

Chapter 3 順次処理とトレース

一番オーソドックスな順次処理のトレース練習をやっていきます。下記のプログラムをトレースしていきましょう。最初の方は簡単です。

Section 1 初期化処理

<input type="radio"/> プログラム名 : PG1-1 <input type="radio"/> 整数型 : SU1 ● SU1 ← 0	SU					
	0					

<input type="radio"/> プログラム名 : PG1-2 <input type="radio"/> 整数型 : SU2 ● SU2 ← 100	SU2					
	100					

<input type="radio"/> プログラム名 : PG1-3 <input type="radio"/> 論理型 : SW1 ● SW1 ← true	SW1					
	true					

<input type="radio"/> プログラム名 : PG1-4 <input type="radio"/> 論理型 : SW2 ● SW2 ← false	SW2					
	false					

<input type="radio"/> プログラム名 : PG1-5 <input type="radio"/> 整数型 : Sum, Cnt ● Sum ← 0 ● Cnt ← 0	Sum	Cnt				
	0	0				

○ プログラム名 :PG1－6 ○ 整数型 :SU1 , SU2 ○ 論理型 :Flg <ul style="list-style-type: none"> ● SU1 ← 10 ● SU2 ← 7 ● Flg ← false 	SU1	SU2	Flg			
	10	7	false			

Section 2 通常の格納

○ プログラム名 :PG2－1 ○ 整数型 :SU1 , SU2 <ul style="list-style-type: none"> ● SU1 ← 5 ● SU2 ← SU1 	SU1	SU2				
	5	5				

○ プログラム名 :PG2－2 ○ 整数型 :SU1 , SU2 <ul style="list-style-type: none"> ● SU1 ← 2 ● SU2 ← 4 ● SU1 ← SU2 	SU1	SU2				
	2	4				
	4					

○ プログラム名 :PG2－3 ○ 論理型 :SW1 , SW2 <ul style="list-style-type: none"> ● SW1 ← true ● SW2 ← SW1 	SW1	SW2				
	true	true				

○ プログラム名 :PG2－4 ○ 論理型 :SW1 , SW2 <ul style="list-style-type: none"> ● SW1 ← false ● SW2 ← true ● SW2 ← SW1 	SW1	SW2				
	false	true				
		false				

○ プログラム名 :PG2－5 ○ 整数型:SU1, SU2 ○ 論理型:SW1, SW2 ● SW1 ← false ● SU2 ← 5 ● SW1 ← true ● SW2 ← SW1 ● SU1 ← SU2 ● SU2 ← 4 ● SW2 ← false	SU1	SU2	SW1	SW2		
	5	5	false	true		
		4	true	false		

Section 3 計算結果の格納

○ プログラム名 :PG3－1 ○ 整数型:Sum ● Sum ← 5 + 10	Sum					
	15					

○ プログラム名 :PG3－2 ○ 整数型:Sum ● Sum ← 0 ● Sum ← Sum + 6	Sum					
	0					
	6					

○ プログラム名 :PG3－3 ○ 整数型:Sum, SU ● SU ← 20 ● Sum ← SU - 10	Sum	SU				
	10	20				


- プログラム名:PG4-3
- 整数型:SU1, SU2, wk1
- 論理型:SW1, SW2, wk2
- SW1 ← true
- SU1 ← 5
- SW2 ← false
- SU2 ← 8
- wk2 ← SW1
- wk1 ← SU2
- SW1 ← SW2
- SU2 ← SU1
- Sw2 ← wk2
- SU1 ← wk1
- wk1 ← SU2
- wk2 ← SW1
- SU2 ← SU1
- SW1 ← SW2
- SW2 ← wk2
- SU1 ← wk1


[illegible]


Chapter 4 条件分岐処理とトレース

『条件式』によって処理実行の有無を判断したり、実行する処理を選択したりします。この条件分岐で躓いている人もいますが、簡単なプログラムから練習していきましょう。

Section 1 条件分岐①

○ プログラム名 : PG5-1 ○ 整数型 : SU ● SU ← 100  SU < 10 ● SU ← SU + 10	SU					
	100					

○ プログラム名 : PG5-2 ○ 整数型 : SU ● SU ← 0  SU < 10 ● SU ← SU + 10	SU					
	0					
	10					

○ プログラム名 : PG5-3 ○ 整数型 : SU1, SU2 ● SU1 ← 5 ● SU2 ← 3  SU1 > SU2 ● SU1 ← SU1 - SU2	SU1	SU2				
	5	3				
	2					

○ プログラム名 :PG5－4 ○ 整数型:SU1, SU2 ● SU1 ← 7 ● SU2 ← 8 ↑ (SU1 + SU2) > 10 ↓ ● SU1 ← SU1 + SU2	SU1	SU2				
	7	8				
	15					

○ プログラム名 :PG5－5 ○ 整数型:SU ○ 論理型:SW ● SW ← true ● SU ← 0 ↑ SW ↓ ● SU ← 10	SU	SW				
	0	true				
	10					

○ プログラム名 :PG5－6 ○ 整数型:SU ○ 論理型:SW ● SW ← false ↑ SW ↓ ● SU ← 10	SU	SW				
	?	false				

○ プログラム名 :PG5－7 ○ 整数型:SU ○ 論理型:SW ● SU ← 15 ● SW ← false ↑ SU > 10 ↓ ● SW ← true ↑ SW ↓ ● SU ← SU - 1	SU	SW				
	15	false				
	14	true				

Chapter 5 繰返し処理とトレース

繰返し処理です。ここから本格的に分からなくなってくる人が出てくると思います。とにかく、アルゴリズムは理解ではなく慣れが重要ですので1行1行実行してみましょう。

Section 1 繰返し処理①

<p>○ プログラム名 : PG8-1</p> <p>○ 整数型 : Res, Cnt</p> <ul style="list-style-type: none"> ● Res ← 0 ● Cnt ← 0 <p>■ Cnt < 10</p> <p>■</p> <ul style="list-style-type: none"> ● Res ← Res + Cnt ● Cnt ← Cnt + 1 	Res	Cnt				
	0	0				
	0	1				
	1	2				
	3	3				
	6	4				
	10	5				
	15	6				
	21	7				
	28	8				
	36	9				
	45	10				

<p>○ プログラム名 : PG8-2</p> <p>○ 整数型 : Res, Cnt</p> <ul style="list-style-type: none"> ● Res ← 0 ● Cnt ← 1 <p>■ Cnt < 10</p> <p>■</p> <ul style="list-style-type: none"> ● Res ← Res + Cnt ● Cnt ← Cnt + 1 	Res	Cnt				
	0	1				
	1	2				
	3	3				
	6	4				
	10	5				
	15	6				
	21	7				
	28	8				
	36	9				
	45	10				

○ プログラム名 :PG8－3

○ 整数型:Res, Cnt

● Res ← 0

● Cnt ← 0

■ Cnt ≤ 10

● Res ← Res + Cnt

● Cnt ← Cnt + 1

Res	Cnt				
0	0				
0	1				
1	2				
3	3				
6	4				
10	5				
15	6				
21	7				
28	8				
36	9				
45	10				
55	11				

○ プログラム名 :PG8－4

○ 整数型:Res, Cnt

● Res ← 0

● Cnt ← 1

■ Cnt ≤ 10

● Res ← Res + Cnt

● Cnt ← Cnt + 1

Res	Cnt				
0	1				
1	2				
3	3				
6	4				
10	5				
15	6				
21	7				
28	8				
36	9				
45	10				
55	11				

<p>○ プログラム名 : PG8-5</p> <p>○ 整数型 : Res, Cnt</p> <ul style="list-style-type: none"> ● Res ← 0 ● Cnt ← 1 <p>■ Cnt ≤ 5000</p> <ul style="list-style-type: none"> ● Res ← Res + Cnt ● Cnt ← Cnt * 3 	Res	Cnt				
	0	1				
	1	3				
	4	9				
	13	27				
	40	81				
	121	243				
	364	729				
	1093	2187				
	3280	6561				

<p>○ プログラム名 : PG8-6</p> <p>○ 整数型 : SU1, SU2, SU3, Cnt</p> <ul style="list-style-type: none"> ● SU1 ← 0 ● SU2 ← 0 ● SU3 ← 0 ● Cnt ← 1 <p>■ Cnt ≤ 5</p> <ul style="list-style-type: none"> ● SU1 ← SU1 + 3 ● SU2 ← SU2 + 1 ● SU3 ← SU3 + 2 ● SU1 ← SU1 + SU3 ● SU2 ← SU2 + SU1 ● SU3 ← SU3 + SU2 ● Cnt ← Cnt + 1 	SU1	SU2	SU3	Cnt		
	0	0	0	1		
	3	1	2			
	5	6	8	2		
	8	7	10			
	18	25	35	3		
	21	26	37			
	58	84	121	4		
	61	85	123			
	184	269	392	5		
	187	270	394			
	581	851	1245	6		

<p>○ プログラム名 :PG8－7</p> <p>○ 整数型 :SU, Sum, Avg, Cnt</p> <ul style="list-style-type: none"> ● SU ← 3 ● Sum ← 0 ● Cnt ← 1 <p>■ SU < 30</p> <ul style="list-style-type: none"> ● Sum ← Sum + SU ● SU ← SU + 3 ● Cnt ← Cnt + 1 <p>● Avg ← Sum / Cnt</p>	SU	Sum	Avg	Cnt		
	3	0	?	1		
	6	3		2		
	9	9		3		
	12	18		4		
	15	30		5		
	18	45		6		
	21	63		7		
	24	84		8		
	27	108		9		
	30	135	13	10		

<p>○ プログラム名 :PG8－8</p> <p>○ 整数型 :SU1, SU2, Data</p> <ul style="list-style-type: none"> ● SU1 ← 4321 ● SU2 ← 1234 ● Data ← SU1 % SU2 <p>■ Data ≠ 0</p> <ul style="list-style-type: none"> ● SU1 ← SU2 ● SU2 ← Data ● Data ← SU1 % SU2 	SU1	SU2	Data			
	4321	1234	619			
	1234	619	615			
	619	615	4			
	615	4	3			
	4	3	1			
	3	1	0			

<p>○ プログラム名 :PG9-3</p> <p>○ 整数型:SU, Sum</p> <ul style="list-style-type: none"> ● SU ← 1 ● Sum ← 0 <p>■ SU ≤ 15</p> <p>↑ (SU % 3) = 0</p> <ul style="list-style-type: none"> ● Sum ← Sum + SU <p>↓</p> <ul style="list-style-type: none"> ● SU ← SU + 1 	SU	Sum				
	1	0				
	2					
	3	3				
	4					
	5					
	6	9				
	7					
	8					
	9	18				
	10					
	11					
	12	30				
	13					
	14					
	15	45				
	16					

<p>○ プログラム名 :PG9-4</p> <p>○ 整数型:SU, Sum, Data</p> <p>○ 論理型:SW</p> <ul style="list-style-type: none"> ● SU ← 0 ● Sum ← 0 ● SW ← true <p>■ SW</p> <ul style="list-style-type: none"> ● SU ← SU + 1 ● Sum ← Sum + SU <p>↑ SU ≥ 15</p> <ul style="list-style-type: none"> ● Data ← Sum / SU ● SW ← false 	SU	Sum	Data	SW		
	0	0	?	true		
	1	1				
	2	3				
	3	6				
	4	10				
	5	15				
	6	21				
	7	28				
	8	36				
	9	45				
	10	55				
	11	66				
	12	78				
	13	91				
	14	105				
	15	120	8	false		

○ プログラム名 : PG9-5
 ○ 整数型 : SU, Sum1, Sum2, Sum3, Sum4

○ 論理型 : SW

- SU ← 0
- Sum1 ← 0
- Sum2 ← 0
- Sum3 ← 0
- Sum4 ← 0
- SW ← true

■ SW

- SU ← SU + 1
- ↑ (SU % 2) = 0
- ↓ ● Sum1 ← Sum1 + SU
- ↑ (SU % 3) = 0
- ↓ ● Sum2 ← Sum2 + SU
- ↑ (SU % 4) = 0
- ↓ ● Sum3 ← Sum3 + SU
- ↑ (SU % 5) = 0
- ↓ ● Sum4 ← Sum4 + SU
- ↑ SU > 30
- ↓ ● SW ← false

SU	Sum1	Sum2	Sum3	Sum4	SW
0	0	0	0	0	true
1					
2	2				
3		3			
4	6		4		
5				5	
6	12	9			
7					
8	20		12		
9		18			
10	30			15	
11					
12	42	30	24		
13					
14	56				
15		45		30	
16	72		40		
17					
18	90	63			
19					
20	110		60	50	
21		84			
22	132				
23					
24	156	108	84		
25				75	
26	182				
27		135			
28	210		112		
29					
30	240	165		105	
31					false

<p>○ プログラム名 : PG9-6</p> <p>○ 整数型 : SU, Sum1, Sum2</p> <ul style="list-style-type: none"> ● SU ← 1 ● Sum1 ← 0 ● Sum2 ← 0 <p>■ SU ≤ 15</p> <div> <p>↑ (SU % 2) = 0</p> <ul style="list-style-type: none"> ● Sum1 ← Sum1 + SU <hr/> <ul style="list-style-type: none"> ● Sum2 ← Sum2 + SU ● SU ← SU + 1 </div>	SU	Sum1	Sum2			
1	0	0				
2	2	1				
3		4				
4	6					
5		9				
6	12					
7		16				
8	20					
9		25				
10	30					
11		36				
12	42					
13		49				
14	56					
15		64				
16						

<p>○ プログラム名 : PG9-7</p> <p>○ 整数型 : SU, Sum1, Sum2</p> <p>○ 論理型 : SW</p> <ul style="list-style-type: none"> ● SU ← 1 ● Sum1 ← 0 ● Sum2 ← 0 ● SW ← true <p>■ SU < 15</p> <div> <p>↑ SW</p> <ul style="list-style-type: none"> ● Sum2 ← Sum1 + SU ● SW ← false <hr/> <ul style="list-style-type: none"> ● Sum1 ← Sum2 + SU ● SW ← true </div> <p>↓</p> <ul style="list-style-type: none"> ● SU ← SU + 1 	SU	Sum1	Sum2	SW		
1	0	0	true			
2		1	false			
3	3		true			
4		6	false			
5	10		true			
6		15	false			
7	21		true			
8		28	false			
9	36		true			
10		45	false			
11	55		true			
12		66	false			
13	78		true			
14		91	false			
15	105		true			

Section 3

繰返し処理③

ループなど一回動きが分かれば大したことない！！所詮同じことの繰り返しだ！！！！

[illegible]



SU				表示
0				FizzBuzz
1				1
2				2
3				Fizz
4				4
5				Buzz
6				Fizz
7				7
8				8
9				Fizz
10				Buzz
11				11
12				Fizz
13				13
14				14
15				FizzBuzz
16				16
17				17
18				Fizz
19				19
20				Buzz
21				Fizz
22				22
23				23
24				Fizz
25				Buzz
26				26
27				Fizz
28				28
29				29
30				FizzBuzz
31				

○ プログラム名 : PG10-3

○ 整数型 : SU, Sum1, Sum2, Cnt

● SU ← 1

● Cnt ← 0

● Sum1 ← 0

● Sum2 ← 0

■ SU < 8

↑ (SU % 2) = 0

■ Cnt < SU

● Sum1 ← Sum1 + SU

● Cnt ← Cnt + 1

■ Cnt < SU

● Sum2 ← Sum2 + SU

● Cnt ← Cnt + 1

● Cnt ← 0

● SU ← SU + 1

SU	Cnt	Sum1	Sum2		
1	0	0	0		
	1		1		
2	0				
	1	2			
	2	4			
3	0				
	1		4		
	2		7		
	3		10		
4	0				
	1	8			
	2	12			
	3	16			
	4	20			
5	0				
	1		15		
	2		20		
	3		25		
	4		30		
	5		35		
6	0				
	1	26			
	2	32			
	3	38			
	4	44			
	5	50			
	6	56			
7	0				
	1		42		
	2		49		
	3		56		
	4		63		
	5		70		
	6		77		
	7		84		
8	0				

Chapter 6 配列とトレース

まだまだこれから！！！！やれ！！

Section 1 配列(1次元配列)

配列とは、複数の値を格納できる変数です。変数を何十個と定義するより、一つの配列にまとめると、プログラムをより効率的に作ることができます。

①配列の定義

配列を利用するには、宣言部に記述します。

変数と同じように型を指定します。また、要素数(何個の値を格納できるか)も記述します。

ex) 整数を5個格納できる変数をarrayという名前 で定義する

○ 整数型:array[5]

ex) 文字を10個格納できる変数をcharsという名前 で定義する

○ 文字型:chars[10]

②配列の利用

配列は小さな箱を集めたようなものです。

整数型:array[5] と定義した配列は以下のようなイメージとなります。

	[1]	[2]	[3]	[4]	[5]
array					

この配列arrayに値を代入するには、複数ある要素(1つ1つの箱)の中から、1つを選ばなければなりません。1つの要素を選ぶには、要素番号(添え字)を利用します。

ex) arrayの要素に値を代入する

- array[1] ← 10
- array[3] ← 30
- array[5] ← array[1] + array[3]

処理後の配列arrayの中身は以下となります。

	[1]	[2]	[3]	[4]	[5]
array	10		30		40

○ プログラム名:PG11-1

○ 整数型:T[5]

- T[1] ← 0
- T[2] ← 1
- T[3] ← 2
- T[4] ← 3
- T[5] ← 4

	[1]	[2]	[3]	[4]	[5]
T	0				
T	0	1			
T	0	1	2		
T	0	1	2	3	
T	0	1	2	3	4

<p>○ プログラム名 :PG11-8</p> <p>○ 整数型:T[5]={4, 6, 3, 8, 5}, sum, i</p> <ul style="list-style-type: none"> ● sum ← 0 ● i ← 1 ■ i ≤ 5 <ul style="list-style-type: none"> ● sum ← sum + T[i] ● i ← i + 1 	sum	i				
	0	1				
	4	2				
	10	3				
	13	4				
	21	5				
	26	6				

<p>○ プログラム名 :PG11-9</p> <p>○ 整数型:T[5], su, i</p> <ul style="list-style-type: none"> ● su ← 1 ● i ← 1 ■ i ≤ 5 <div> <div> ↑ su % 2 = 0 </div> <div> ● T[i] ← su </div> <div> ↓ </div> </div> <ul style="list-style-type: none"> ● i ← i + 1 ● su ← su + 1 	su	i		[1]	[2]	[3]	[4]	[5]
	1	1	T					
	2	2	T	2				
	3		T					
	4	3	T	2	4			
	5		T					
	6	4	T	2	4	6		
	7		T					
	8	5	T	2	4	6	8	
	9		T					
	10	6	T	2	4	6	8	10
	11		T					
			T					
			T					
			T					

PG11-12

[illegible]

PG11-13

[illegible]

PG11-14

[illegible]

PG12-1

i	j	A	[1]	[2]	[3]	[4]	[5]
1	1	[1]	0	0	0	0	0
	2	[2]	0	0	0	0	0
	3	[3]	0	0	0	0	0
	4	[4]	0	0	0	0	0
	5	[5]	0	0	0	0	0
2	1						
	2						
	3						
	4						
	5						
3	1						
	2						
	3						
	4						
	5						
4	1						
	2						
	3						
	4						
	5						
5	1						
	2						
	3						
	4						
	5						
6							

PG12-2

i	j	A	[1]	[2]	[3]	[4]	[5]
1	1	[1]	1	2	3	4	5
	2	[2]	2	4	6	8	10
	3	[3]	3	6	9	12	15
	4	[4]	4	8	12	16	20
	5	[5]	5	10	15	20	25
	6						
2	1						
	2						
	3						
	4						
	5						
	6						
3	1						
	2						
	3						
	4						
	5						
	6						
4	1						
	2						
	3						
	4						
	5						
	6						
5	1						
	2						
	3						
	4						
	5						
6							

PG12-3

i	j	A[i][6]
1	1	2
	2	10
	3	17
	4	22
	5	25
	6	
2	1	4
	2	6
	3	11
	4	19
	5	28
	6	
3	1	2
	2	7
	3	11
	4	18
	5	25
	6	
4		

PG12-4

i	j	A[j][6]
1	1	1
	2	7
	3	7
	4	
2	1	9
	2	16
	3	15
	4	
3	1	18
	2	20
	3	19
	4	
4	1	19
	2	23
	3	27
	4	
5	1	27
	2	25
	3	29
	4	
6		

PG12-5

i	j	C	[1]	[2]	[3]
1	1	[1]	2	8	9
	2	[2]	5	8	7
	3	[3]	9	5	8
	4				
2	1				
	2				
	3				
	4				
3	1				
	2				
	3				
	4				
4					

PG12-6

i	j	A	[1]	[2]	[3]	[4]
1	1	[1]	1	8	9	18
	2	[2]	1	8	7	16
	3	[3]	9	4	3	16
	4	[4]	11	20	19	50
2	1					
	2					
	3					
	4					
3	1					
	2					
	3					
	4					
4						

PG12-7

i	j	w	A	[1]	[2]	[3]	[4]	[5]
1	1	1	[1]	5	4	3	2	1
	2	2	[2]	10	9	8	7	6
	3		[3]	15	14	13	12	11
2	1	6	[4]	20	19	18	17	16
	2	7	[5]	25	24	23	22	21
	3							
3	1	11						
	2	12						
	3							
4	1	16						
	2	17						
	3							
5	1	21						
	2	22						
	3							
6								

PG12-8

i	j	w	A	[1]	[2]	[3]	[4]	[5]
1	1	1	[1]	21	22	23	24	25
	2	2	[2]	16	17	18	19	20
	3	3	[3]	11	12	13	14	15
	4	4	[4]	6	7	8	9	10
	5	5	[5]	1	2	3	4	5
	6							
2	1	6						
	2	7						
	3	8						
	4	9						
	5	10						
	6							
3								

PG12-9

i	j	w	A	[1]	[2]	[3]	[4]	[5]
1	1	1	[1]	5	10	15	20	25
	2	2	[2]	4	9	14	19	24
	3	3	[3]	3	8	13	18	23
	4	4	[4]	2	7	12	17	22
	5		[5]	1	6	11	16	21
2	2	7						
	3	8						
	4							
3								

PG12-10

i	j	w	A	[1]	[2]	[3]	[4]	[5]
1	1	1	[1]	21	16	11	6	1
	2	2	[2]	22	17	12	7	2
	3	3	[3]	23	18	13	8	3
	4	4	[4]	24	19	14	9	4
	5		[5]	25	20	15	10	5
2	2	7						
	3	8						
	4							
3								

PG13-1

i	SP
0	0
1	1
2	2
3	3
4	4
5	5
0	4
1	3
2	2
3	1
4	0
5	

Stack

[0]	[1]	[2]	[3]	[4]
A				
A	B			
A	B	C		
A	B	C	D	
A	B	C	D	E

Text

[0]	[1]	[2]	[3]	[4]
A	B	C	D	E
A	B	C	D	E
A	B	C	D	E
A	B	C	D	E
A	B	C	D	E
A	B	C	D	E
E	B	C	D	E
E	D	C	D	E
E	D	C	D	E
E	D	C	B	E
E	D	C	B	A

PG14-1

i	j	w
1		
	2	
	3	2
	4	
	5	
	6	
2	3	8
	4	7
	5	
	6	
3	4	8
	5	7
	6	
4	5	8
	6	
5		

	[1]	[2]	[3]	[4]	[5]
A	7	8	2	3	5
A					
A	2	8	7	3	5
A					
A					
A					
A	2	7	8	3	5
A	2	3	8	7	5
A					
A					
A	2	3	7	8	5
A	2	3	5	8	7
A					
A	2	3	5	7	8
A					
A					

PG14-2

i	j	w
1		
	2	2
	3	
	4	
	5	
	6	
2	3	2
	4	
	5	
	6	
3	4	2
	5	4
	6	
4	5	2
	6	
5		

	[1]	[2]	[3]	[4]	[5]
A	2	8	7	4	5
A	8	2	7	4	5
A					
A					
A					
A					
A	8	7	2	4	5
A					
A					
A					
A	8	7	4	2	5
A	8	7	5	2	4
A					
A	8	7	5	4	2
A					
A					

PG14-3

i	j	w
5		
	4	
	3	
	2	
	1	
	0	
4	3	
	2	
	1	
	0	
3	2	9
	1	8
	0	
2	1	9
	0	
1		

	[1]	[2]	[3]	[4]	[5]
A	4	8	9	3	2
A					
A					
A					
A					
A					
A					
A					
A					
A					
A	4	9	8	3	2
A	8	9	4	3	2
A					
A	9	8	4	3	2
A					
A					

PG14-4

i	j	w	min
1	2		1
	3		3
	4		
	5		
	6	1	
2	3		2
			3
	4		
	5		
	6	8	
3	4		3
	5		5
	6	8	
4	5		4
			5
	6	9	
5			

	[1]	[2]	[3]	[4]	[5]
A	1	8	0	9	6
A					
A					
A					
A	0	8	1	9	6
A					
A					
A					
A					
A	0	1	8	9	6
A					
A					
A	0	1	6	9	8
A					
A					
A	0	1	6	8	9
A					

PG14-5

i	j	w		[1]	[2]	[3]	[4]	[5]
1	1		A	2	0	7	8	5
		2	A	0	2	7	8	5
	2		A					
	3		A					
	4	8	A	0	2	7	5	8
	5		A					
2	1		A					
	2		A					
	3	7	A	0	2	5	7	8
	4		A					
3	1		A					
	2		A					
	3		A					
4	1		A					
	2		A					
5			A					

PG14-6

i	j	w	sw		[1]	[2]	[3]	[4]	[5]
1			0	A	9	2	3	7	8
	1	9	1	A	2	9	3	7	8
	2	9	0	A	2	3	9	7	8
	3	9		A	2	3	7	9	8
	4	9		A	2	3	7	8	9
	5			A					
2	1		1	A					
	2			A					
	3			A					
	4			A					
3				A					

PG14-7

i	j	w		[1]	[2]	[3]	[4]	[5]
2			A	6	3	8	4	7
	5	7	A	6	3	8	7	4
	4		A					
	3	8	A	6	8	3	7	4
	2	8	A	8	6	3	7	4
	1		A					
3	5		A					
	4	7	A	8	6	7	3	4
	3	7	A	8	7	6	3	4
	2		A					
4	5	4	A	8	7	6	4	3
	4		A					
	3		A					
5	5		A					
	4		A					
6			A					

PG14-8

i	j	w		[1]	[2]	[3]	[4]	[5]
2			A	5	7	4	8	2
	2		A					
3	3	4	A	5	4	7	8	2
	2	4	A	4	5	7	8	2
	1		A					
4	4		A					
5	5	2	A	4	5	7	2	8
	4	2	A	4	5	2	7	8
	3	2	A	4	2	5	7	8
	2	2	A	2	4	5	7	8
	1		A					
6			A					

PG14-9

i	j	tmp		[1]	[2]	[3]	[4]	[5]
2			A	4	7	5	2	8
	2	7	A	4	7	5	2	8
3	3	5	A	4	7	7	2	8
	2		A	4	5	7	2	8
4	4	2	A	4	5	7	7	8
	3		A	4	5	5	7	8
	2		A	4	4	5	7	8
	1		A	2	4	5	7	8
5			A					

PG14-10

i	j	tmp		[1]	[2]	[3]	[4]	[5]
2			A	2	7	8	5	4
	2	7	A	2	7	8	5	4
3	3	8	A	2	7	8	5	4
4	4	5	A	2	7	8	8	4
	3		A	2	7	7	8	4
	2		A	2	5	7	8	4
5	5	4	A	2	5	7	8	8
	4		A	2	5	7	7	8
	3		A	2	5	5	7	8
	2		A	2	4	5	7	8
6			A					