

初めてのさんのJ言語 (PARTV)

統計数理研究所(名誉教授) 鈴木義一郎

ボックスで 与えた要素の 組合せ 片側動詞の カタログ({ Catalogue)なり  
中カッコ({} 左で与えた インデックスの アイテムを取る 両側関数({ From}  
カッコ閉じ()) 左で与えた インデックスの アイテム修正 両側関数  
修正値と インデックスを左に 入力すれば 右引数の値を 修正す() Amend)

【「{」 { Catalogue : 片側形】

<pre>{ 0 1;0 1</pre> <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;">0 1</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;"></td></tr> <tr><td style="padding: 2px;">1</td><td style="padding: 2px;">1 1</td></tr> <tr><td style="padding: 2px;">0</td><td style="padding: 2px;"></td></tr> </table>	0	0 1	0		1	1 1	0		<pre>{'a';'bc';'def'</pre> <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <tr><td style="padding: 2px;">abd</td><td style="padding: 2px;">abe</td><td style="padding: 2px;">abf</td></tr> <tr><td style="padding: 2px;">acd</td><td style="padding: 2px;">ace</td><td style="padding: 2px;">acf</td></tr> </table>	abd	abe	abf	acd	ace	acf	<pre>{3</pre> <p style="margin-left: 20px;">3 アトムに対しては“ボックス(&lt;)” と同じ結果を出力する。</p>																																									
0	0 1																																																								
0																																																									
1	1 1																																																								
0																																																									
abd	abe	abf																																																							
acd	ace	acf																																																							
<pre>]Card='CDHS';'23456789TJQKA'</pre>																																																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">CDHS</td> <td style="padding: 2px;">23456789TJQKA</td> <td style="padding: 2px;">{ Card</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr><td>C2</td><td>C3</td><td>C4</td><td>C5</td><td>C6</td><td>C7</td><td>C8</td><td>C9</td><td>CT</td><td>CJ</td><td>CQ</td><td>CK</td><td>CA</td></tr> <tr><td>D2</td><td>D3</td><td>D4</td><td>D5</td><td>D6</td><td>D7</td><td>D8</td><td>D9</td><td>DT</td><td>DJ</td><td>DQ</td><td>DK</td><td>DA</td></tr> <tr><td>H2</td><td>H3</td><td>H4</td><td>H5</td><td>H6</td><td>H7</td><td>H8</td><td>H9</td><td>HT</td><td>HJ</td><td>HQ</td><td>HK</td><td>HA</td></tr> <tr><td>S2</td><td>S3</td><td>S4</td><td>S5</td><td>S6</td><td>S7</td><td>S8</td><td>S9</td><td>ST</td><td>SJ</td><td>SQ</td><td>SK</td><td>SA</td></tr> </table> <p style="text-align: right; margin-right: 20px;">《52 枚のカード(俗称トランプ)》</p>			CDHS	23456789TJQKA	{ Card	C2	C3	C4	C5	C6	C7	C8	C9	CT	CJ	CQ	CK	CA	D2	D3	D4	D5	D6	D7	D8	D9	DT	DJ	DQ	DK	DA	H2	H3	H4	H5	H6	H7	H8	H9	HT	HJ	HQ	HK	HA	S2	S3	S4	S5	S6	S7	S8	S9	ST	SJ	SQ	SK	SA
CDHS	23456789TJQKA	{ Card																																																							
C2	C3	C4	C5	C6	C7	C8	C9	CT	CJ	CQ	CK	CA																																													
D2	D3	D4	D5	D6	D7	D8	D9	DT	DJ	DQ	DK	DA																																													
H2	H3	H4	H5	H6	H7	H8	H9	HT	HJ	HQ	HK	HA																																													
S2	S3	S4	S5	S6	S7	S8	S9	ST	SJ	SQ	SK	SA																																													

【「{」 From : 両側形】

<pre>{ 'abcde'</pre> <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <tr><td style="padding: 2px;">abcde</td></tr> </table>	abcde	<pre>{"0 'abcde'</pre> <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <tr><td style="padding: 2px;">a</td><td style="padding: 2px;">b</td><td style="padding: 2px;">c</td><td style="padding: 2px;">d</td><td style="padding: 2px;">e</td></tr> </table>	a	b	c	d	e	<pre>{ L=:&gt;:i.2 3</pre> <table border="1" style="margin: 5px auto; border-collapse: collapse;"> <tr><td style="padding: 2px;">1</td><td style="padding: 2px;">2</td><td style="padding: 2px;">4</td><td style="padding: 2px;">5</td></tr> <tr><td style="padding: 2px;">3</td><td style="padding: 2px;"></td><td style="padding: 2px;">6</td><td style="padding: 2px;"></td></tr> </table>	1	2	4	5	3		6		<pre>]T=:&gt;:i.3 4</pre> <table style="margin: 5px auto;"> <tr><td style="padding: 2px;">1</td><td style="padding: 2px;">2</td><td style="padding: 2px;">3</td><td style="padding: 2px;">4</td></tr> <tr><td style="padding: 2px;">5</td><td style="padding: 2px;">6</td><td style="padding: 2px;">7</td><td style="padding: 2px;">8</td></tr> <tr><td style="padding: 2px;">9</td><td style="padding: 2px;">10</td><td style="padding: 2px;">11</td><td style="padding: 2px;">12</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12
abcde																													
a	b	c	d	e																									
1	2	4	5																										
3		6																											
1	2	3	4																										
5	6	7	8																										
9	10	11	12																										

<pre> 1 3 { 'abcde' bd </pre>	<pre> 0 { T 1 2 3 4 0 {"1 T 1 5 9 </pre>	<pre> 0 2 { T 1 2 3 4 9 10 11 12 </pre>	<pre> &lt;1 2;2 3 </pre> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>2</td></tr> <tr><td>2</td><td>3</td></tr> </table> </div> (二重ボ	1	2	2	3				
1	2										
2	3										
<pre> (h=:1 1 2 0)}T 5 6 11 4 </pre>	<pre> ]k=:box h </pre>	<pre> ]k=:box h </pre>	<pre> (二重ボックス) &lt;1 2;2 3){ T 7 8 11 12 </pre>								
<pre> box=:[:&lt;"1],.i.@# </pre>	<pre> k { T 5 6 11 4 </pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <table border="1" style="border-collapse: collapse;"> <tr><td>1 0</td><td>1</td><td>2 2</td><td>0 3</td></tr> <tr><td></td><td>1</td><td></td><td></td></tr> </table> </div>	1 0	1	2 2	0 3		1			
1 0	1	2 2	0 3								
	1										

【「{.」 Head.Take】

<pre> ]K=:&gt;:i.3 1 2 3 </pre>	<pre> {. K 1 </pre>	<pre> 1{. K 1 </pre>	<pre> 2{.K 1 2 </pre>
<pre> _1{.K=:&gt;:i.3 3 </pre>	<pre> _2{. K 2 3 </pre>	<pre> 「{.」の片側形は先頭要素の取り。両側形は正(負)なら先頭(末尾)から個数分の取り。 </pre>	

【「{:」 Tail(片側形のみ)】

<pre> {:K=:&gt;:i.3 3 </pre>	<pre> 2 }. K 3 </pre>	<pre> (([:&lt;:#)}.) K 3 </pre>	<pre> 「{:」はアイテムの末尾の要素を出力する。 </pre>
------------------------------	-----------------------	---------------------------------	------------------------------------

【「[]」 Item Amend.Amend】

<pre> ]A=:2 4\$'abcdefgh' abcd efgh </pre>	<pre> 1 0 1 0 } A ebgd </pre>	<pre> 1 0 1 1 } A ebgh </pre>	<pre> 1 0 1 0 } &gt;:i.2 4 5 2 7 4 </pre>
<pre> ]B=: 'abcde' abcde </pre>	<pre> 'BD' (1 3)}B aBcDe </pre>	<pre> B abcde </pre>	<pre> ]B=: 'BD' (1 3)}B aBcDe </pre>

J言語では、訂正したものを定義し直さないと変更されない点に注意。

【「}.」 Behead.Drop : 「}:」 Curtail(片側形のみ)】

<pre> ]K=:1+i.3 1 2 3 </pre>	<pre> }. K 2 3 </pre>	<pre> 1 }.K 2 3 </pre>	<pre> 2 }.K 3 </pre>
<pre> _1}.K </pre>	<pre> _2}.K </pre>	<pre> }: K </pre>	<pre> 「}:」はアイテムの末尾の要素を除去する。 </pre>

1 2	1	1 2	
-----	---	-----	--

【乱数の生成 (? : Roll) と (? . : Deal)】

? 10\$10 8 3 8 1 2 8 0 0 2 1	? . 7\$10 6 5 9 2 4 9 0	? . 8\$10 6 5 9 2 4 9 0	「?」の片側形は重複を許さぬ乱数の生成。「? .」はシードを固定
10 ? 10 5 9 6 0 7 3 4 8 2 1	7 ? . 10 6 9 1 4 0 2 3	7 ? . 10 6 9 1 4 0 2 3	「?」の両側形は重複を許した乱数の生成。「? .」はシードを固定
? 2 8 \$ 10 5 7 8 8 2 1 9 7 9 9 1 4 3 9 7 2	? . 2 7 \$ 10 6 5 9 2 4 9 7 0 4 6 8 3 8 1	? . 2 7 \$ 10 6 5 9 2 4 9 7 0 4 6 8 3 8 1	0 から 9 までの重複乱数を 2×7 の形で生成している。「? .」を使えばシードを固定。

両側動詞に ウエーブ (~) つけりゃ 右引数を左にも  左右に数値がある場合には左右の引数を交換す ウエーブ・ピリ (~.nub) の片側形は 重複要素を 排除する ウエーブ・コロ				
---	--	--	--	--

<p>ン(~:)の片側形はダブリの位置に“0”を与える</p> <p>ウェーブ・コロン(~:)の両側形は各要素毎の不一致に“1”</p> <p>マイナス・コロン(-:)の両側形は引数マトメテ一致に“1”(match)</p>			

**【「~」Refrex.Passive Evoke】**

+/~ a=:1+i.3	a +/ a	*/~ a	a */ a
2 3 4	2 3 4	1 2 3	1 2 3
3 4 5	3 4 5	2 4 6	2 4 6
4 5 6	4 5 6	3 6 9	3 6 9
1 2 3 -- 5	5 - 1 2 3	5 %~ i.5	(i.5) % 5
4 3 2	4 3 2	0 0.2 0.4 0.6 0.8	0 0.2 0.4 0.6 0.8

**【(「~.」Nub/). (「~:」Nub Sieve/Not-Equal)】**

a =: 1 2 1 3 3 2 1	~. a	~: a	nub_s a	(~:a) #
nub_s=:[:+/:[:</\ "1=	1 2 3	1 1 0 1 0 0 0	1 1 0 1 0 0 0	a 1 2 3

「~.nub」の片側形は重複要素の排除、「~: nub sieve」は重複の箇所に「0」を与える。

(1 2)~:2 1	(1 2)~:1 2	「~:」の両側形は各要素毎の不一致に「1」
------------	------------	-----------------------

1 1	0 0	そうでなければ「0」を与える論理演算
(1 2)--:2 1	(1 2)--:1 2	「-:」の両側形は引数マトメテ一致に「1」
0	1	

【 ( < Less than) . ( > Larger than) . ( <: Lesser or equal) . ( >: Lager or equal)】

3 > 3.14	3 < 3.14	3 > 3	3 < 3	これらはいずれも「論理演算」で、「コロン(:)」の付いたほうは「=」がついてる場合である。
0	1	0	0	
3 >: 3.14	3 <: 3.14	3 >: 3	3 <: 3	
0	1	1	1	

【 (["."] Do.Number) : ([":"] Defort Format.Format) 】

<pre>1+":2 domain error (数値と文字の足算 はエラーになる)</pre>	<pre>1+".:2 3 (数値化してからな ら演算可能)</pre>	<pre>4j2 ": 3.14159 3.14 5j3 ": 3.14159 3.142</pre>	<p>「” :」は文字化で、両側形は書式関数。</p> <p>「” .」は数値化</p>
<pre>a=: '1+2+3'</pre>	<pre>". a 6</pre>	<p>「” .(do)」は文字で記述された演算内容を実行する。</p>	
<pre>]b=: '1 2 3', '4 5', ':' 1 2 3 4 5 \$ b 3 5</pre>	<pre>8 ". b 1 2 3 4 5 8 8 8 8</pre>	<pre>1j1 ". b 1 2 3 4 5 1j1 1j1 1j1 1j1</pre>	<p>「” .」の両側形は隙間に左引数で与えた数値を挿入する。</p>
<pre>]d=:3+i.5 3 4 5 6 7</pre>	<pre>". '5*d' 15 20 25 30 35</pre>	<p>「5*d」という演算結果を出力する。</p>	

セイム(same [,])は 左右のいずれかを 出力させる 便利な動詞  
 キャップ([:])はなんとも 不思議な動詞 何もしないで フォークを作る

【 [,] (Same) . [: (Cap) 】

<pre>2 3 [ 4 5 2 3</pre>	<pre>2 3 ] 4 5 4 5</pre>	<pre>(+:2 3) [4+5 4 6</pre>	<pre>(+:2 3) ] 4 + 5 9</pre>
<pre>([:*+:)2 3 4 16 36 64</pre>	<pre>([:&gt;+:)2 3 4 5 7 9</pre>	<pre>*:&amp;+: 2 3 4 16 36 64</pre>	<pre>&gt;:&amp;+: 2 3 4 5 7 9</pre>
<pre>2(+*-)1 3</pre>	<pre>]a=: (2+1)*(2-1) 3</pre>	<pre>2([:&gt;+*-)1 4</pre>	<pre>&gt;: a 4</pre>
<p>キャップは 演算結果に 関係せず</p>			