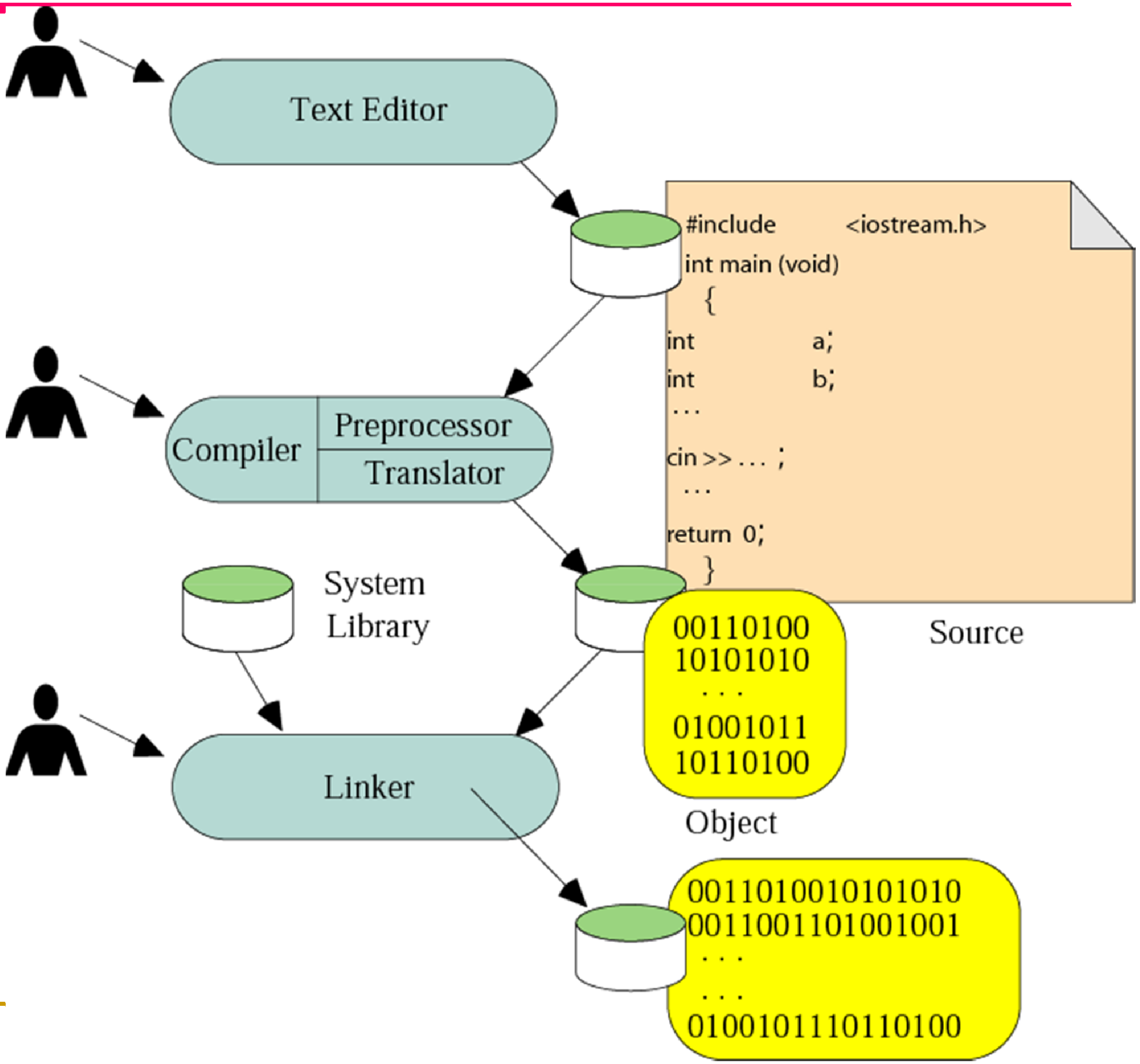
BUILDING A PROGRAM

Building a program

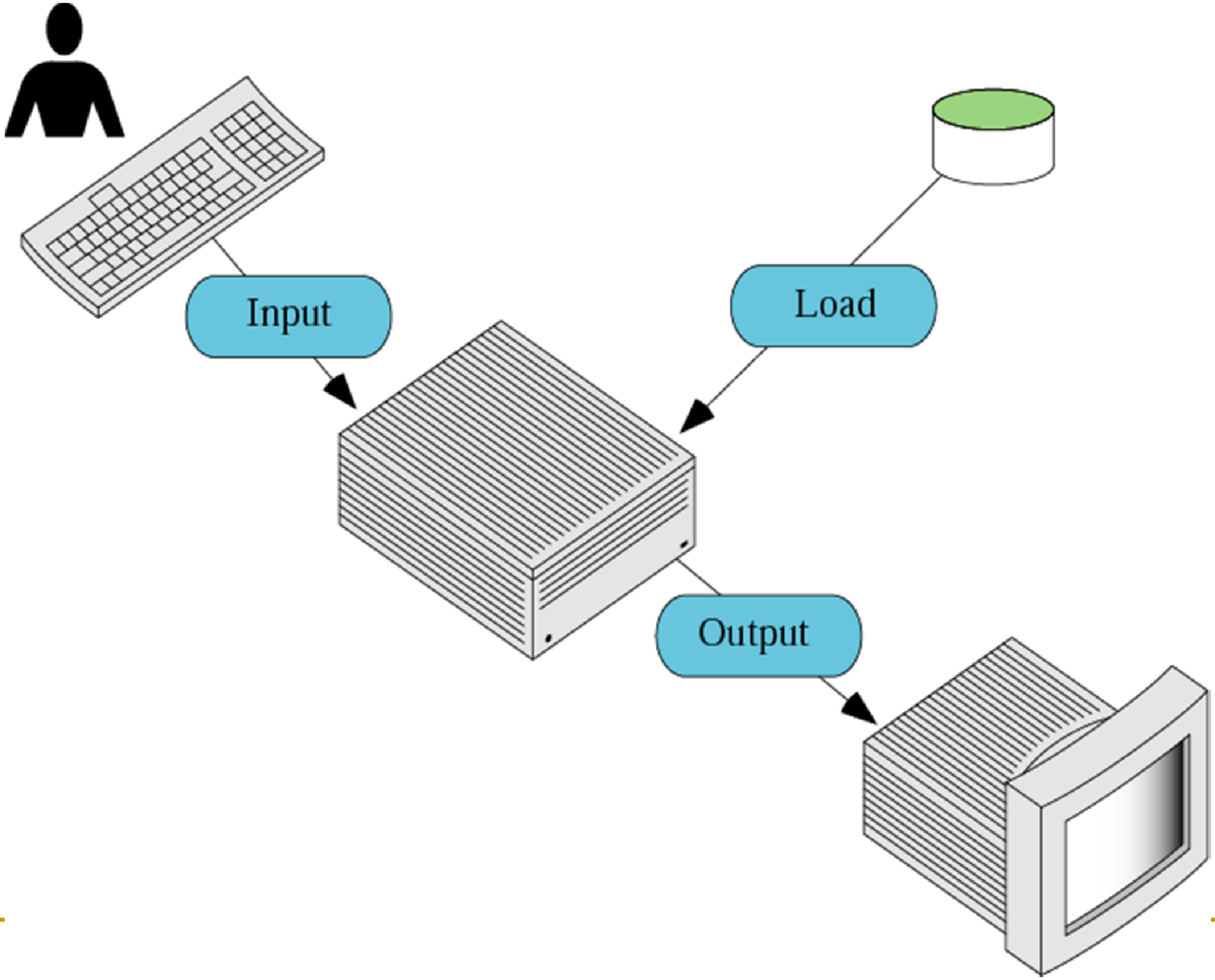
- The steps to building a program include writing, editing, compiling, and linking code.



```
#include        <iostream.h>
int main (void)
  {
int             a;
int             b;
 . . .
cin >> . . . ;
  . . .
return 0;
  }
```

PROGRAM EXECUTION

Program execution



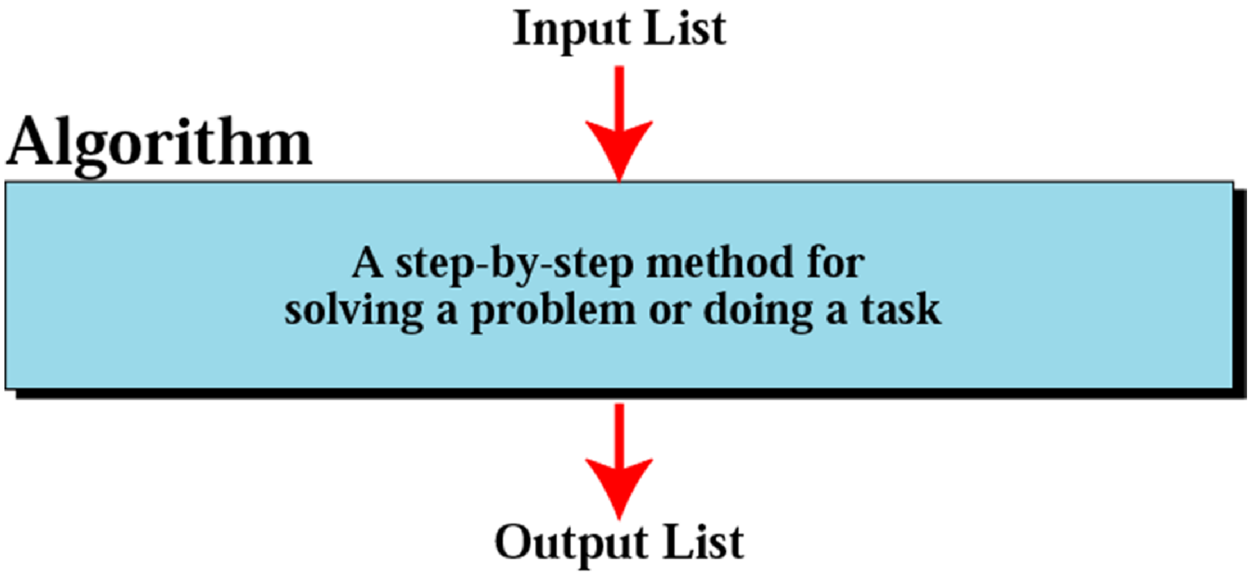
Algorithms + Data Structures = Programs

- Niklaus Wirth, 1975.

CONCEPT

Informal definition

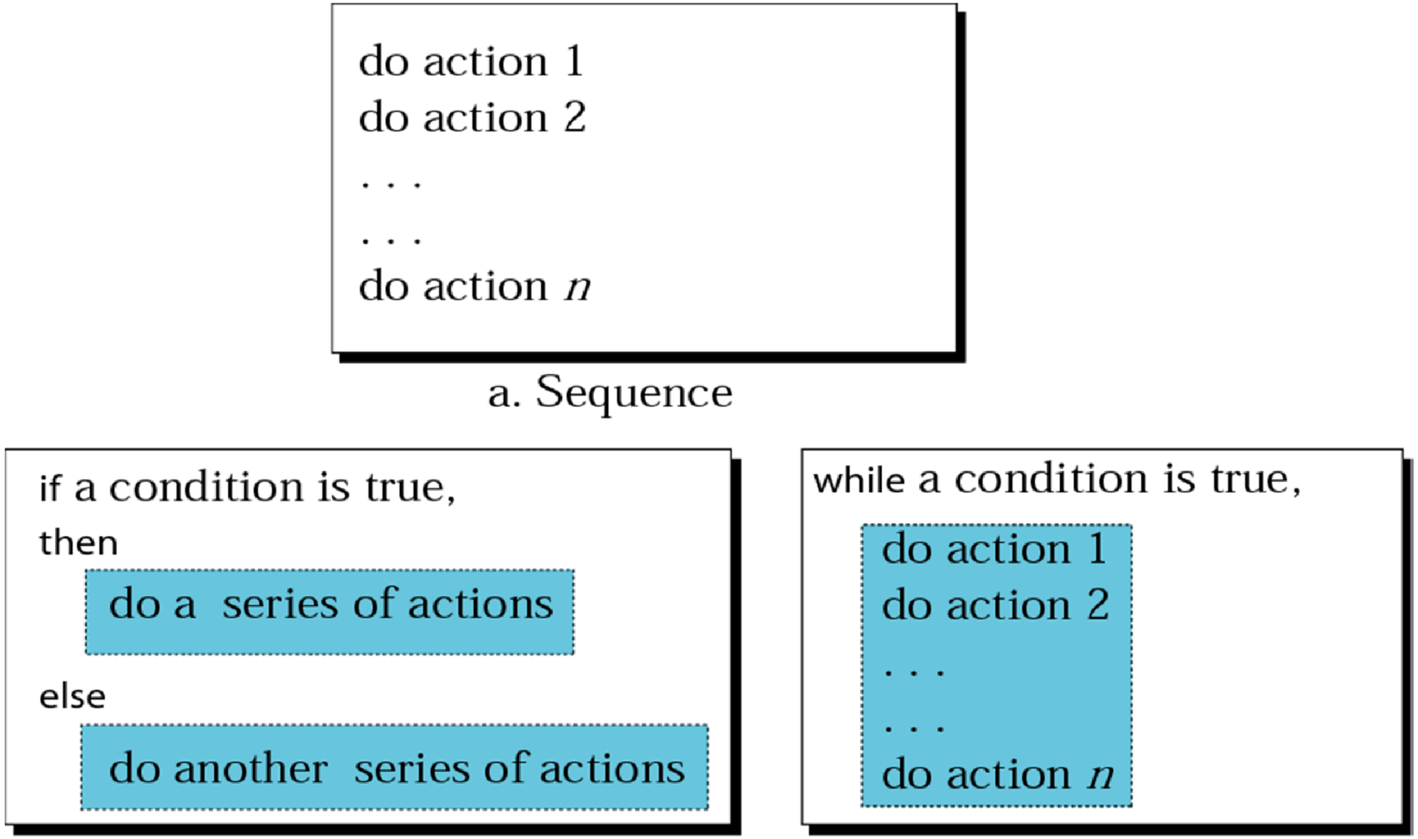
- Informally, an algorithm is a step-by-step method for solving a problem or doing a task.
- An algorithm accepts an input list of data and creates an output list of data.



THREE CONSTRUCTS

Three constructs

- A program is a combination of sequence constructs, decision constructs, and repetition constructs.



```
do action 1
do action 2
. . .
. . .
do action n
```

a. Sequence

```
if a condition is true,
then
    do a  series of actions
else
    do another  series of actions
```

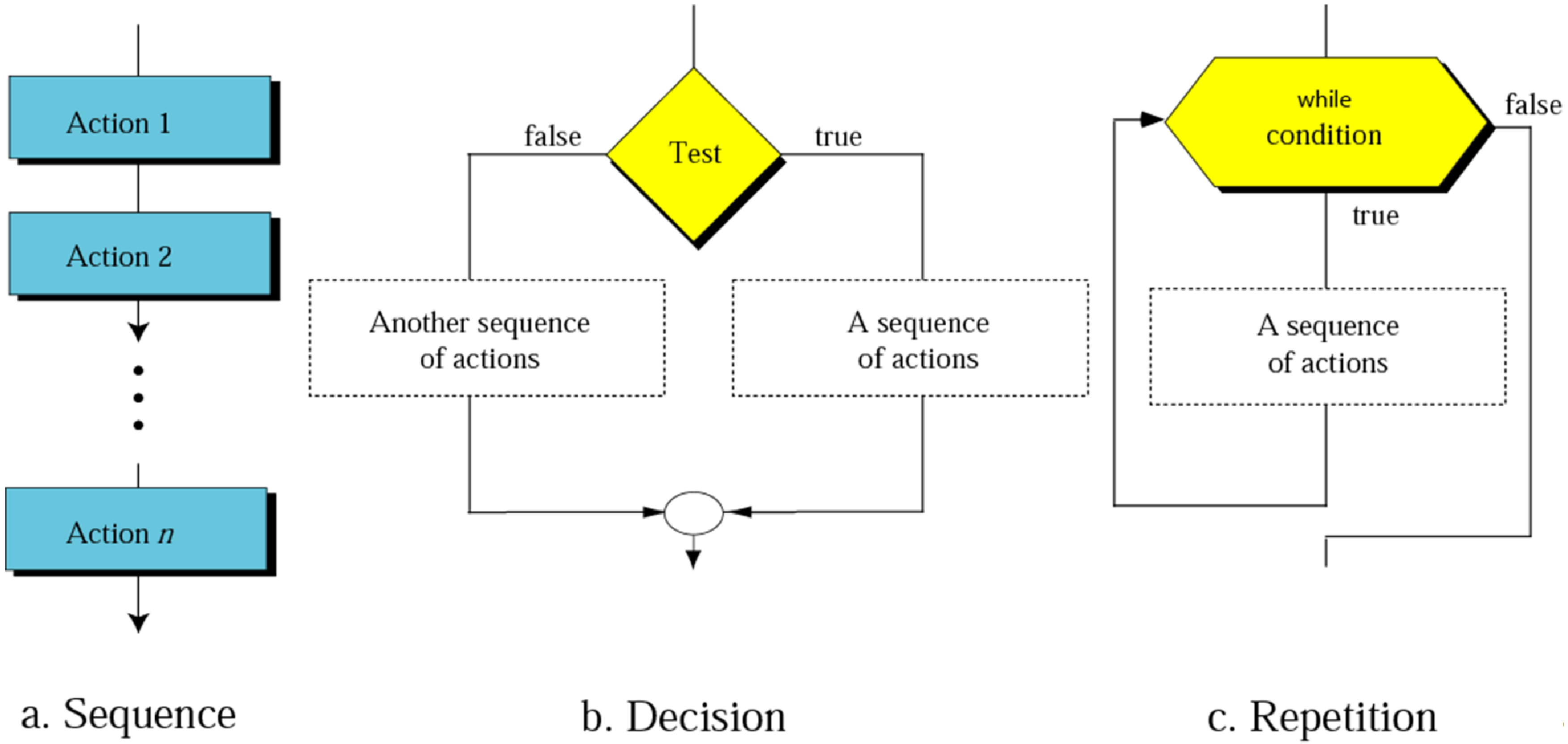
```
while a condition is true,
    do action 1
    do action 2
    . . .
    . . .
    do action n
```

b. Decision

c. Repetition

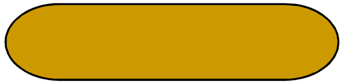
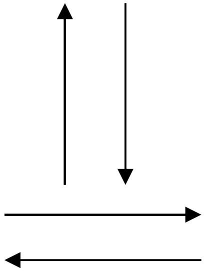
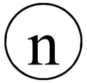
Flowcharts for three constructs

- A flowchart is a pictorial representation of an algorithm.

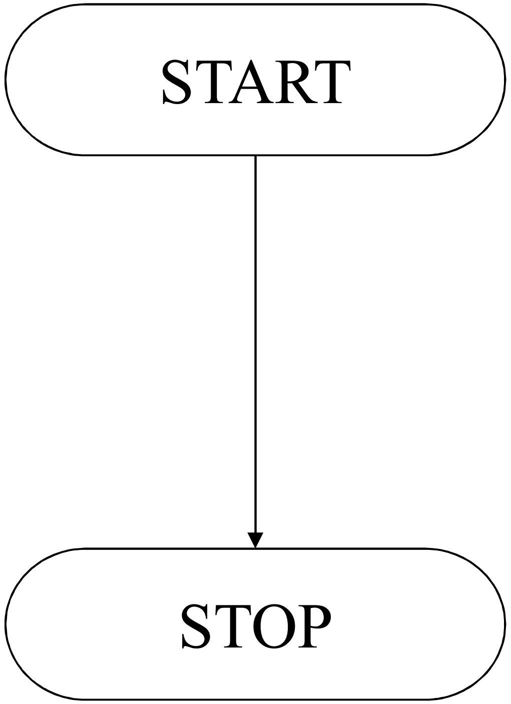


a. Sequence b. Decision c. Repetition

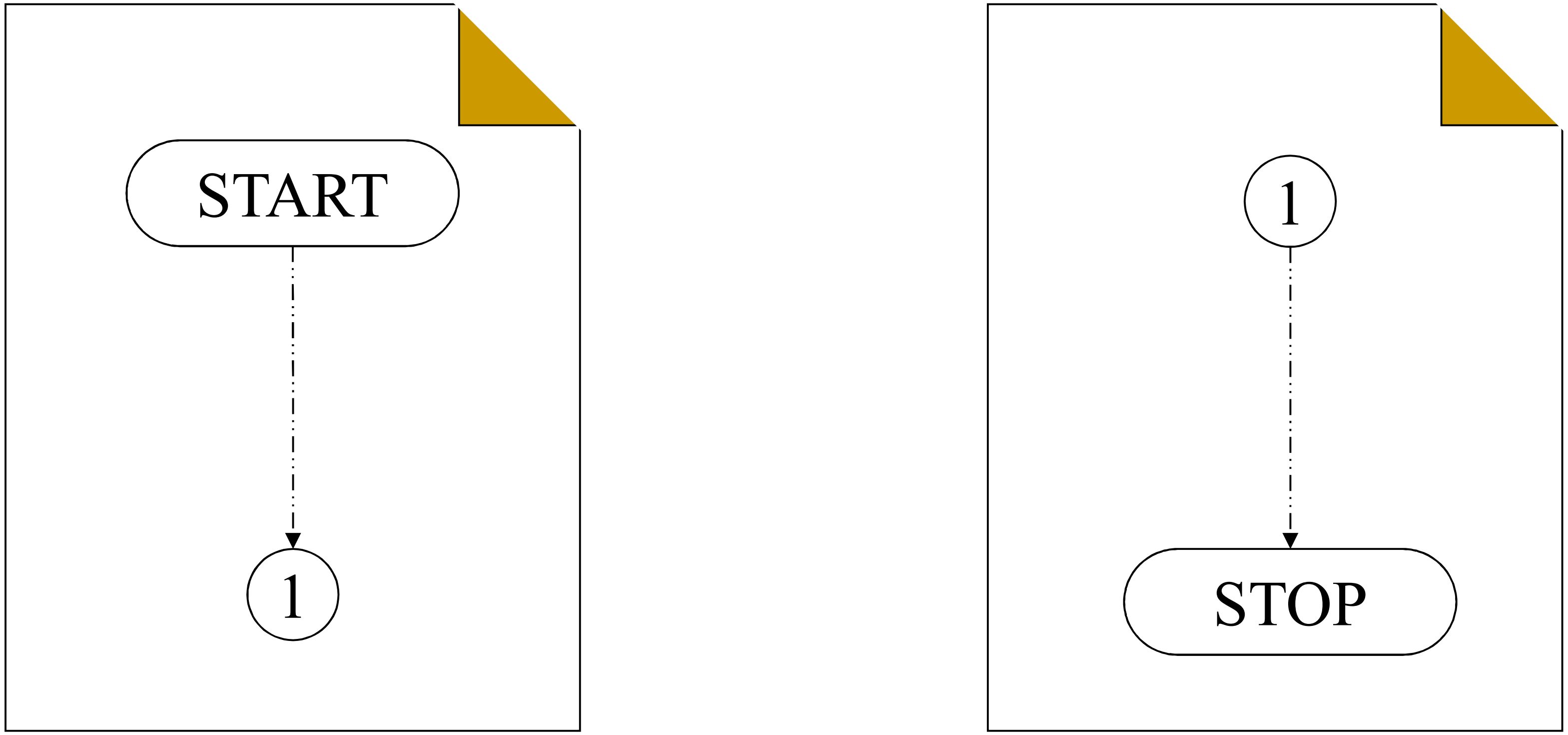
Appendix : Flowcharts

SYMBOL	NAME	APPLICATION
	Terminal	Shows the beginning or end of an algorithm
	Flow Lines	Show the action order in an algorithm
	Connector	Shows the continuity of the algorithm on the next page

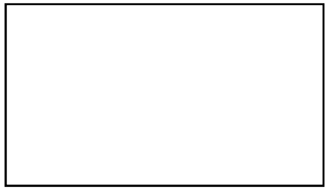
START and STOP



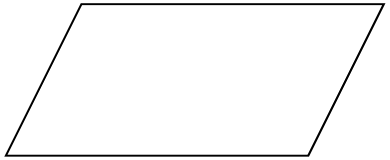
Connectors



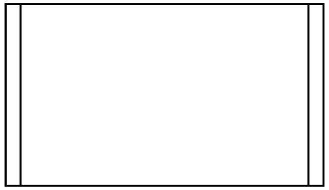
Sequence Symbols



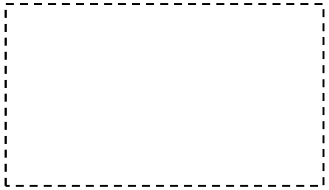
Assignment statement



Input/output statement



Module call

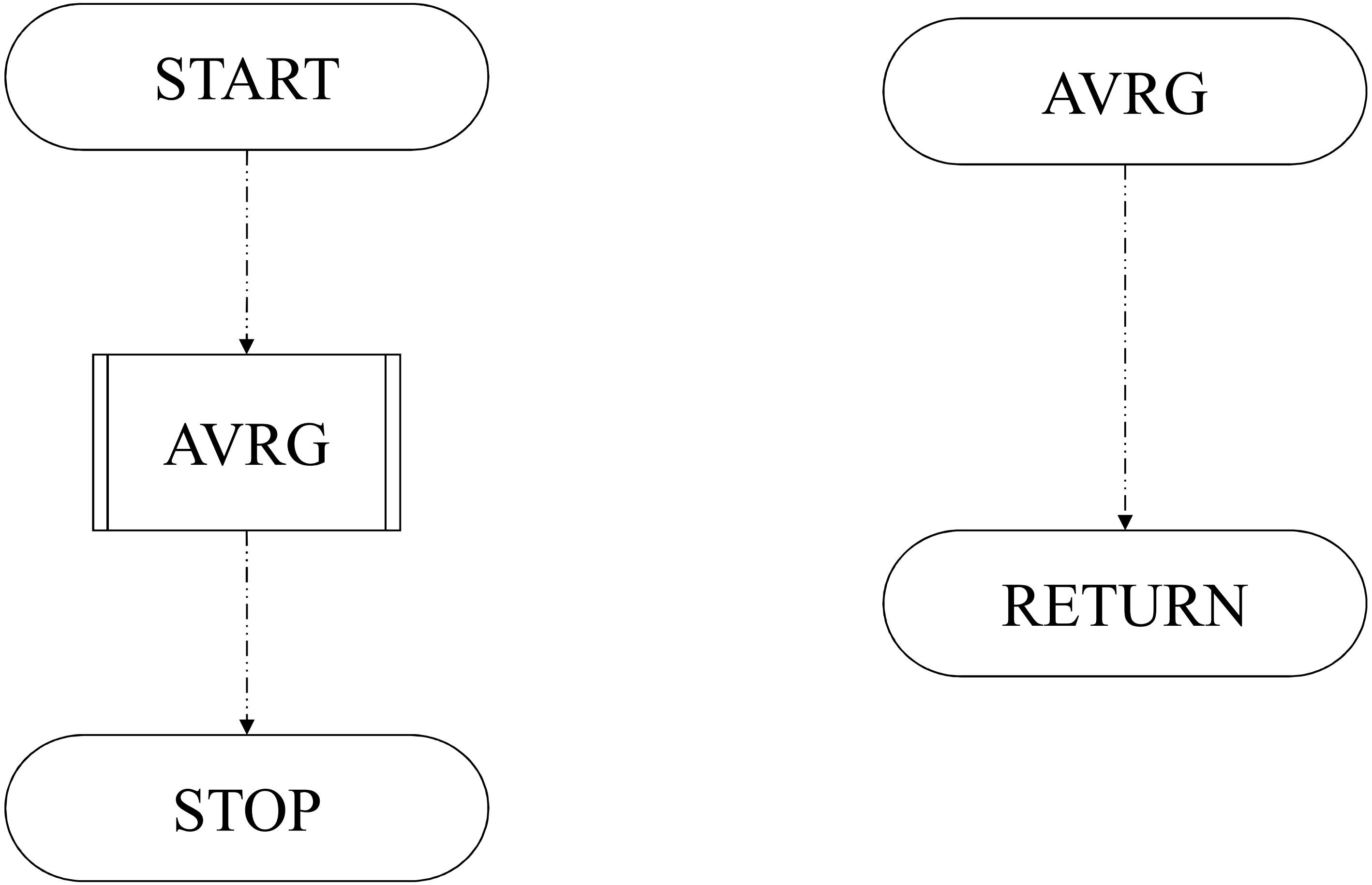


Compound statement

Assignment statement

variable ← expression

Module-Call Statement



START

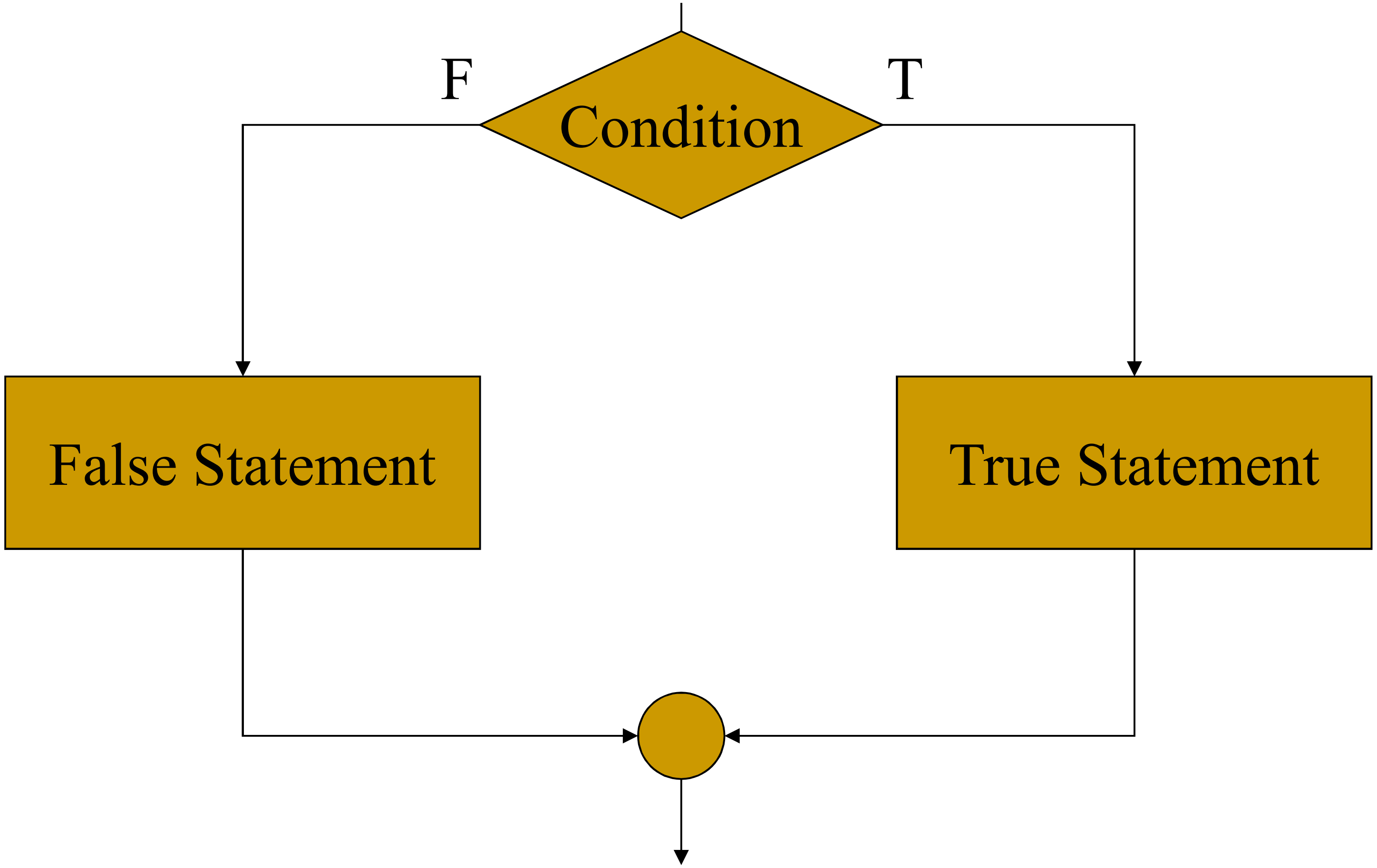
AVRG

STOP

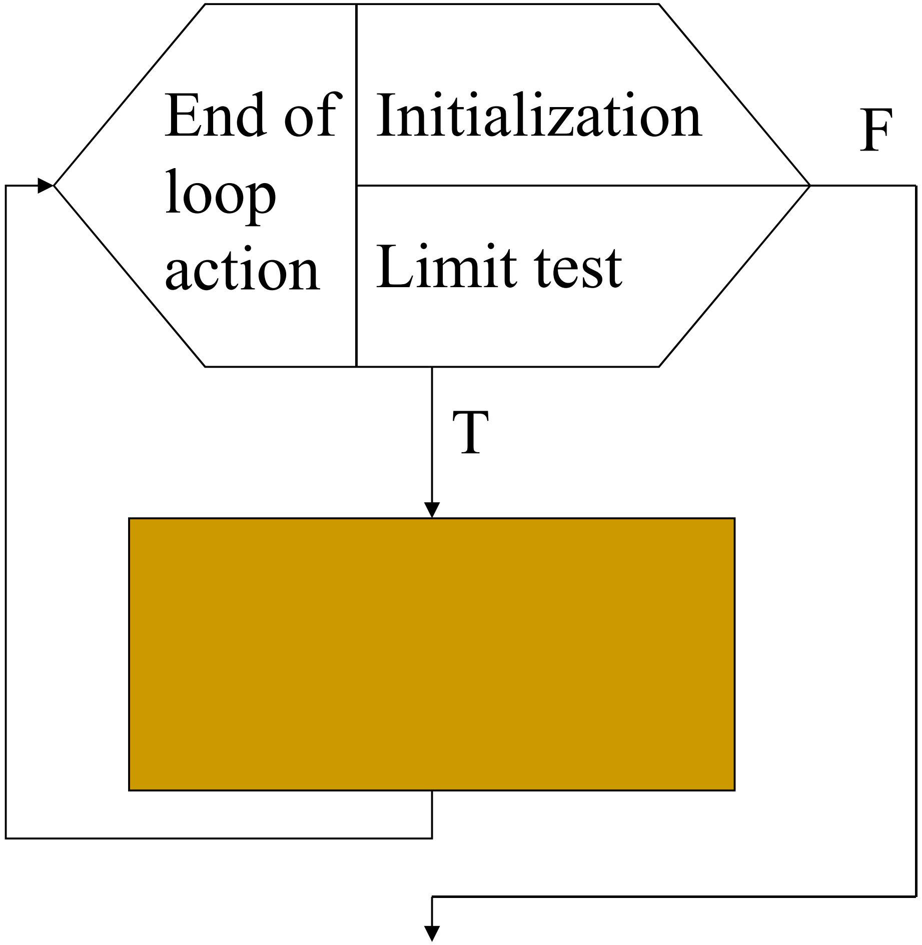
AVRG

RETURN

Two-Way Selection



for Loop

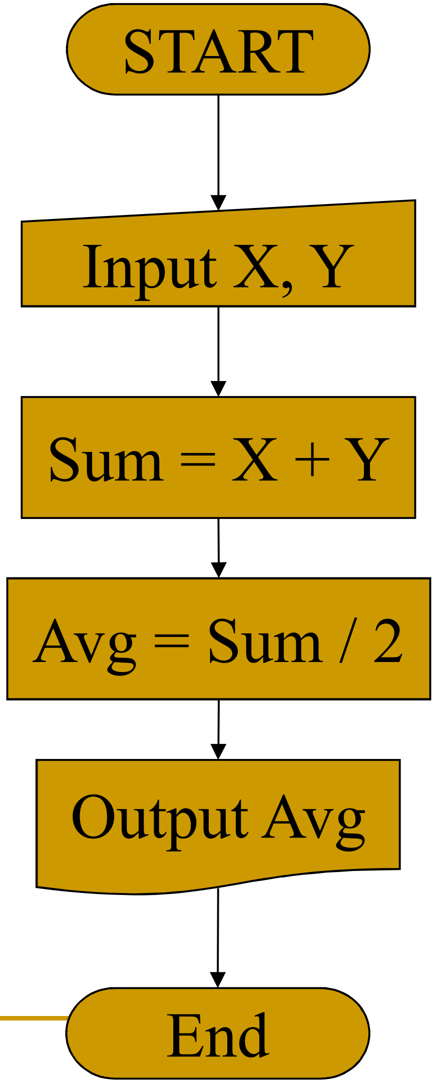


Example 1

Write an algorithm that finds the average of two numbers

Algorithm 1: Average of two

```
    AverageOfTwo
    Input: Two numbers
1.  Add the two numbers
2.  Divide the result by 2
3.  Return the result of Step 2
    End
```



START

Input X, Y

Sum = X + Y

Avg = Sum / 2

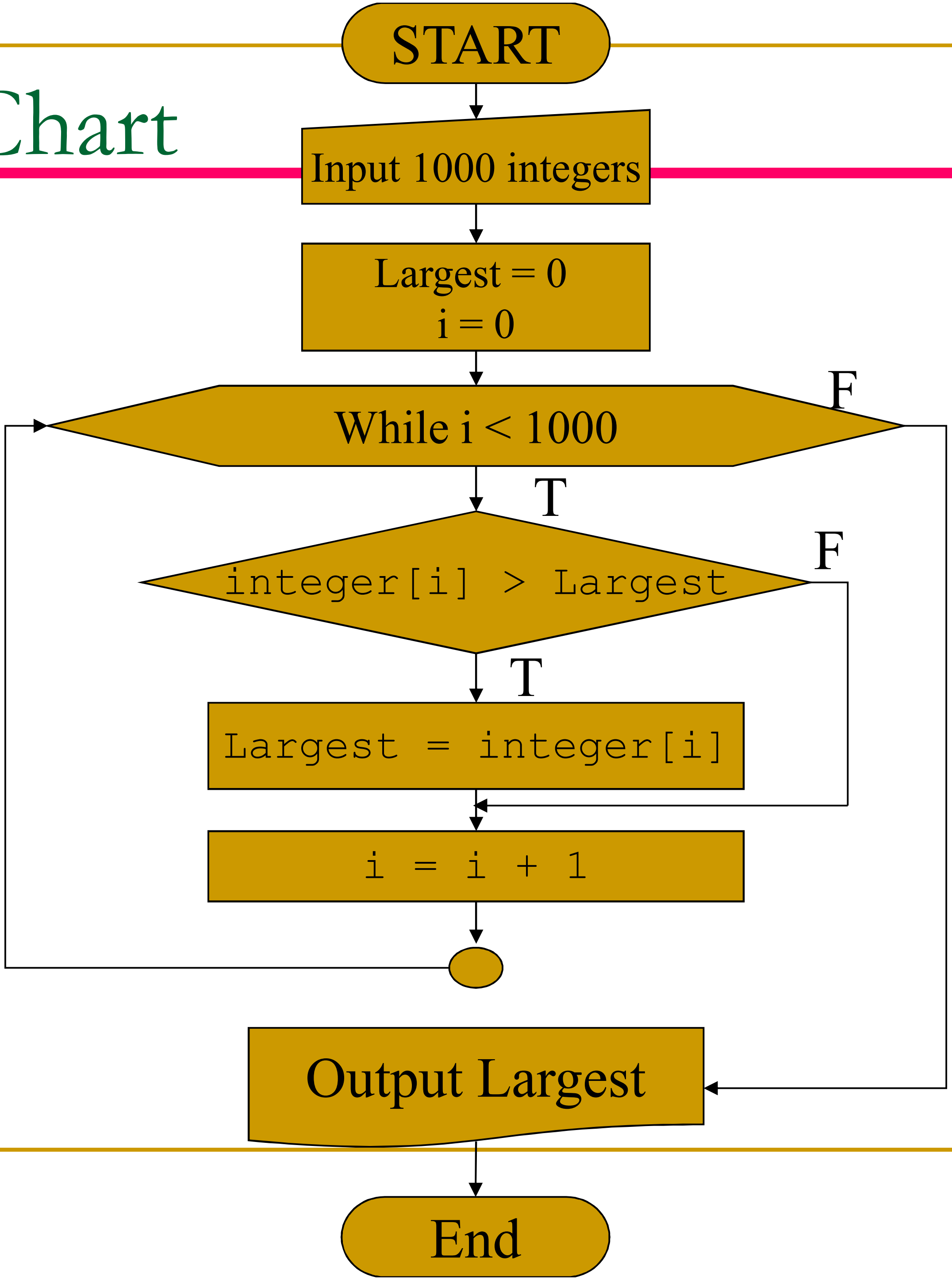
Output Avg

End

Example 2

Write an algorithm to find the largest of 1000 numbers.

Flow Chart

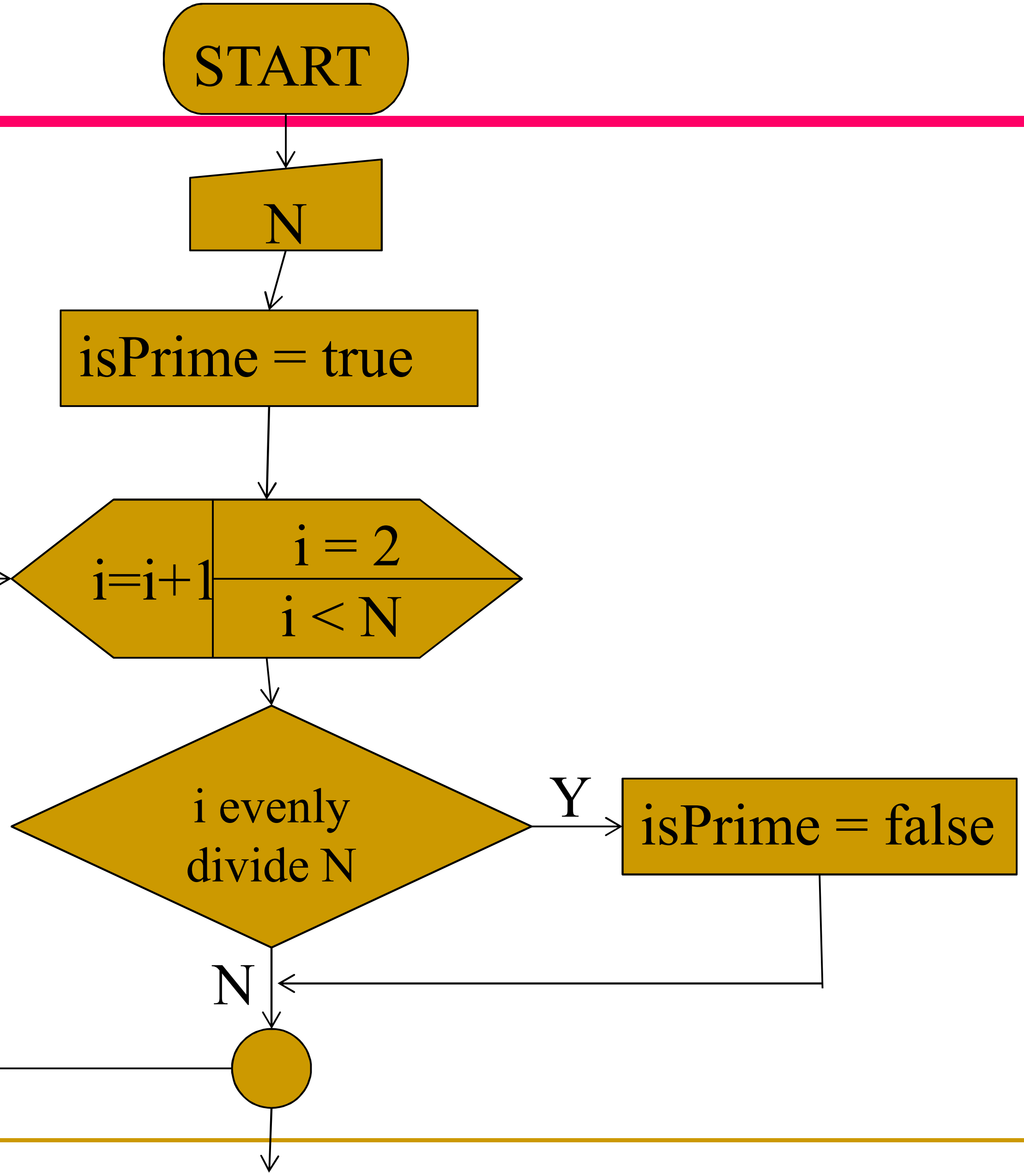

START
Input 1000 integers
Largest = 0
i = 0
While i < 1000
F
T
integer[i] > Largest
F
T
Largest = integer[i]
i = i + 1
Output Largest
End

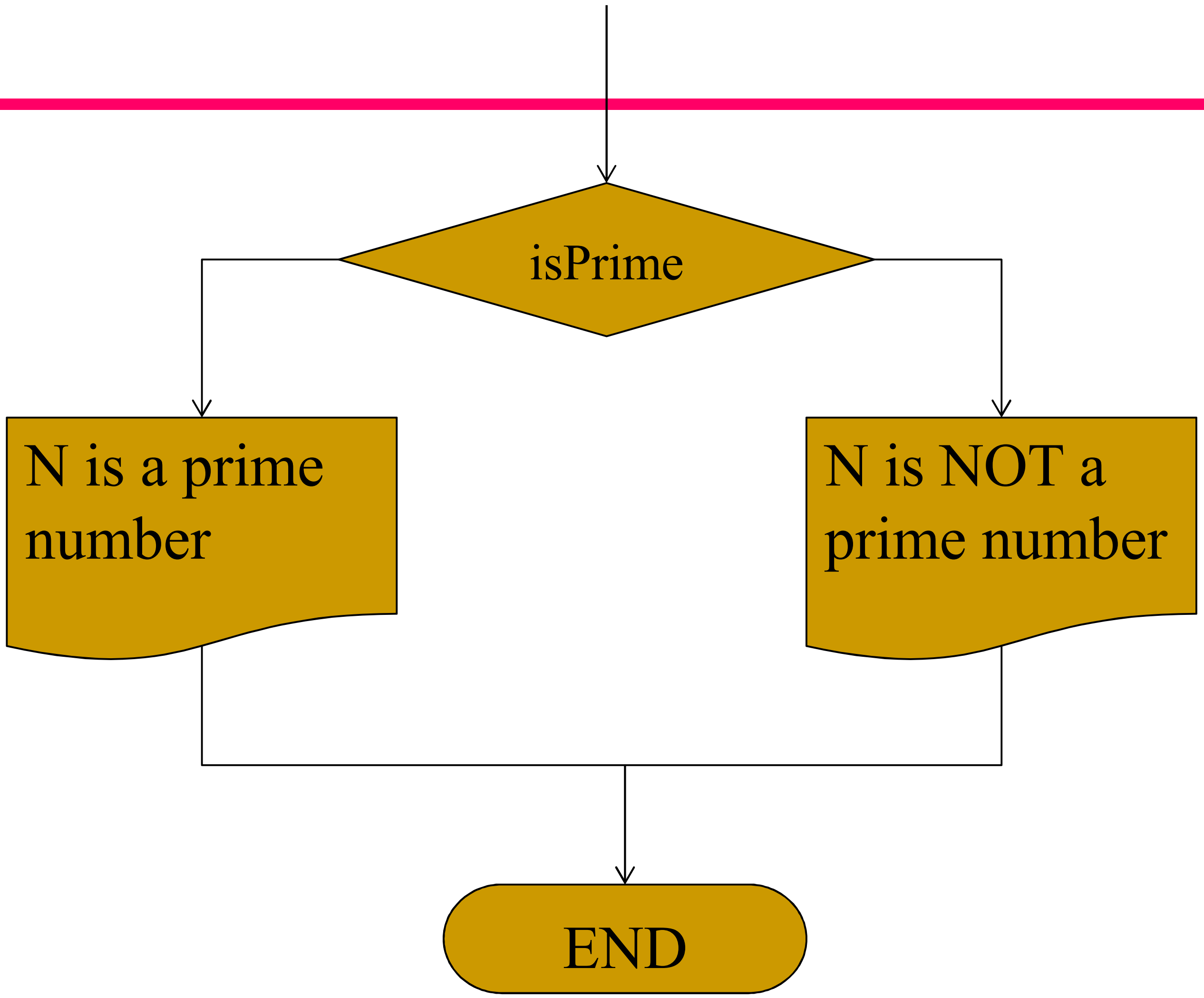
Algorithm 2: *Find largest of 1000 numbers*

```
    FindLargest
    Input: 1000 positive integers
1.  Set Largest to 0
2.  Set Counter to 0
3.  while (Counter less than 1000)
     3.1  if (the integer is greater than Largest)
              then
                3.1.1  Set Largest to the value of the integer
            End if
      3.2  Increment Counter
    End while
4.  Return Largest
    End
```

Example 3: Prime Number Test

- Given a natural number N, where N > 1.
    - If there exists an integer i, 1 < i < N, such that i can evenly divide N, then N is a composite number.
    - Otherwise, N is a prime number.

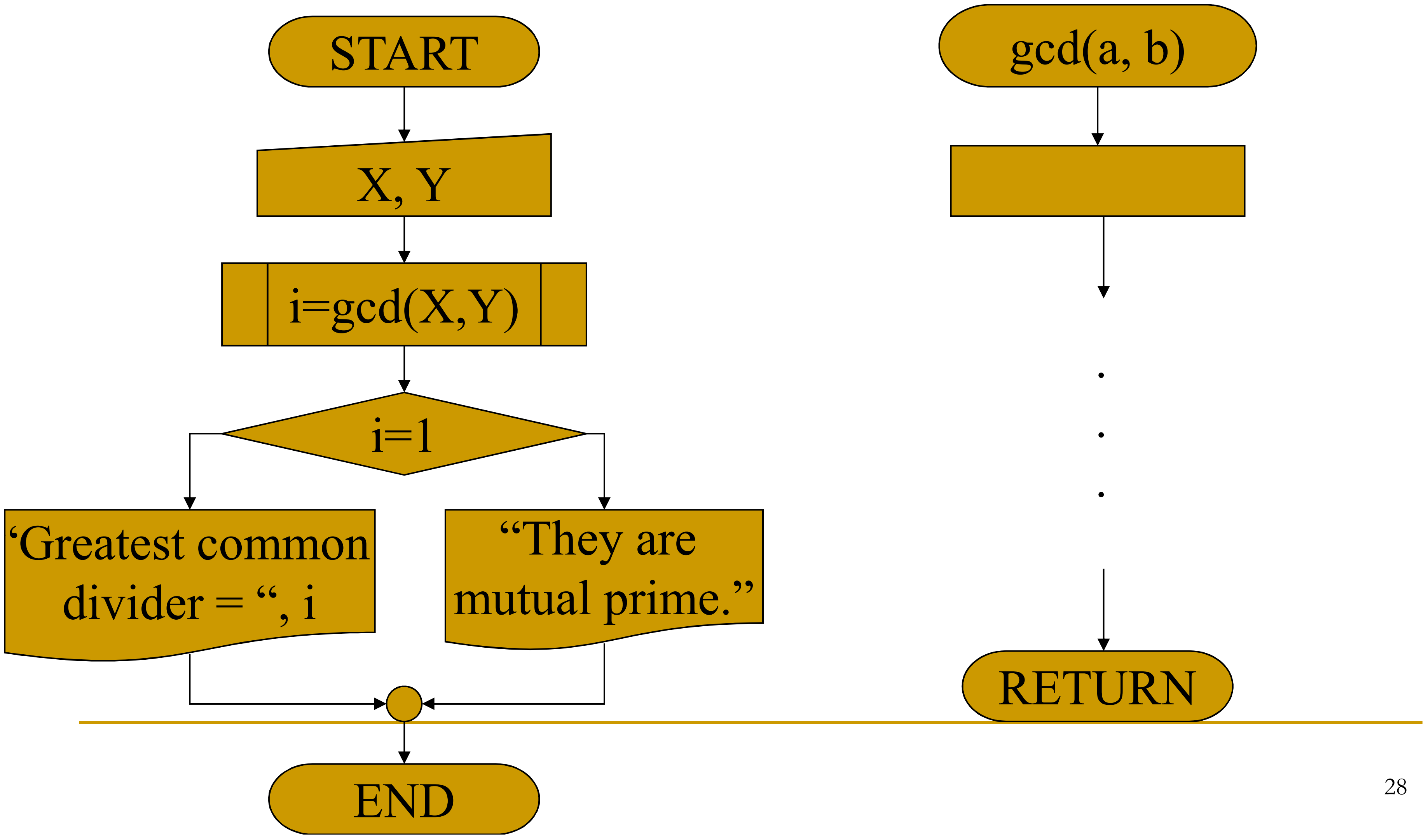

START
N
isPrime = true
i=i+1
i = 2
i < N
i evenly divide N
Y
isPrime = false
N


isPrime
N is a prime number
N is NOT a prime number
END

SUB-ALGORITHMS

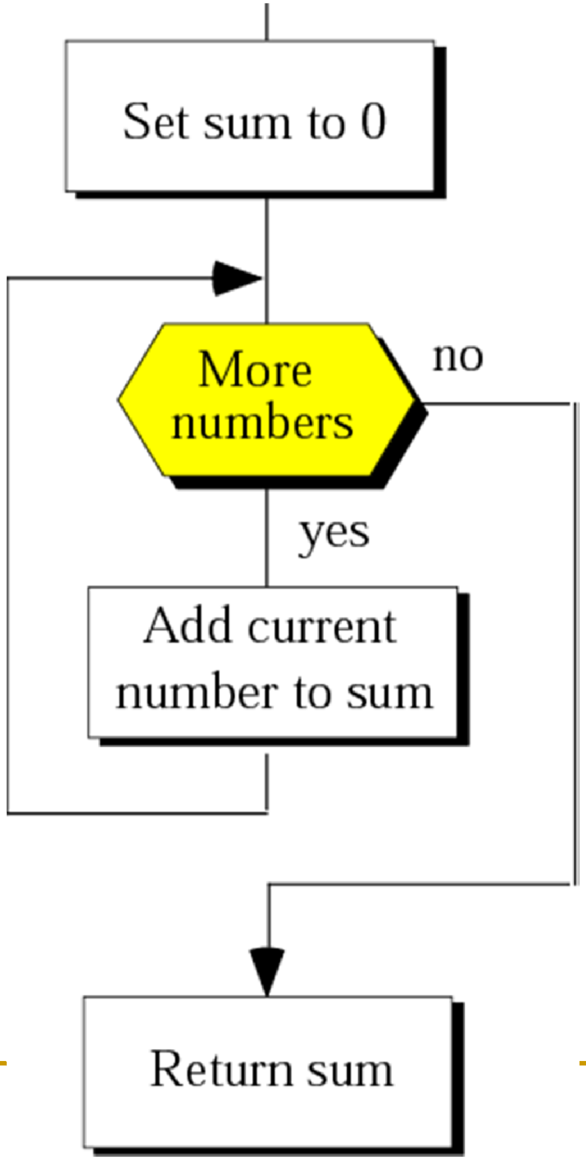
Concept of a subalgorithm

- An algorithm can be broken into smaller units called subalgorithms.

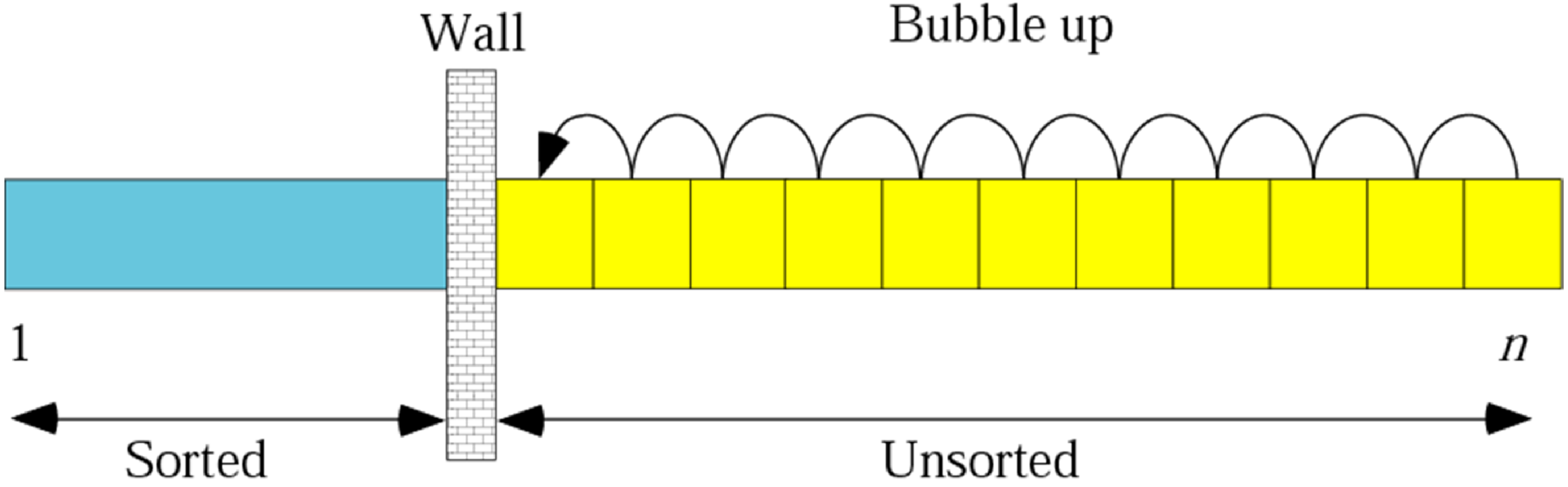


BASIC ALGORITHMS

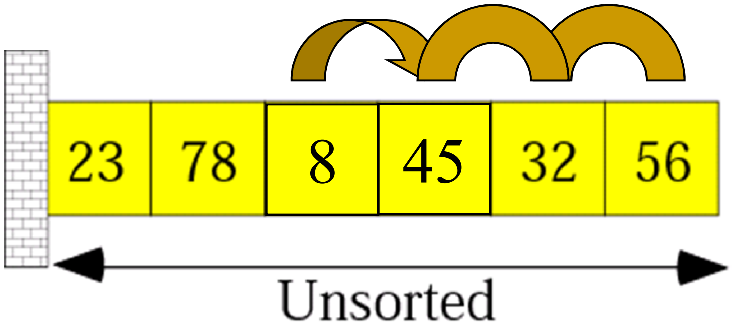
Summation


Set sum to 0
More numbers
no
yes
Add current number to sum
Return sum

Bubble sort

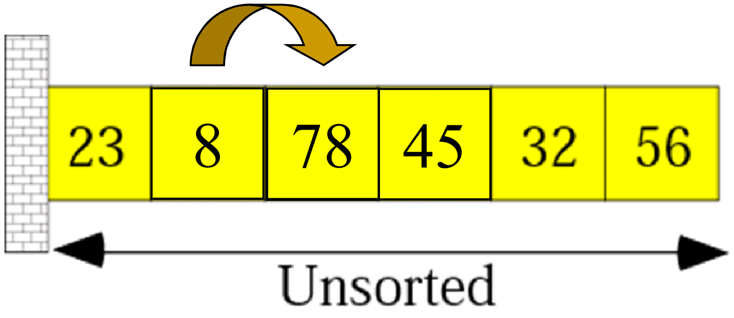


Example of bubble sort



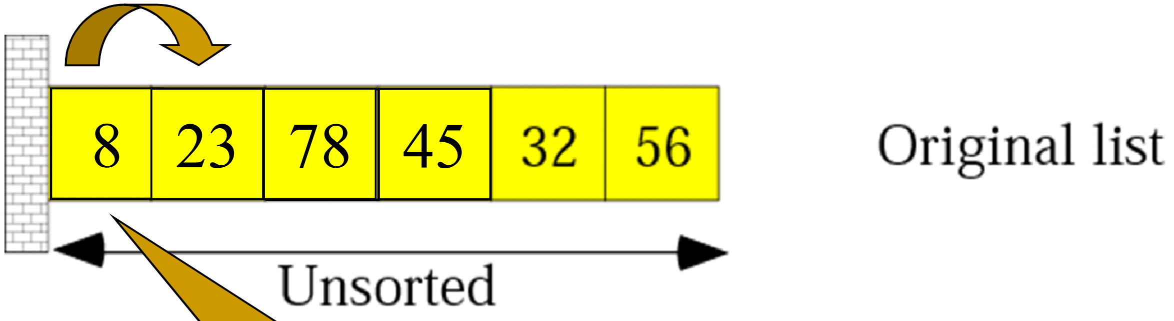
Original list

Example of bubble sort



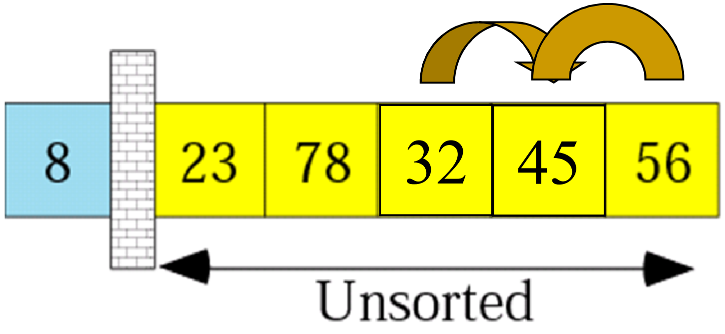
Original list

Example of bubble sort



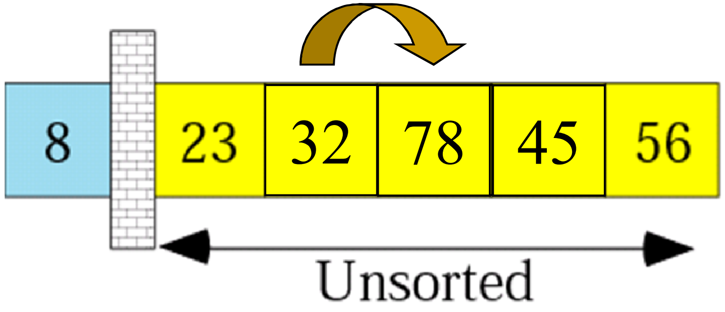
The smallest number is moved to the head.

Example of bubble sort



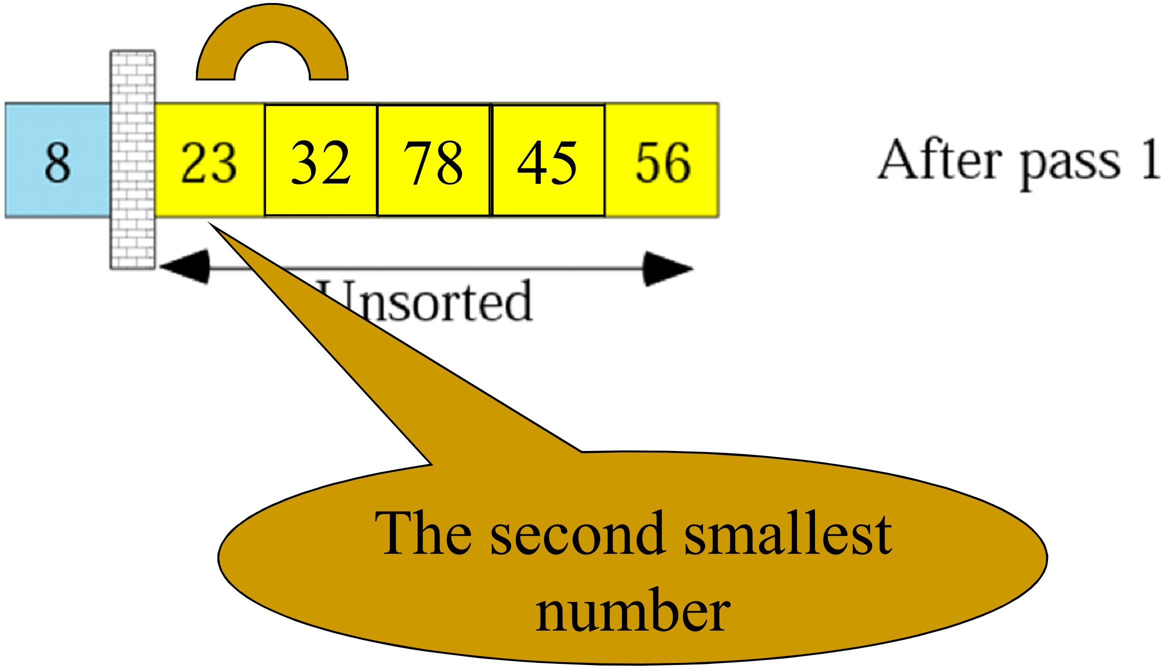
After pass 1

Example of bubble sort



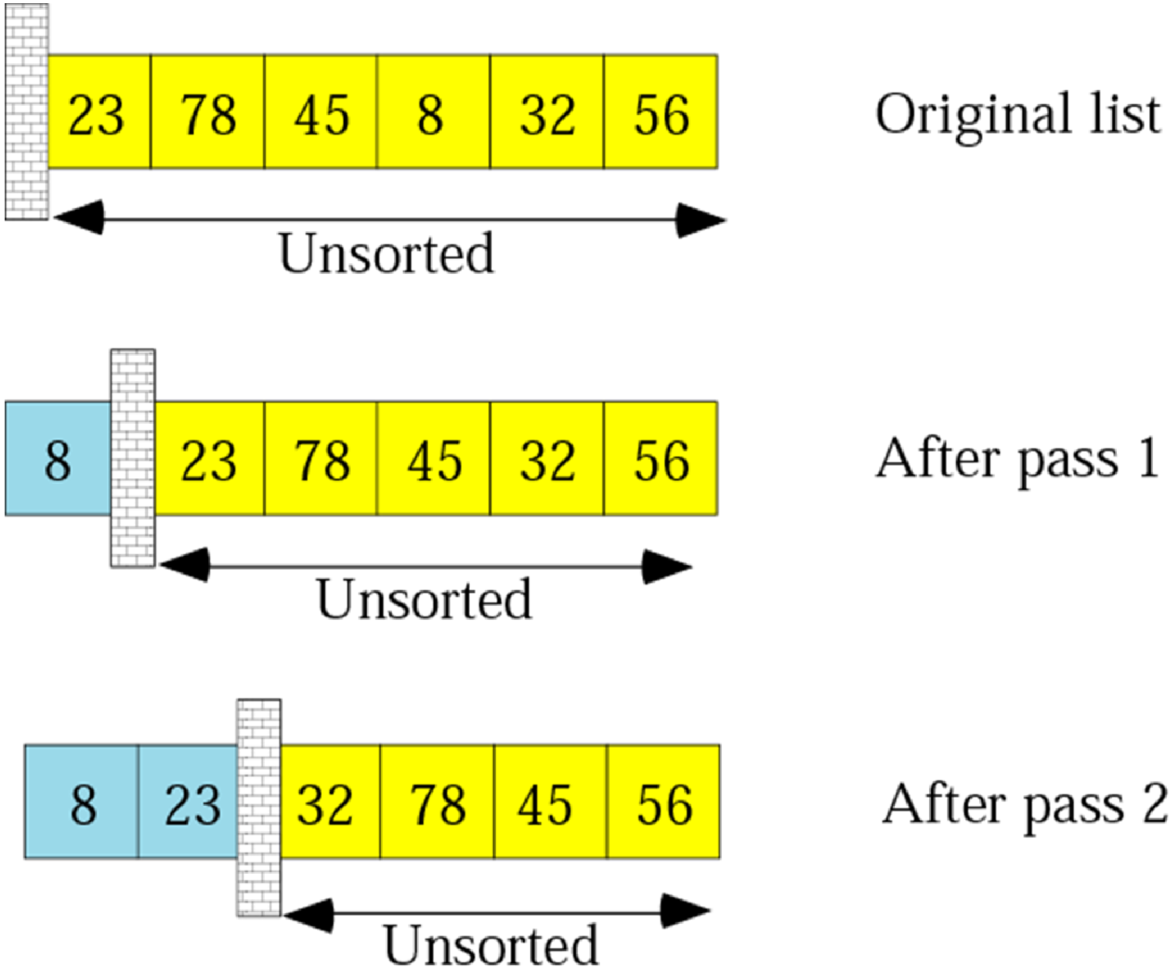
After pass 1

Example of bubble sort



The second smallest number

Example of bubble sort



23 78 45 8 32 56

Unsorted

Original list

8 23 78 45 32 56

Unsorted

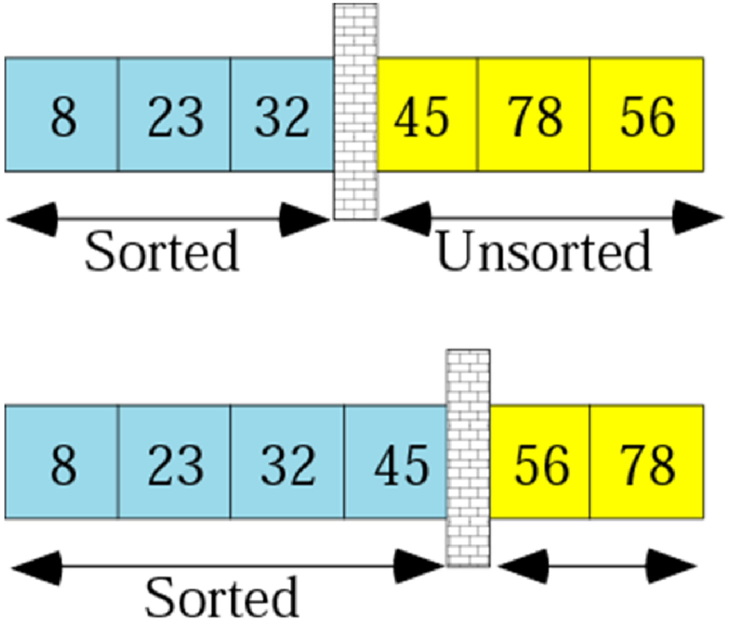
After pass 1

8 23 32 78 45 56

Unsorted

After pass 2

Example of bubble sort



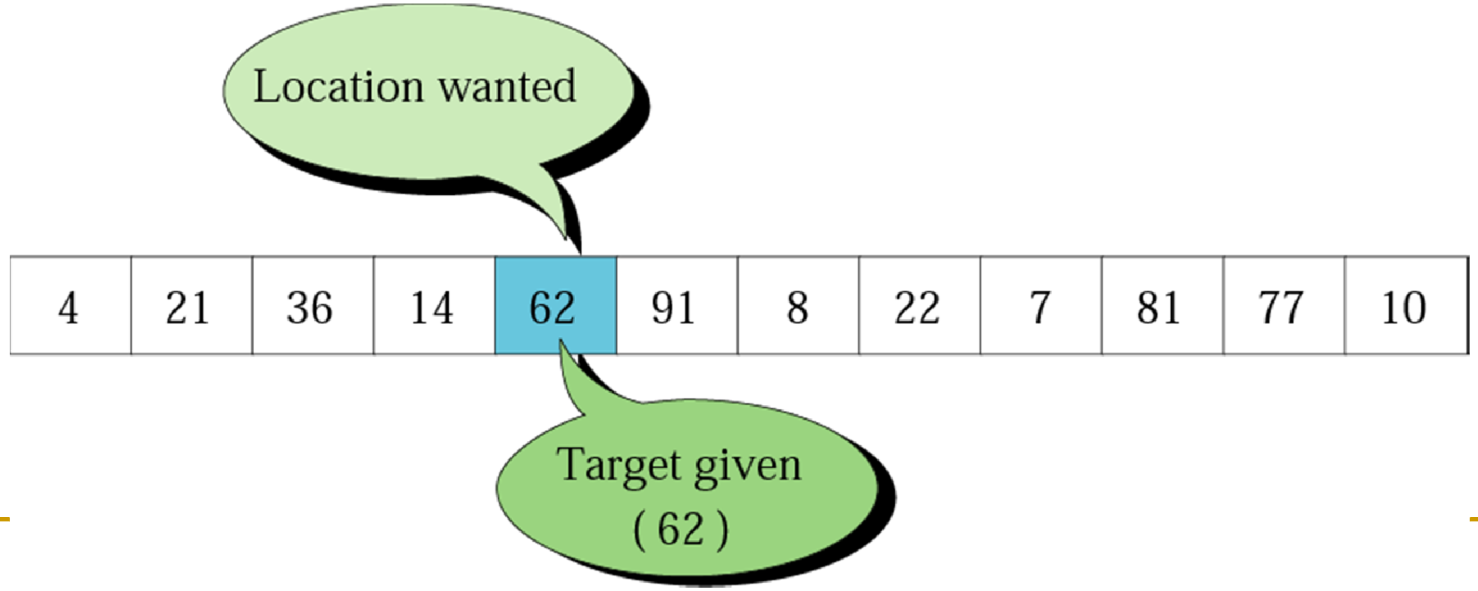
After pass 3

After pass 4
Sorted

And so on …

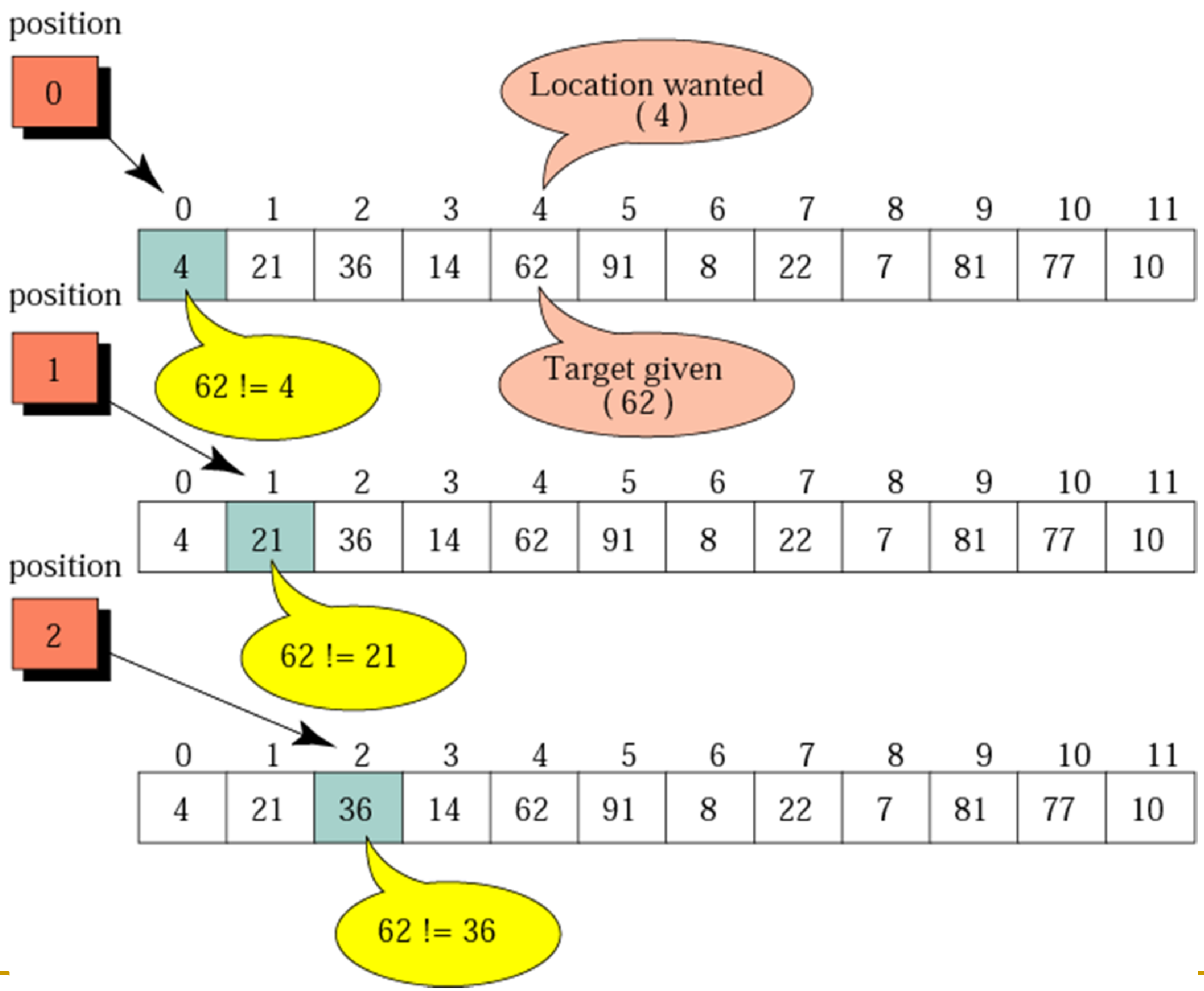
Search concept

- Searching, a process to locate a target in a list of data, is a basic algorithm.
- Sequential search is used for unordered lists.
- Binary search is used for ordered lists.



4	21	36	14	62	91	8	22	7	81	77	10

Example of a sequential search

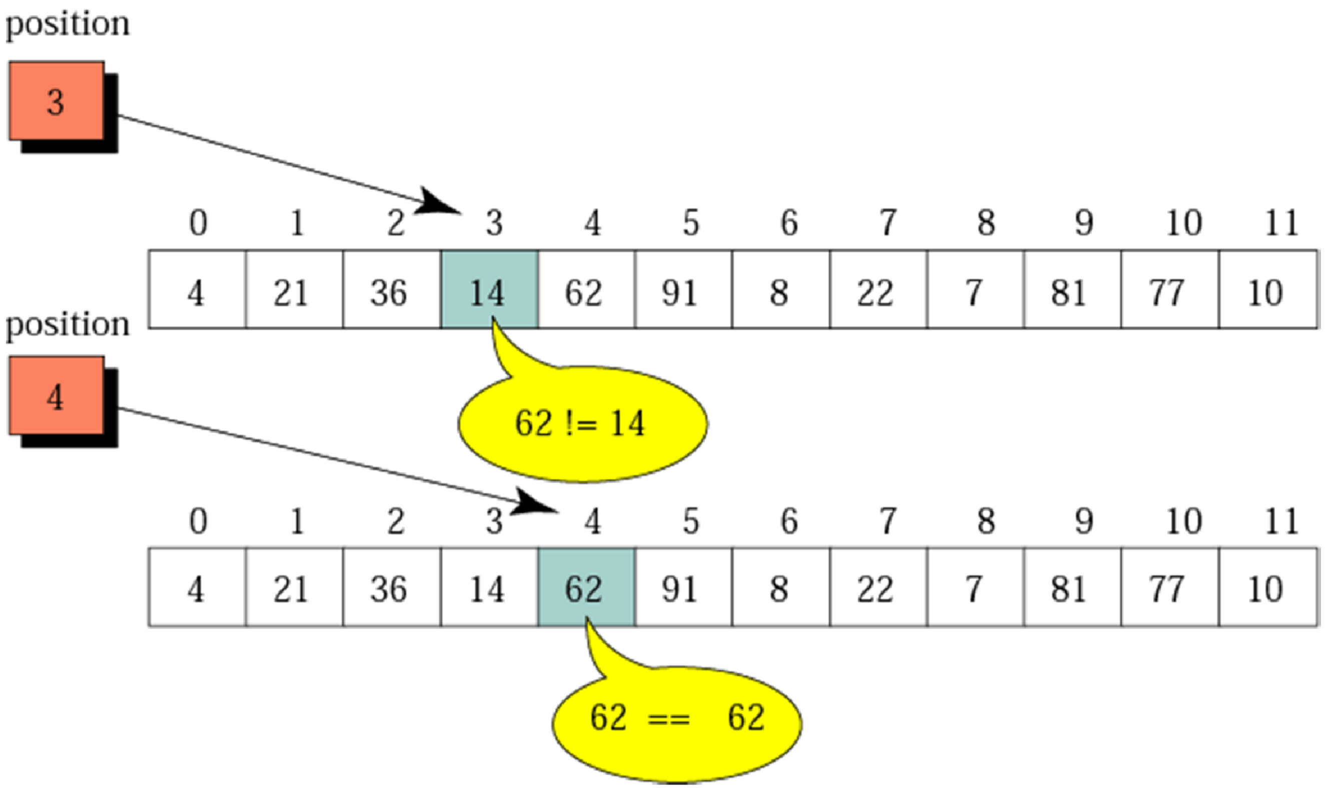


0	1	2	3	4	5	6	7	8	9	10	11
4	21	36	14	62	91	8	22	7	81	77	10

0	1	2	3	4	5	6	7	8	9	10	11
4	21	36	14	62	91	8	22	7	81	77	10

0	1	2	3	4	5	6	7	8	9	10	11
4	21	36	14	62	91	8	22	7	81	77	10

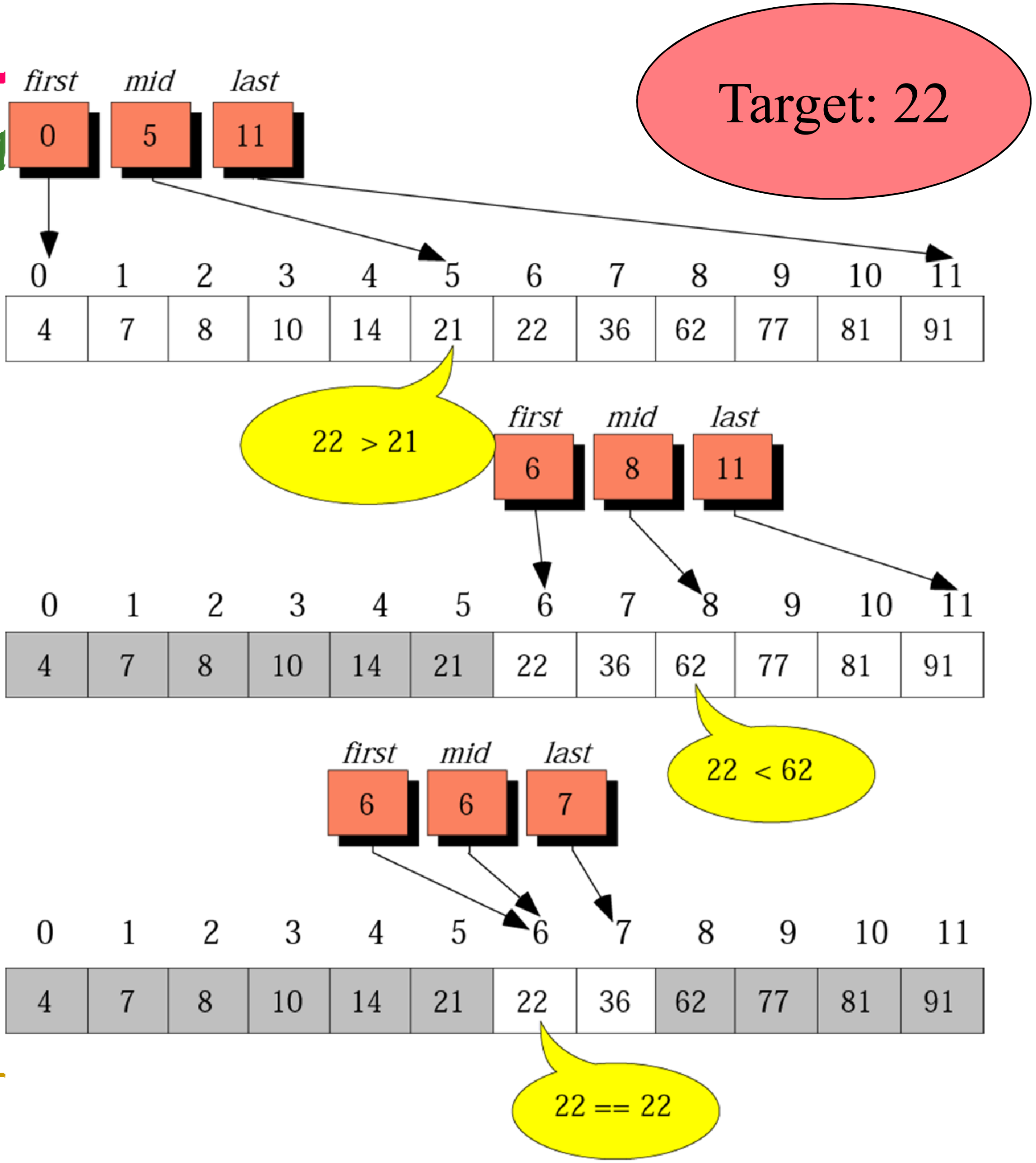
Example of a sequential search



0	1	2	3	4	5	6	7	8	9	10	11
4	21	36	14	62	91	8	22	7	81	77	10

0	1	2	3	4	5	6	7	8	9	10	11
4	21	36	14	62	91	8	22	7	81	77	10

Example of a binary search



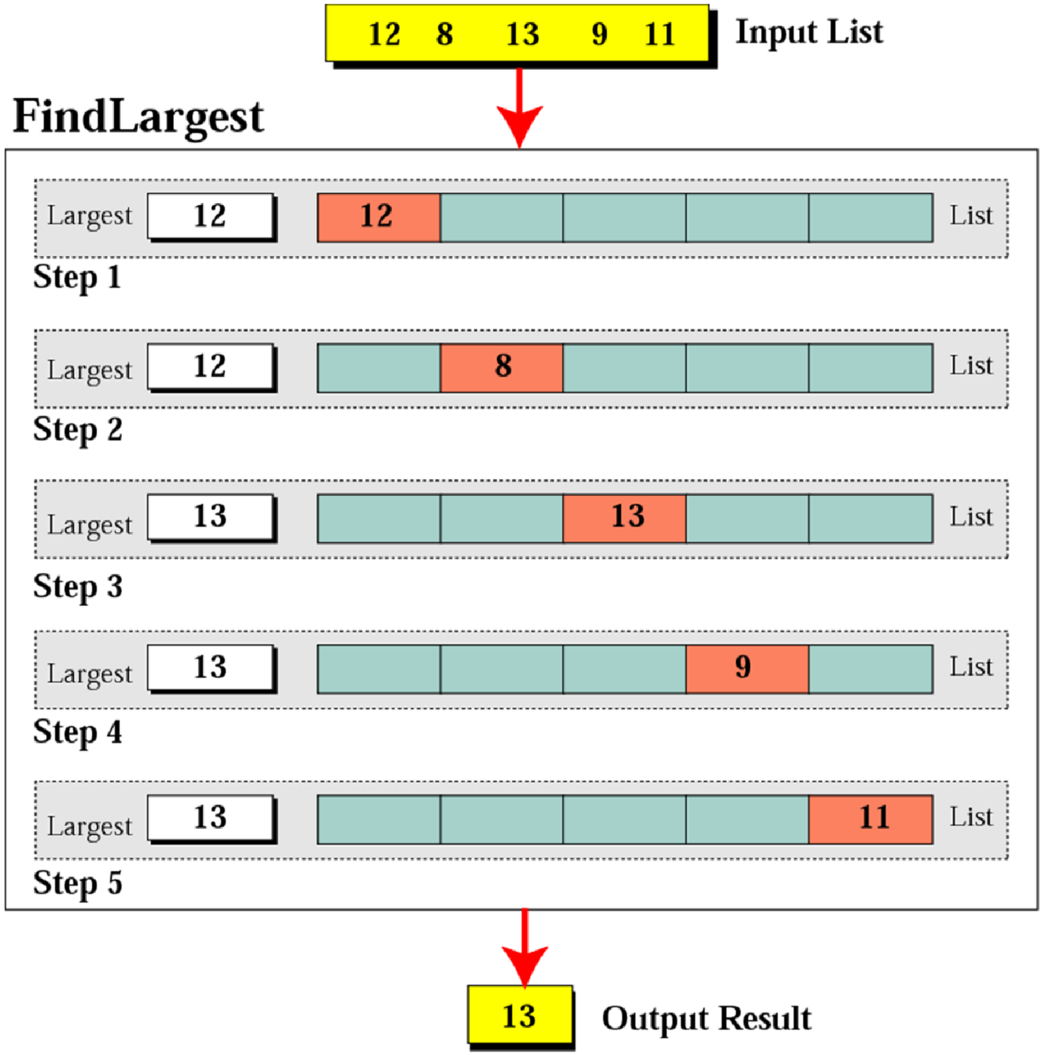
0	1	2	3	4	5	6	7	8	9	10	11
4	7	8	10	14	21	22	36	62	77	81	91

0	1	2	3	4	5	6	7	8	9	10	11
4	7	8	10	14	21	22	36	62	77	81	91

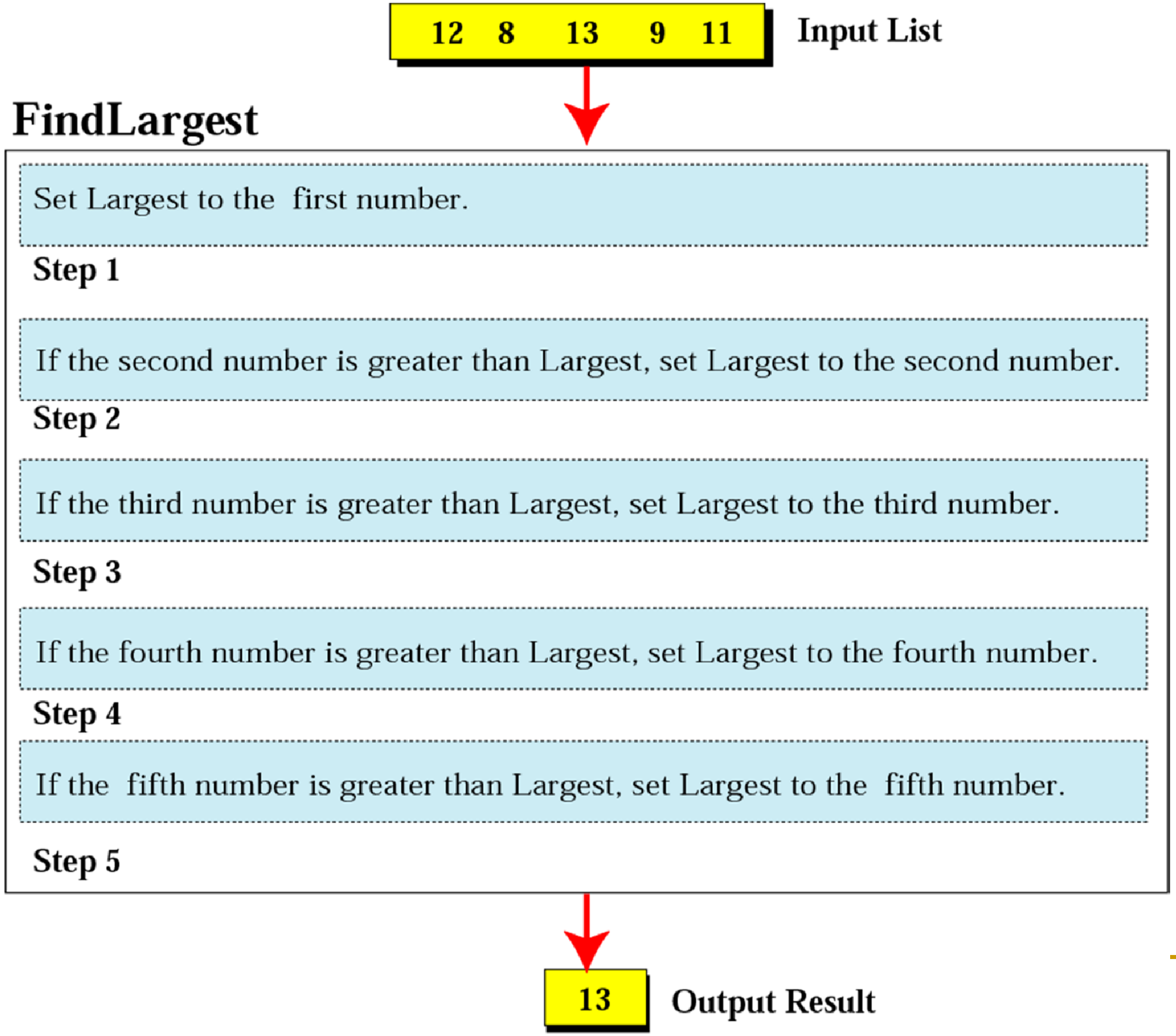
0	1	2	3	4	5	6	7	8	9	10	11
4	7	8	10	14	21	22	36	62	77	81	91

Another Example

The algorithm uses the following five steps to find the largest integer.



Defining actions in FindLargest algorithm



12 8 13 9 11 Input List

FindLargest

Set Largest to the first number.

Step 1

If the second number is greater than Largest, set Largest to the second number.

Step 2

If the third number is greater than Largest, set Largest to the third number.

Step 3

If the fourth number is greater than Largest, set Largest to the fourth number.

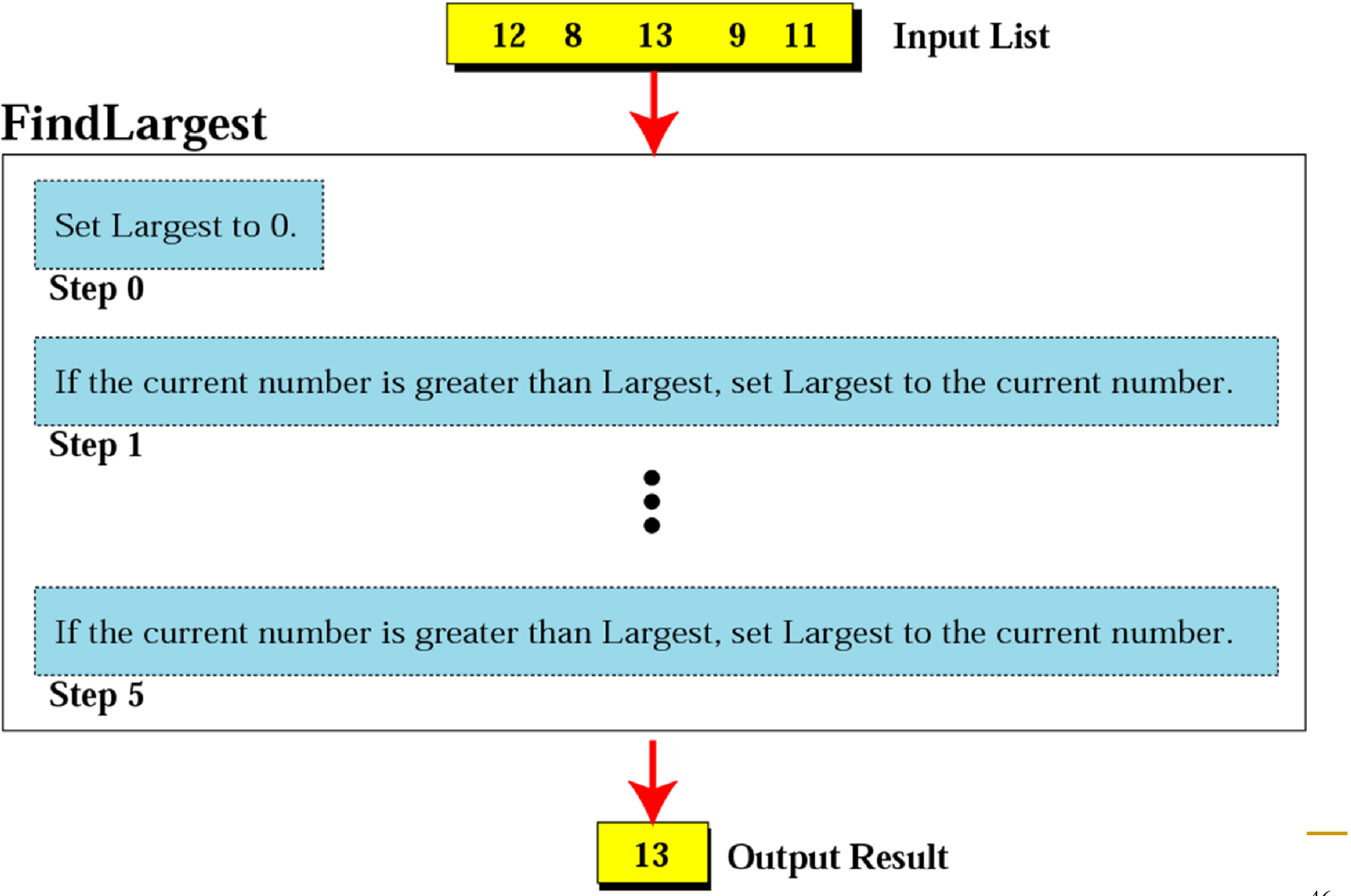
Step 4

If the fifth number is greater than Largest, set Largest to the fifth number.

Step 5

13 Output Result

Refinement



12 8 13 9 11

Input List

FindLargest

Set Largest to 0.

Step 0

If the current number is greater than Largest, set Largest to the current number.

Step 1

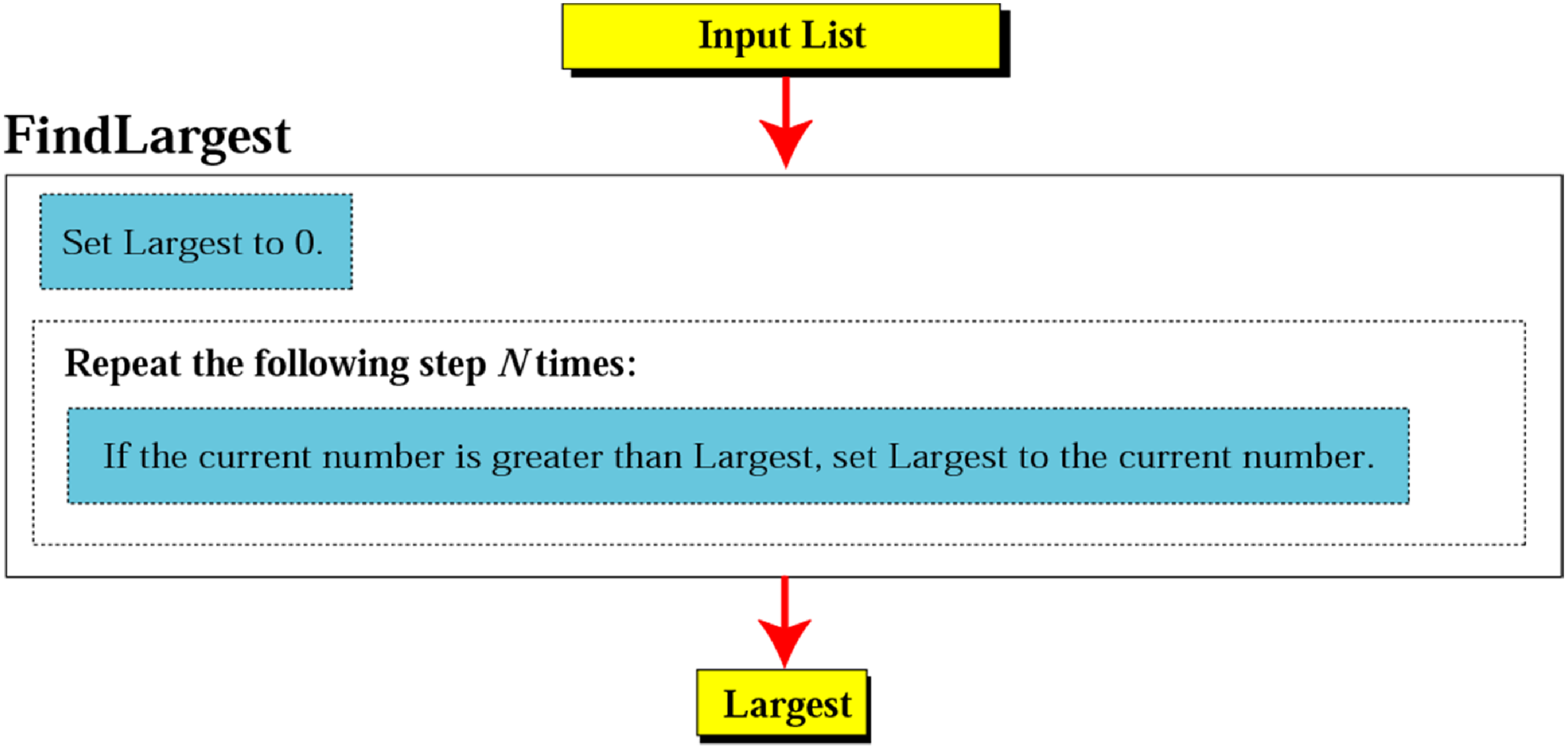
If the current number is greater than Largest, set Largest to the current number.

Step 5

13

Output Result

Generalization



Input List

FindLargest

Set Largest to 0.

Repeat the following step *N* times:

If the current number is greater than Largest, set Largest to the current number.

Largest