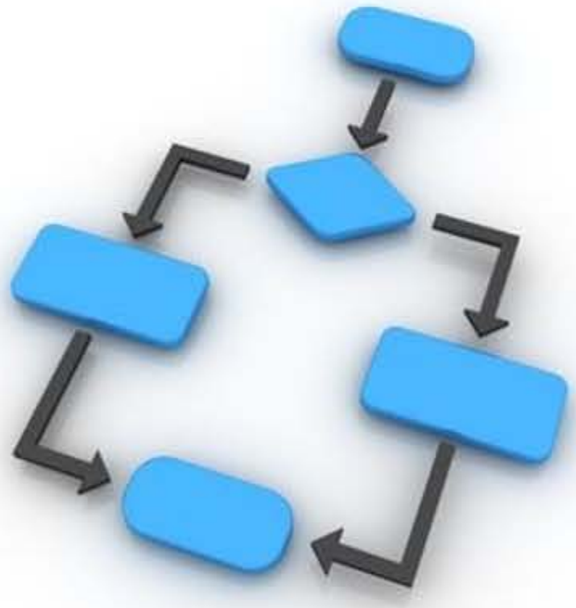
Tracing Algorithms



Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5		

Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5	1	

Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5	1	
		5

Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5	1	
		5
	2	

Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5	1	
		5
	2	
		10

Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5	1	
		5
	2	
		10
	3	

Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5	1	
		5
	2	
		10
	3	
		15

Tracing an Algorithm

Tracing allows us to manually test the inner workings of an algorithm to ensure everything works as intended.

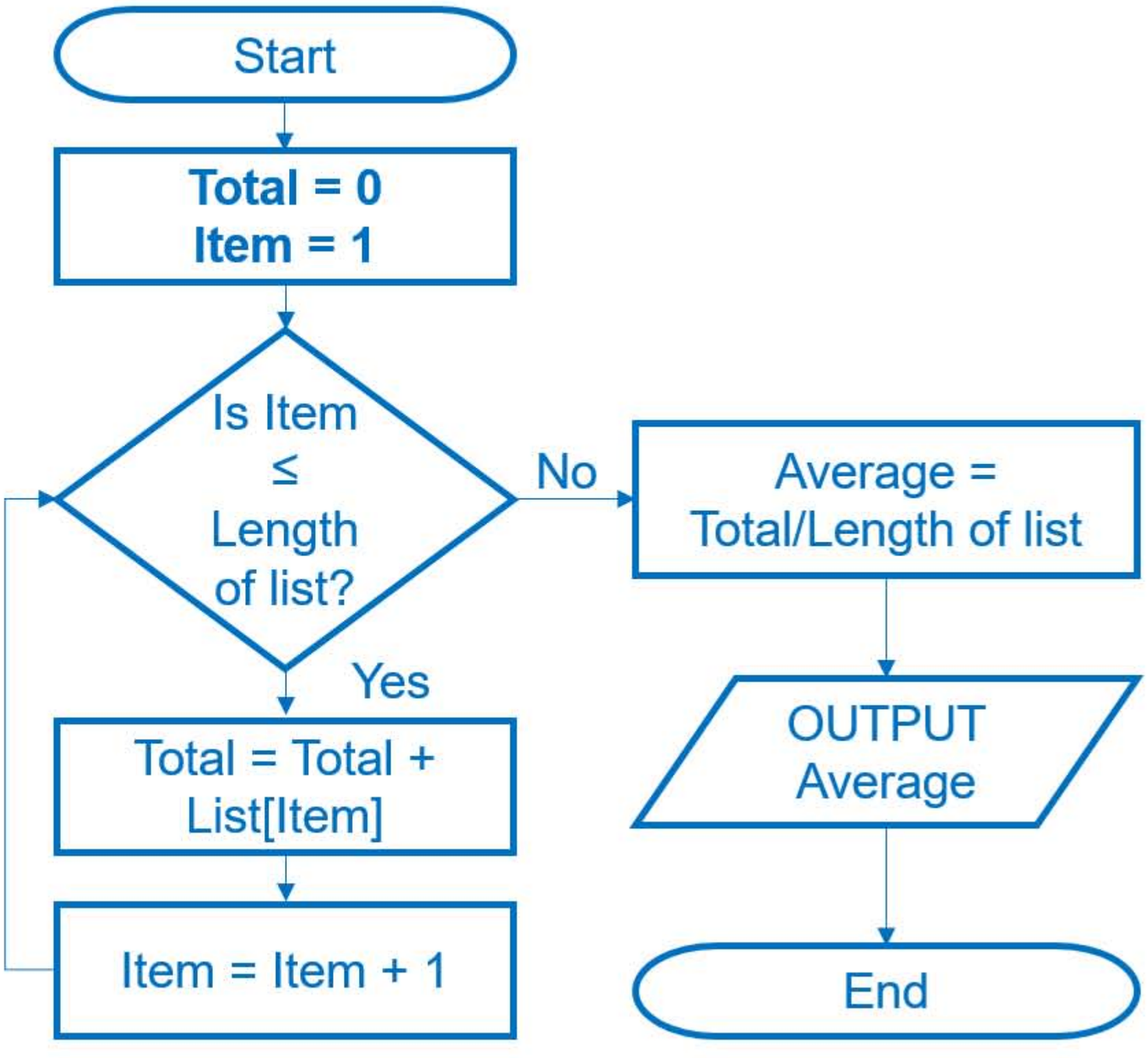
Tracing is carried out using a trace table that has a column for each variable. There may also be a column for the output.

Each time a variable changes, its value is placed in a new row of the trace table.

```
Num ← 5
Index ← 1
REPEAT
    OUTPUT Index*Num
    Index ← Index + 1
UNTIL c > 3
```

Num	Index	Output
5	1	
		5
	2	
		10
	3	
		15
	4	

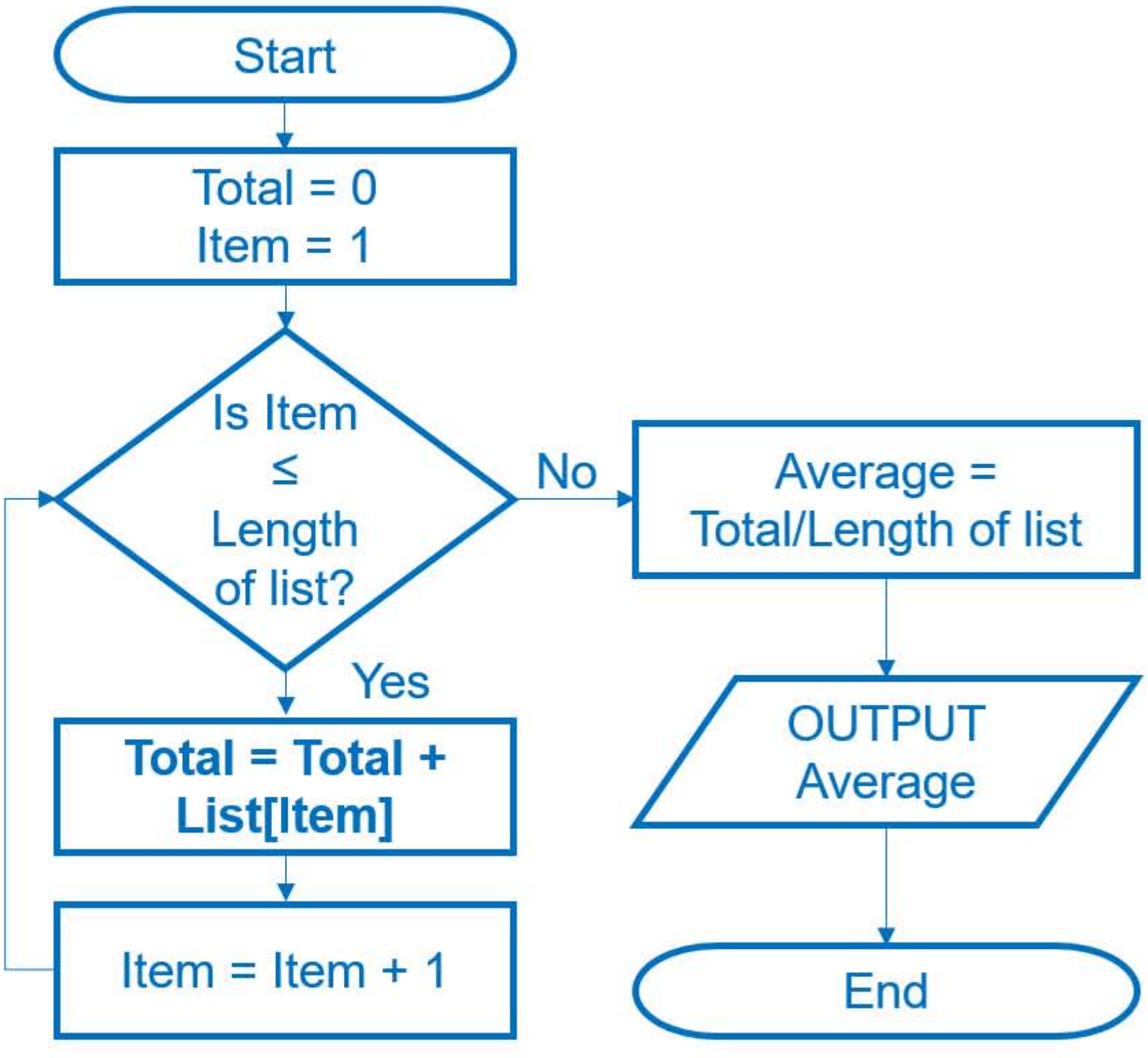
Tracing a Flow Chart



List	5	6	4	5

Total	Item	Average	Output
0	1		

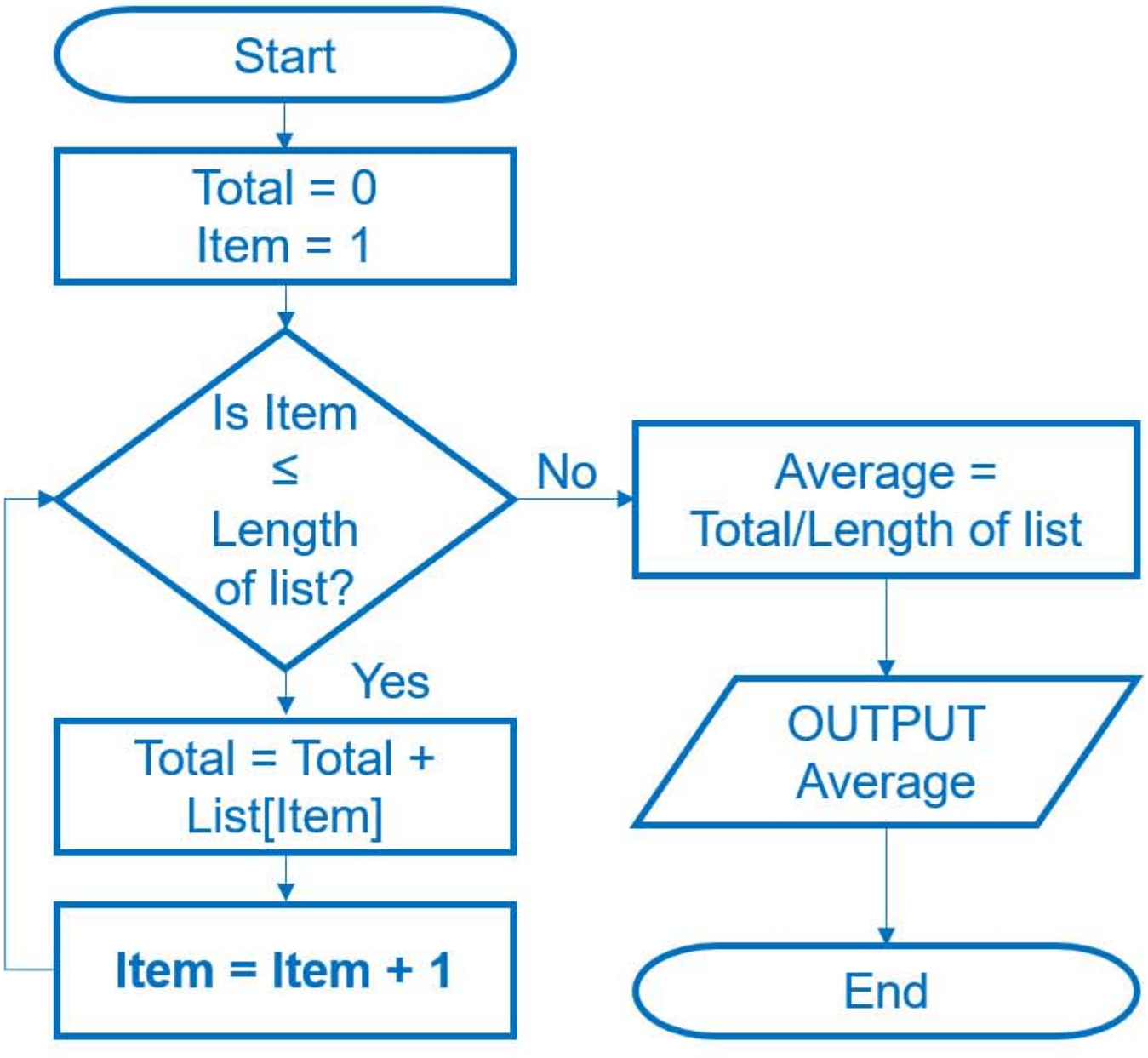
Tracing a Flow Chart



Total	Item	Average	Output
0	1		
5			

List	**5**	6	4	5

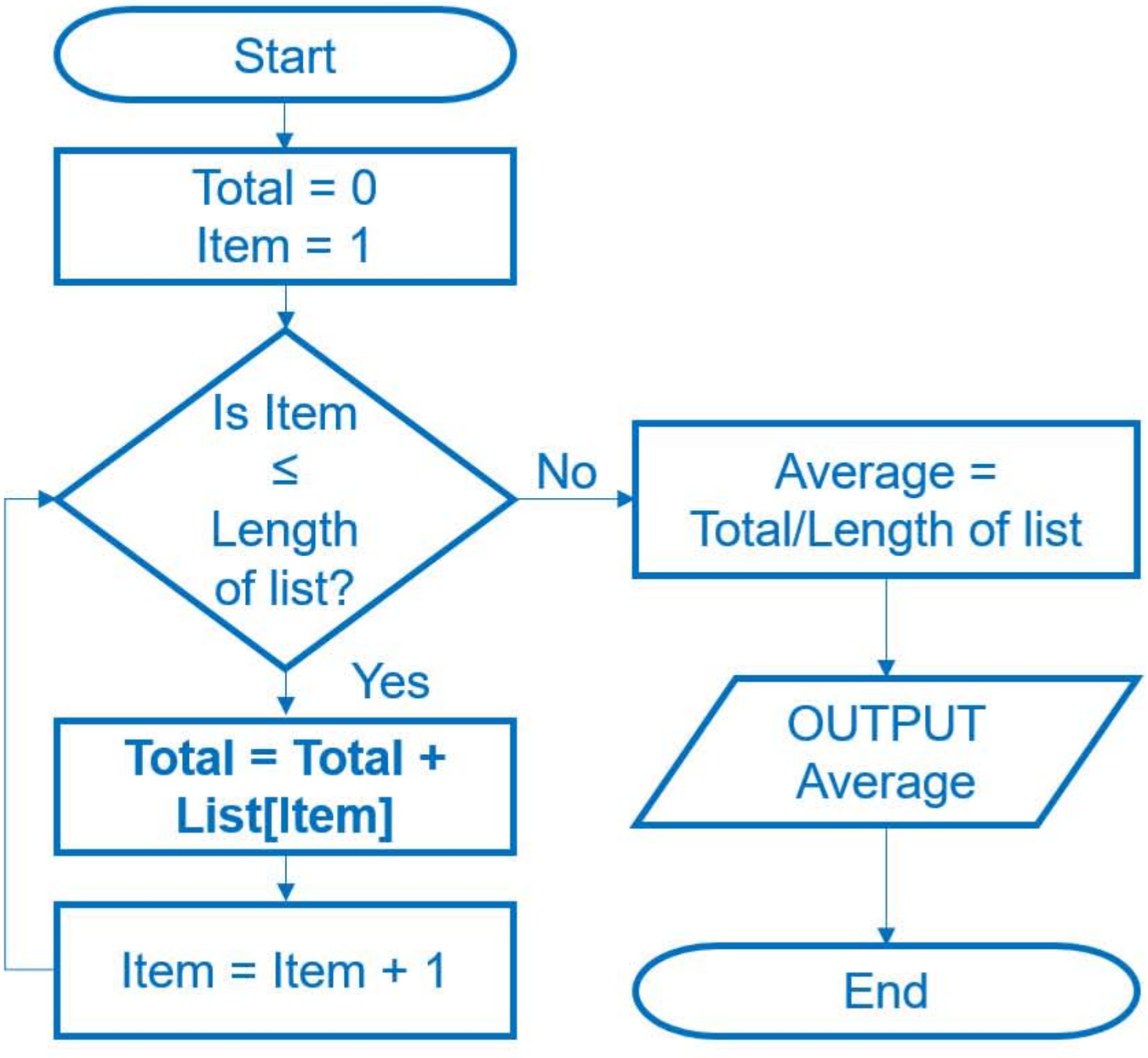
Tracing a Flow Chart



Total	Item	Average	Output
0	1		
5			
	2		

List	5	6	4	5

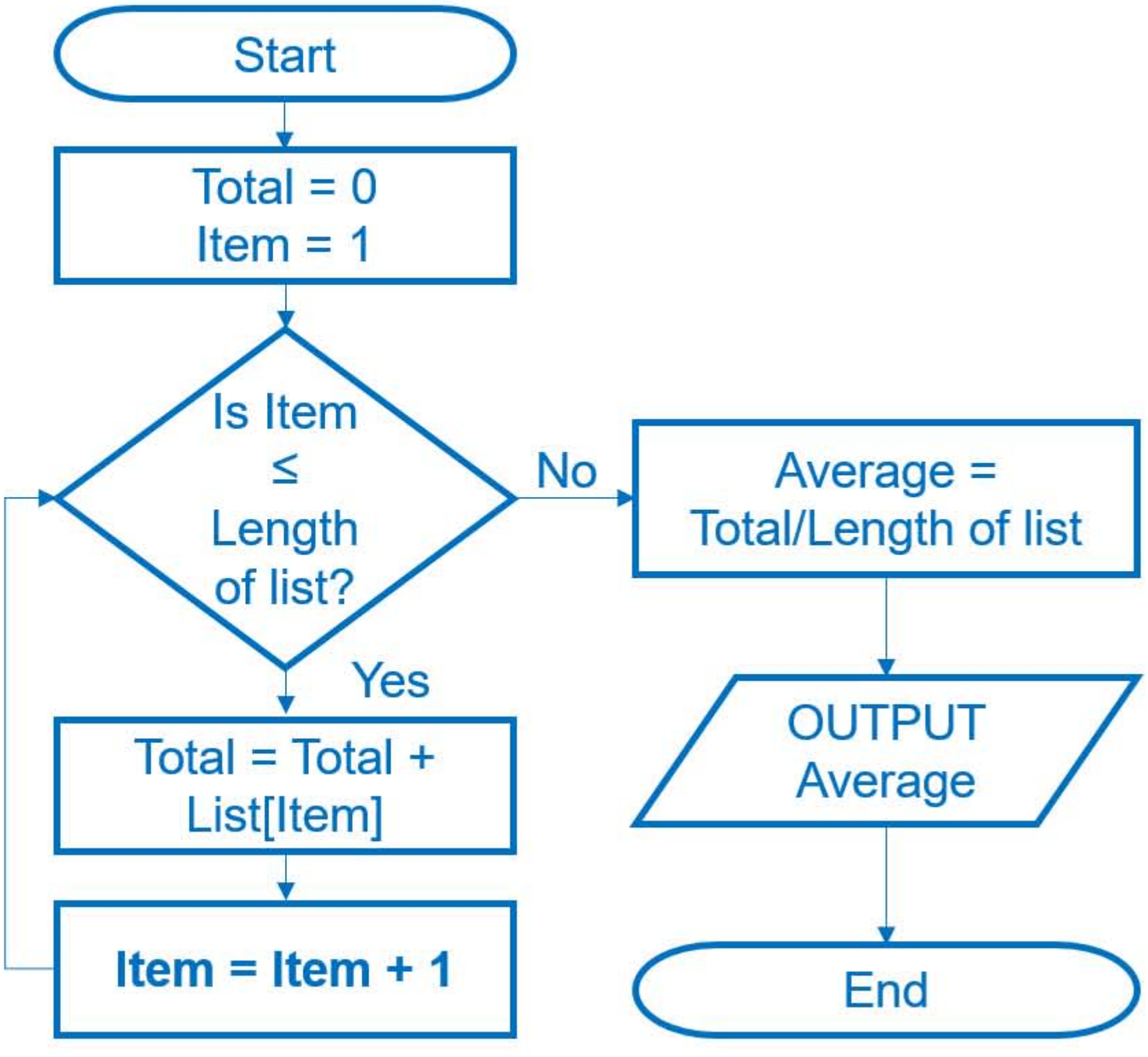
Tracing a Flow Chart



List	5	**6**	4	5

Total	Item	Average	Output
0	1		
5			
	2		
11			

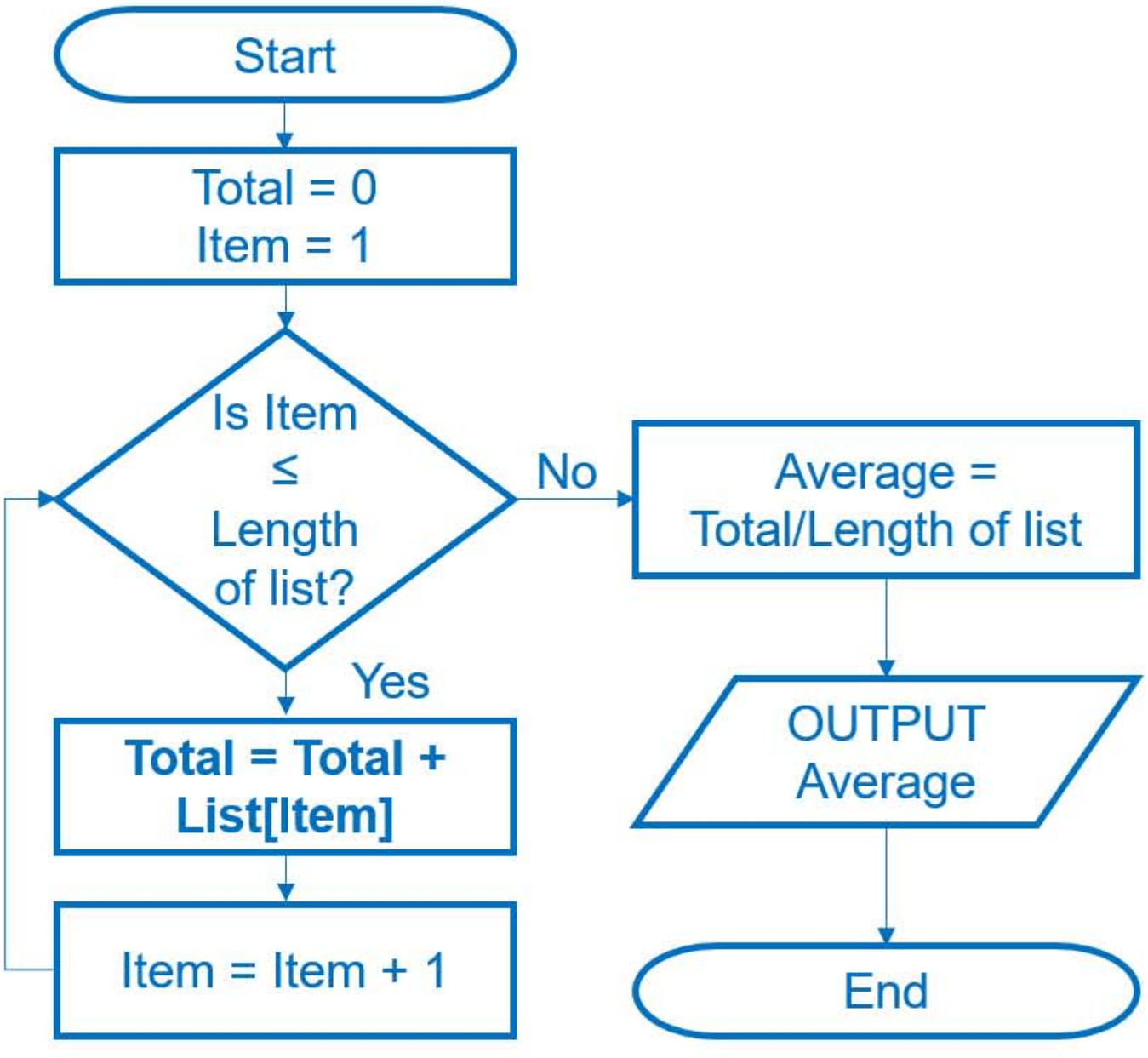
Tracing a Flow Chart



List	5	6	4	5

Total	Item	Average	Output
0	1		
5			
	2		
11			
	3		

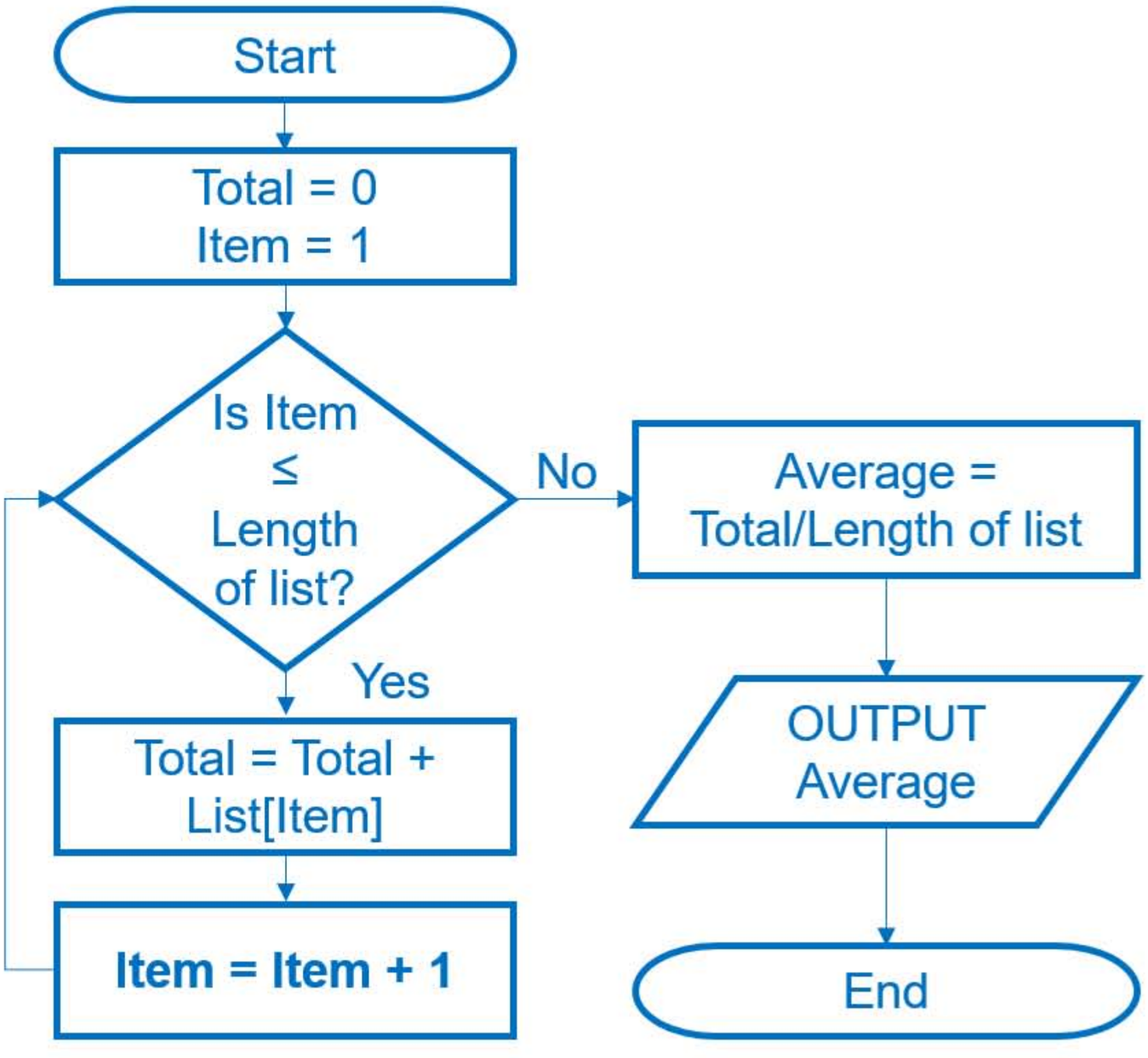
Tracing a Flow Chart



Total	Item	Average	Output
0	1		
5			
	2		
11			
	3		
15			

List	5	6	**4**	5

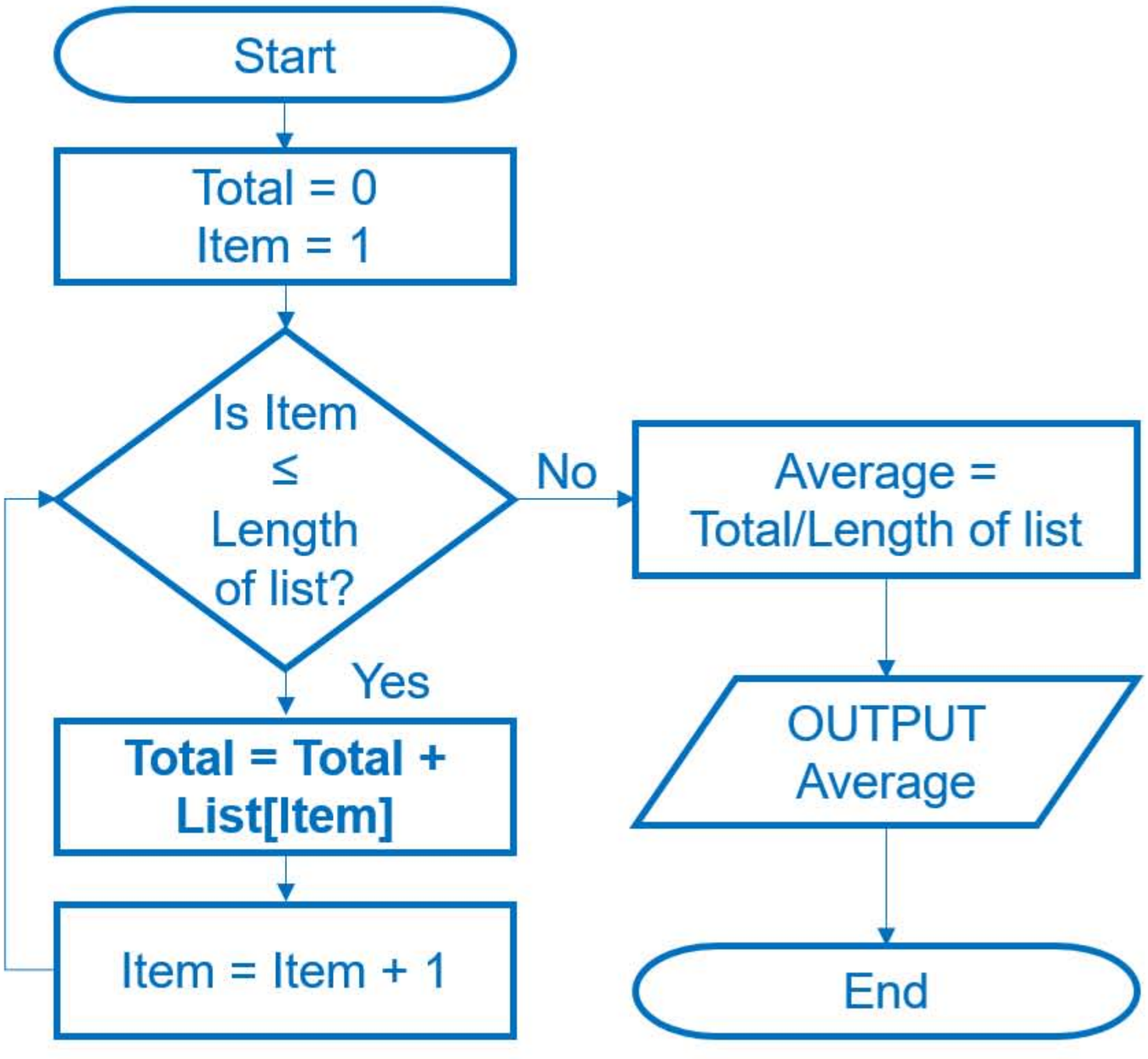
Tracing a Flow Chart



List	5	6	4	5

Total	Item	Average	Output
0	1		
5			
	2		
11			
	3		
15			
	4		

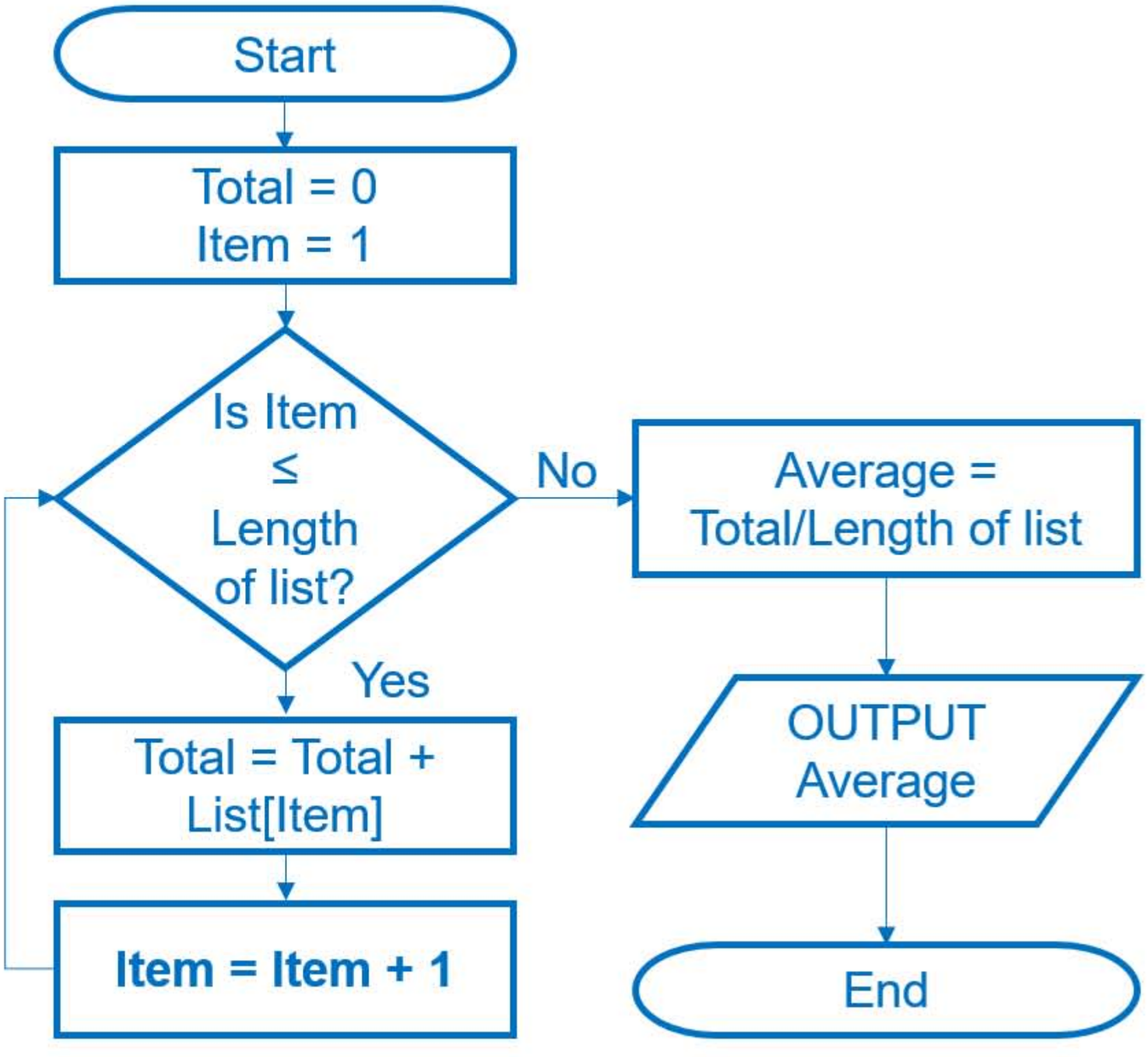
Tracing a Flow Chart



List	5	6	4	**5**

Total	Item	Average	Output
0	1		
5			
	2		
11			
	3		
15			
	4		
20			

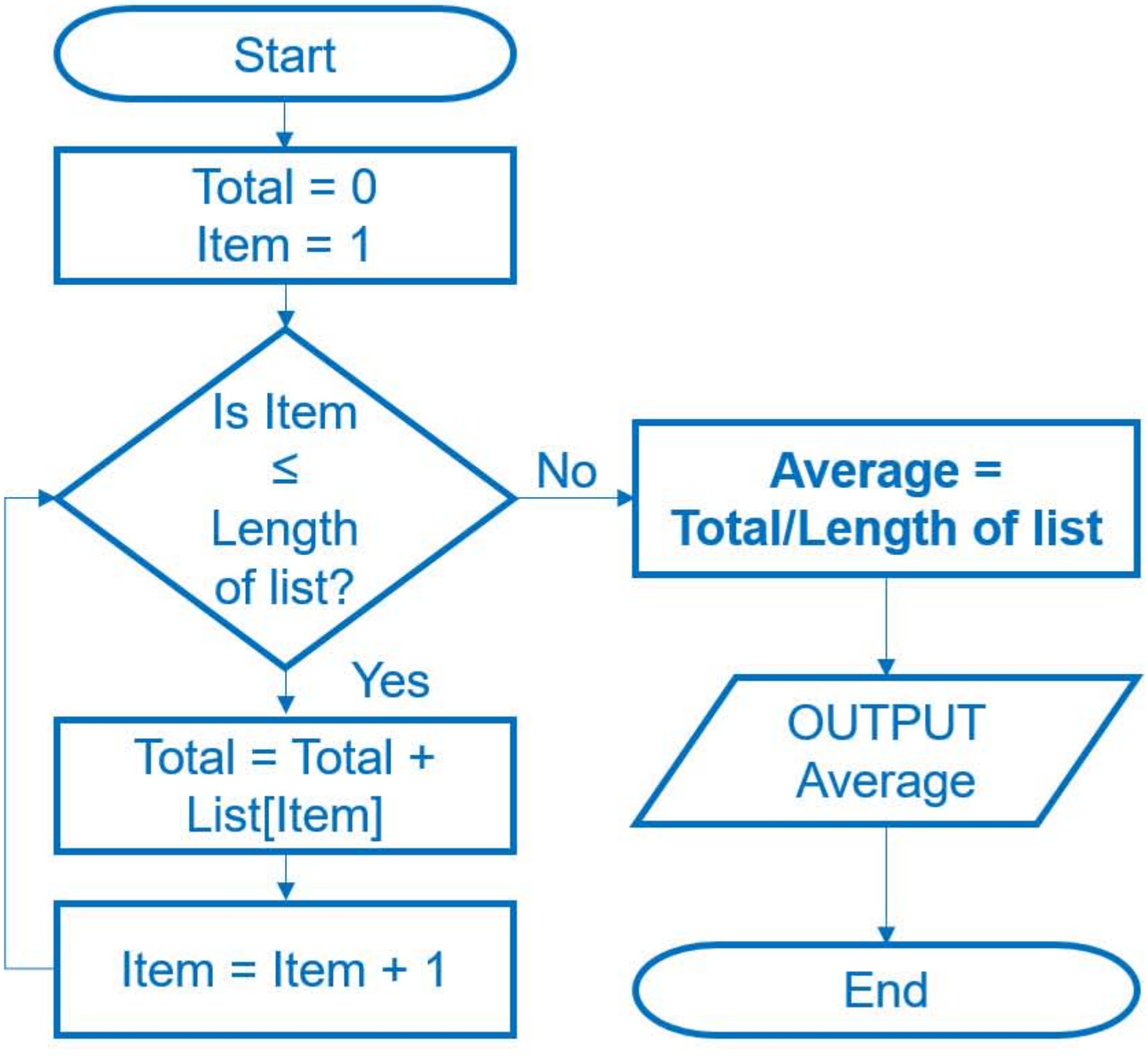
Tracing a Flow Chart



List	5	6	4	5

Total	Item	Average	Output
0	1		
5			
	2		
11			
	3		
15			
	4		
20			
	5		

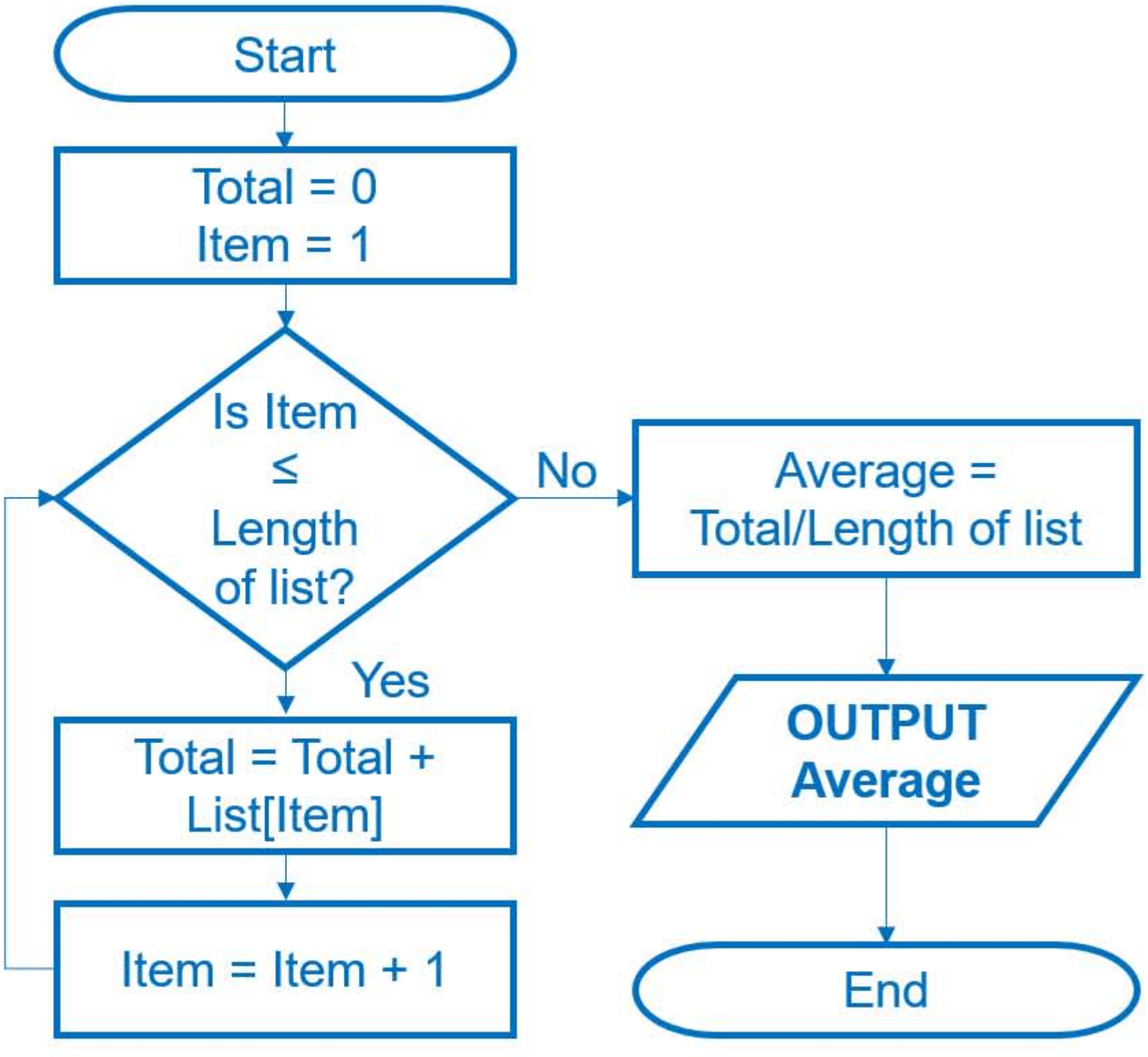
Tracing a Flow Chart



List	5	6	4	5

Total	Item	Average	Output
0	1		
5			
	2		
11			
	3		
15			
	4		
20			
	5		
		5	

Tracing a Flow Chart



List	5	6	4	5

Total	Item	Average	Output
0	1		
5			
	2		
11			
	3		
15			
	4		
20			
	5		
		5	
			5

Arrays

When tracing algorithms that iterate through an array, you may be expected to show the current value of each element in the array.

If multiple variables are declared or changed in one block of code, their values can be placed in the same row.

```
Scores ← [34,76,21,93]
i ← 0
Count ← 0
Len ← LEN(Scores)
WHILE i < Len
    IF Scores[i] ≥ 75 THEN
        Count ← Count + 1
    ENDIF
    i ← i + 1
ENDWHILE
```

i	Count	Len	Score[i]
0	0	4	
			34
1			
			76
	1		
2			
			21
3			
			93
	2		
4			