

Handbook of SAS® Data Step Programming

Execution Phase of Program 6.11

Arthur Li

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

At the beginning of the 1st iteration:

At the beginning of the 1 st iteration:								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		1			0		.	

ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G	
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
.		

ARRAY
TRACKING

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
→ set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration:

1st iteration:

						G[1]		G[2]		G[3]							
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		0		.	
ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]							
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K						
.								

G [J]

ALL_G 1

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration:

		G[1]		G[2]		G[3]											
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		0		.	

ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G	
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
.		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  → i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration:

1st iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		.	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
.		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (1st DO loop):

1 st iteration (1 st DO loop):								G[1]		G[2]		G[3]					
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
.		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]		M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (1st DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (1st DO loop):

1st iteration (1st DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (2nd DO loop):

1 st iteration (2 nd DO loop):								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]		M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    ➔ all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (2nd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		2	

ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G	
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		

G [J]		G1		G2		G3	
		1		2		3	
ALL_G	[I,J]	1	M_G1	M_G2	M_G3	2	F_G1
		2	F_G1	F_G2	F_G3	3	F_G2

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (2nd DO loop):

1st iteration (2nd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (3rd DO loop):

1 st iteration (3 rd DO loop):								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]		M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (3rd DO loop):

1st iteration (3rd DO loop):

								G[1]	G[2]		G[3]						
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	

G [J]		G1		G2		G3	
		1		2		3	
ALL_G [I,J]		M_G1	M_G2	M_G3			
	2	F_G1	F_G2	F_G3			

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (3rd DO loop):

1 st iteration (3 rd DO loop):								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration (4th DO loop):

1st iteration (4th DO loop):

								G[1]		G[2]		G[3]					
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		4	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]		M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  → if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

1st iteration:

1st iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
1		1		0		1		A		B		F		1		4	



ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
set dat1;
by id;
array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
array g[3];
retain all_g;
if first.id then i = 0;
i + 1;
do j = 1 to 3;
    all_g[i,j] = g[j];
end;
if last.id;
run;
    
```

↓ ↓ ↓ ↓

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration:

2nd iteration:

								G[1]		G[2]		G[3]							
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D		
2		1		0		1		A		B		F		1		.			
↑		↑		↑		↑		↑		↑		↑		↑		↑			
ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G									
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]									
																G [J]			

M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	
↑		↑		↑		↑		↑		↑	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
→ set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration:

2nd iteration:

						G[1]		G[2]		G[3]							
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		1		.	
ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]							
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K						
A		B		F		.		.		.							

G [J]

ALL_G 1

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		1		.	

ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G	
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration:

2nd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		.	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (1st DO loop):

2nd iteration (1st DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		.		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    ➔ all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (1st DO loop):

2nd iteration (1st DO loop):

								G[1]	G[2]		G[3]						
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (1st DO loop):

2nd iteration (1st DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (2nd DO loop):

2nd iteration (2nd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		.		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (2nd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		2	

ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G	
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G	[I,J]	1	M_G1	M_G2
		2	F_G1	F_G2

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (2nd DO loop):

2nd iteration (2nd DO loop):

				G[1]		G[2]		G[3]									
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		.	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (3rd DO loop):

2nd iteration (3rd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		.	

G [J]				
		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    → all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (3rd DO loop):

2nd iteration (3rd DO loop):

								G[1]		G[2]		G[3]					
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1		G2		G3	
		1		2		3	
ALL_G [I,J]	1	M_G1		M_G2		M_G3	
	2	F_G1		F_G2		F_G3	

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (3rd DO loop):

2nd iteration (3rd DO loop):

		G[1]		G[2]		G[3]											
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration (4th DO loop):

2nd iteration (4th DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		4	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		4	



ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

2nd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		4	



ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

2nd iteration:

2 nd iteration:								G[1]		G[2]		G[3]					
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
2		0		1		1		B		A		C		2		4	

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration:

3rd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		0		1		1		B		A		C		2		.	



ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
➔ set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration:

3rd iteration:

		G[1]		G[2]		G[3]											
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		2		.	
ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]							
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K						
A		B		F		B		A		C							

G [J]

ALL_G 1

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  → if first.id then i = 0;
    i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration:

3rd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		0		.	

ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G	
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1			G2			G3		
		1			2			3		
ALL_G	[I,J]	1	M_G1	M_G2	M_G3	2	F_G1	F_G2	F_G3	
		2	F_G1	F_G2	F_G3					

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  → i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration:

3rd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		.	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (1st DO loop):

3rd iteration (1st DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
A		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (1st DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		1	

ALL_G [1,1]	ALL_G [1,2]	ALL_G [1,3]	ALL_G [2,1]	ALL_G [2,2]	ALL_G [2,3]
M_G1 K	M_G2 K	M_G3 K	F_G1 K	F_G2 K	F_G3 K
B	B	F	B	A	C

G [J]	G1	G2	G3
	1	2	3
ALL_G [I,J]	1 M_G1	M_G2	M_G3
	2 F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (1st DO loop):

3rd iteration (1st DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (2nd DO loop):

3rd iteration (2nd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		B		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    ➔ all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (2nd DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (2nd DO loop):

3rd iteration (2nd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		F		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (3rd DO loop):

3rd iteration (3rd DO loop):

				G[1]		G[2]		G[3]									
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		F		B		A		C	

G [J]				
		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    ➔ all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (3rd DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (3rd DO loop):

3rd iteration (3rd DO loop):

				G[1]		G[2]		G[3]									
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1			G2			G3		
		1			2			3		
ALL_G [I,J]	1	M_G1			M_G2			M_G3		
	2	F_G1			F_G2			F_G3		

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration (4th DO loop):

3rd iteration (4th DO loop):

				G[1]		G[2]		G[3]									
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		4	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]		M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

3rd iteration:

3rd iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
3		1		0		2		B		A		D		1		4	



ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration:

4th iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		1		0		2		B		A		D		1		.	

ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G	
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
→ set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration:

4th iteration:

								G[1]		G[2]		G[3]									
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D				
4		0		1		2		C		B		C		1		.					
ALL_G		ALL_G		ALL_G		ALL_G		ALL_G		ALL_G											
[1,1]		[1,2]		[1,3]		[2,1]		[2,2]		[2,3]											

G [J]

ALL_G 1

M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration:

4th iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		1		.	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1			G2			G3		
		1			2			3		
ALL_G [I,J]	1	M_G1	M_G2	M_G3	2	F_G1	F_G2	F_G3	3	
	2	F_G1	F_G2	F_G3						

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  → i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration:

4th iteration:

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		.	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1			G2			G3		
		1			2			3		
ALL_G [I,J]	1	M_G1	M_G2	M_G3	2	F_G1	F_G2	F_G3		
	2	F_G1	F_G2	F_G3						

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (1st DO loop):

4th iteration (1st DO loop):

								G[1]		G[2]		G[3]					
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		B		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    ➔ all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (1st DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		1	

<div></div>											
ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (1st DO loop):

4th iteration (1st DO loop):

								G[1]		G[2]		G[3]					
N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		1	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```
data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;
```

Dat2:

	ID	M_G1	M_G2
1	1	A	B

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (2nd DO loop):

N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		A		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    → all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (2nd DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		B		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (2nd DO loop):

4th iteration (2nd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		2	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		B		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (3rd DO loop):

4th iteration (3rd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		B		C	

G [J]				
		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    → all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (3rd DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		3	

ALL_G	ALL_G	ALL_G	ALL_G	ALL_G	ALL_G
[1,1]	[1,2]	[1,3]	[2,1]	[2,2]	[2,3]
M_G1	K	M_G2	K	M_G3	K
F_G1	K	F_G2	K	F_G3	K
B	A	D	C	B	C

G [J]	G1	G2	G3
	1	2	3
ALL_G	M_G1	M_G2	M_G3
[I,J]	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (3rd DO loop):

4th iteration (3rd DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		3	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		B		C	

G [J]		G1			G2			G3		
		1			2			3		
ALL_G [I,J]	1	M_G1			M_G2			M_G3		
	2	F_G1			F_G2			F_G3		

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (4th DO loop):

4th iteration (4th DO loop):

								G[1]		G[2]		G[3]					
_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		4	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		B		C	

G [J]				
		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C

4th iteration (4th DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		4	



ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		B		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3

Execution Phase of Program 6.11

```

data dat2 (drop = i j g1 - g3);
  set dat1;
  by id;
  array all_g [2,3] $ m_g1 - m_g3 f_g1 - f_g3;
  array g[3];
  retain all_g;
  if first.id then i = 0;
  i + 1;
  do j = 1 to 3;
    all_g[i,j] = g[j];
  end;
  if last.id;
run;

```

Dat1:

	ID	G1	G2	G3
1	1	A	B	F
2	1	B	A	C
3	2	B	A	D
4	2	C	B	C

Dat2:

	ID	M_G1	M_G2	M_G3	F_G1	F_G2	F_G3
1	1	A	B	F	B	A	C
2	2	B	A	D	C	B	C

4th iteration (4th DO loop):

_N	D	FIRST.ID	D	LAST.ID	D	ID	K	G1	D	G2	D	G3	D	I	D	J	D
4		0		1		2		C		B		C		2		4	

ALL_G [1,1]		ALL_G [1,2]		ALL_G [1,3]		ALL_G [2,1]		ALL_G [2,2]		ALL_G [2,3]	
M_G1	K	M_G2	K	M_G3	K	F_G1	K	F_G2	K	F_G3	K
B		A		D		C		B		C	

G [J]		G1	G2	G3
		1	2	3
ALL_G [I,J]	1	M_G1	M_G2	M_G3
	2	F_G1	F_G2	F_G3