

Handbook of SAS® Data Step Programming

Execution Phase of Program 4.5

Arthur Li

Execution Phase of Program 4.5

```
→ data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		1				



1st iteration:

❖ $_N_ \leftarrow 1$

Execution Phase of Program 4.5

```
→ data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		1				

1st iteration:

❖ $_N_ \leftarrow 1$

❖ $\text{FIRST.ID} \leftarrow 1, \text{LAST.ID} \leftarrow 1$

Execution Phase of Program 4.5

```
→ data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		1				




1st iteration:

- ❖ $_N_ \leftarrow 1$
- ❖ $\text{FIRST.ID} \leftarrow 1, \text{LAST.ID} \leftarrow 1$
- ❖ Other variables \leftarrow missing

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  → set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:



	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		0		A01		1		3		.		.		.	




1st iteration:

❖ The SET statement copies the 1st observation → PDV

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  → set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:



	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		0		A01		1		3		.		.		.	



1st iteration:

- ❖ The SET statement copies the 1st observation → PDV
- ❖ FIRST.ID ← 1 since this is the 1st observation for A01
- ❖ LAST.ID ← 0 since this is not the last observation for A01

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  ➔ by id;  
  ➔ retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		0		A01		1		3		.		.		.	

1st iteration:

❖ Both BY and RETAIN statements are declarative statements


Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  → if first.id then do;  
  →   s1 = .; s2 = .; s3 = .;  
  → end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		0		A01		1		3		.		.		.	



1st iteration:

❖ Since FIRST.ID =1, S1 – S3 are set to missing

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  → if time = 1 then s1 = score;  
  → else if time = 2 then s2 = score;  
  → else s3 = score;  
    if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		0		A01		1		3		3		.		.	



1st iteration:

❖ Since TIME = 1, S1 ← SCORE (3)

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  → if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
1		1		0		A01		1		3		3		.		.	



1st iteration:

- ❖ Since LAST.ID ≠ 1, (the subsetting IF statement is false), no further statements are processed for the current observation. SAS immediately returns to the beginning of the DATA step

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		1		0		A01		1		3		3		.		.	



2nd iteration:

❖ _N_ ↑2

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		1		0		A01		1		3		3		.		.	

2nd iteration:

❖ $_N_ \uparrow 2$

❖ FIRST.ID and LAST.ID are retained; they are automatic variables

Execution Phase of Program 4.5

```
→ data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		1		0		A01		1		3		3		.		.	



2nd iteration:

- ❖ $_N_ \uparrow 2$
- ❖ FIRST.ID and LAST.ID are retained; they are automatic variables
- ❖ ID, TIME, SCORE are retained; they are from input dataset

Execution Phase of Program 4.5

```
→ data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		1		0		A01		1		3		3		.		.	




2nd iteration:

- ❖ $_N_ \uparrow 2$
- ❖ FIRST.ID and LAST.ID are retained; they are automatic variables
- ❖ ID, TIME, SCORE are retained; they are from input dataset
- ❖ S1, S2, and S3 are retained because of the RETAIN statement

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  ➔ set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:



	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		0		0		A01		2		4		3		.		.	




2nd iteration:

❖ The SET statement copies the 2nd observation to the PDV

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  → set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:



	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		0		0		A01		2		4		3		.		.	

2nd iteration:

- ❖ The SET statement copies the 2nd observation to the PDV
- ❖ FIRST.ID ← 0; this is not the first observation for A01
- ❖ LAST.ID ← 0; this is not the last observation for A01 either

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  → if first.id then do;  
  →   s1 = .; s2 = .; s3 = .;  
  → end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		0		0		A01		2		4		3		.		.	



2nd iteration:

❖ Since FIRST.ID ≠ 1, no execution

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  ➔ if time = 1 then s1 = score;  
  ➔ else if time = 2 then s2 = score;  
  ➔ else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		0		0		A01		2		4		3		4		.	



2nd iteration:

❖ Since TIME = 2, S2 ← SCORE (4)

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  → if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
2		0		0		A01		2		4		3		4		.	



2nd iteration:

- ❖ Since LAST.ID ≠ 1, (the subsetting IF statement is false), SAS immediately returns to the beginning of the DATA step

Execution Phase of Program 4.5

```
→ data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
3		0		0		A01		2		4		3		4		.	



3rd iteration:

❖ _N_ ↑3

❖ The rest of the variables are retained

Execution Phase of Program 4.5

```
data wide (drop=time score);  
➔ set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2



N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
3		0		1		A01		3		5		3		4		.	



3rd iteration:

❖ The SET statement copies the 3rd observation → PDV

Execution Phase of Program 4.5

```
data wide (drop=time score);  
➔ set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2



N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
3		0		1		A01		3		5		3		4		.	



3rd iteration:

- ❖ The SET statement copies the 3rd observation → PDV
- ❖ FIRST.ID ← 0; this is not the first observation for A01
- ❖ LAST.ID ← 1; this is the last observation for A01

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  → if first.id then do;  
  →   s1 = .; s2 = .; s3 = .;  
  → end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
3		0		1		A01		3		5		3		4		.	



3rd iteration:

❖ Since FIRST.ID ≠ 1, no execution

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  → if time = 1 then s1 = score;  
  → else if time = 2 then s2 = score;  
  → else s3 = score;  
    if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
3		0		1		A01		3		5		3		4		5	



3rd iteration:

❖ Since TIME = 3, S3 ← SCORE (5)

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  → if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
3		0		1		A01		3		5		3		4		5	



3rd iteration:

- ❖ Since LAST.ID = 1 (the subsetting IF statement is true), SAS continues to execute the remaining statements

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
→ run;
```


LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
3		0		1		A01		3		5		3		4		5	



3rd iteration:

❖ SAS reaches the end of the 3rd iteration,

- ❑ The implicit OUTPUT statement executes
- ❑ SAS returns to the beginning of the DATA step to begin the 4th iteration

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
4		0		1		A01		3		5		3		4		5	



4th iteration:

❖ _N_ ↑4

❖ The rest of the variables are retained

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  → set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5



N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
4		1		0		A02		1		4		3		4		5	



4th iteration:

❖ The SET statement copies the 4th observation → PDV

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  → set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5



N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
4		1		0		A02		1		4		3		4		5	



4th iteration:

- ❖ The SET statement copies the 4th observation → PDV
- ❖ FIRST.ID ← 1; this is the first observation for A02
- ❖ LAST.ID ← 0; this is not the last observation for A02

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  → if first.id then do;  
  →   s1 = .; s2 = .; s3 = .;  
  → end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```





LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
4		1		0		A02		1		4		.		.		.	



4th iteration:

❖ Since FIRST.ID = 1, S1 – S3 are set to *missing*

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  → if time = 1 then s1 = score;  
  → else if time = 2 then s2 = score;  
  → else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
4		1		0		A02		1		4		4		.		.	



4th iteration:

❖ Since TIME = 1, S1 ← SCORE (4)

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  → if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
4		1		0		A02		1		4		4		.		.	



4th iteration:

- ❖ Since LAST.ID ≠ 1, (the subsetting IF statement is false), SAS immediately returns to the beginning of the DATA step

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
5		1		0		A02		1		4		4		.		.	



5th iteration:

❖ _N_ ↑5

❖ The rest of the variables are retained

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  → set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5



N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
5		0		1		A02		3		2		4		.		.	



5th iteration:

❖ The SET statement copies the 5th observation → PDV

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  → set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5



N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
5		0		1		A02		3		2		4		.		.	



5th iteration:

- ❖ The SET statement copies the 5th observation → PDV
- ❖ FIRST.ID ← 0; this is not the first observation for A02
- ❖ LAST.ID ← 1; this is the last observation for A02

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  → if first.id then do;  
  →   s1 = .; s2 = .; s3 = .;  
  → end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
5		0		1		A02		3		2		4		.		.	



5th iteration:

❖ Since FIRST.ID ≠ 1, no execution

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  → if time = 1 then s1 = score;  
  → else if time = 2 then s2 = score;  
  → else s3 = score;  
    if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
5		0		1		A02		3		2		4		.		2	



5th iteration:

❖ Since TIME = 3, S3 ← SCORE (2)

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  → if last.id;  
run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
5		0		1		A02		3		2		4		.		2	



5th iteration:

- ❖ Since LAST.ID = 1 (the subsetting IF statement is true), SAS continues to execute the remaining statements

Execution Phase of Program 4.5

```
data wide (drop=time score);  
  set long;  
  by id;  
  retain s1 - s3;  
  if first.id then do;  
    s1 = .; s2 = .; s3 = .;  
  end;  
  if time = 1 then s1 = score;  
  else if time = 2 then s2 = score;  
  else s3 = score;  
  if last.id;  
→ run;
```

LONG:

	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

WIDE:

	ID	S1	S2	S3
1	A01	3	4	5
2	A02	4	.	2

N	D	FIRST.ID	D	LAST.ID	D	ID	K	TIME	D	SCORE	D	S1	K	S2	K	S3	K
5		0		1		A02		3		2		4		.		2	

5th iteration:

- ❖ SAS reaches the end of the 5th iteration,
- ❖ The implicit OUTPUT statement copies variables marked with (K) to the data set