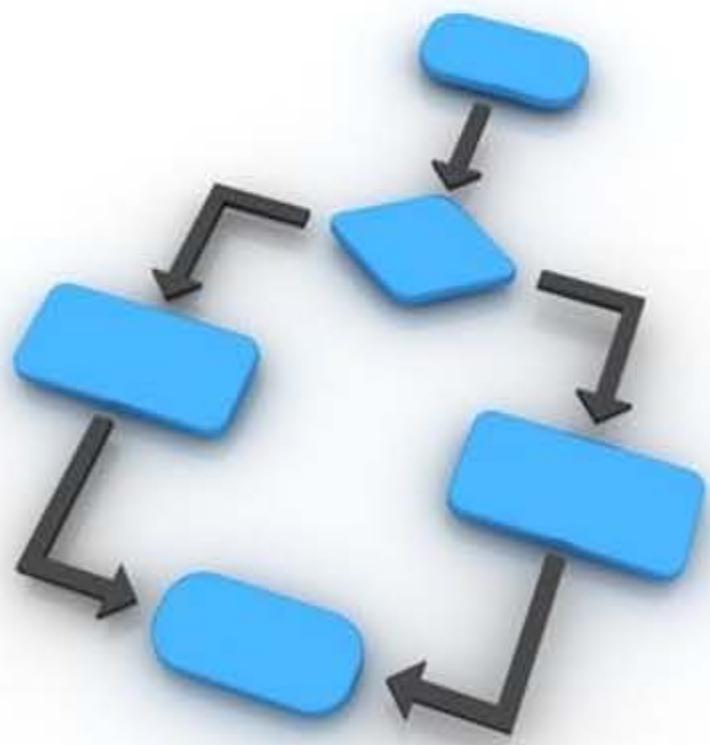


Designing Solutions



Flow Charts

Flow charts are used to represent algorithms visually, in the form of a diagram.

A number of standard symbols are used in flow charts:

Start/Stop

Used to indicate the start or end of an algorithm.

Process

Used to indicate a process; for example, performing a calculation.

Input/Output

Used when data needs to be inputted or outputted.

Decision

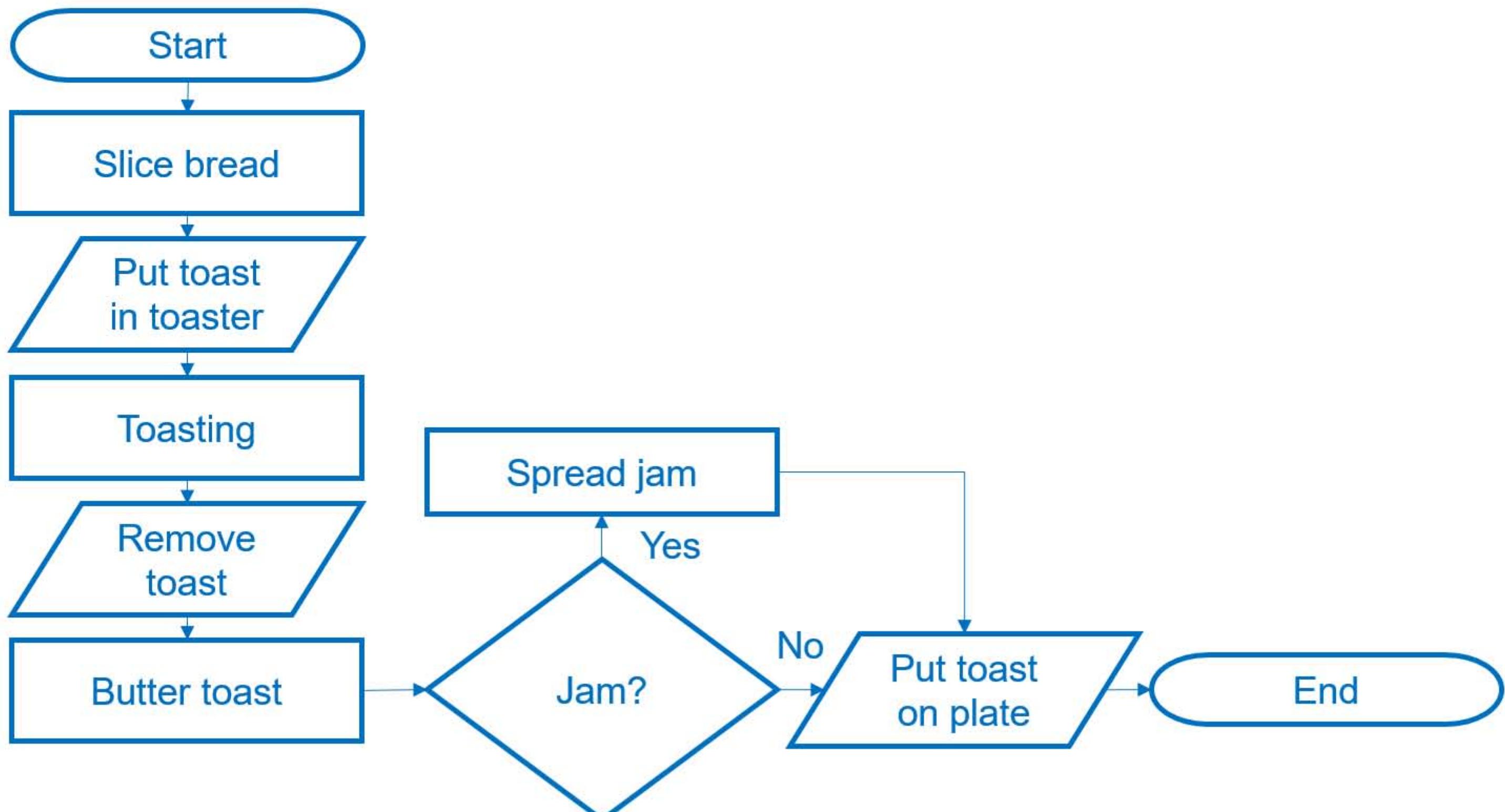
Used to control the path taken through an algorithm based on the result of a condition.

Subroutine

Used to call a predefined algorithm.

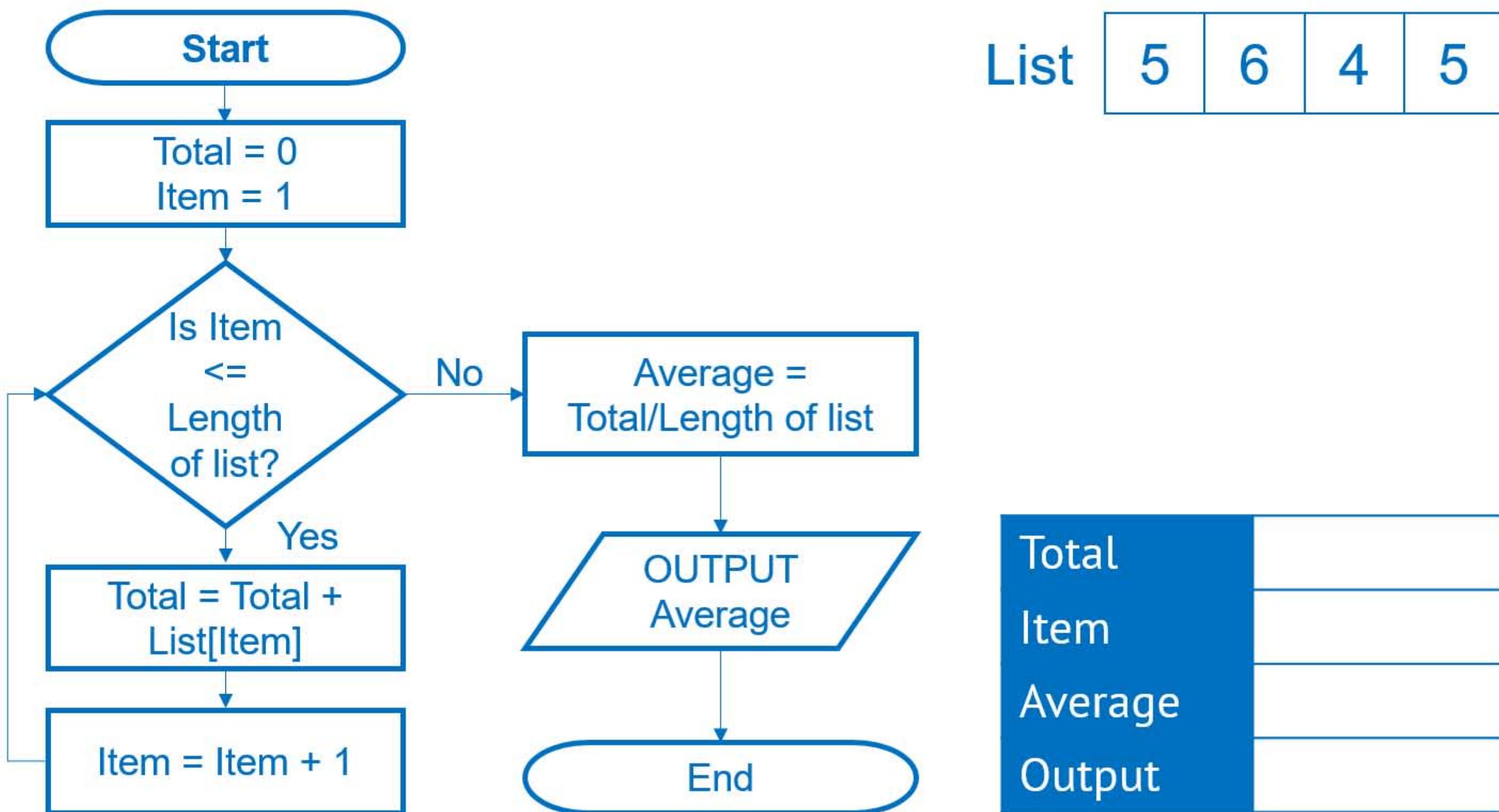
Making Toast Flow Chart

Here is the toast algorithm represented in the form of a flow chart:



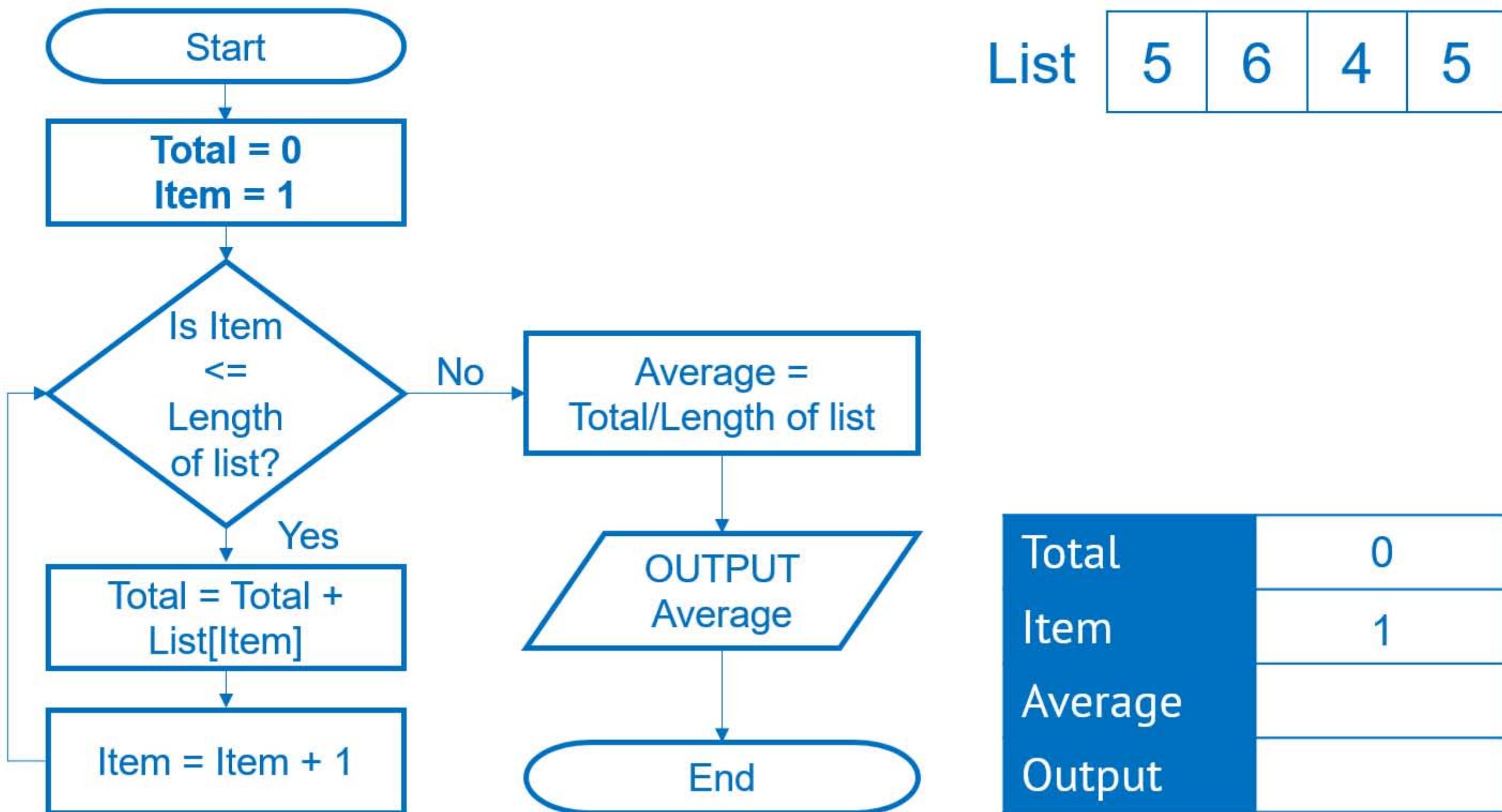
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



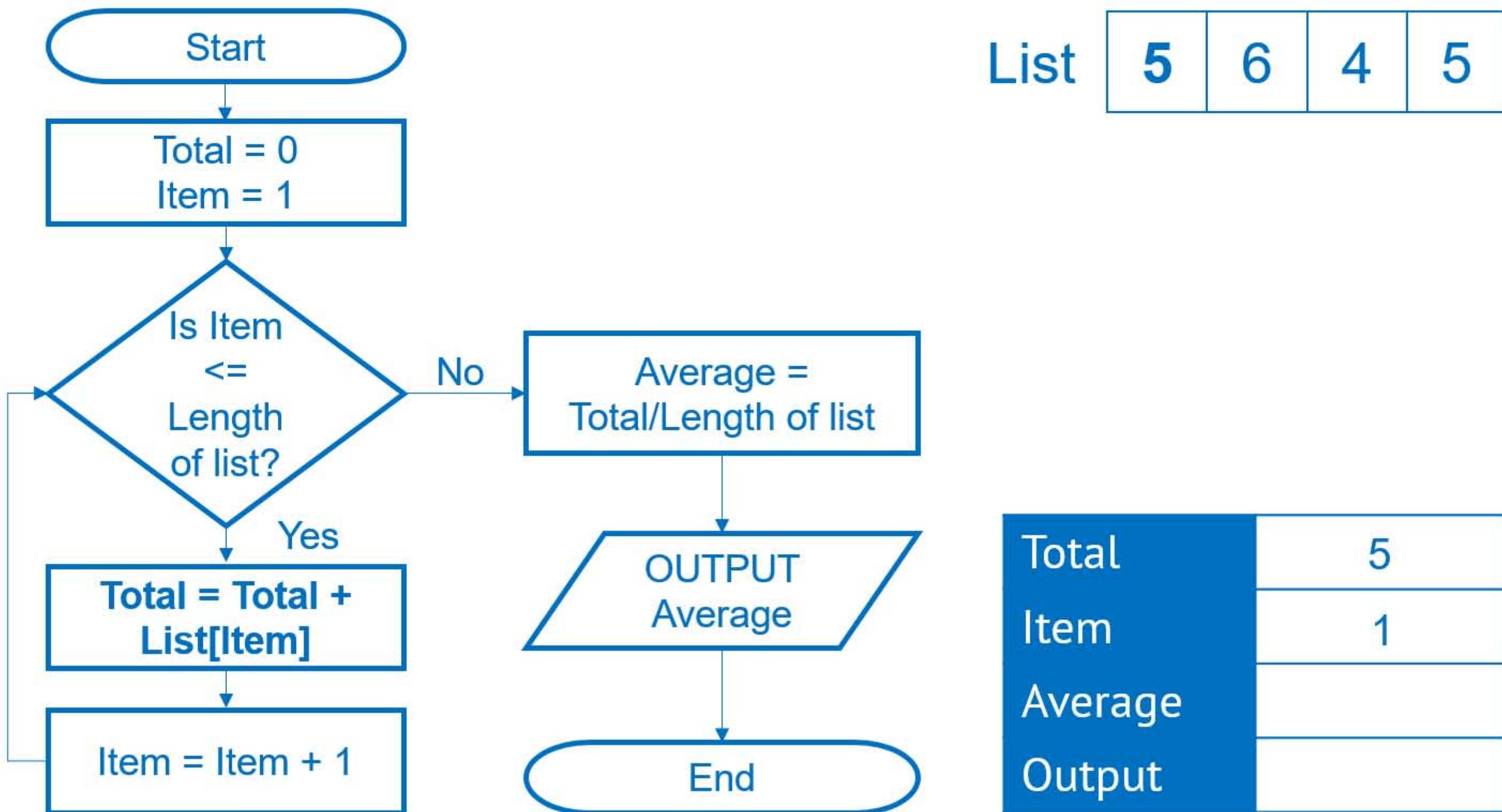
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



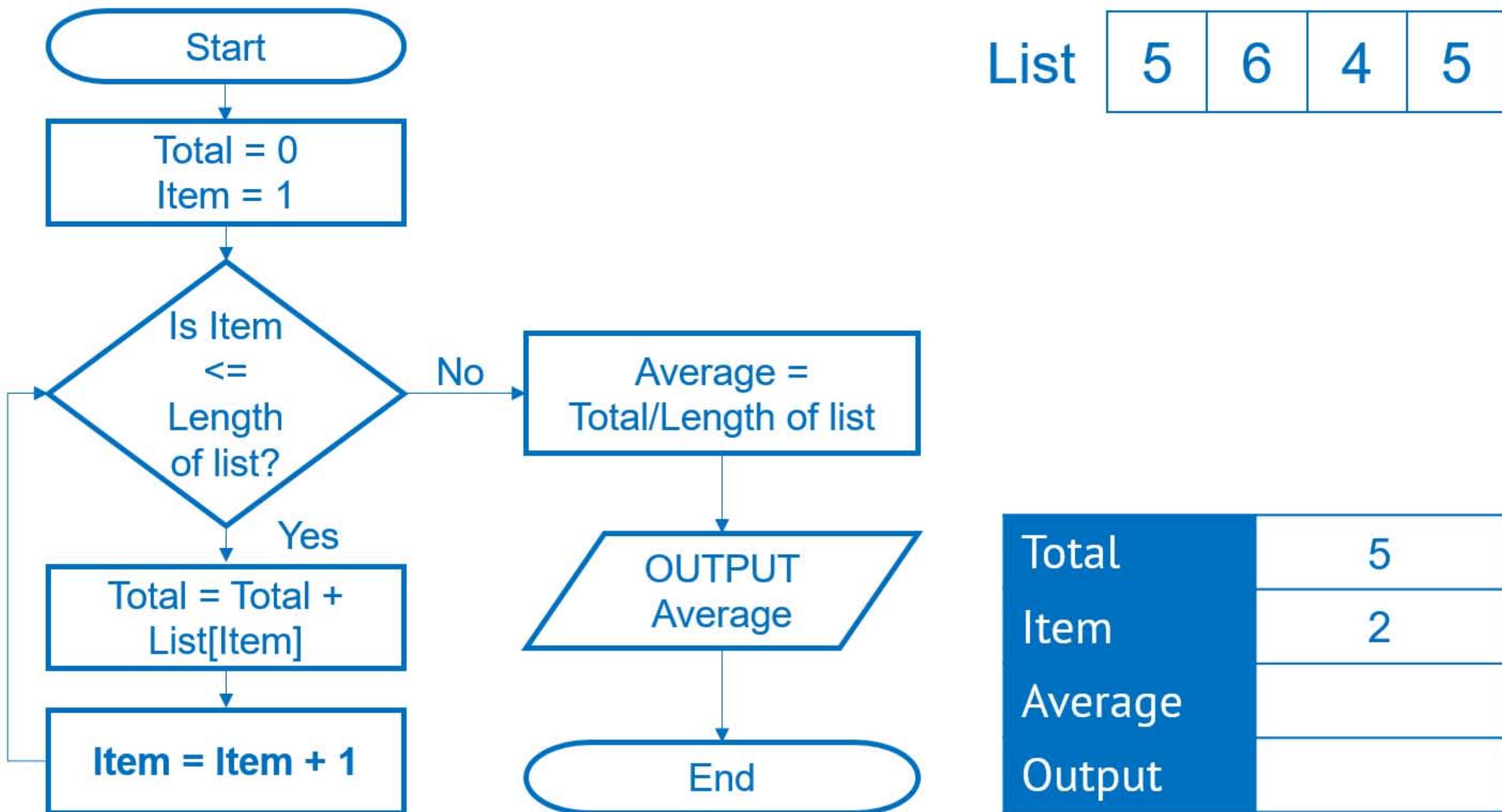
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



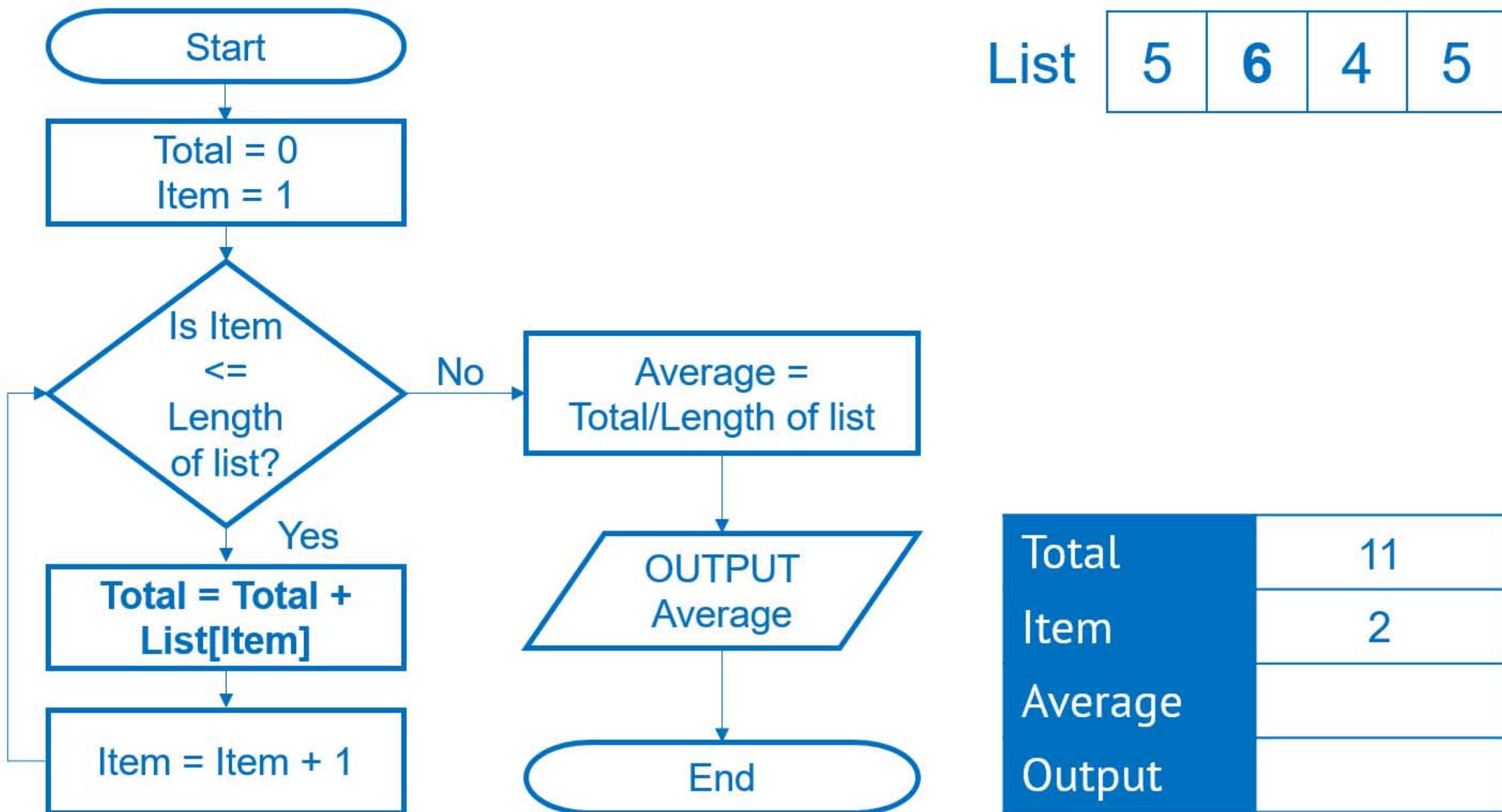
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



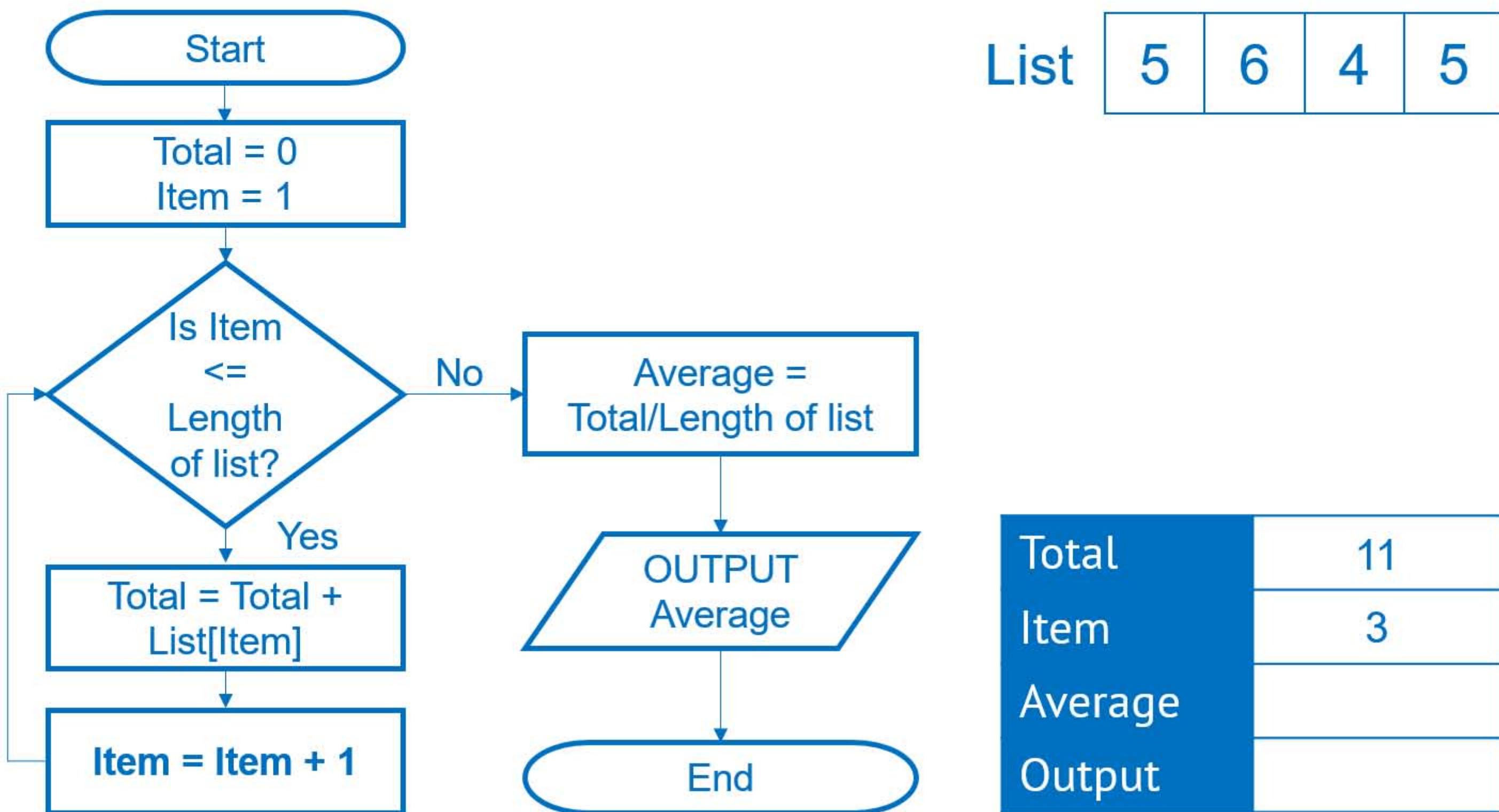
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



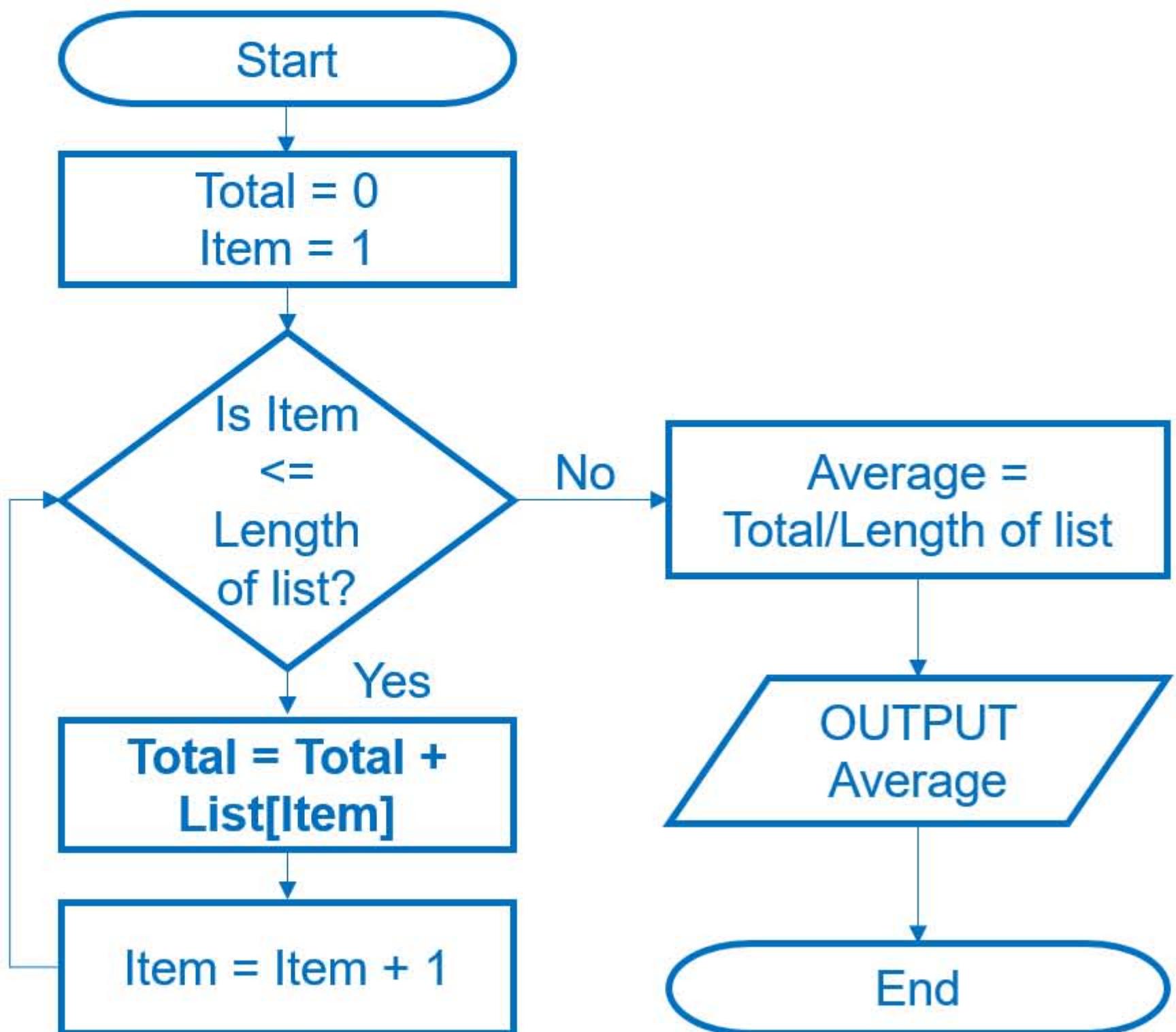
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



Example Flow Chart

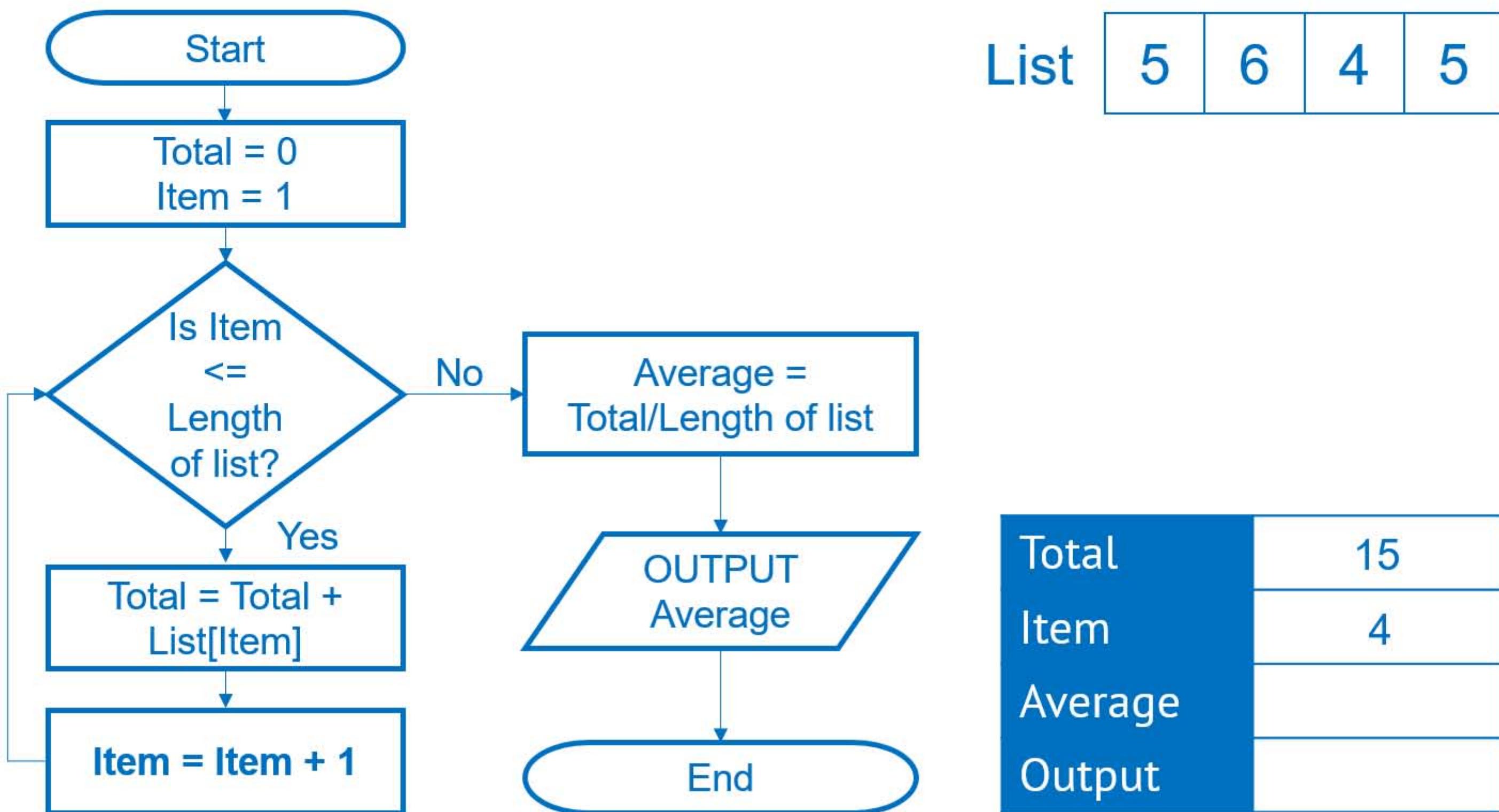
This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



Total	15
Item	3
Average	
Output	

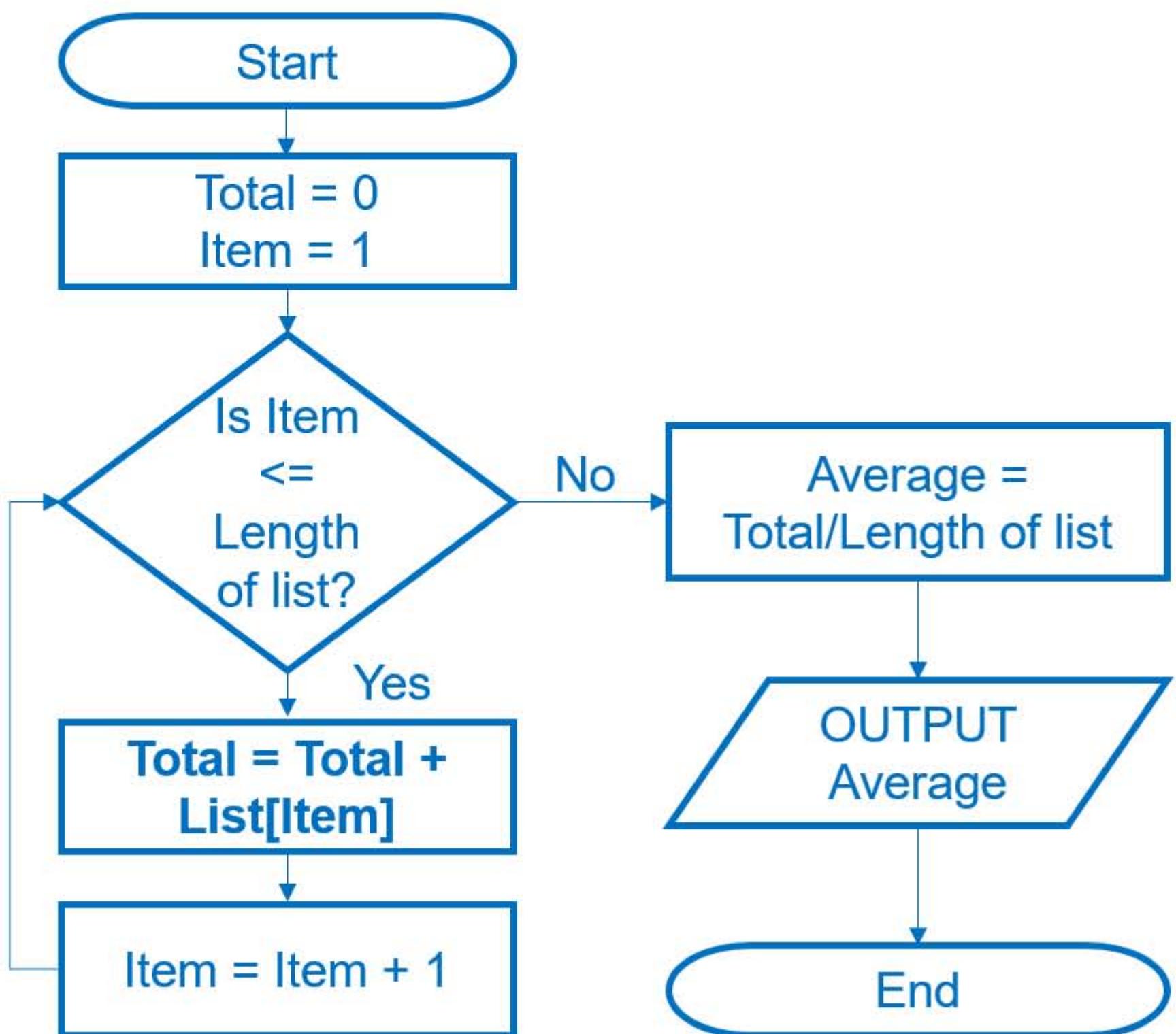
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



Example Flow Chart

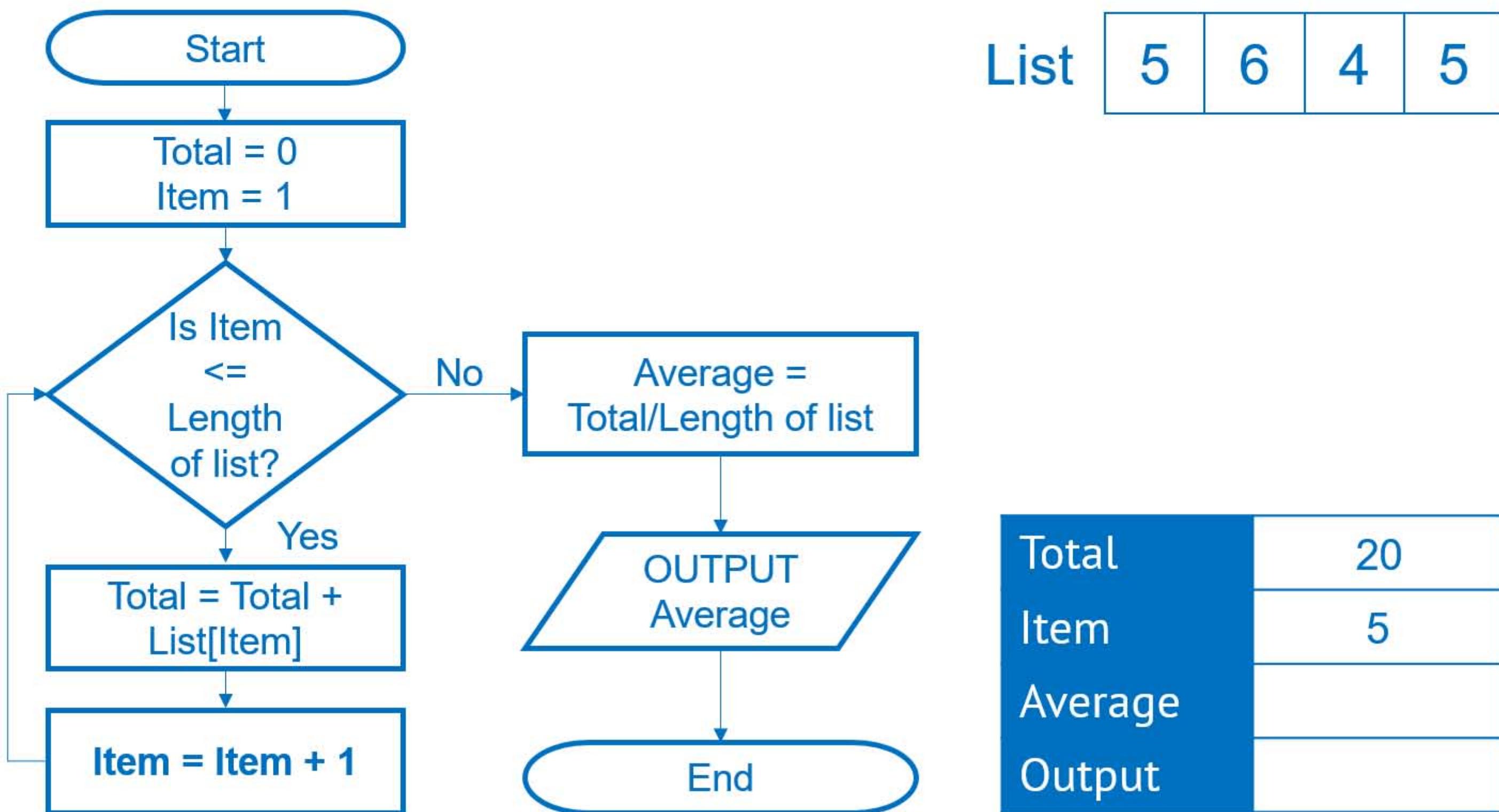
This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



Total	20
Item	4
Average	
Output	

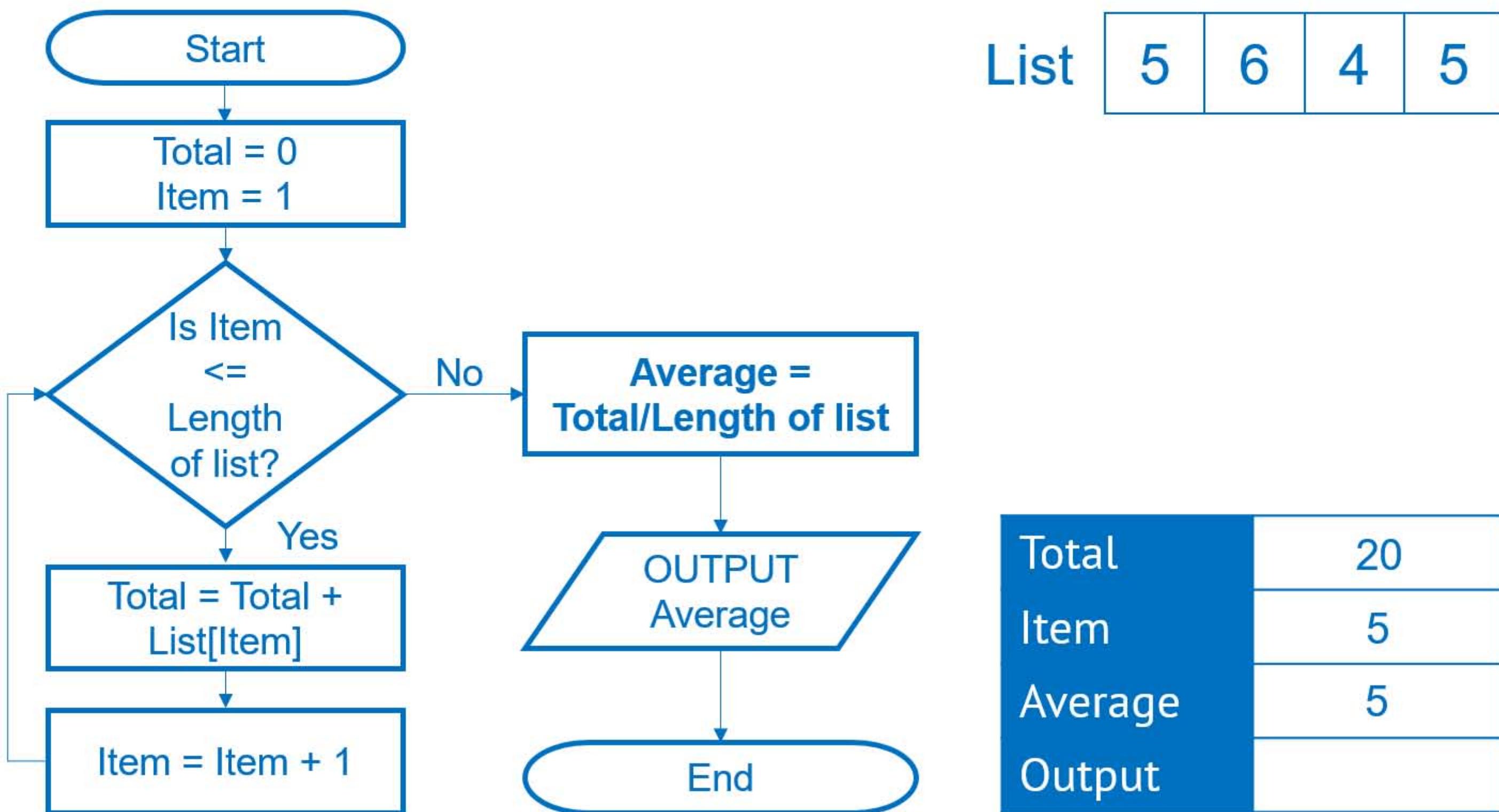
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



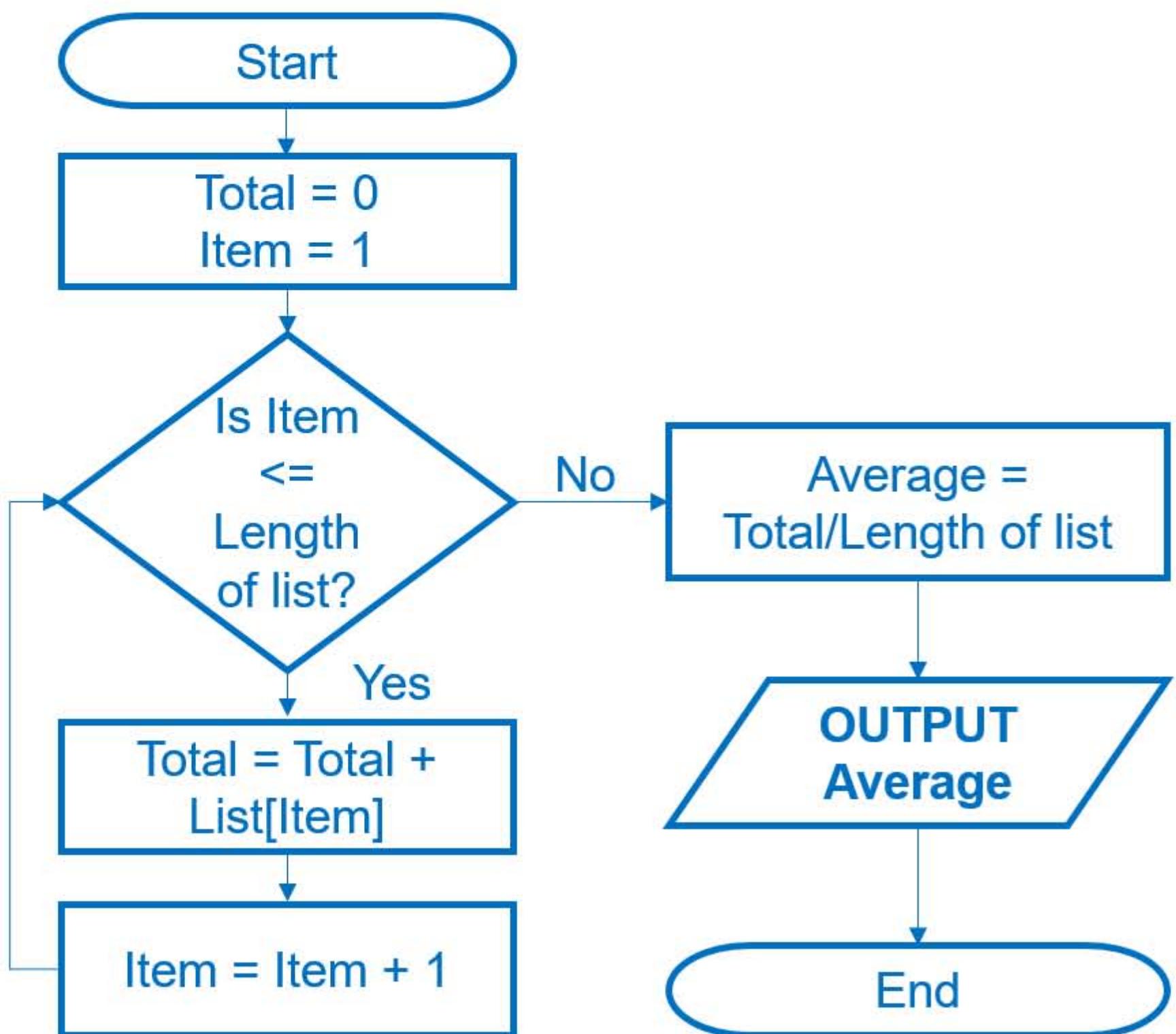
Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



Example Flow Chart

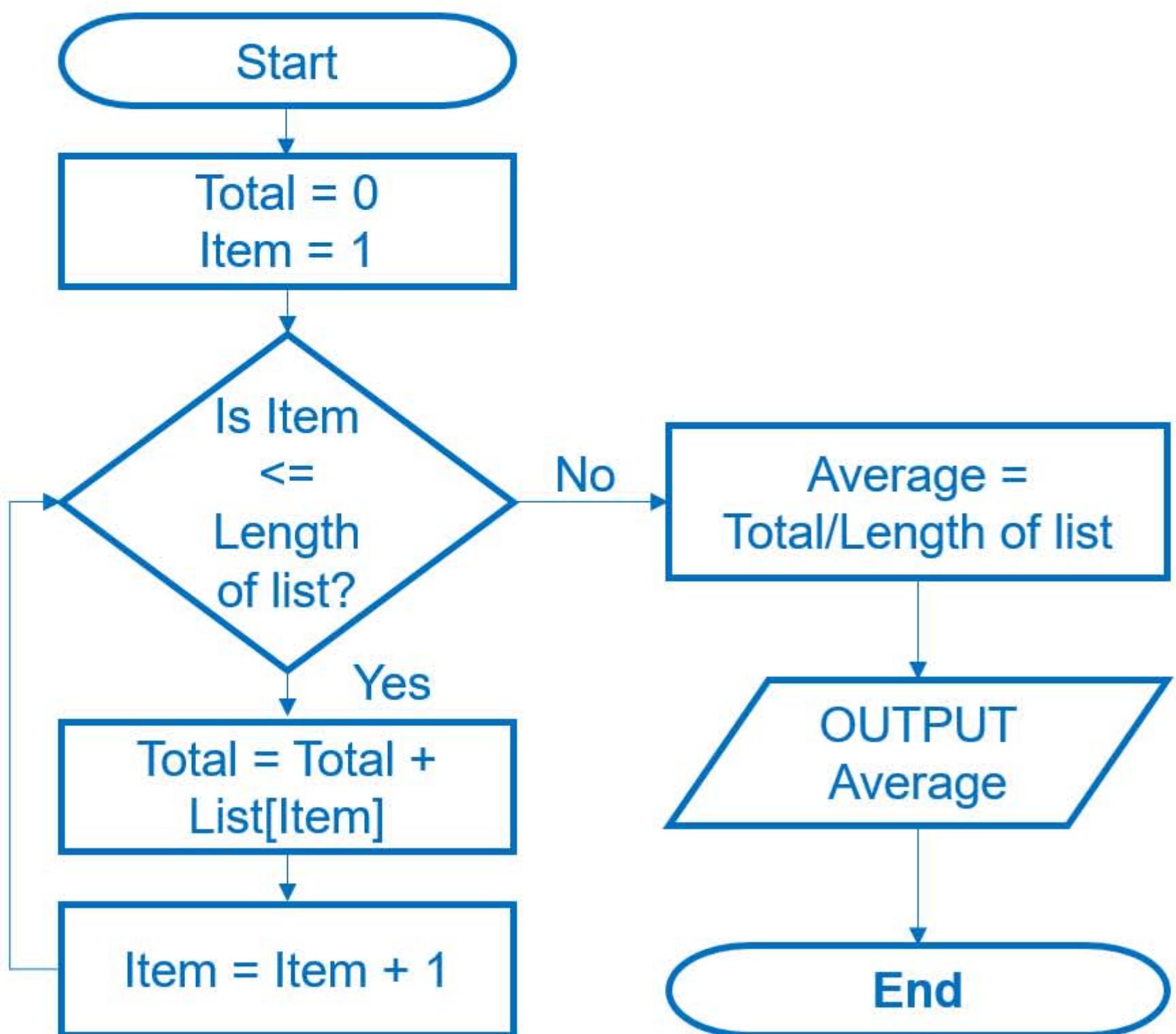
This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



Total	20
Item	5
Average	5
Output	5

Example Flow Chart

This is an example algorithm represented using a flow chart, it is designed to calculate the average of a list of values.



Total	20
Item	5
Average	5
Output	5

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	0
Item	
ListLength	
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	0
Item	1
ListLength	
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	0
Item	1
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	5
Item	1
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	5
Item	2
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	11
Item	2
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	11
Item	3
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	15
Item	3
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	15
Item	4
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```

Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
    
```

List	5	6	4	5
------	---	---	---	---

Total	20
Item	4
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	20
Item	5
ListLength	4
Average	
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	20
Item	5
ListLength	4
Average	5
Output	

Pseudocode

Pseudocode is a way of designing an algorithm using a syntax that looks similar to a real programming language.

There are no specific rules to writing pseudocode; the important thing is to be consistent.

```
Total ← 0
Item ← 1
ListLength ← List.length
WHILE Item <= ListLength
    Total ← Total + List[Item]
    Item ← Item + 1
ENDWHILE
Average ← Total/ListLength
PRINT Average
```

List	5	6	4	5
------	---	---	---	---

Total	20
Item	5
ListLength	4
Average	5
Output	5

Another Example

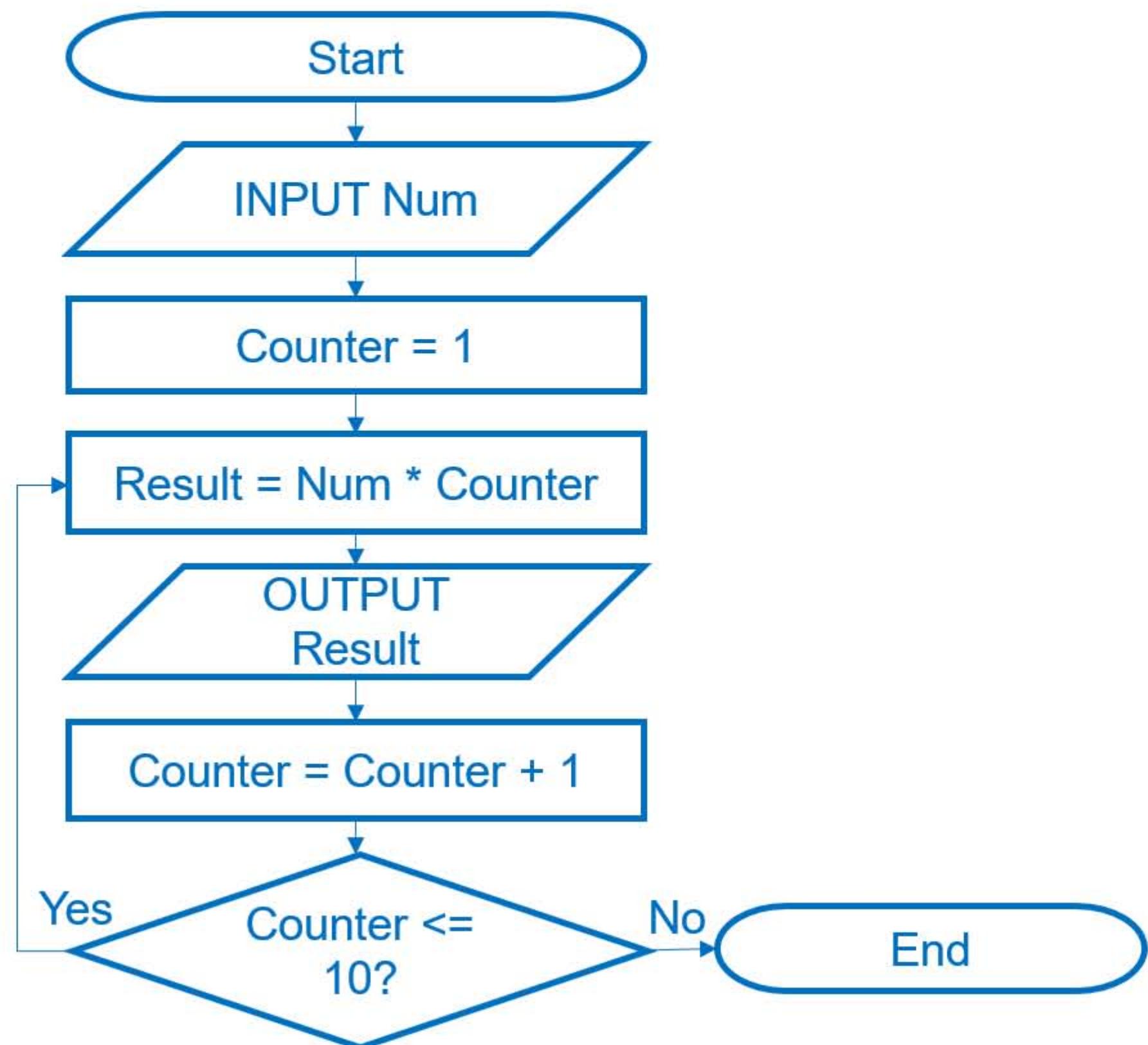
Here is another example of a flow chart program design:

Ask the user to input a number, create a counter and set it to 1.

Multiply the number by the counter and output the result.

Add 1 to the counter.

Repeat while the counter is less than or equal to 10.



Structure Diagrams

Structure diagrams are another way of designing software.

When creating a structure diagram we focus on the main tasks that need to be performed before breaking them down further if needed. This approach is known as top-down design.

