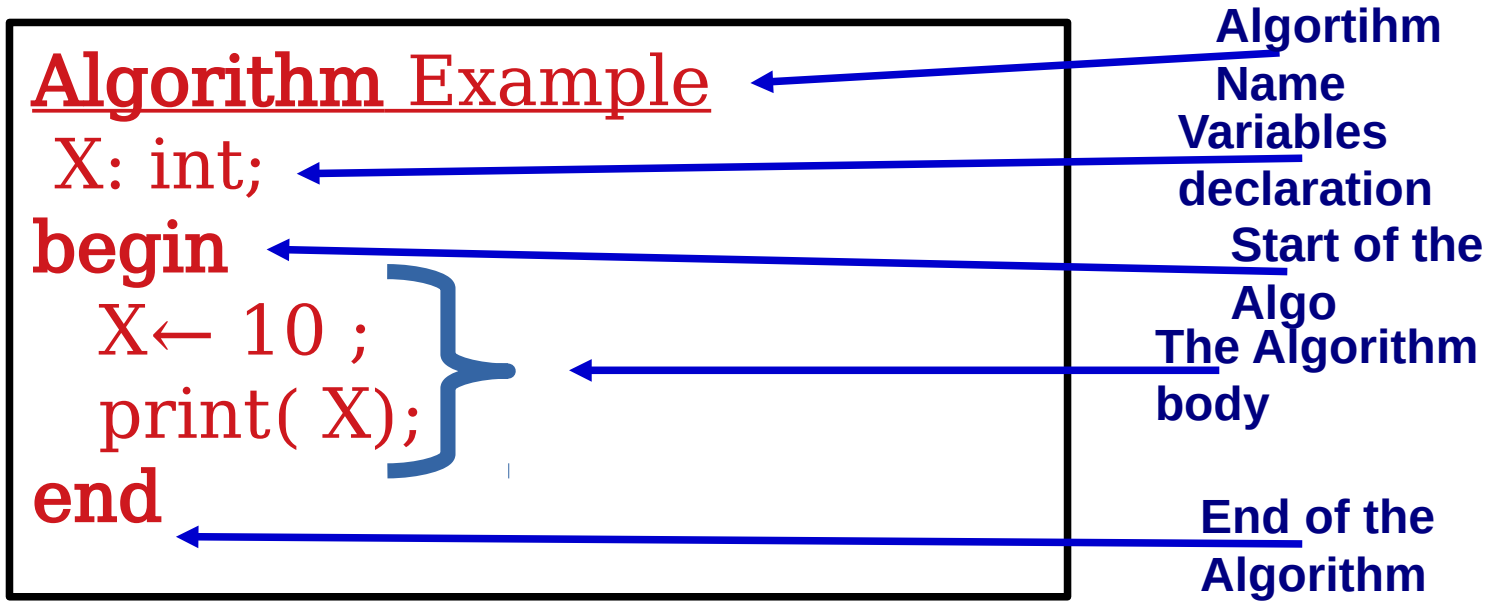
Algorithms and Data Structure 01

Dr. Sabri Ghazi
sabri.ghazi@univ-annaba.dz
Computer Science Department
Badji Mokhtar Annaba University

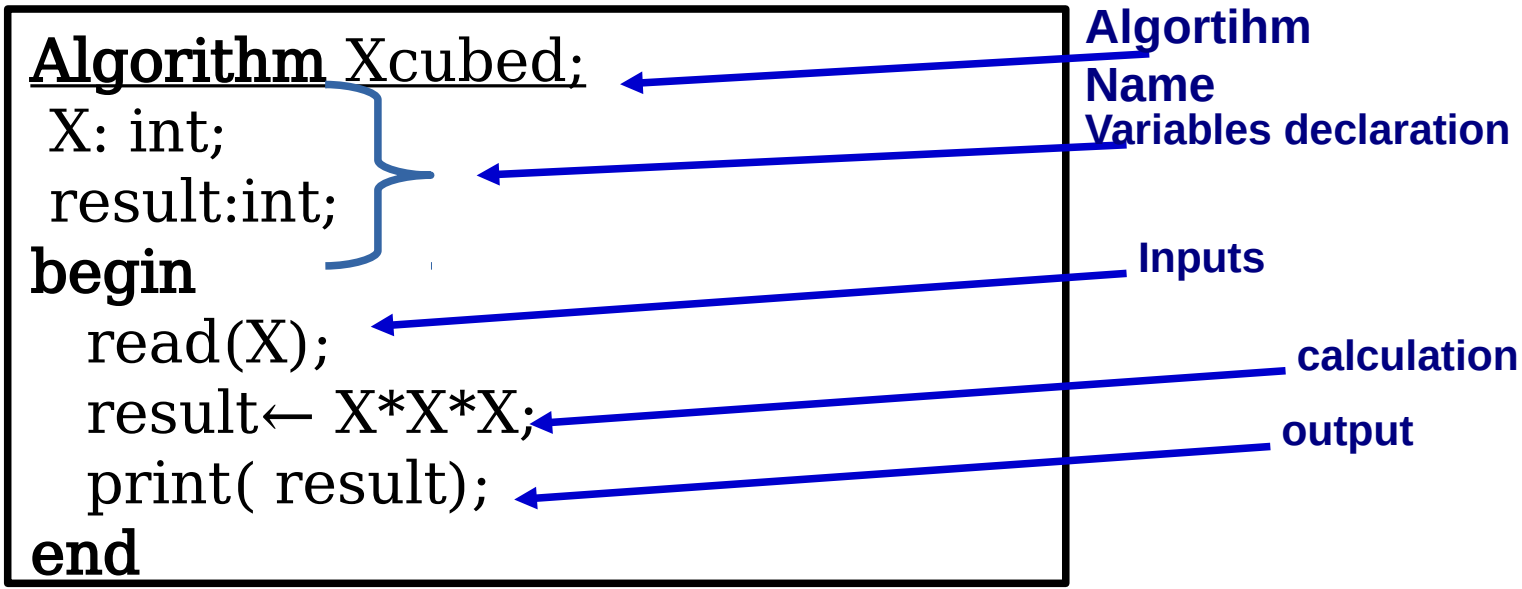
Syntax to write Algorithms

- When we write Algorithms we need to respect the following structure:



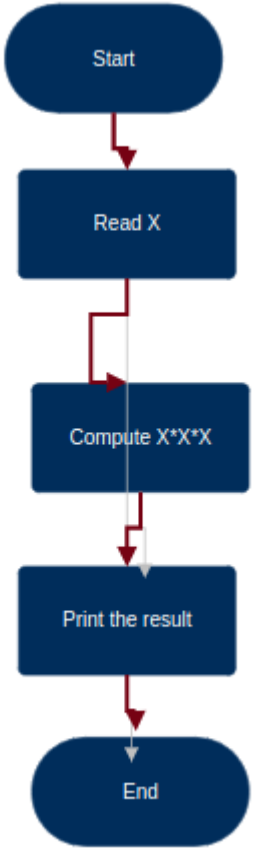
Example, Algorithm X cubed.

- Write an Algorihtm which compute the **X^3**.
  - First let us find the inputs and the outputs.
  - Then declare the needed variables.

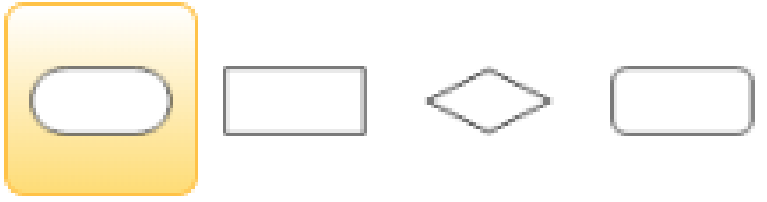


Algorithm Structure

An Algorithm can also be presented in form of **FlowChart**; where each instruction Is represented in rectangle from the starting point to the end.
And algorithm which compute X cubed, can be represented as :



Start and end are represented by :

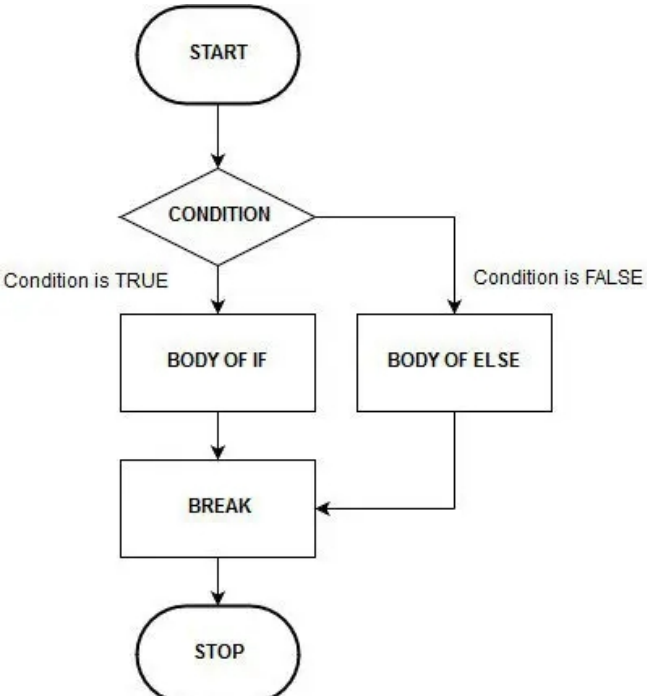


And instruction by :

Conditional Instruction

As we discussed previously in this course, all the algorithm instructions are executed
In sequence. In some cases, we can manipulate the execution by telling the computer
Which part of our algorithm to be executed and this according to specific condition.
If the condition is True we execute a bloc of instruction
If it is not true we execute another bloc.

```
Algorithm FindBigestNumber;
 X: int;
 Y: int;
begin
   read(X);
   read(Y)
   if( X > Y)
        Print(X);
   else print(Y);
end
```



Conditional Instruction

if (**logical Expression**)

*Block of Instructions to be executed if the expression is **True***

else

*Block of Instructions to be executed if the expression is **False***

Example of logical expression:

X > Y

(X*2) > 200

Age > 10 and size <30

Date<2000 or ville ==’Annaba’

Logical operators

Till now we used numeric data type and char, there is another data type nammed **Boolean**, this type can have only two possible values **True** or **False**

Let us examine this flowchart, as
you can see it is an algorithm
Who can tell me what this algorithm do ?

We need first to understand logic comparison ?

Operator	Description
&&	AND
| |	OR
!	NOT
!=	NOT EQUAL TO
&	BITWISE AND
|	BITWISE OR
^	BITWISE XOR
&=	AND EQUAL
|=	OR EQUAL
^=	XOR EQUAL

Truth Table

x	y	x and y
false	false	false
false	true	false
true	false	false
true	true	true

Logical and (&&)

x	y	x or y
false	false	false
false	true	true
true	false	true
true	true	true

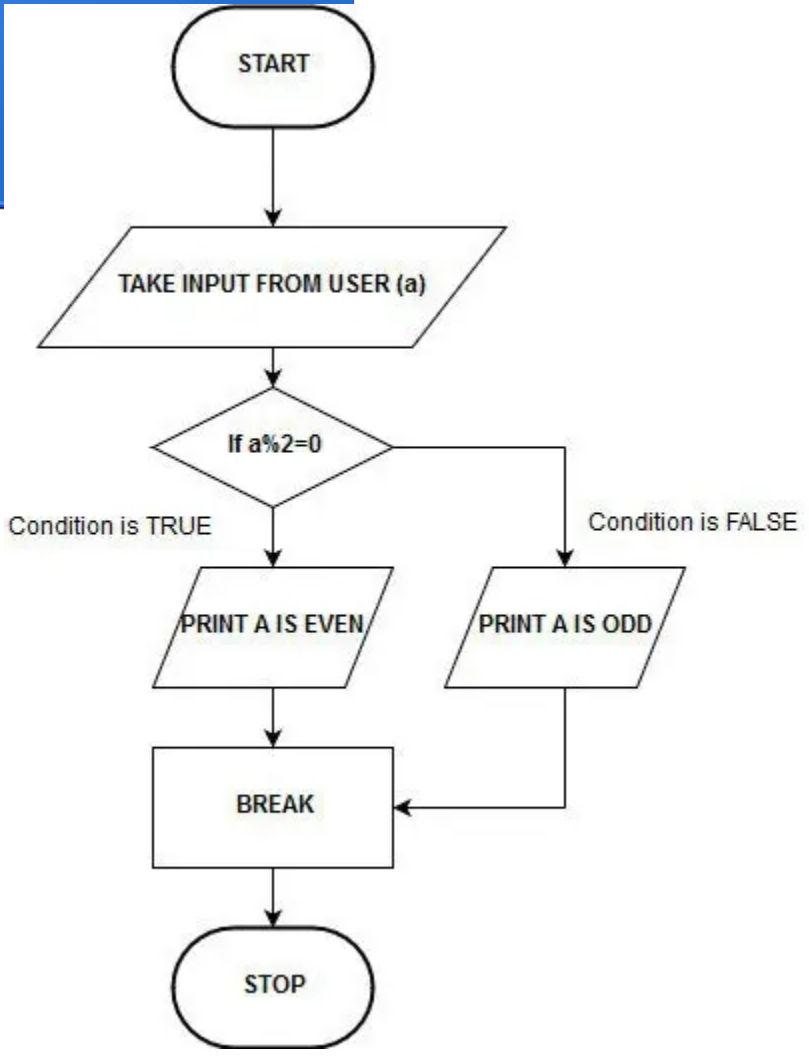
Logical or (||)

x	not x
false	true
true	false

Logical not (!)

Conditional Instruction demonstration

```
Algorithm OddEvenNumbers;
 X: int;
begin
   read(X);
   if( X % 2== 0)
        print("X is even number");
   else print("X is odd number");
end
```



Combining Conditional Instruction

if (**logical Expression1**)

*Instructions to be executed if the **expression1** is **True***

else if (**logical Expression2**)

*Instructions to be executed if the **expression1** is **False but Expression2 is True***

else if ((**logical Expression3**)

*Instructions to be executed if the **expression1** is **False and Expression2 is False but expression3 is True***

else

Instructions to be executed if all the expression are False !

Combining Conditional Instruction

Consider the following problem of shoes store is selling shoes according to a promotion :

- Buy a quantity less than 5 , the price is unchanged 250$ per unit.
- Buy more than 5 and less than 20, the price is 225$ per unit.
- Buy more than 20 and less than 40 , the price is 200$ per unit.
- Buy more than 40, the price is 180$ per unit.
-
- Knowing that a tax of 7% is applied to the final price, write an algorithm which helps with the sales.

Consider the following second order equation

$$A X^2 + B X + d = 0$$

Identify Input and outputs of this problem.
Write an algorithm that solve any variant of this equation.

NB: use determinant Delta= $B^2 - 4 A d$!

A case or switch statement

```
switch   ( variable)
{
case value1:
      Instructions;
        break;
case value1:
        instructions
        break;
case value1:
        instructions;
        break;
case value1:
        instructions;
        break;
Default:
        instructions;
       Break;
 }
```