

# **L#5. Problem Solving Tools and Program Design in Computer Programming: The Repetition Structure**

Problems with Solutions Requiring Repetition

Summer 2015

# Repetition

What if we need to repeat several lines of an algorithm over and over?



Que aburrido prof:  
¿solo da vueltas?



# Program length?

*Example: Write a small program that will display the numbers 1 - 100.*

For so a simple task you would need one hundred lines.  
It must be another way.

*it's one!!*

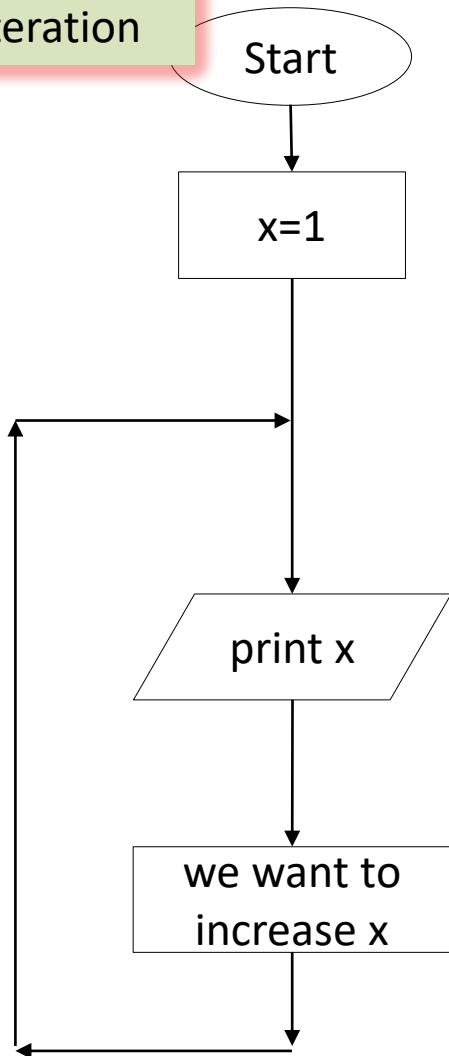
```
PRINT 1  
PRINT 2  
PRINT 3  
PRINT 4  
PRINT 5  
...  
...  
...  
PRINT 99  
PRINT 100
```



Without knowledge of repetition, You would most likely write a program that uses individual lines of code that print out each number.

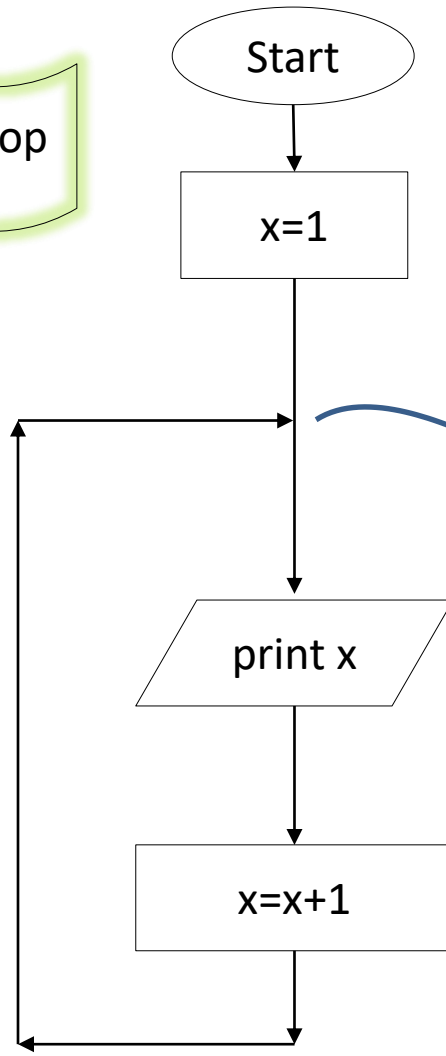
one iteration

A



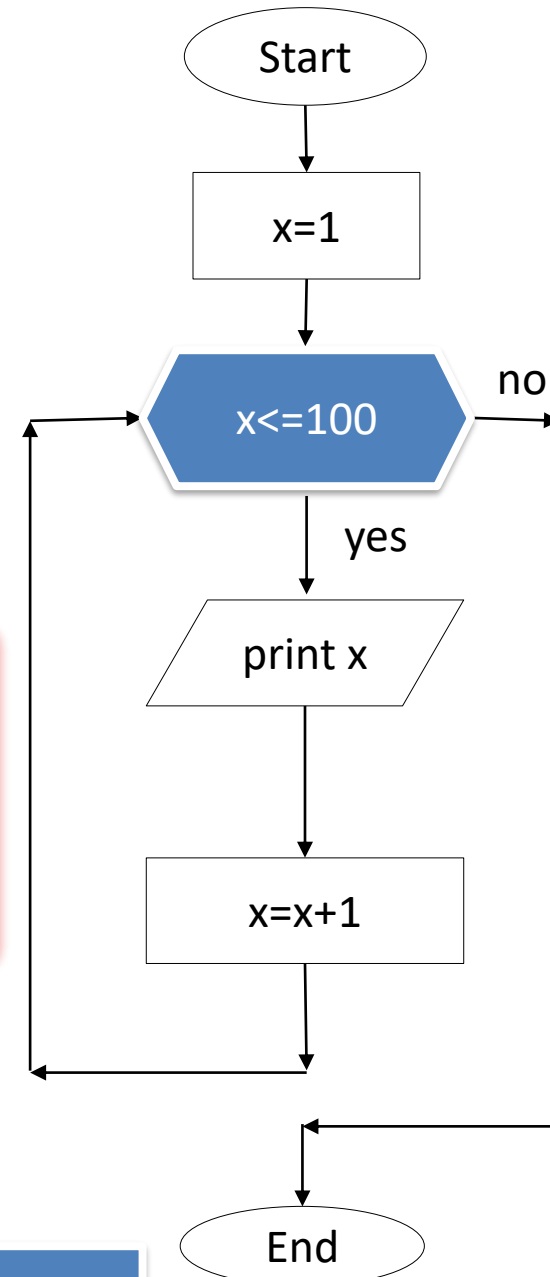
1

Developing a loop



2

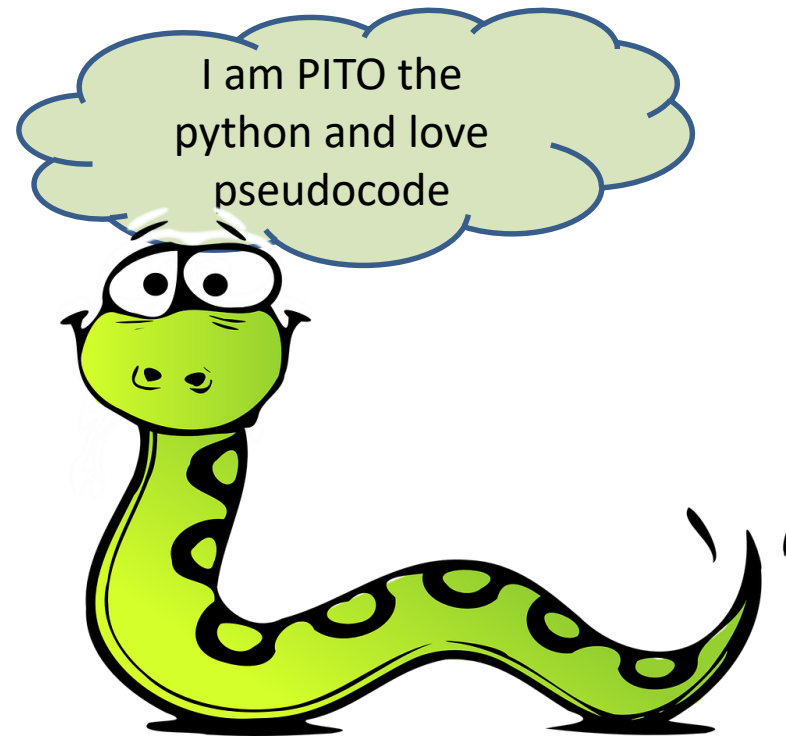
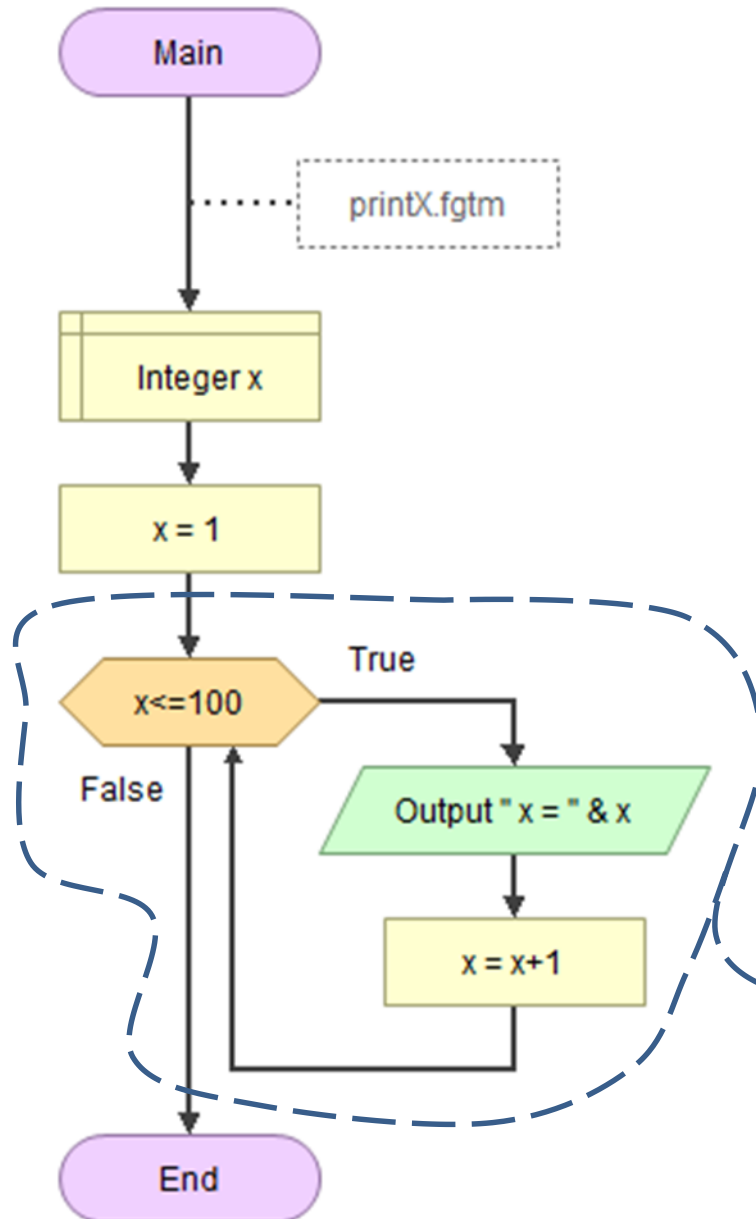
we want to stop looping



3

4

this means looping



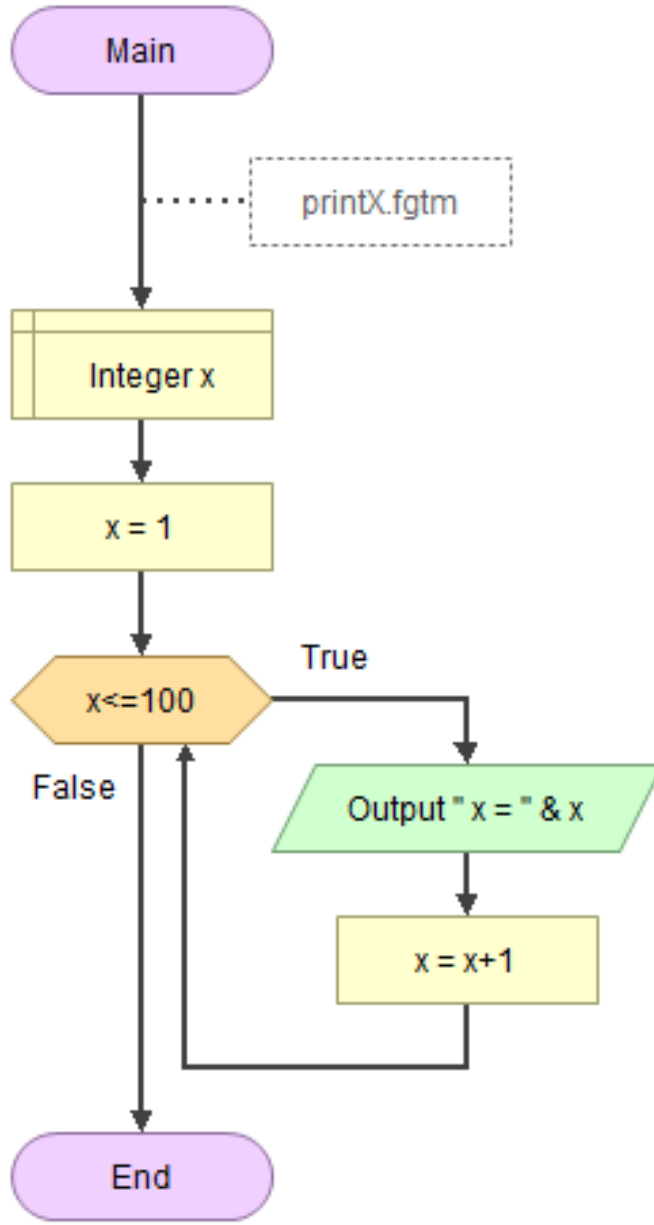
A LOOP STRUCTURE in pseudocode:

```
SET x=1
WHILE x<=100
  PRINT "x=", x
  ADD x=x+1
```

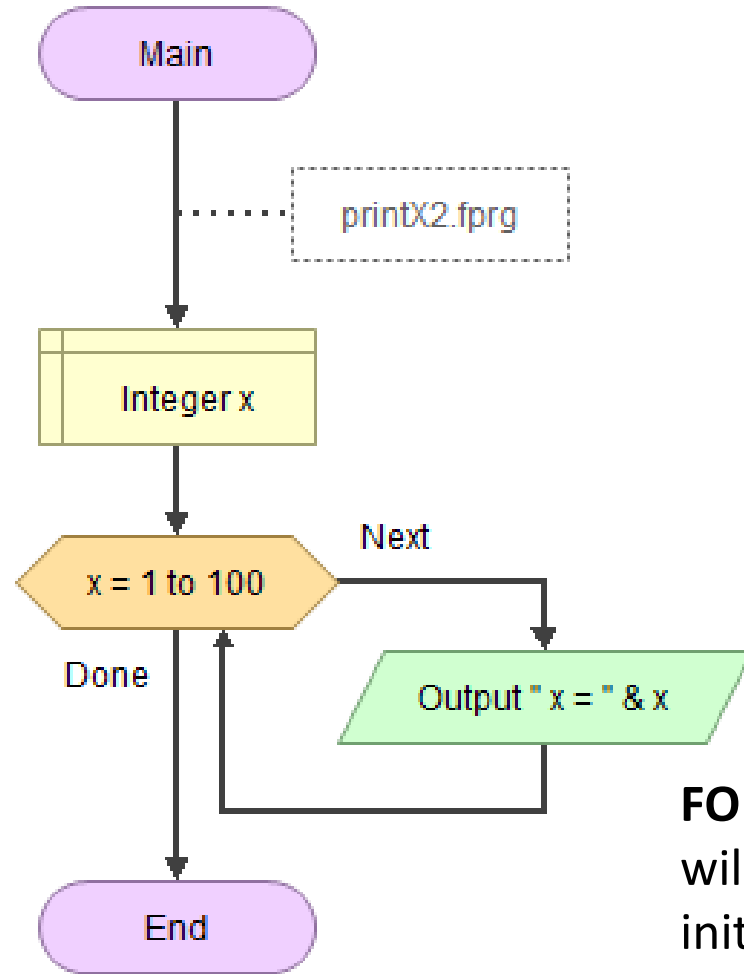
**Mientras** la condicion sea cierta las instrucciones de adentro se repiten

**"WHILE"** indicates that a certain chunk of code will be repeated (within the loop or indented code) as long as the condition ("x <= 100") is true. Once this condition becomes false, e.g., when x = 101, we skip to the **next** line after the indentation, after the loop.

While loop:



For loop:



# WHILE & FOR loops:

While loop:

```
SET x=1
WHILE x<=100
  PRINT "x=", x
  ADD x=x+1
```

For loop:

```
FOR x=1 to 100, Step 1
  PRINT "x=", x
```

**FOR** indicates that a certain chunk of code will be repeated (indented code) from the initial up to the final value of x, jumping by Step. After the last iteration at x = 100, we skip to the **next** line after the indentation, after the loop.

## Iteration Decomposition: Identifying the need of a loop

For example, see if you can determine which of the following problems might be best solved using a loop:

- **A.** Solving the equation  $y=2x^2 + x + 5$  for all  $x$  between 5 and 10, in steps of 1.
- **B.** Summing entered integers until the user keys -1
- **C.** The user enters in the current year and then his/her birth year. Your program computes the user's age. Perform this task again for several friends.

**A** The solution of

# Iteration Decomposition: Paper and Pencil

Solving the equation  
 $y=2x^2 + x + 5$  for all  $x$   
between 5 and 10

First, be  
sure you  
can solve it  
by paper  
and pencil?

$$x = 5$$

$$y = 2(5)^2 + 5 + 5 = 60$$

$$x = 6$$

$$y = 2(6)^2 + 6 + 5 = 83$$

$$x = 7$$

$$y = 2(7)^2 + 7 + 5 = 110$$

$$x = 8$$

$$y = 2(8)^2 + 8 + 5 = 141$$

$$x = 9$$

$$y = 2(9)^2 + 9 + 5 = 176$$

$$x = 10$$

$$y = 2(10)^2 + 10 + 5 = 215$$

STOP

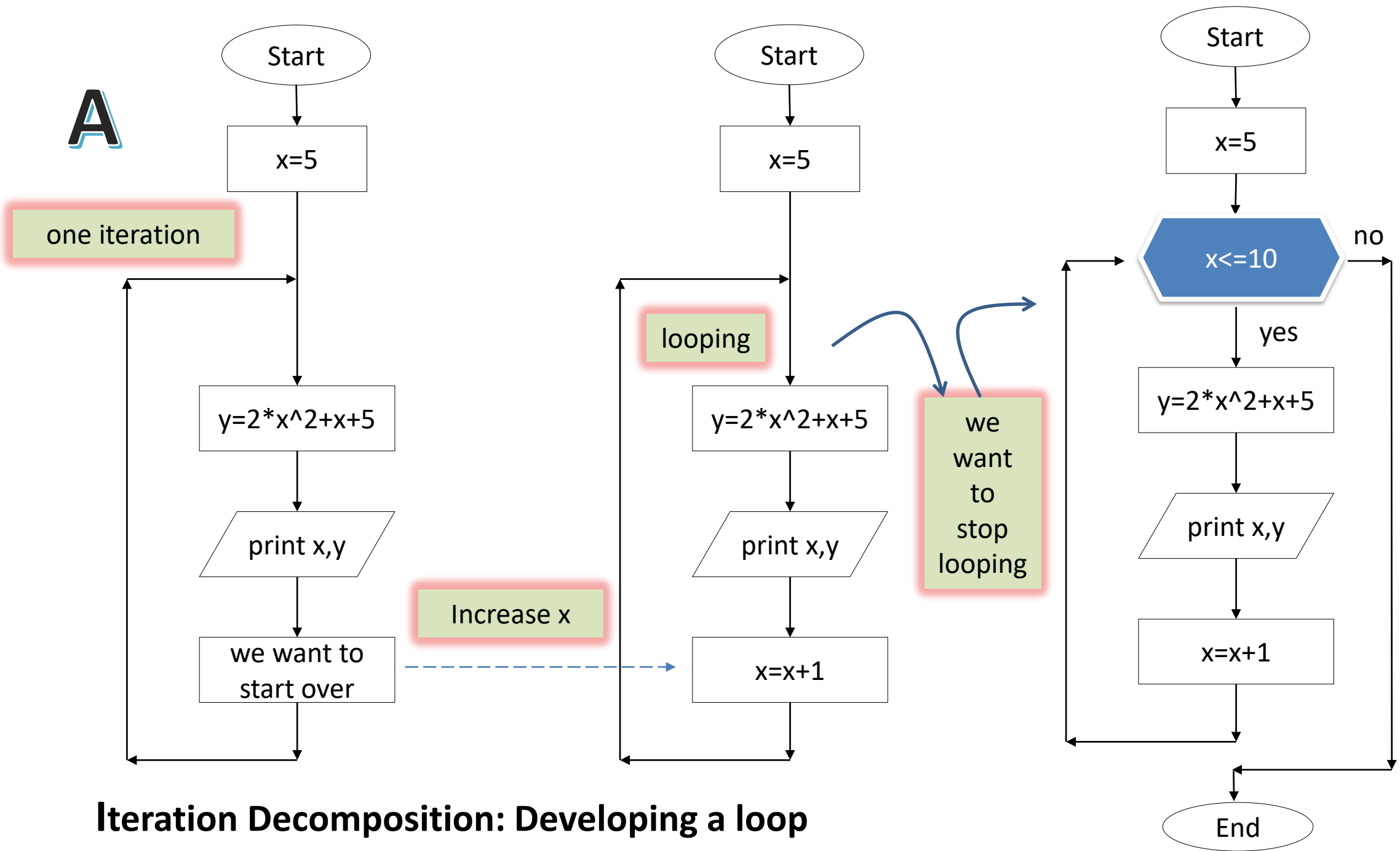
in table format

x	y
5	60
6	83
7	110
8	141
9	176
10	215

Next 3 slides shows the  
development of a loop with a  
flowchart and pseudocode.

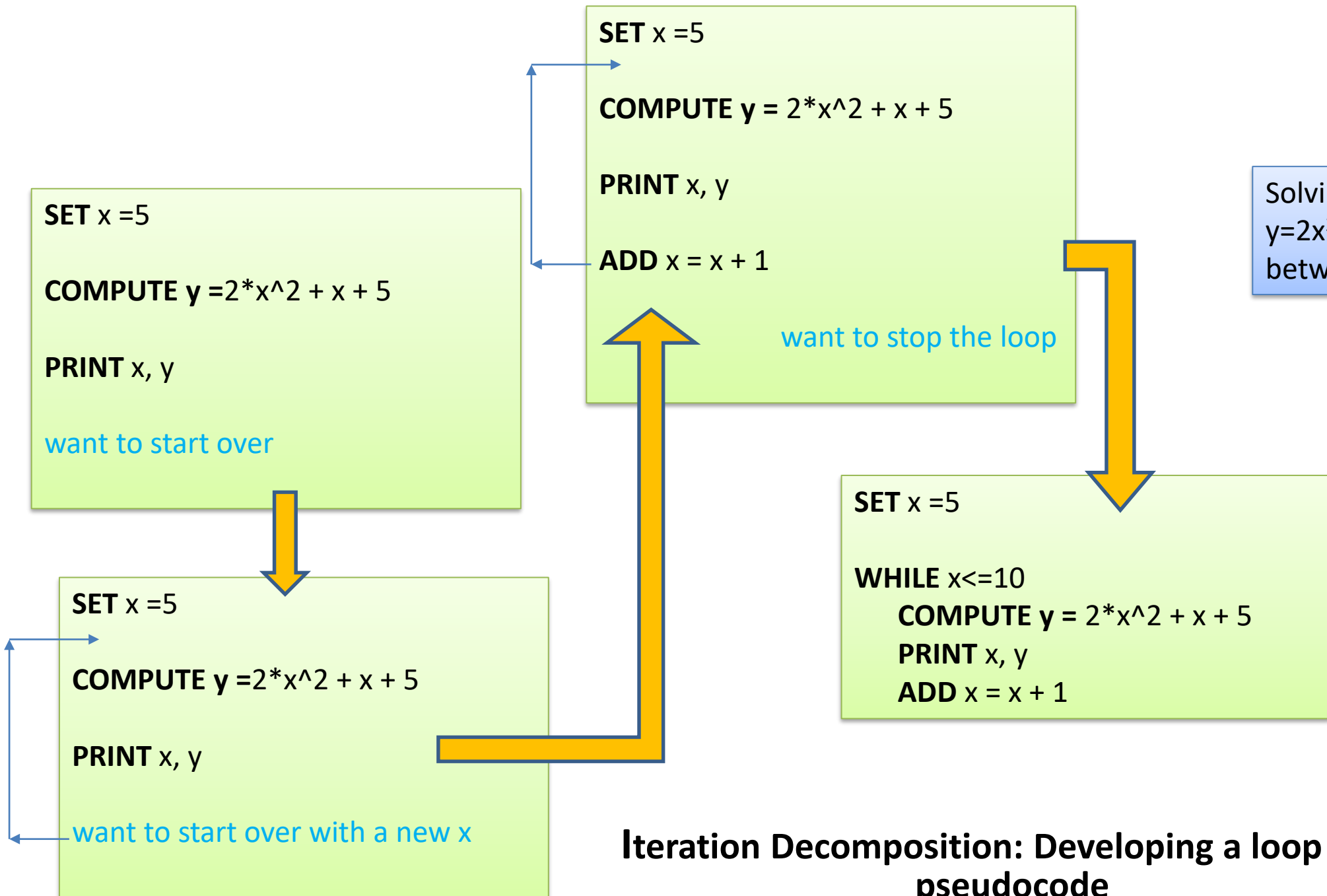


A



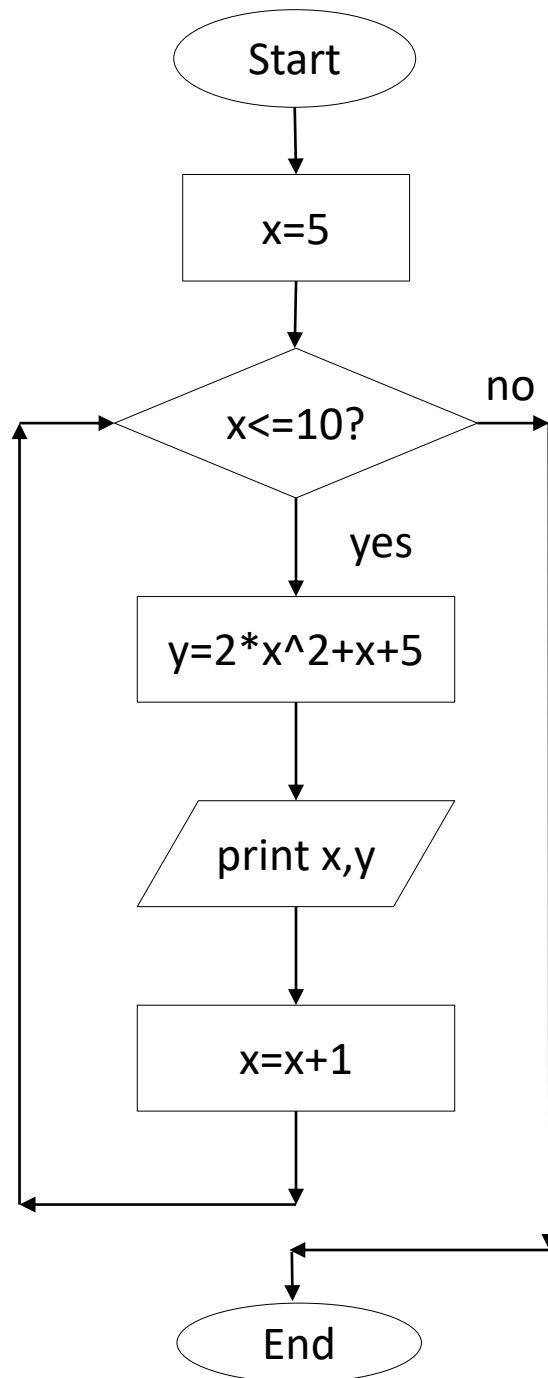
Iteration Decomposition: Developing a loop

Solving the equation  
 $y=2x^2 + x + 5$  for all  $x$   
between 5 and 10



Iteration Decomposition: Developing a loop with  
pseudocode

# A



## Final Flowchart and Pseudocode

Solving the equation  $y=2x^2 + x + 5$   
for all x between 5 and 10

**SET** x  $\leftarrow$  5  
**WHILE** x <= 10  
    **COMPUTE** y  $\leftarrow$  2\*x^2 + x + 5  
    **PRINT** x, y  
    **ADD** x  $\leftarrow$  x+1

### Discuss

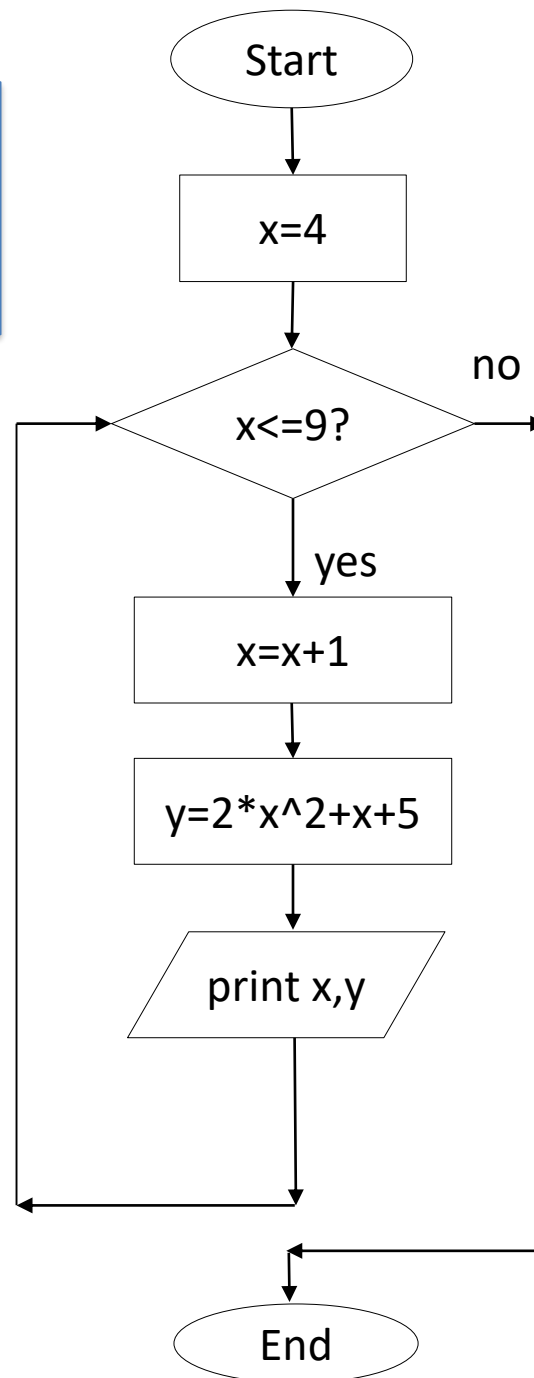
- ❖ Cycles or iterations
- ❖ Number of cycles
- ❖ Sequence of statements
- ❖ Arrows or equal sign in pseudocode
- ❖ Do I have always to develop the iteration process?

A

Solving the equation  
 $y=2x^2 + x + 5$  for all  $x$   
between 5 and 10



**This is another  
algorithm for the  
same purpose**



```
SET x ← 4
LOOP WHILE x <= 9
  ADD x ← x+1
  COMPUTE y ← 2*x^2 + x + 5
  PRINT x,y
```

# Iteration Decomposition



The skill you need to acquire is how you solve the problem by **breaking it down into steps** that are so simple that you can understand and the computer can execute them.

For a problem that needs iteration (i.e., loops, repetition). First solve a few instances of the problem by hand and think about what you did to find the partial solution. For example if the task is computing a table, compute few rows of the table by hand and repeat this process until you have computed the whole table.

# Trace Tables

- A technique used to test algorithms to make sure that no logical errors occur.
- Construct a table
  - Follow the development of each variable by writing it as heading for each row or column in a table.
  - Each *row or column* represents a variable or statement in the code and they should follow the sequence
  - This include input, output, and intermediate variables
- Follow the code flow and looping sequence by tracing the variable changes within the code.

En español esta técnica le llaman también “Prueba de Escritorio”

A

Develop a trace table for the following code:

SET  $x = 5$

WHILE  $x \leq 10$

    COMPUTE  $y = 2 * x^2 + x + 5$

    PRINT  $x, y$

    ADD  $x = x + 1$

EXCEL

# Trace Table

Paper & pencil (or Excel)

	D	E	F	G	H	I
31	iter	x	$x \leq 10?$	y	print x,y	$x = x + 1$
32	none	5				
33	1		TRUE	60	5,60	6
34	2		TRUE	83	6,83	7
35	3		TRUE	110	7,110	8
36	4		TRUE	141	8,141	9
37	5		TRUE	176	9,176	10
38	6		TRUE	215	10,215	11
39	EXIT		FALSE			

$G33 = 2 * E32^2 + E32 + 5$

$H33 = E32 \& ", "& G33$

$I33 = E32 + 1$

$G34 = 2 * I33^2 + I33 + 5$

$I34 = I33 + 1$

A



# Trace Table

Same problem, different algorithm:

SET x to 4

WHILE x <= 9

ADD x=x+1

COMPUTE  $y=2*x^2 + x + 5$

PRINT x,y

EXCEL

Paper and pencil, or Excel ?

	D	E	F	G	H	I
31	iter	x	x<=9?	x=x+1	y	print x,y
32	none	4				
33	1		TRUE	5	60	5,60
34	2		TRUE	6	83	6,83
35	3		TRUE	7	110	7,110
36	4		TRUE	8	141	8,141
37	5		TRUE	9	176	9,176
38	6		TRUE	10	215	10,215
39	EXIT		FALSE			

G33=E32+1

H33=2\*G33^2+G33+5

I33 =G33&"", "&H33



# Suggested Exercise

Hands on: Now, on your own, design the programs for the recommended exercises: A, B, and C, through the steps:

- Iteration decomposition
- Trace table
- Flowcharting
- Pseudocode

# Synthesis of Tools for (Programming) Problem Solving

