Algorithms, Flowchart and Pseudocode

Marie Yvette B. de Robles
Institute of Computer Science
University of the Philippines Los Baños

Objectives

At the end of this meeting, students should be able to:

- solve a simple problem and illustrate it using a flowchart
- explain the sequence control for algorithms

From Specifications to Algorithms to Programs

- **Specification** – **What** do we have to do?

  Precise statement of what the problem is about: What are the inputs? What should the output be?
- **Algorithm** – **How** do we do it?

  Sequence of steps to solve the problem
- **Program**

  The algorithm written in a programming language, ready for compilation and execution

Algorithms in Everyday Tasks

Describe informal instructions for these:

- How do you find a specific book, journal article or map in the main library?
- How do you add, delete, change a subject in your registration for this semester?
- How do you compute your general weighted average for all the subjects you have taken in your curriculum?

Flowcharts and Pseudocode

Flowcharts and pseudocode are two commonly used tools to help document the algorithm.

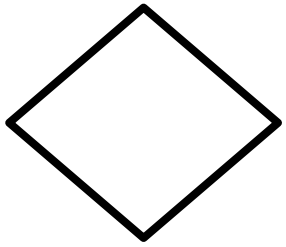
- **Pseudocode** is an artificial and informal language that helps programmers develop algorithms. Pseudocode is very similar to everyday English.
- A **flowchart** is a graphical representation of an algorithm.

Flowchart Symbols

Start or end of the program

Process to be carried out
e.g. addition, subtraction, division, etc.

Input or output operation



Decision making and branching

Sequence Control for Algorithms

Sequence

perform several steps in the given sequence

Selection (or branching)

perform some group of statements depending on some condition

Iteration (or looping)

perform some group of statements repeatedly

Functions

group together a block of instructions as a single, named logical unit

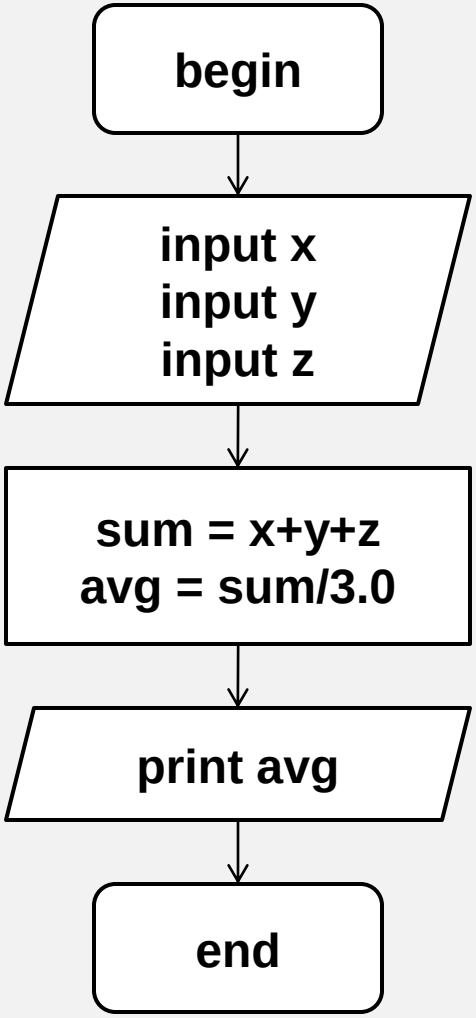
Sequence

- Do a block of steps in the given order

```
{
   wake_up;
   dress_up;
   go_to_school;
}
```

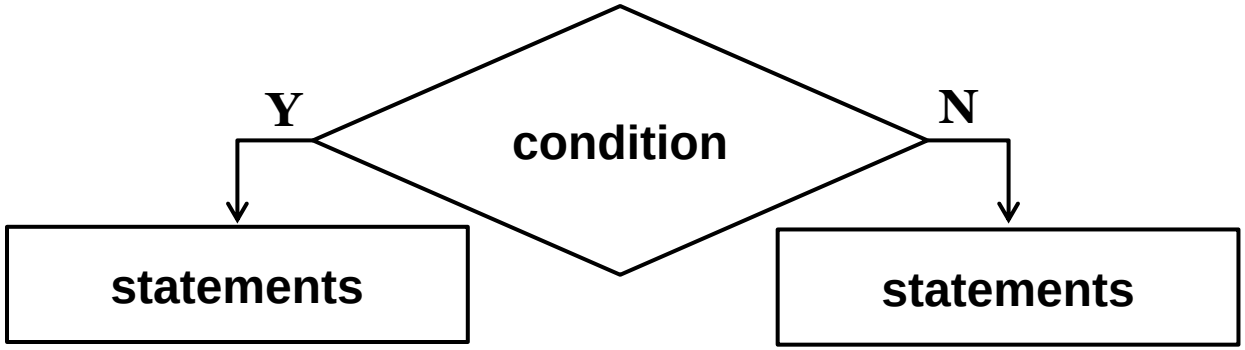
Sequence

Problem 1: Write an algorithm to determine a student's final grade. The final grade is calculated as the average of three marks.

Flowchart	Pseudocode
	• Read *x*, *y*, *z* • Compute *sum* as *x* + *y* + *z* • Compute average as *sum / 3* • Write the average

Selection (or branching)

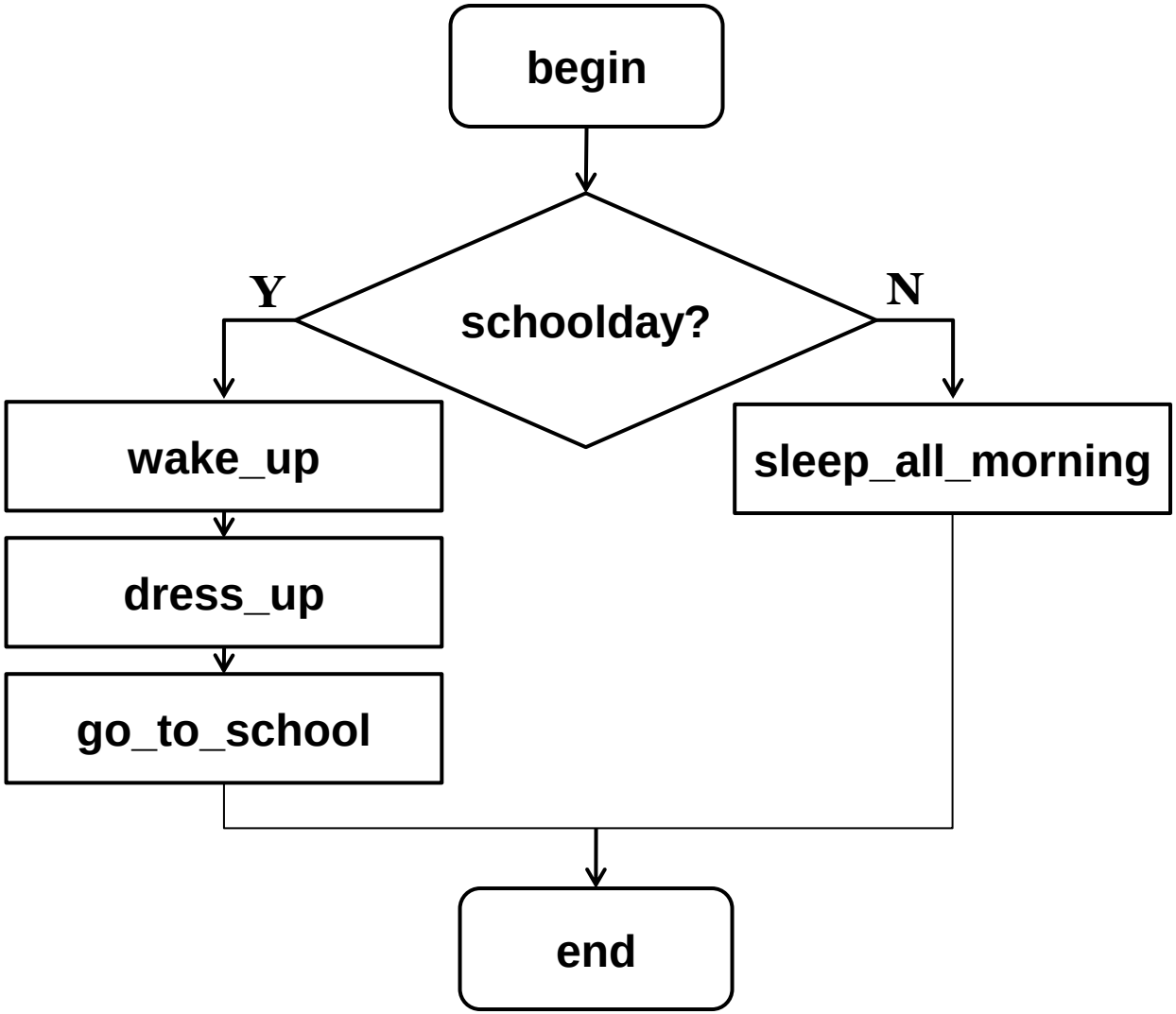
Evaluate the ***condition*** and perform a block of steps if the condition is true (otherwise perform the other block of steps)



Selection (or branching)

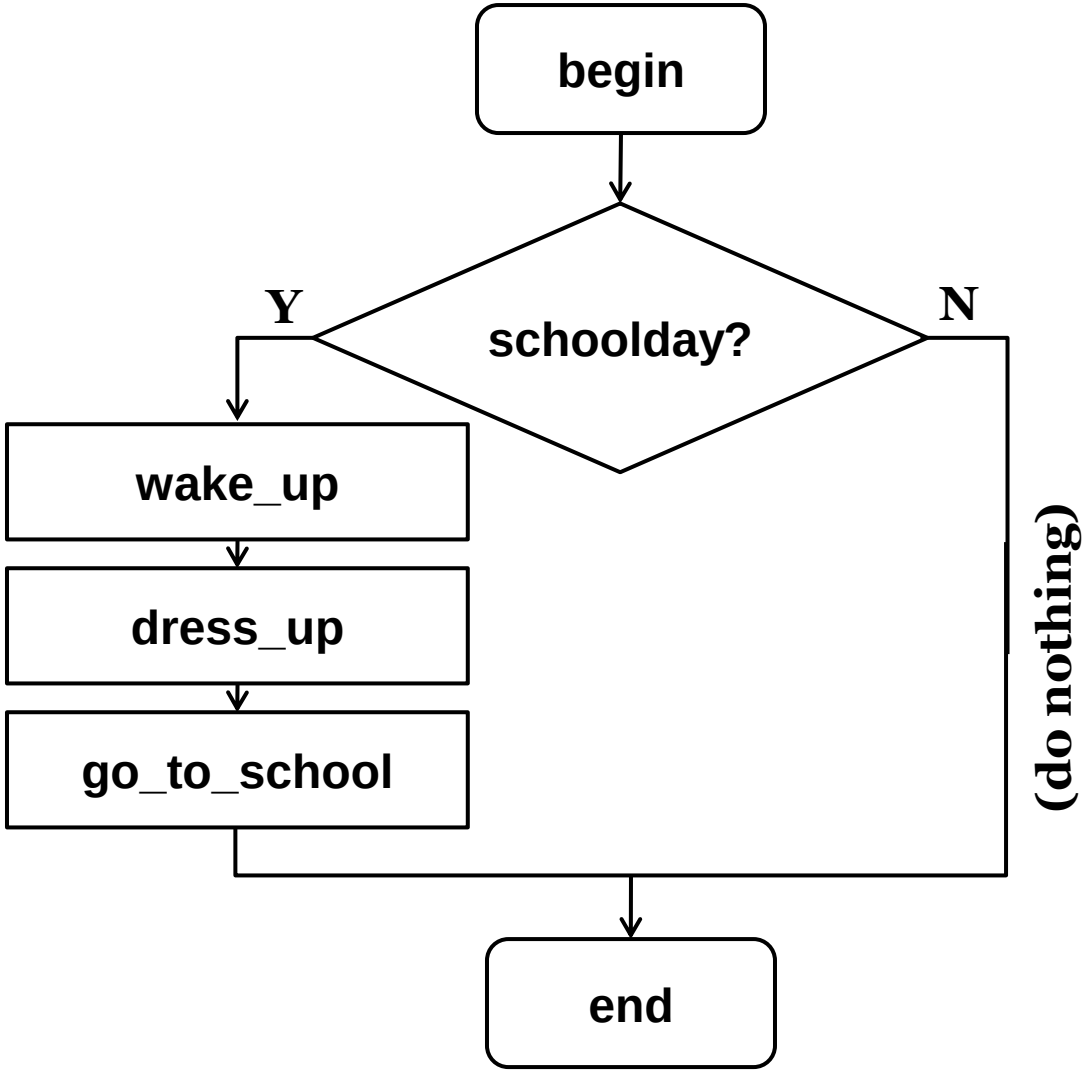
```
/* condition that may be T or F */
if ( today_is_a_school_day ) {
   wake_up;                    // if a school day …
   dress_up;
   go_to_school;
}
else{
   sleep_all_morning;    // if not a school day …
}
```

Selection (or branching)



A branch can be empty

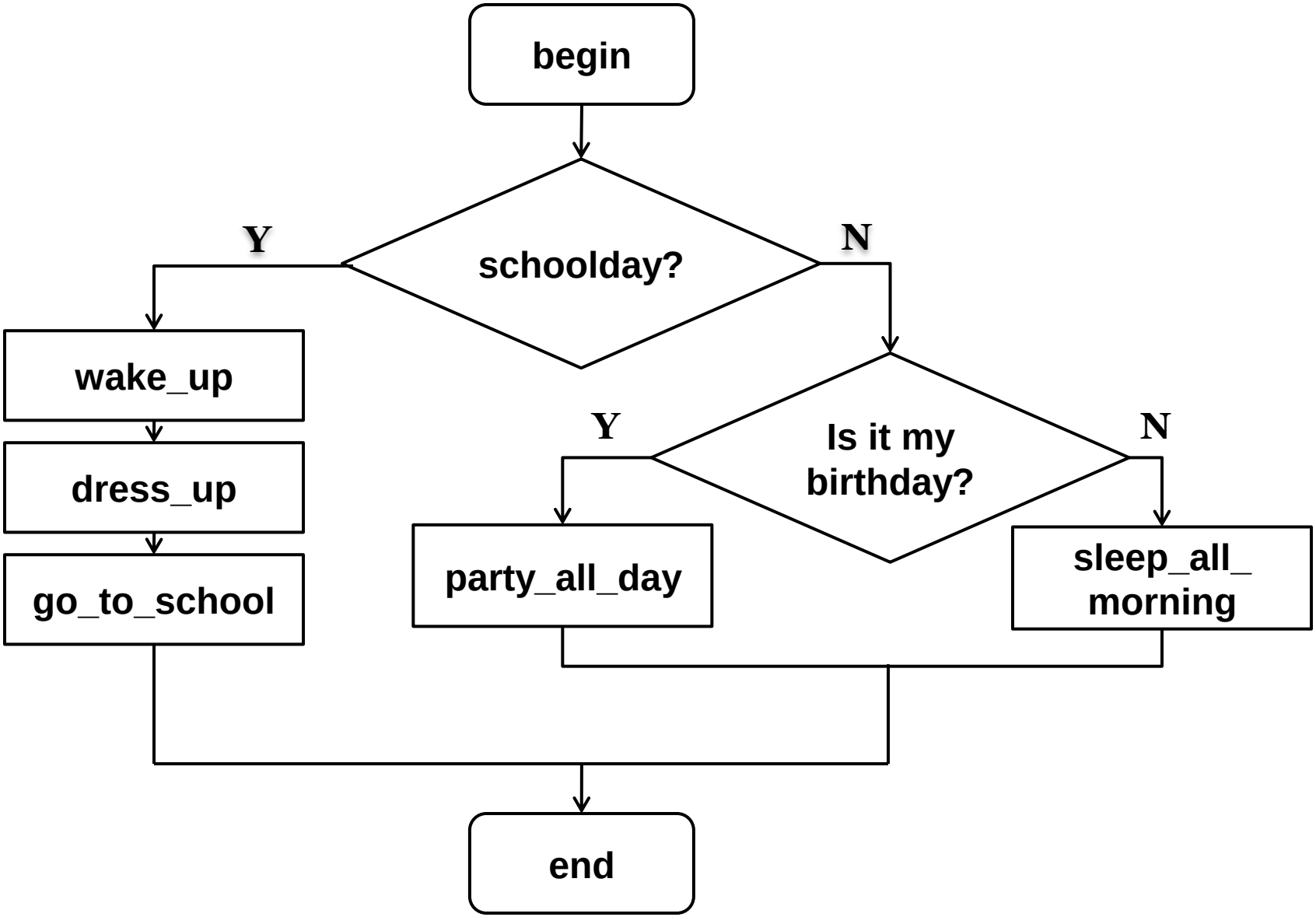
```
if(schoolday){
   wake_up;
   dress_up;
   go_to_school;
}
```



A branch can have more branches

```
if ( today_is_a_school_day )
{
    wake_up;
    dress_up;
    go_to_school;
}
```

```
else{
    if ( my_birthday_today ){
            party_all_day;
    }
    else{
            sleep_all_morning;
    }
}
```


begin
schoolday?
Y
N
wake_up
dress_up
go_to_school
Is it my birthday?
Y
N
party_all_day
sleep_all_ morning
end

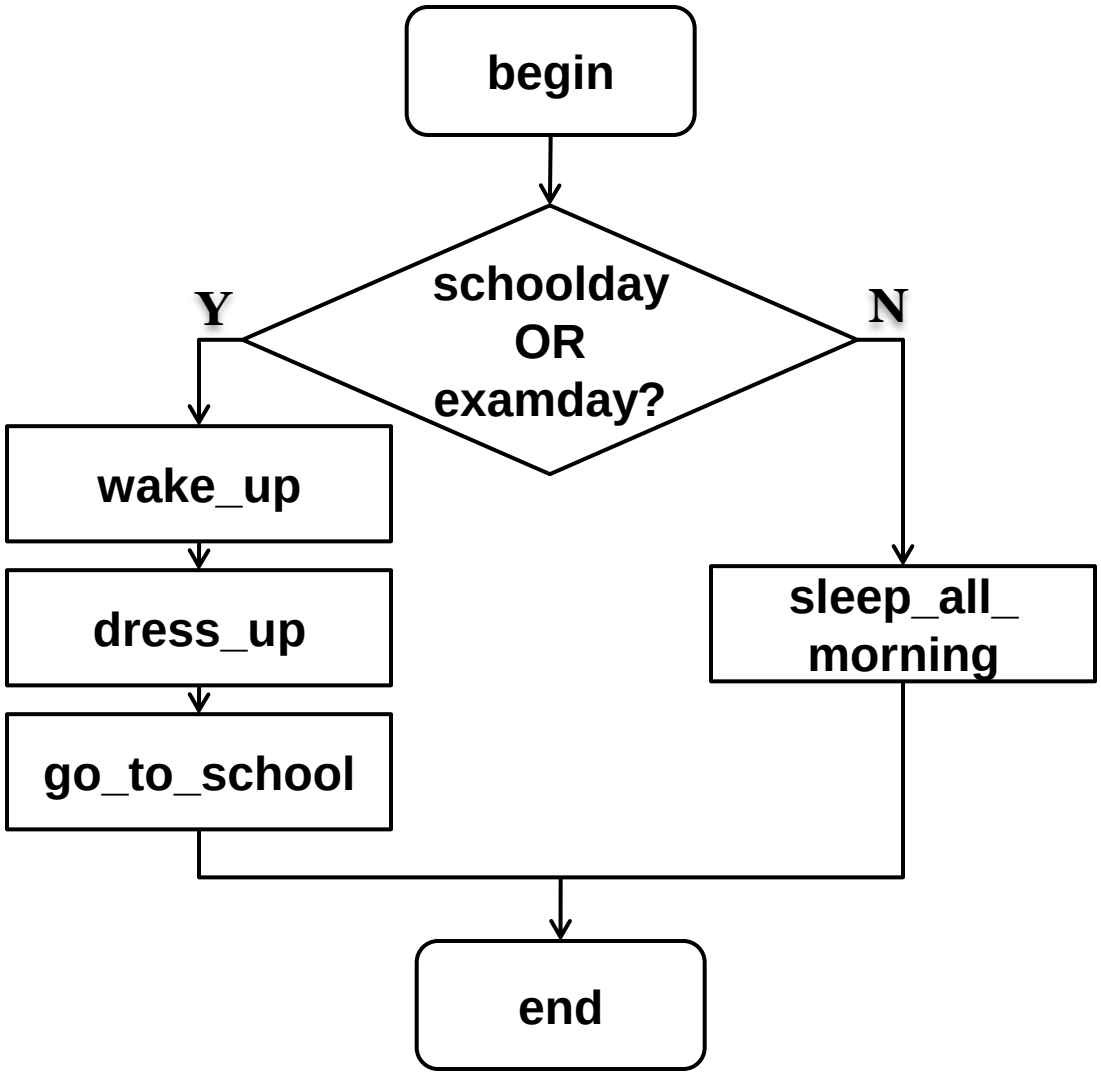
Conditions can be simple, or complex with the use of logical operators

- **!** Means **NOT**
  - ***(!A) is true*** if and only if ***A is false***
- **&&** means **AND**
  - ***(A && B) is true*** if and only if ***both A and B are true***
- **||** means **OR**
  - ***(A || B)*** is true if and only if ***at least one of A or B is true***

A	B	A && B	A || B
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

A	!A
T	F
F	T

```
if(schoolday || examday)
{
    wake_up;
    dress_up;
    go_to_school;
}
else
{
    sleep_all_morning;
}
```



Selection (or branching)

Problem 2: Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of three marks.

Pseudocode

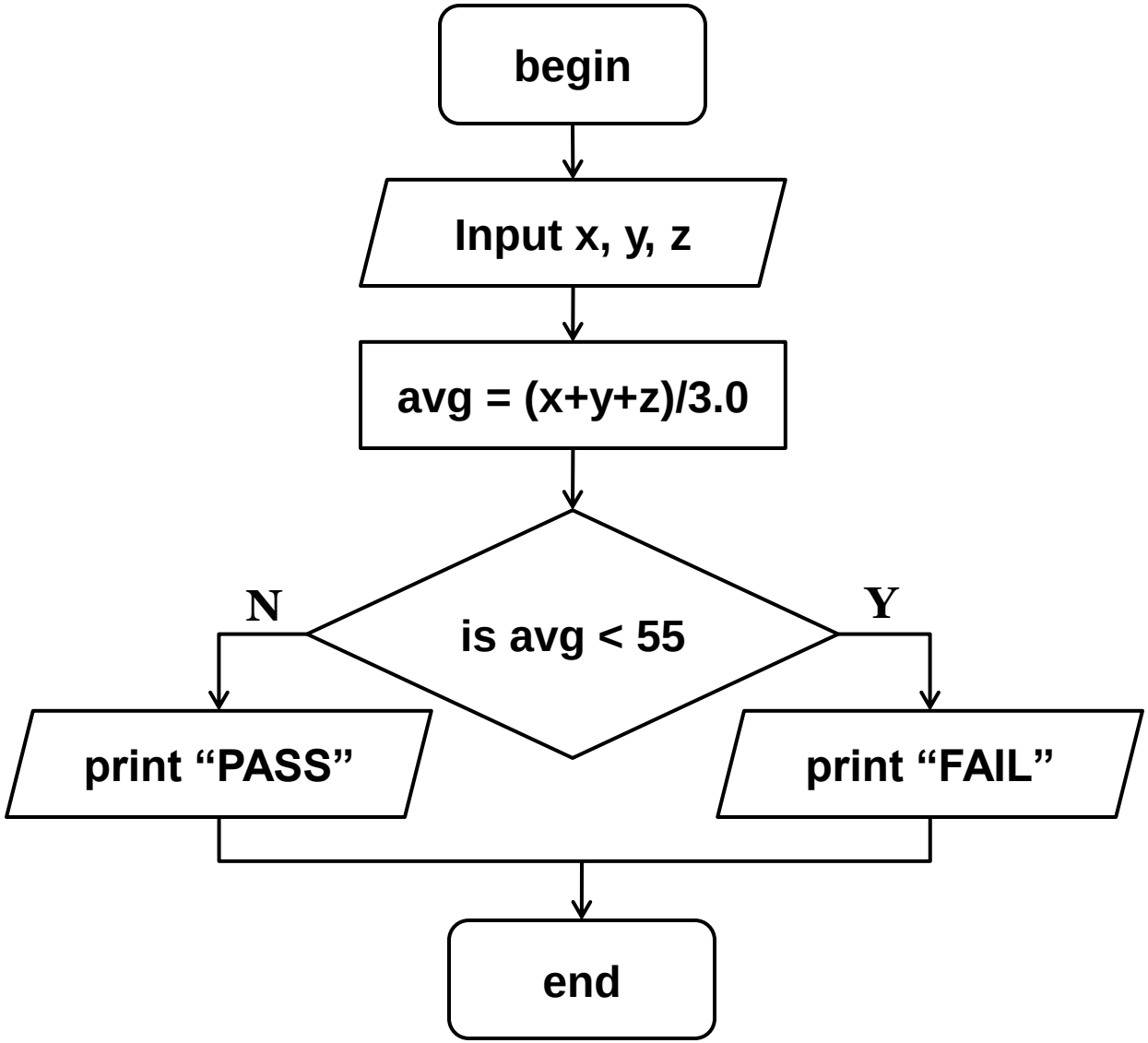
- Input a set of three marks
- Calculate their average by summing and dividing by 3
- if average is below 55

  Print “FAIL”

  else

  Print “PASS”

Flowchart


begin
Input x, y, z
avg = (x+y+z)/3.0
is avg < 55
N
Y
print “PASS”
print “FAIL”
end

Objectives

At the end of the meeting, students should be able to:

- Explain the concept of iteration
- Design algorithms for problems involving iteration

Sequence Control for Algorithms

Sequence

perform several steps in the given sequence

Selection (or branching)

perform some group of statements depending on some condition

Iteration (or looping)

perform some group of statements repeatedly

Functions

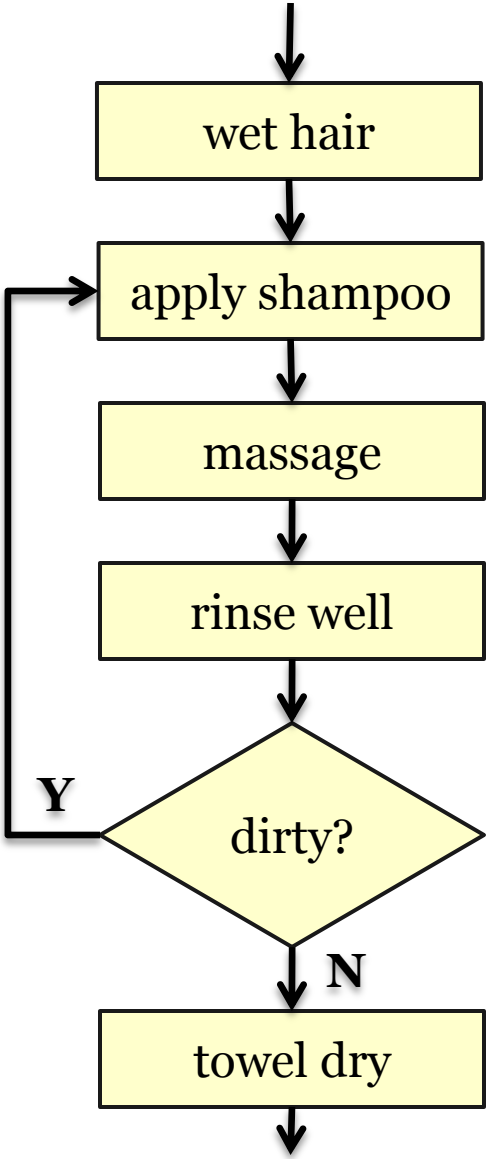
group together a block of instructions as a single, named logical unit

Iteration (or looping)

- Loops allow some block of statements to be performed repeatedly.

```
{ // instructions for a nerdy brand of shampoo
  wet_hair;
  do {
        apply_shampoo;
        massage_into_hair_and_scalp;
        rinse_well;
  } while ( dirty );       // repeat if necessary
  towel_dry;
}
```

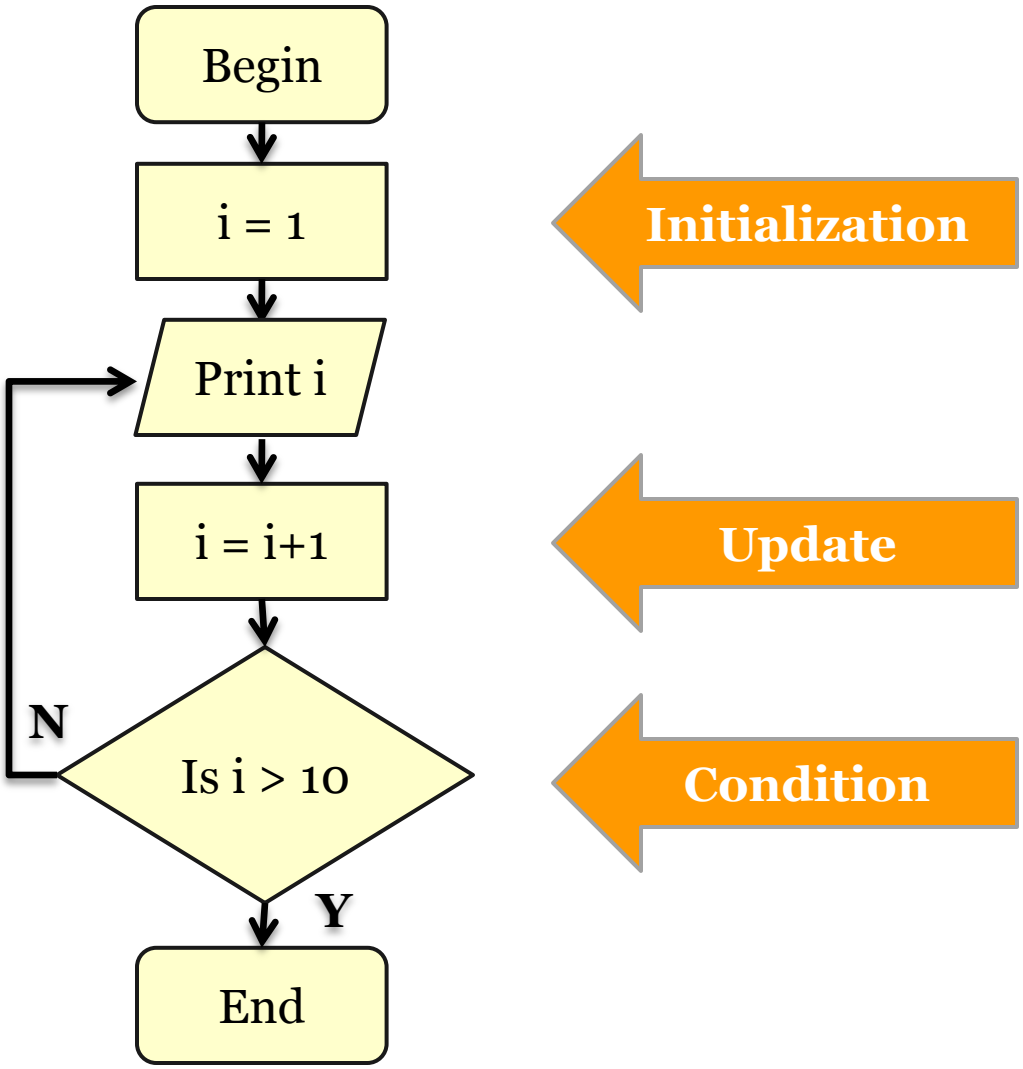
Iteration (or looping)



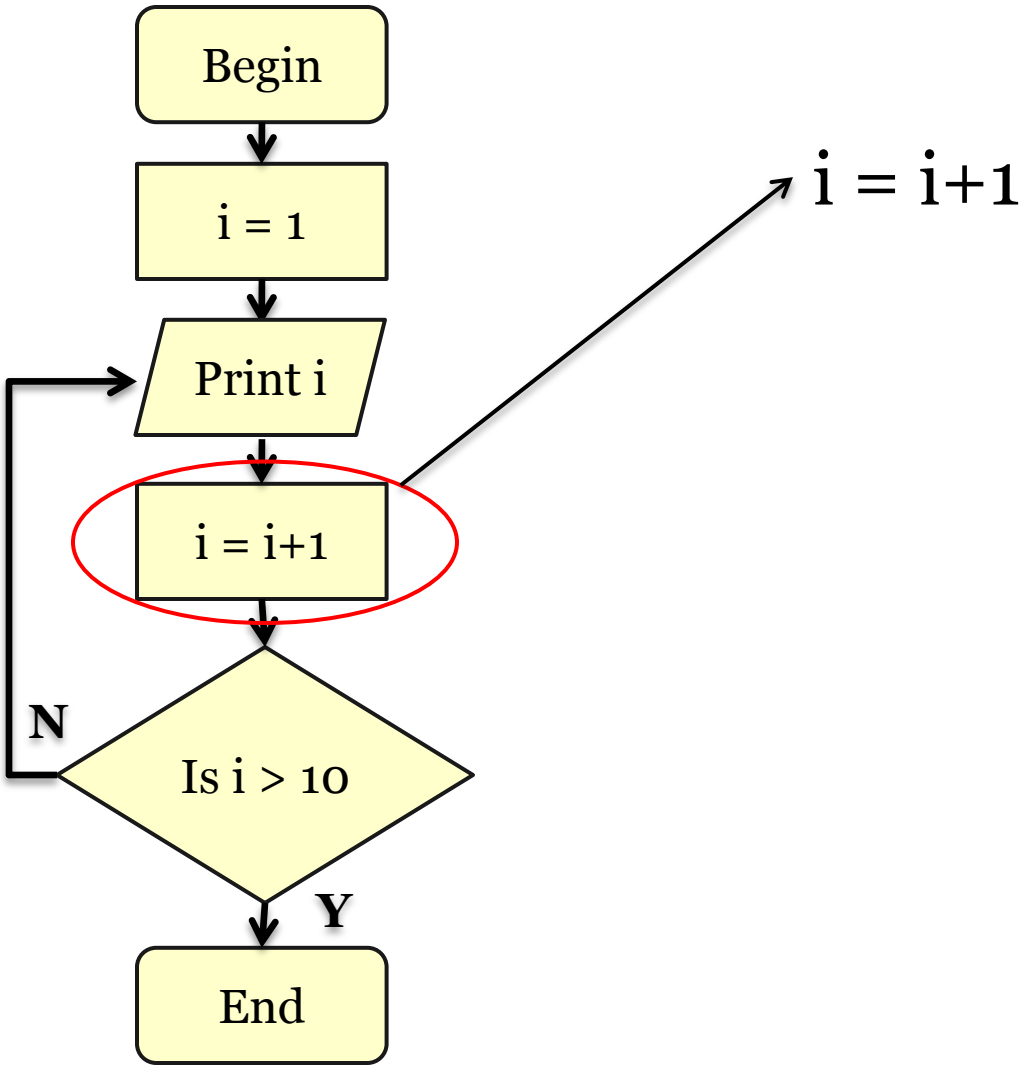
Iteration (or looping)

- ***Problem 1:*** Print numbers 1 to 10.

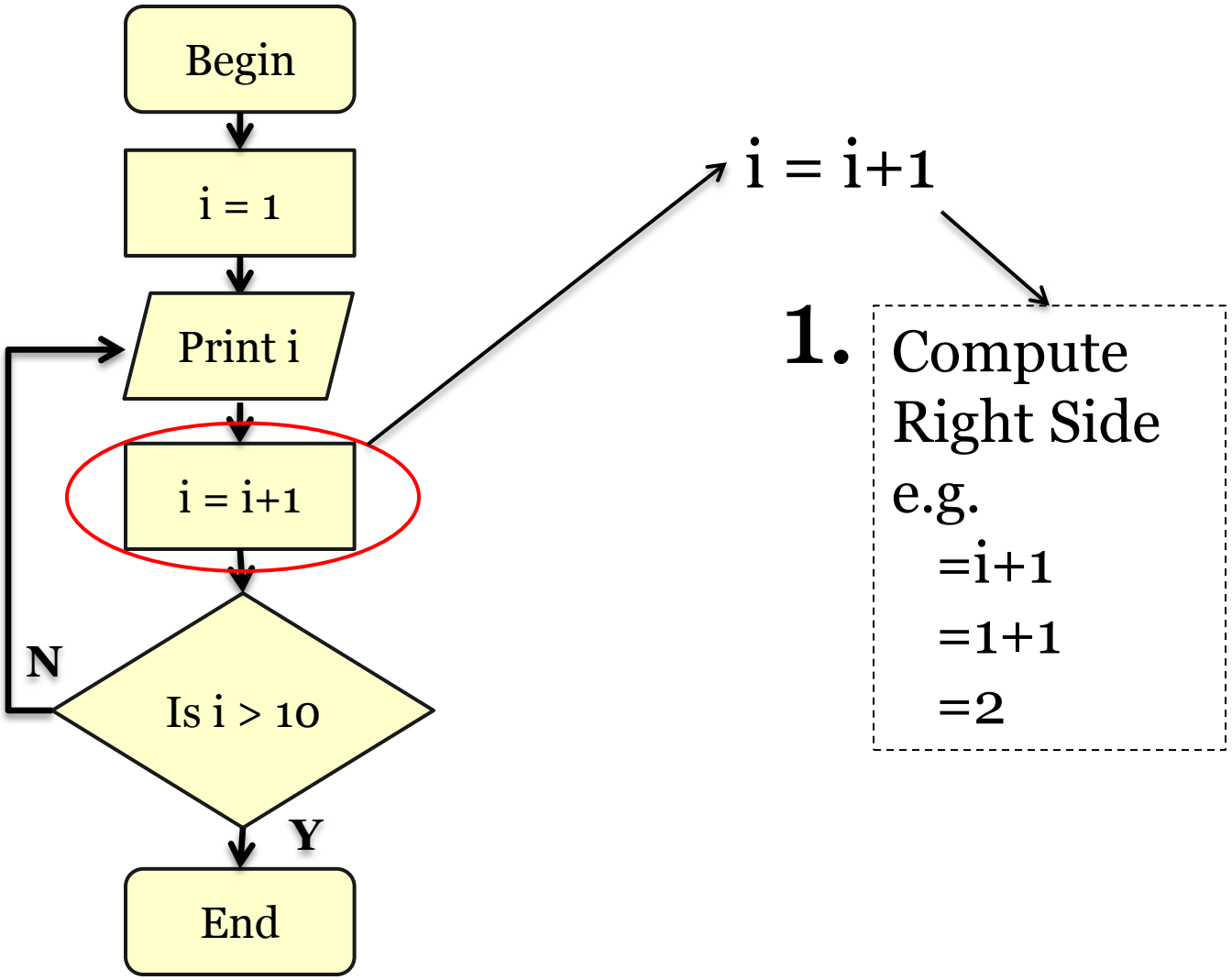
Types of Loops: Count Loops



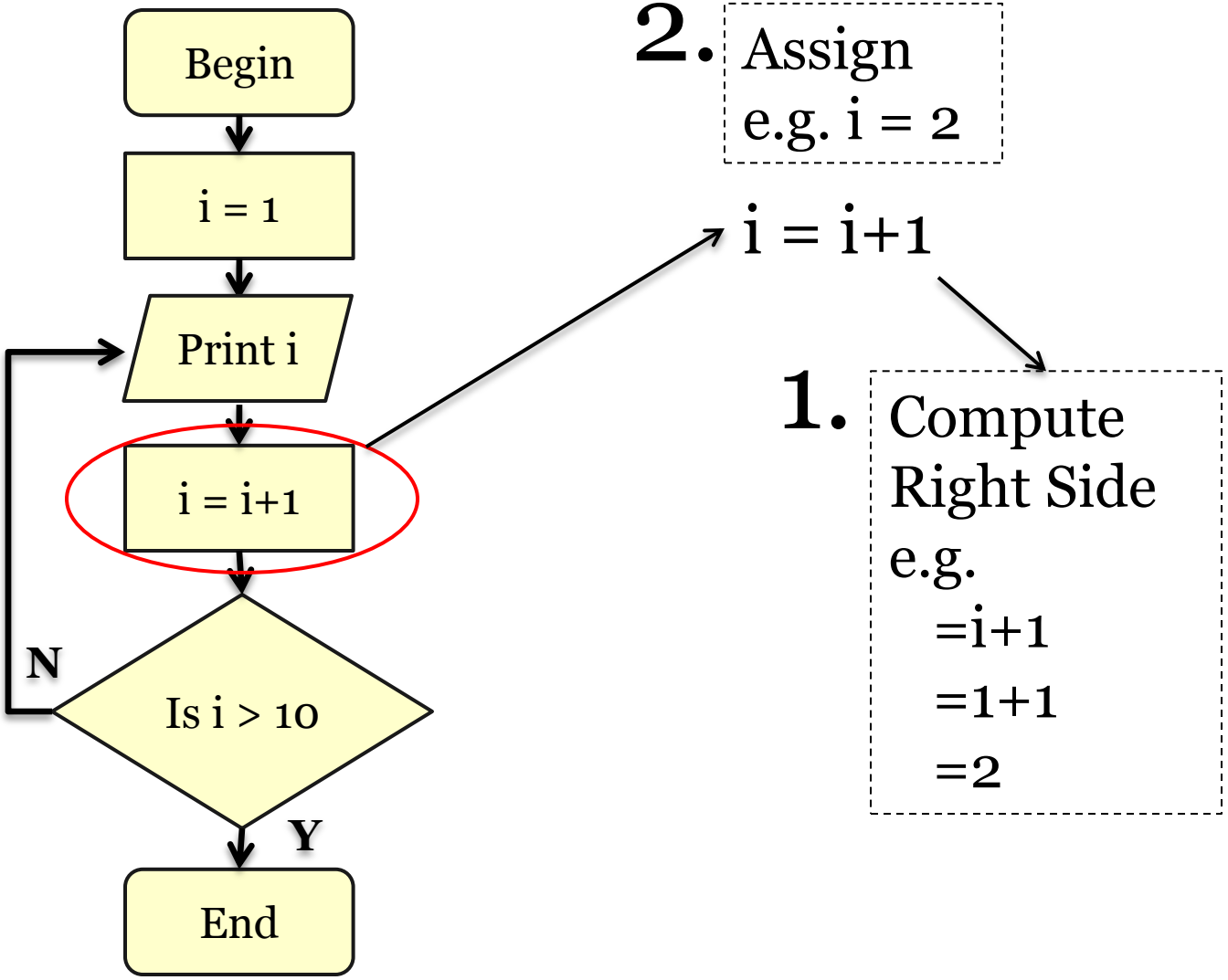
Types of Loops: Count Loops



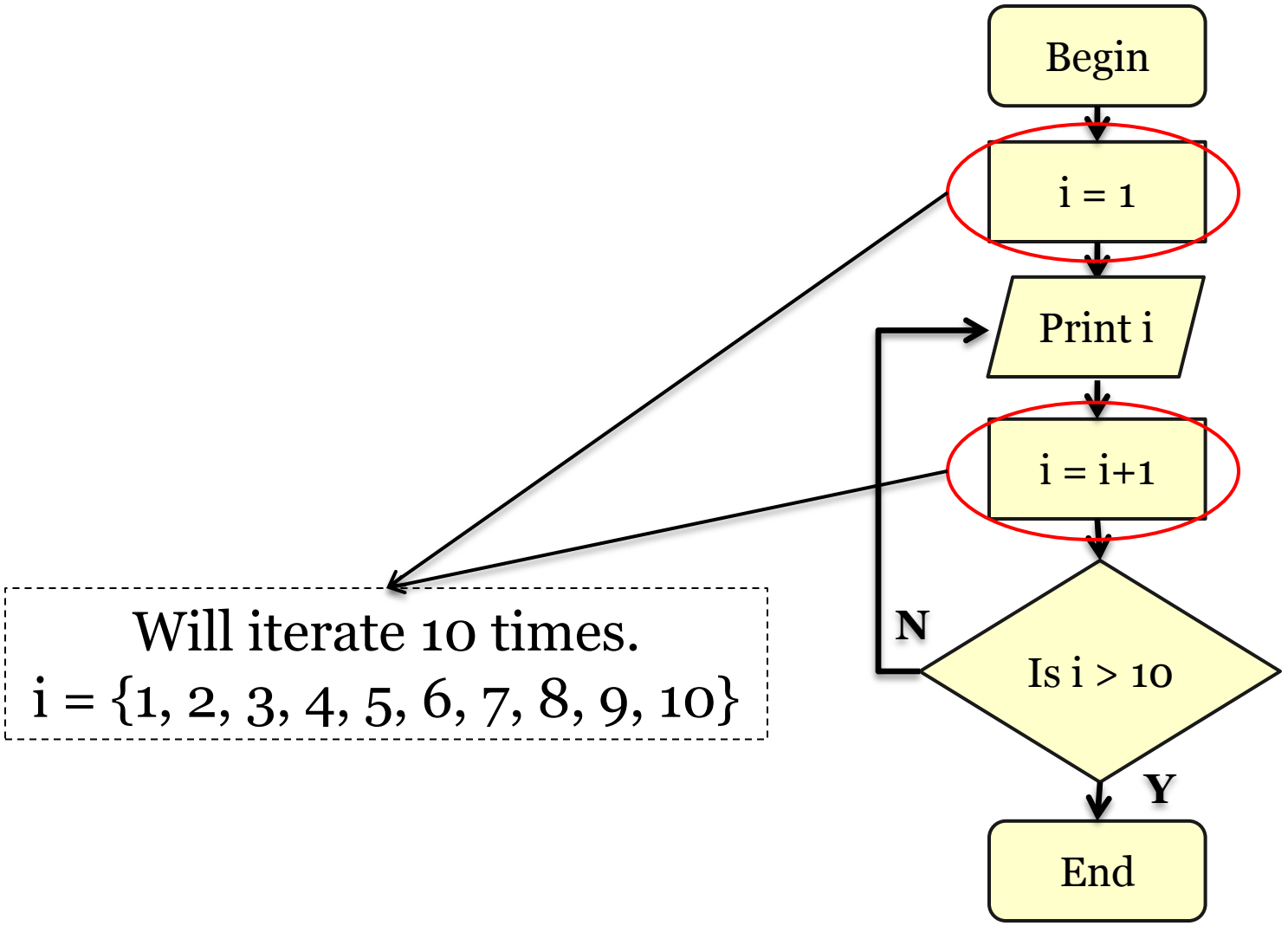
Types of Loops: Count Loops



Types of Loops: Count Loops

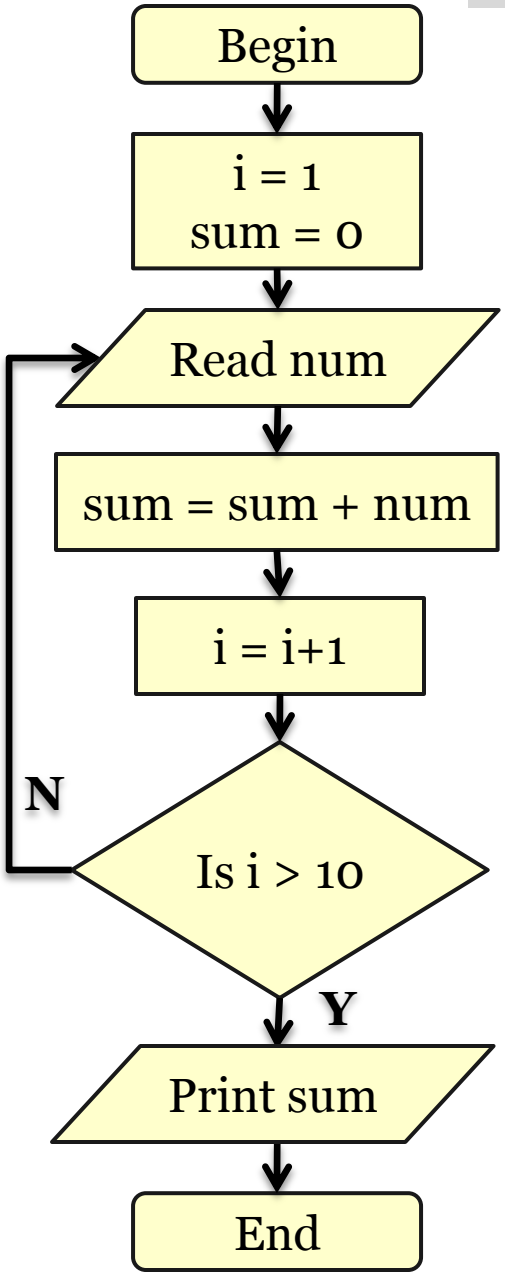


Types of Loops: Count Loops



Iteration (or looping)

- ***Problem 2:*** *Get the sum of 10 integers.*


Begin
i = 1
sum = 0
Read num
sum = sum + num
i = i+1
Is i > 10
N
Y
Print sum
End

Iteration (or looping)

- ***Problem 3:*** *Get the least common multiple of two integers a and b (smallest positive integer that is divisible by both a and b).*

Least Common Multiple

Example: What is the LCM of 6 and 8?

Multiples of 6 are:

6, 12, 18, 24, 30, …

Multiples of 8 are:

8, 16, 24, 32, 40, …

Least Common Multiple

Example: What is the LCM of 6 and 8?

Multiples of 6 are:

6

Multiples of 8 are:

8

Least Common Multiple

Example: What is the LCM of 6 and 8?

Multiples of 6 are:

6, 12

Multiples of 8 are:

8

Least Common Multiple

Example: What is the LCM of 6 and 8?

Multiples of 6 are:

6, 12

Multiples of 8 are:

8, 16

Least Common Multiple

Example: What is the LCM of 6 and 8?

Multiples of 6 are:

6, 12, 18

Multiples of 8 are:

8, 16

Least Common Multiple

Example: What is the LCM of 6 and 8?

Multiples of 6 are:

6, 12, 18

Multiples of 8 are:

8, 16, 24

Least Common Multiple

Example: What is the LCM of 6 and 8?

Multiples of 6 are:

6, 12, 18, 24

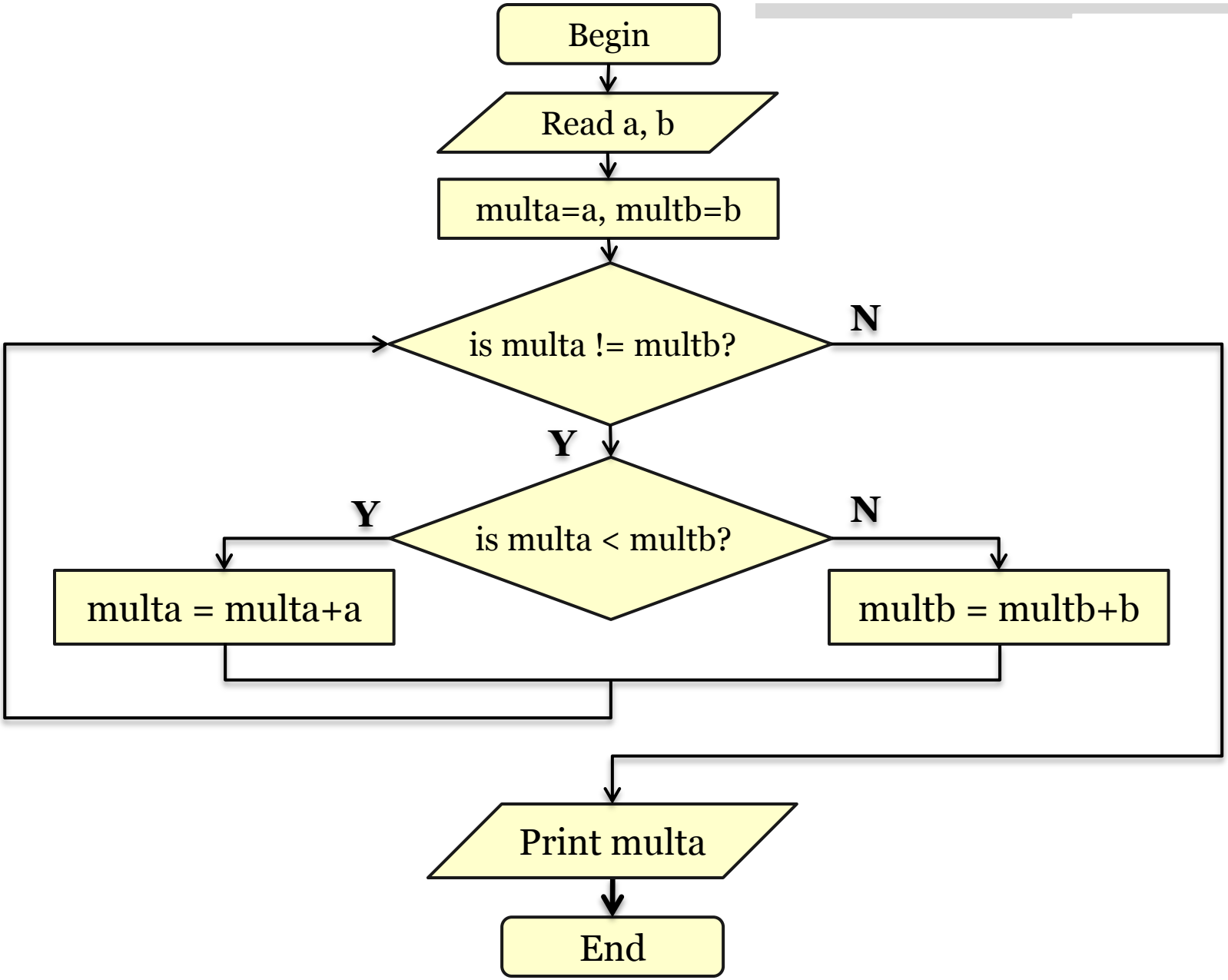
Multiples of 8 are:

8, 16, 24

Types of Loops: Conditional Loops

Start with *multa*=*a* and *multb*=*b*

```
while(multa != multb){
        if(multa < multb){
                multa = multa+a;
        }
        else{
                multb = multb+b;
        }
}
```


Begin
Read a, b
multa=a, multb=b
is multa != multb?
N
Y
is multa < multb?
Y
N
multa = multa+a
multb = multb+b
Print multa
End

Iteration (or looping)

- ***Problem 4:*** *Get the greatest common factor of two integers a and b (highest number that divides both a and b exactly).*

  (Hint: use the modulo operation)

Modulo Operation

- The modulo operation finds the remainder of division of one number by another.
- Example:

  $5 \bmod 2 = 1$

  $9 \bmod 3 = 0$

So What?

if $a \bmod b = 0$, then a is divisible by b

Greatest Common Factor

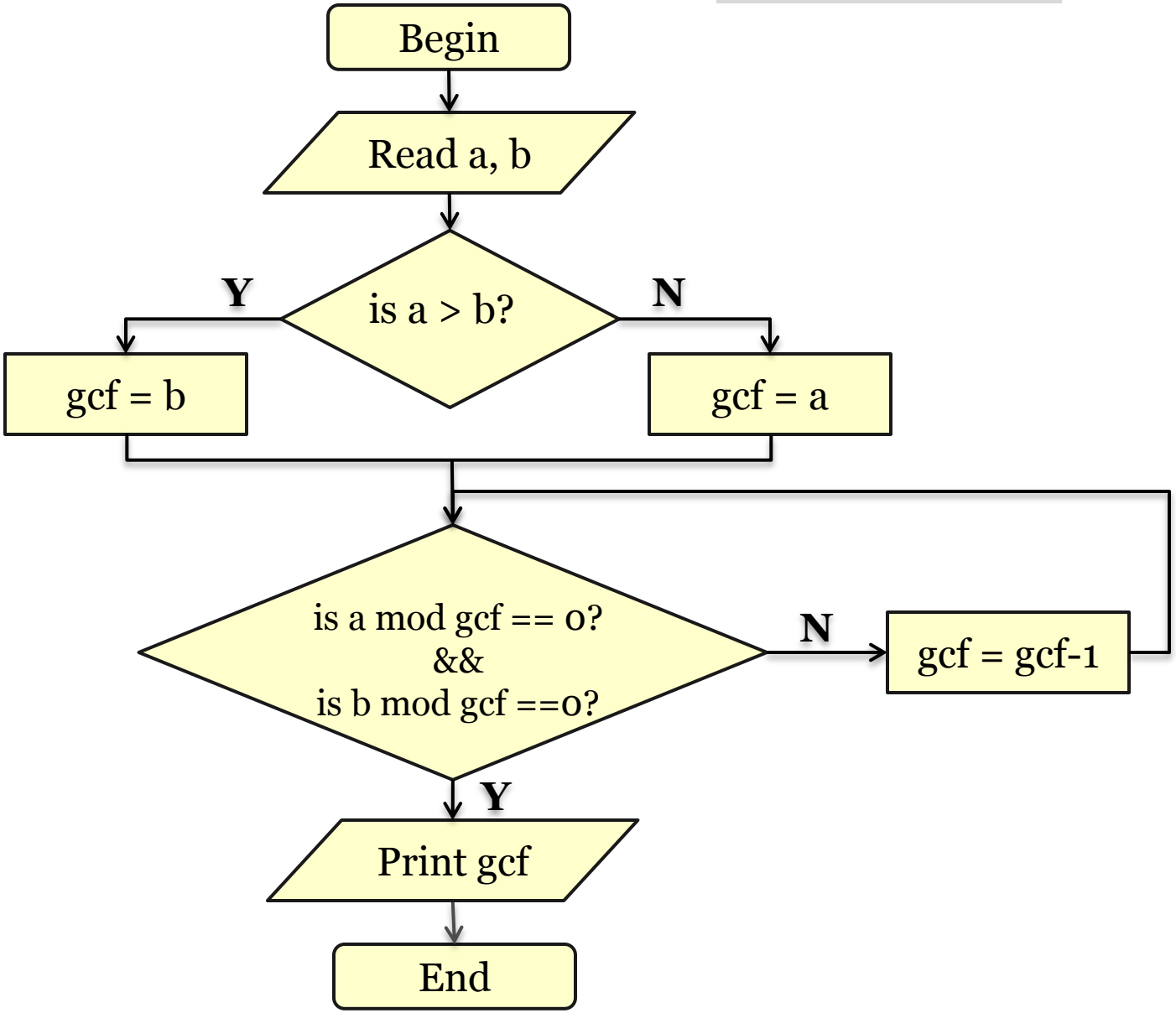
Example:
What is the GCF of 32 and 12?

Why start with 12 and not 32? 1? 2?

Is 32%12 equal to 0? **&&** 12%12 equal to 0? NO
Is 32%11 equal to 0? **&&** 12%11 equal to 0? NO
Is 32%10 equal to 0? **&&** 12%10 equal to 0? NO

…

When will the iteration stop?


Begin
Read a, b
is a > b?
Y
N
gcf = b
gcf = a
is a mod gcf == 0?
&&
is b mod gcf ==0?
N
gcf = gcf-1
Y
Print gcf
End

Problem #1: Start to End

Draw a flowchart which asks the user for a starting value and an ending value and then writes all the integers (inclusive) between those two values.

Example:

Start: 5
End: 9

5
6
7
8
9

Problem #2: Factorial of a Number

Draw a flowchart which computes *n*!.

n! = n*(n-1)*(n-2)*…*1

Example:

n = 5

5! = 5*4*3*2*1 = 120

Problem #3: Power of a Number

Draw a flowchart which computes a^b.
Assume: $a, b > 0$

$$a^b = a * a * a * \ldots * a$$

$$3^5 = 3*3*3*3*3 = 243$$

Problem #4: Adding up Squares

Draw a flowchart which adds up the squares of integers from 1 to *N*, where *N* is entered by the user:

Example:

N = 5

The sum of Squares is 55.

(1+4+9+16+25 = 55)

Problem #5: Average

Draw a flowchart which computes the average of N numbers.

Example: $N = 6$

4
5
2
6
3
4
Average: 4

Problem #6: Sum-the-Odds

Draw a flowchart which computes the sum of the odd numbers from 1 to *N*.

Example:

$N = 6$

Sum: $1+3+5 = 9$