



# Algorithm

**Abdallah Karakra**

Computer Science Department

Comp 132

# Algorithm & Pseudocode

- An algorithm is a procedure or formula for solving a problem.
- Pseudocode is a kind of structured English for describing algorithms. It allows the designer to focus on the logic of the algorithm without being distracted by details of language syntax.

# Example

- Let's say that you have a friend arriving at the airport, and your friend needs to get from the airport to your house. Here are three different algorithms that you might give your friend for getting to your home:



# Example Cont.

- **The taxi algorithm:**
  - Go to the taxi stand.
  - Get in a taxi.
  - Give the driver my address.



# Example Cont.

- **The call-me algorithm:**
  - When your plane arrives, call my cell phone.
  - Meet me outside baggage claim.



# Example Cont.

- **The bus algorithm:**
  - Outside baggage claim, catch bus number 70.
  - Transfer to bus 14 on Rukab Street.
  - Get off on Jerusalem street.
  - Walk two blocks north to my house.





# مقلوبة



# Common Action Keywords

- Input: READ , OBTAIN, GET
- Output: PRINT, DISPLAY, SHOW
- Compute: COMPUTE, CALCULATE
- Initialize: SET
- Add one: INCREMENT



# Types of Algorithm operations

Sequential

Conditional

Iterative

# Sequential

- Write an algorithm to find and print the sum of two integers ?
  1. Ask user to enter first integer
  2. Read the integer and save as integer\_1
  3. Ask user to enter the second integer
  4. Read second integer and save as integer\_2
  5. Add integer\_1 to integer\_2 and save result as sum
  6. Print sum to screen

```
"E:\C programs\Spring2015\Fisrt_Algorithm\bin\Release\Fisrt_Algorithm.exe"  
Please Enter the First Integer  
5  
Please Enter the Second Integer  
6  
Result is 11
```

# Sequential

- Write an algorithm to find and print the area of rectangle.
  1. Ask user to enter the height of rectangle.
  2. Read height and save as `rectangle_height`.
  3. Ask user to enter the width of rectangle.
  4. Read width and save as `rectangle_width`.
  5. Multiply `rectangle_height` by `rectangle_width` and save the result as `area`.
  6. Display `area`.

# Sequential

- Write an algorithm to reverse any two digits number.
  1. Ask user to enter two digits number.
  2. Read number and save as num.
  3. Divide num by ten and save result as tens.
  4. Divide num by ten and save remainder as rem.
  5. Multiply rem by ten and save the result as rev.
  6. Add tens to rev.
  7. Print rev.

```
Suppose num=12
tens=num /10 =12/10→tens=1
rem=num%10=12%10→rem=2
rev=rem*10=2*10→rev=20
rev=rev+tens=20+1→rev=21
```

# Conditional

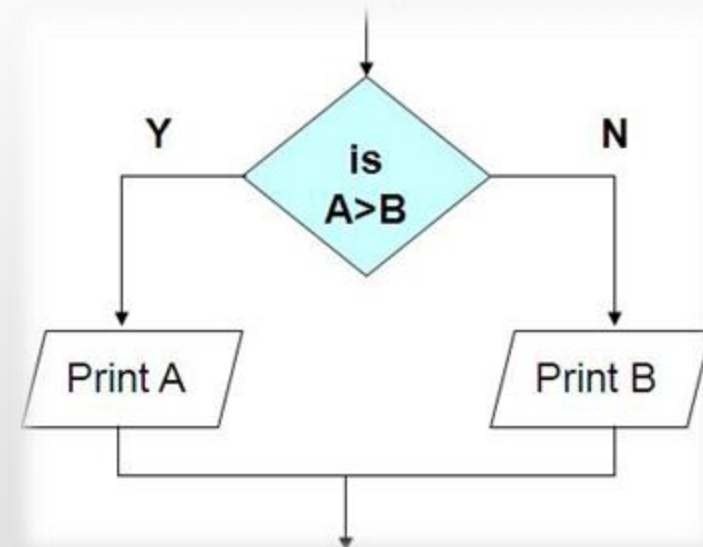
- Selection logic
- Case

# Conditional

- Ask questions and choose alternative actions based on the answers.

## Example

```
if A is greater than B then  
    print A  
else  
    print B  
end if
```



# Conditional

**ELSE** keyword is **optional**

```
IF condition THEN
    Sequence
END IF
```

```
IF condition THEN
    Sequence 1
ELSE IF condition THEN
    Sequence 2
ELSE IF condition THEN
    Sequence 3
ELSE
    Sequence 4
END IF
```



# Conditional

## Logical Operators :

- AND
- OR

## Relational Operators :

- Greater than
- Greater than or equal
- Less than
- Less than or equal
- Equal
- Not Equal

# Conditional

Write an algorithm to print **passed** or **failed** based on the student grade.

1. Ask user to enter student grade.
2. Read grade and save as `student_grade`.
3. If `student_grade` greater than or equal sixty then  
    print "passed"  
    else  
        print "failed"  
    end if

# Conditional

Write an algorithm to find and print the maximum element of a set of 3 integers.

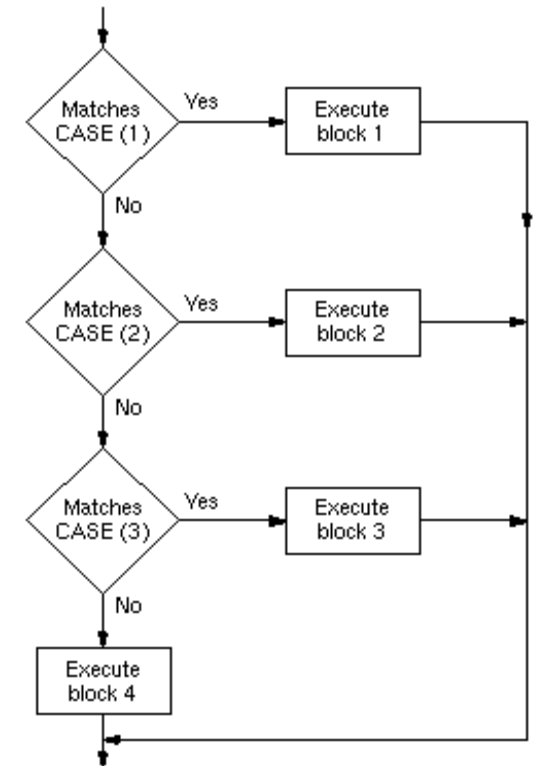
1. Ask user to enter the first integer.
2. Read number and save as first\_integer.
3. Ask user to enter the second integer.
4. Read number and save as second\_integer.
5. Ask user to enter third integer.
6. Read number and save as third\_integer.
7. Let max equal to the first\_integer.
8. If max less than second\_integer then  
    set max to second\_integer  
end if
9. If max less than third\_integer then  
    set max to third\_integer  
end if
10. Print “the maximum integer is” max

# Conditional

- CASE: multi way branch based on conditions that are mutually exclusive.

```
CASE expression OF
  Condition 1: sequence 1
  Condition 2: sequence 2
  ..... : .....
  Condition n: sequence n
OTHERS:
  default sequence
END CASE
```

```
SELECT CASE (TEST 2)
CASE (1)
  block 1
CASE (2)
  block 2
CASE (3)
  block 3
CASE DEFAULT
  block 4
END SELECT
```



# Conditional

Write an algorithm to find and print the smallest of three given numbers (**assume all numbers are different**).

1. Ask user to enter first number
2. Read number and save as num1
3. Ask user to enter second number
4. Read number and save as num2
5. Ask user to enter third number
6. Read number and save as num3
7. **If** num1 **less than** num2 **and** num1 **less than** num3 **then**  
    print num1 "is the smallest"  
**else if** num2 **less than** num1 **and** num2 **less than** num3 **then**  
    print num2 "is the smallest "  
**else**  
    print num3 "is the smallest "  
**end if**

Rules for logical **And** operations

T	T	T
T	F	F
F	T	F
F	F	F

# Conditional

**Write an algorithm to read a number x and display its sign.**

1. Ask user to enter a number
2. Read number and save as X
3. If x is **greater than** zero **then**  
    print x “is positive”  
    else if x is **equal** zero **then**  
        print x “is zero”  
    else  
        print x “is negative”  
**end if**

# Conditional

Write an algorithm that will input student average. If the average is greater than or equal to 60 and less than or equal to 70, the algorithm should display “Passed”. If it is greater than 70 and less than or equal to 80, print “Good”. If it is greater than 80 and less than 90, print “Very good”. If it is greater than or equal 90 , print “Excellent”. If it is less than 60 the prints “Fail”.



# Conditional

1. Ask user to enter student average
2. Read average and save as ag
3. If ag is **greater than or equal to sixty** and ag is **less than or equal to seventy** then  
    print "Pass"  
    else if ag is **greater than seventy** and ag is **less than or equal to eighty** then  
    print "Good"  
    else if ag is **greater than eighty** and ag is **less than ninety** then  
    print "Very good"  
    else if ag is **greater than or equal to ninety** then  
    print "Excellent"  
    else  
    print "Fail"  
    end if

Rules for logical <b>OR</b> operations		
T	T	T
T	F	T
F	T	T
F	F	F

# Iterative

- Perform “looping” behavior; repeating actions until a continuation condition becomes false

(1)

**WHILE** condition  
sequence  
**END WHILE**

(2)

**REPEAT**  
sequence  
**UNTILE** condition

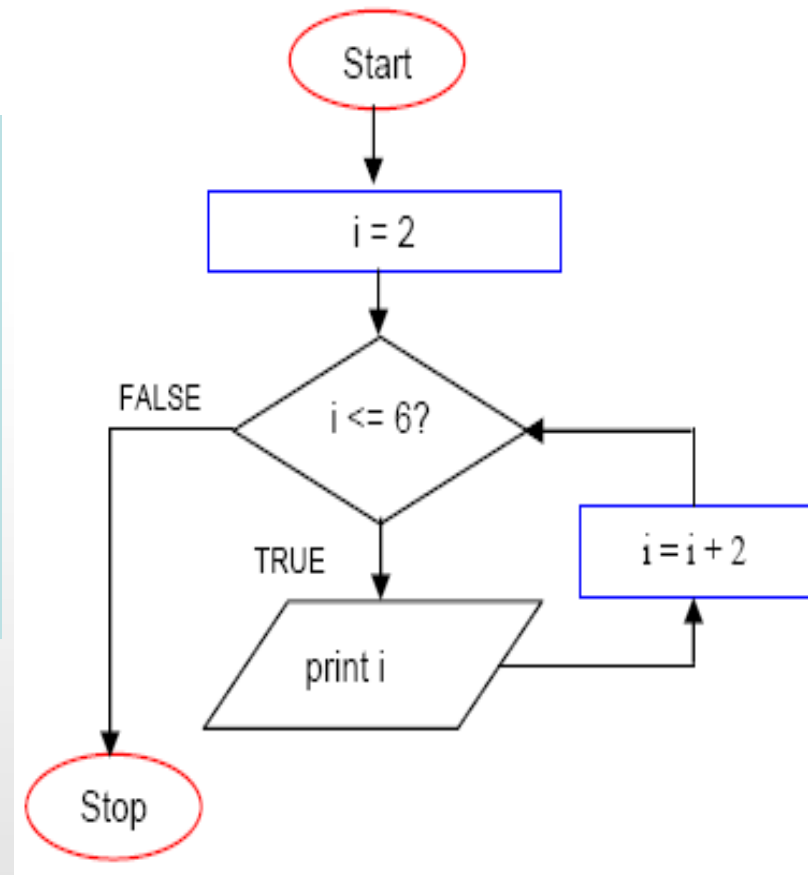
(3)

**FOR** iteration bounds  
sequence  
**END FOR**

# Iterative

1. Set  $i$  equal to two
2. While  $i$  less than or equal six  
    print  $i$   
    add two to  $i$   
end while

Output:  
2 4 6

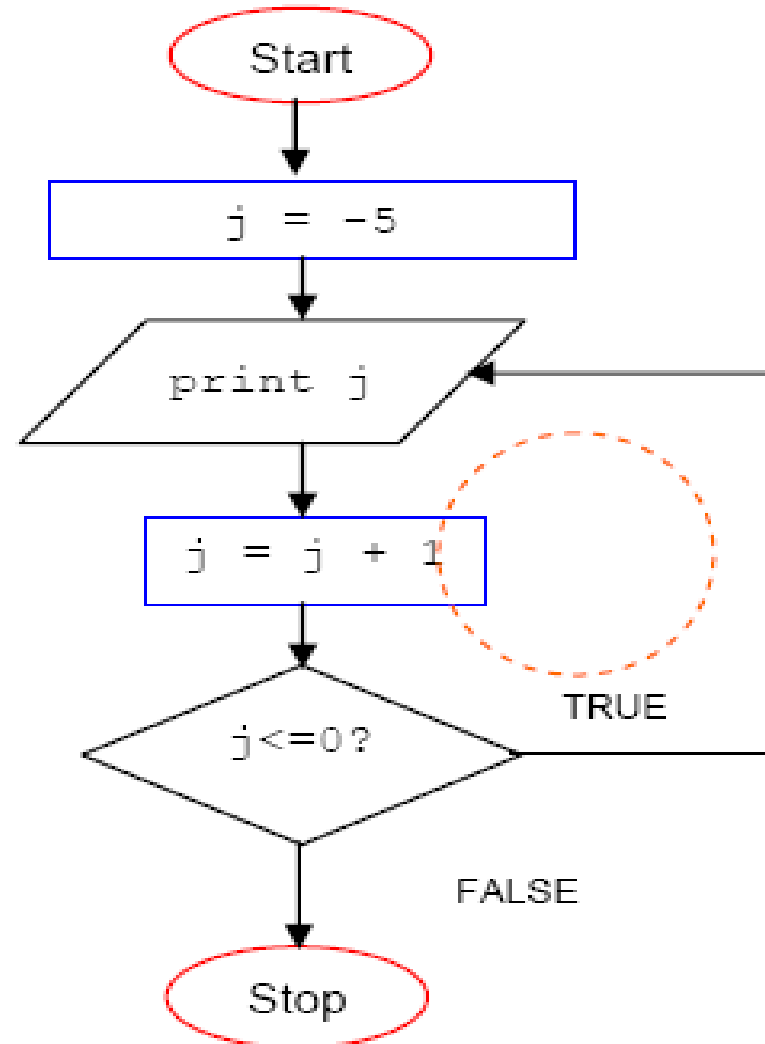


# Iterative

1. Set  $j$  equal to negative five
2. Repeat
  - print  $j$
  - increment  $j$until  $j$  less than or equal to zero

Output:

-5 -4 -3 -2 -1 0



# Iterative

- Write an algorithm to calculate the average of a set of 10 students.

## Solution 1

1. Set counter to **zero**
2. Set total to zero
3. While counter **is less than ten**  
    Ask user to enter grade  
    Read grade and save as gd  
    Add the gd into the total  
    increment counter  
end while
4. Set the average to the total **divided by counter**
5. Print “the average is ” average

## Solution 2

1. Set counter to **one**
2. Set total to zero
3. While counter **is less than or equal ten**  
    Ask user to enter grade  
    Read grade and save as gd  
    Add the gd into the total  
    increment counter  
end while
4. Set the average to the total **divided by 10**
5. Print “the average is ” average

# Iterative

Write an algorithm that will count the number of student pass in a class and the amount failed. The pass mark is more than or equal to 65. Suppose the number of students are 30 . The algorithm should output the amount fail and passed.

# Iterative

1. Set counter to zero
2. Set numberOfStudents to thirty
3. Set passCounter to zero
4. Set failureCounter to zero
5. While counter less than numberOfStudents  
    Ask user to enter student average  
    Read average and save as ag  
    if ag greater than or equal sixty five then  
        increment passCounter  
    else  
        increment failureCounter  
    end if  
    increment counter  
end while

Message

Value

6. Print "pass counter =" **passCounter** "and failure counter =" **failureCounter**



# *Extra Exercises*

1. Write an algorithm that takes 20 integers and decides and prints the number of integers divisible by 3 and the number of integers not divisible by 3.
2. Write an algorithm that will accept the values of the sides of a square and display its area where the formula is :  $\text{area} = \text{side} * \text{side}$

# *Extra Exercises*

Solutions:

[Extra Exercises\\_1.txt](#)

[Extra Exercises\\_2.txt](#)