) = 0$$

$$\text{firstrows}(\text{matint}(5),1) = (1 \;\; 6 \;\; 11 \;\; 16 \;\; 21) \qquad \text{firstrows}(\text{fillint1}(5,7),2) = \begin{pmatrix} 1 & 6 & 11 & 16 & 21 & 26 & 31 \\ 2 & 7 & 12 & 17 & 22 & 27 & 32 \end{pmatrix}$$

- **lastrows(*A*,*n*):** extracts (at most) the last ***n*** rows of array ***A***.

$$\text{lastrows}(\text{fillint1}(5,5),0) = 0$$

$$\text{lastrows}(\text{matint}(5),1) = (5 \;\; 10 \;\; 15 \;\; 20 \;\; 25) \qquad \text{lastrows}(\text{fillint1}(5,7),2) = \begin{pmatrix} 4 & 9 & 14 & 19 & 24 & 29 & 34 \\ 5 & 10 & 15 & 20 & 25 & 30 & 35 \end{pmatrix}$$

- **firstcols(*A*,*n*):** extracts (at most) the first ***n*** columns of array ***A***.

$$\text{firstcols}(\text{fillint1}(5,5),0) = 0 \qquad \text{firstcols}(\text{fillint1}(5,5),1) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \qquad \text{firstcols}(\text{fillint1}(5,5),2) = \begin{pmatrix} 1 & 6 \\ 2 & 7 \\ 3 & 8 \\ 4 & 9 \\ 5 & 10 \end{pmatrix}$$

- **lastcols(*A*,*n*):** extracts (at most) the last ***n*** columns of array ***A***.

$$\text{lastcols}(\text{fillint1}(5,5),0) = 0 \qquad \text{lastcols}(\text{fillint1}(5,5),1) = \begin{pmatrix} 21 \\ 22 \\ 23 \\ 24 \\ 25 \end{pmatrix} \qquad \text{lastcols}(\text{fillint1}(5,5),2) = \begin{pmatrix} 16 & 21 \\ 17 & 22 \\ 18 & 23 \\ 19 & 24 \\ 20 & 25 \end{pmatrix}$$

- **rowends(*a*,*n*):** extracts (at most) the first ***n*** rows and the last ***n*** rows of array ***a***, separating them with a line of dashes if ***n*** is negative. returns ***a*** if 2***n*** covers all of the rows.

$$\mathrm{rowends(matint(5),1)} = \begin{pmatrix} 1 & 6 & 11 & 16 & 21 \\ "" & "" & "" & "" & "" \\ 5 & 10 & 15 & 20 & 25 \end{pmatrix} \qquad \mathrm{rowends(fillint1(5,7),-1)} = \begin{pmatrix} 1 & 6 & 11 & 16 & 21 & 26 & 31 \\ 5 & 10 & 15 & 20 & 25 & 30 & 35 \end{pmatrix}$$

```
r := | for k ∈ 1..7
     |   | r[k-1,0] ← rowends(matint(k),2)
     |   | r[k-1,1] ← rowends(matint(k),-2)
     | r
```

$$r^T = \begin{bmatrix} (1) & \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 4 & 7 \\ 3 & 6 & 9 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 1 & 6 & 11 & 16 & 21 \\ 2 & 7 & 12 & 17 & 22 \\ "" & "" & "" & "" & "" \\ 4 & 9 & 14 & 19 & 24 \\ 5 & 10 & 15 & 20 & 25 \end{pmatrix} & \begin{pmatrix} 1 & 7 & 13 & 19 & 25 & 31 \\ 2 & 8 & 14 & 20 & 26 & 32 \\ "" & "" & "" & "" & "" & "" \\ 5 & 11 & 17 & 23 & 29 & 35 \\ 6 & 12 & 18 & 24 & 30 & 36 \end{pmatrix} & \begin{pmatrix} 1 & 8 & 15 & 22 & \cdots \\ 2 & 9 & 16 & 23 & \cdots \\ "" & "" & "" & "" & \cdots \\ 6 & 13 & 20 & 27 & \cdots \\ 7 & 14 & 21 & 28 & \cdots \end{pmatrix} \\ (1) & \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 4 & 7 \\ 3 & 6 & 9 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 1 & 6 & 11 & 16 & 21 \\ 2 & 7 & 12 & 17 & 22 \\ 4 & 9 & 14 & 19 & 24 \\ 5 & 10 & 15 & 20 & 25 \end{pmatrix} & \begin{pmatrix} 1 & 7 & 13 & 19 & 25 & 31 \\ 2 & 8 & 14 & 20 & 26 & 32 \\ 5 & 11 & 17 & 23 & 29 & 35 \\ 6 & 12 & 18 & 24 & 30 & 36 \end{pmatrix} & \begin{pmatrix} 1 & 8 & 15 & 22 & \cdots \\ 2 & 9 & 16 & 23 & \cdots \\ 6 & 13 & 20 & 27 & \cdots \\ 7 & 14 & 21 & 28 & \cdots \end{pmatrix} \end{bmatrix}$$

- **colends(*a*,*n*):** extracts (at most) the first ***n*** columns and the last ***n*** columns of array ***a***, separating them with a line of dashes if ***n*** is negative. returns ***a*** if 2***n*** covers all of the columns.

$$\mathrm{colends(matint(5),1)} = \begin{pmatrix} 1 & "" & 21 \\ 2 & "" & 22 \\ 3 & "" & 23 \\ 4 & "" & 24 \\ 5 & "" & 25 \end{pmatrix} \qquad \mathrm{colends(fillint1(5,7),-1)} = \begin{pmatrix} 1 & 31 \\ 2 & 32 \\ 3 & 33 \\ 4 & 34 \\ 5 & 35 \end{pmatrix}$$

```
r := | for k ∈ 1..7
     |   | r[k-1,0] ← colends(matint(k),2)
     |   | r[k-1,1] ← colends(matint(k),-2)
     | r
```

$$r^T = \begin{bmatrix} (1) & \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 7 \\ 2 & 8 \\ 3 & 9 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 1 & 6 & "" & 16 & 21 \\ 2 & 7 & "" & 17 & 22 \\ 3 & 8 & "" & 18 & 23 \\ 4 & 9 & "" & 19 & 24 \\ 5 & 10 & "" & 20 & 25 \end{pmatrix} & \begin{pmatrix} 1 & 7 & "" & 25 & 31 \\ 2 & 8 & "" & 26 & 32 \\ 3 & 9 & "" & 27 & 33 \\ 4 & 10 & "" & 28 & 34 \\ 5 & 11 & "" & 29 & 35 \\ 6 & 12 & "" & 30 & 36 \end{pmatrix} & \begin{pmatrix} 1 & 8 & "" & 36 & 43 \\ 2 & 9 & "" & 37 & 44 \\ 3 & 10 & "" & 38 & 45 \\ 4 & 11 & "" & 39 & 46 \\ 5 & 12 & "" & 40 & 47 \\ 6 & 13 & "" & 41 & 48 \\ 7 & 14 & "" & 42 & 49 \end{pmatrix} \\ (1) & \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 7 \\ 2 & 8 \\ 3 & 9 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 1 & 6 & 16 & 21 \\ 2 & 7 & 17 & 22 \\ 3 & 8 & 18 & 23 \\ 4 & 9 & 19 & 24 \\ 5 & 10 & 20 & 25 \end{pmatrix} & \begin{pmatrix} 1 & 7 & 25 & 31 \\ 2 & 8 & 26 & 32 \\ 3 & 9 & 27 & 33 \\ 4 & 10 & 28 & 34 \\ 5 & 11 & 29 & 35 \\ 6 & 12 & 30 & 36 \end{pmatrix} & \begin{pmatrix} 1 & 8 & 36 & 43 \\ 2 & 9 & 37 & 44 \\ 3 & 10 & 38 & 45 \\ 4 & 11 & 39 & 46 \\ 5 & 12 & 40 & 47 \\ 6 & 13 & 41 & 48 \\ 7 & 14 & 42 & 49 \end{pmatrix} \end{bmatrix}$$

- **matends(*a*,*m*,*n*):** extracts (at most) the first and last ***m*** rows, and the first and last ***n*** columns of array ***a***, separating the rows and/or column with a line of dashes if either ***m*** or ***n*** is negative. returns all the rows and/or columns of ***a*** if 2***m*** and/or 2***n*** covers all of the rows.

$\text{matends}(\text{matint}(1),2,2) = (1)$ $\quad$ $\text{matends}(\text{matint}(1),-2,2) = (1)$ $\quad$ $\text{matends}(\text{matint}(1),2,-2) = (1)$

$\text{matint}(1) = (1)$

$$\text{matint}(7) = \begin{pmatrix} 1 & 8 & 15 & 22 & 29 & 36 & 43 \\ 2 & 9 & 16 & 23 & 30 & 37 & 44 \\ 3 & 10 & 17 & 24 & 31 & 38 & 45 \\ 4 & 11 & 18 & 25 & 32 & 39 & 46 \\ 5 & 12 & 19 & 26 & 33 & 40 & 47 \\ 6 & 13 & 20 & 27 & 34 & 41 & 48 \\ 7 & 14 & 21 & 28 & 35 & 42 & 49 \end{pmatrix} \qquad \text{matends}(\text{matint}(7),2,2) = \begin{pmatrix} 1 & 8 & "" & 36 & 43 \\ 2 & 9 & "" & 37 & 44 \\ "" & "" & "" & "" & "" \\ 6 & 13 & "" & 41 & 48 \\ 7 & 14 & "" & 42 & 49 \end{pmatrix}$$

```
r :=  | for k ∈ 1..7
      |    | r[k−1,0] ← matends(matint(k),2,2)
      |    | return stack(k)  on error r[k−1,1] ← matends(matint(k),−2,−2)
      | r
```

$$r^T = \begin{bmatrix} (1) & \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 7 \\ 3 & 9 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 1 & 6 & "" & 16 & 21 \\ 2 & 7 & "" & 17 & 22 \\ "" & "" & "" & "" & "" \\ 4 & 9 & "" & 19 & 24 \\ 5 & 10 & "" & 20 & 25 \end{pmatrix} & \begin{pmatrix} 1 & 7 & "" & 25 & 31 \\ 2 & 8 & "" & 26 & 32 \\ "" & "" & "" & "" & "" \\ 5 & 11 & "" & 29 & 35 \\ 6 & 12 & "" & 30 & 36 \end{pmatrix} & \begin{pmatrix} 1 & 8 & "" & 36 & 43 \\ 2 & 9 & "" & 37 & 44 \\ "" & "" & "" & "" & "" \\ 6 & 13 & "" & 41 & 48 \\ 7 & 14 & "" & 42 & 49 \end{pmatrix} \\ (1) & \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix} & \begin{pmatrix} 1 & 7 \\ 3 & 9 \end{pmatrix} & \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix} & \begin{pmatrix} 1 & 6 & 16 & 21 \\ 2 & 7 & 17 & 22 \\ 4 & 9 & 19 & 24 \\ 5 & 10 & 20 & 25 \end{pmatrix} & \begin{pmatrix} 1 & 7 & 25 & 31 \\ 2 & 8 & 26 & 32 \\ 5 & 11 & 29 & 35 \\ 6 & 12 & 30 & 36 \end{pmatrix} & \begin{pmatrix} 1 & 8 & 36 & 43 \\ 2 & 9 & 37 & 44 \\ 6 & 13 & 41 & 48 \\ 7 & 14 & 42 & 49 \end{pmatrix} \end{bmatrix}$$