



# Unit II – Part 02 (Solved Problems)



Lecture Slide

AS2023





# Objectives



By the end of this session, students will be able to:

- Understand algorithm, flowchart and pseudocode
- Write algorithms
- Draw flowcharts
- Write pseudocodes
- Translate algorithms into flowchart and pseudocode



# Example 1



**Problem:** Write an algorithm to compute the sum of two numbers. Translate your algorithm into flowchart and pseudocode.

*Prior to writing algorithms, we need to identify the inputs and outputs for the given problem.*

**Input:** Lets consider  $A$  &  $B$  (*either run time or compilation initialization can be adopted*)

**Output:** the sum of  $A$  &  $B$



# Example 1 (cont..)



## Algorithm:

Step 0: START

Step 1: Read the values of A & B (compile time initialization)

Step 2: Compute the sum of  $A+B$

Step 3: Display the sum

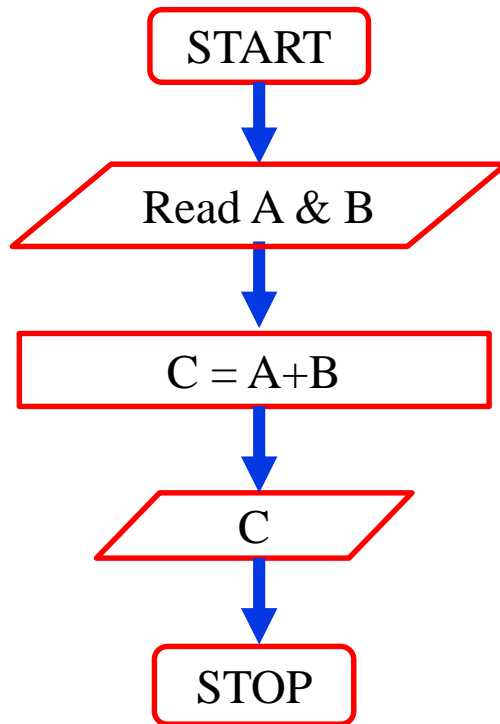
Step 4: STOP



# Example 1 (cont..)



## Flowchart:



## Algorithm:

Step 0: START

Step 1: Read the values of A & B (compile time initialization)

Step 2: Compute the sum of A+B

Step 3: Display the sum

Step 4: STOP



# Example 1 (cont..)



## Pseudocode:

```
READ  A,  B
COMPUTE  C  =  A+B
DISPLAY  C
```

### Algorithm:

Step 0: START

Step 1: Read the values of A & B  
(compile time initialization)

Step 2: Compute the sum of A+B

Step 3: Display the sum

Step 4: STOP

### Keywords:

**Input:** READ, OBTAIN, GET &  
PROMPT

**Output:** PRINT, DISPLAY, and SHOW

**Compute:** COMPUTE, CALCULATE,  
DETERMINE

**Initialize:** SET, INITIALIZE

**Add one:** INCREMENT



# Example 2



**Problem:** Write an algorithm to check whether a number is even or odd. Draw flowchart and write pseudocode for your algorithm.

*Prior to writing algorithms, we need to identify the inputs and outputs for the given problem.*

**Input:** Lets consider A (*either run time or compilation initialization can be adopted*)

**Output:** It should print either *even* or *odd*



# Example 2 (cont..)



## Algorithm:

Step 0: START

Step 1: Get the number from the user (runtime initialization)

Step 2: Compute Remainder ( $R = A \% 2$ )

Step 3: If R is equal 0, then print “the number is even  
otherwise print “ the number is odd”

Step 4: STOP

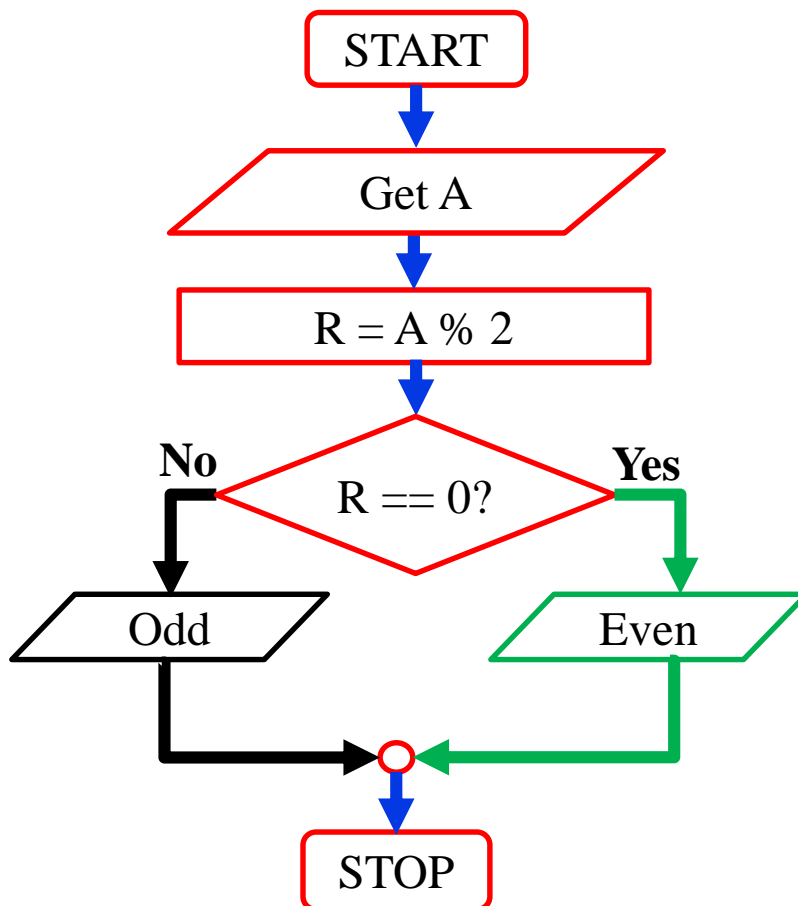




# Example 2 (cont..)



## Flowchart:



## Algorithm:

Step 0: START

Step 1: Get the number from the user  
(runtime initialization)

Step 2: Compute Remainder ( $R = A \% 2$ )

Step 3: If R is equal 0, then print “the  
number is even

otherwise print “ the number is  
odd”

Step 4: STOP



# Example 2 (cont..)



## Pseudocode:

```
PROMPT user enter A
CALCULATE  $R = A \% 2$ 
IF (R==0) :
    THEN DISPLAY "even"
    ELSE DISPLAY "Odd"
ENDIF
```

## Algorithm:

Step 0: START

Step 1: Get the number from the user  
(runtime initialization)

Step 2: Compute Remainder ( $R = A \% 2$ )

Step 3: If R is equal 0, then print "the  
number is even

otherwise print " the number is  
odd"

Step 4: STOP



# Example 3



**Problem:** Write an algorithm to print first  $N$  whole numbers. Draw flowchart and write pseudocode for your algorithm.

*Prior to writing algorithms, we need to identify the inputs and outputs for the given problem.*

**Input:** Lets consider  $N$  (*either run time or compilation initialization can be adopted*)

**Output:** print all first  $N$  whole numbers ( $0, 1, 2, \dots, N-1$ )



# Example 3 (cont..)



## Algorithm:

Step 0: START

Step 1: Enter the value of  $N$

Step 2: set the initial value of  $i$  to 0

Step 3: Print  $i$

Step 4: Increase the value of  $i$  by 1

Step 5: if  $i$  is less than  $N$ , then repeat from **step 3 and 4**  
otherwise go the **next step**

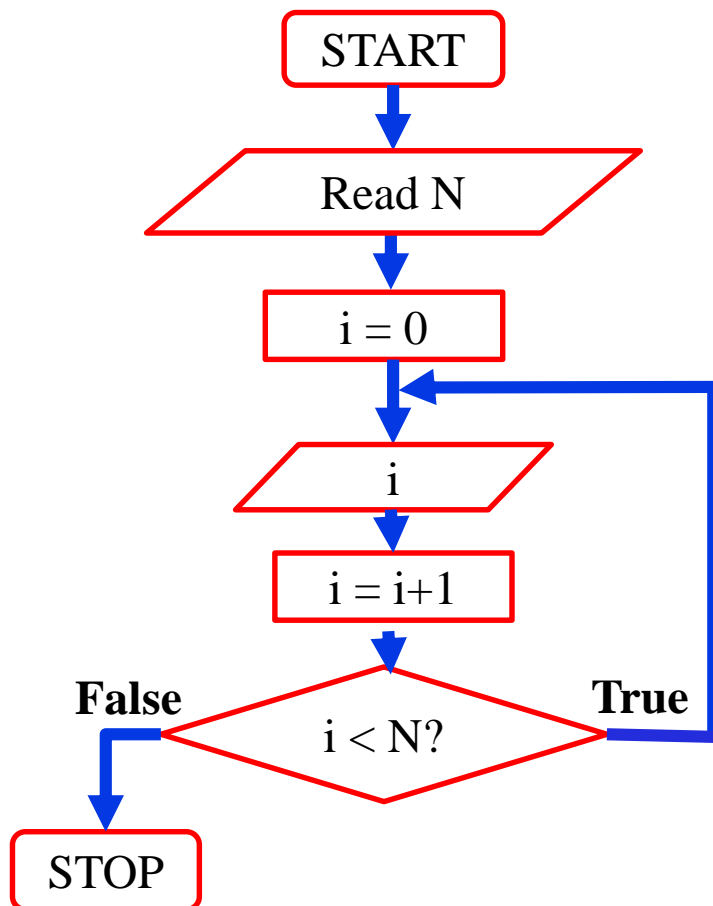
Step 6: STOP



# Example 3 (cont..)



## Flowchart:



## Algorithm:

Step 0: START

Step 1: Enter the value of N

Step 2: set the initial value of i to 0

Step 3: Print i

Step 4: Increase the value of i by 1

Step 5: if i is less than N, then repeat from step 3

otherwise go the next step

Step 6: STOP



# Example 3 (cont..)



## Pseudocode:

GET N

SET  $i = 0$

DO :

    SHOW  $i$

    INCREMENT  $i$  by 1

WHILE ( $i < N$ )

ENDDOWHILE

## Algorithm:

Step 0: START

Step 1: Enter the value of N

Step 2: set the initial value of  $i$  to 0

Step 3: Print  $i$

Step 4: Increase the value of  $i$  by 1

Step 5: if  $i$  is less than N, then repeat  
from step 3

                otherwise go the next step

Step 6: STOP

## Keywords:

**Input:** READ, OBTAIN, GET &  
PROMPT

**Output:** PRINT, DISPLAY, and SHOW

**Compute:** COMPUTE, CALCULATE,  
DETERMINE

**Initialize:** SET, INITIALIZE

**Add one:** INCREMENT



# Example 4



**Problem:** Write an algorithm to print the multiplication table of 2 from 0 to 15. Draw flowchart and write pseudocode for your algorithm.

*Prior to writing algorithms, we need to identify the inputs and outputs for the given problem.*

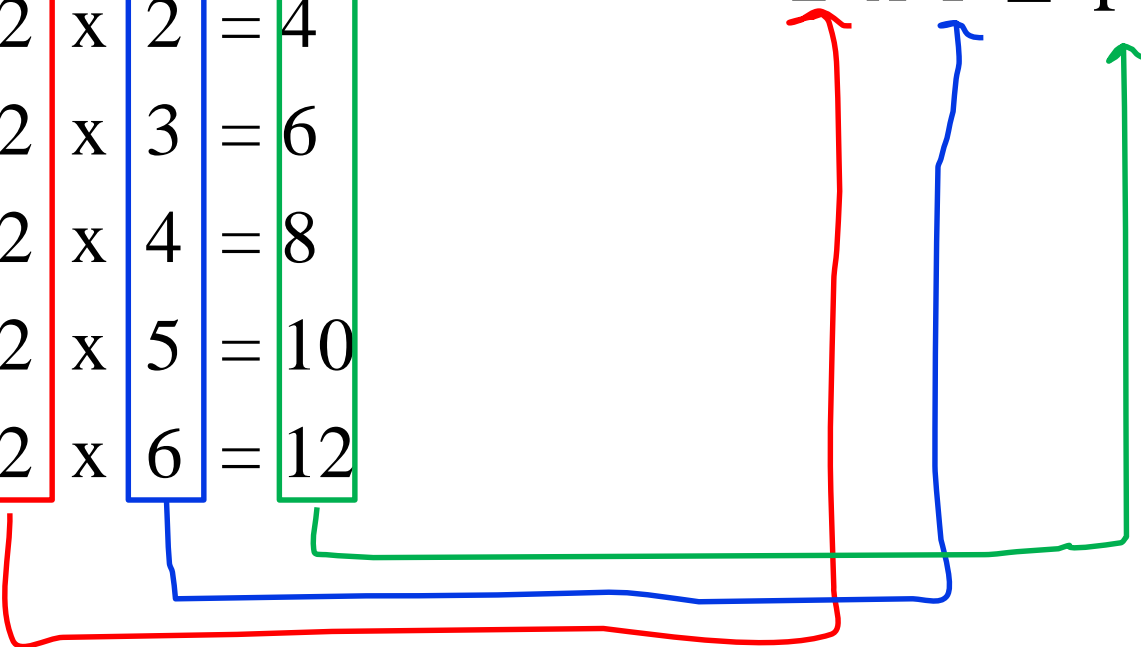
**Input:** the two operands (ie 2 and  $i = 0$  to 15)

**Output:** print multiplication table



2	x	0	=	0
2	x	1	=	2
2	x	2	=	4
2	x	3	=	6
2	x	4	=	8
2	x	5	=	10
2	x	6	=	12

$$2 \times i = p$$







# Example 4 (cont..)



## Algorithm:

Step 0: START

Step 1: set the initial value of  $i$  to 0

Step 2: when  $i$  is less than or equal to 15:

2.1 compute  $p = 2 \times i$

2.2 print “ $2 \times i = p$ ”

2.3 increase the value of  $i$  by 1

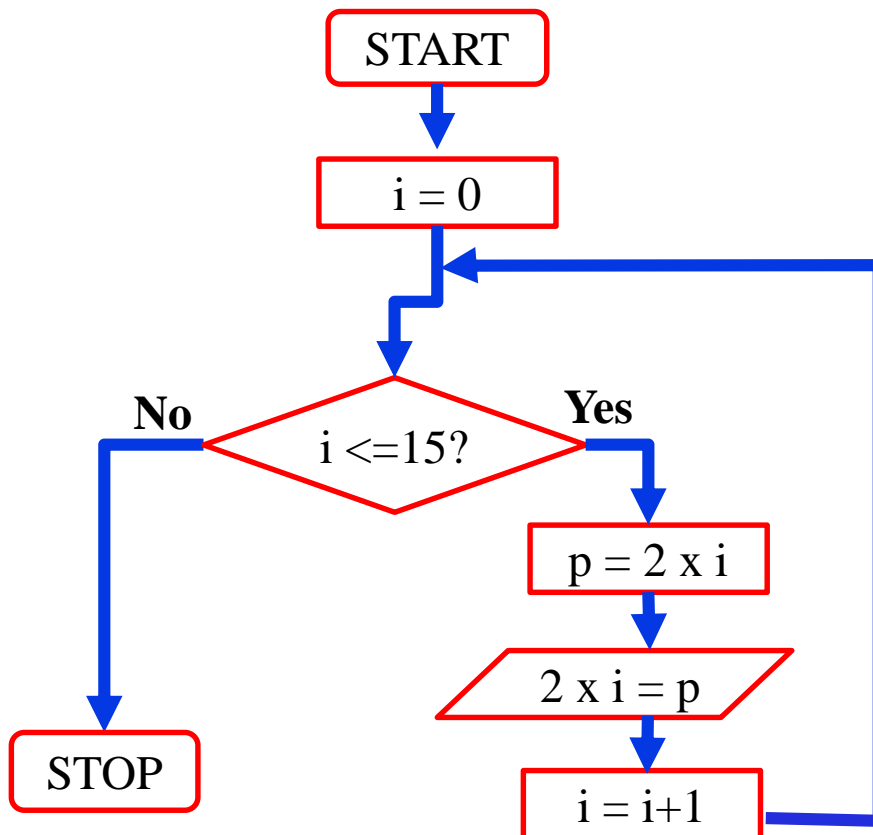
Step 3: STOP



# Example 4 (cont..)



## Flowchart:



## Algorithm:

Step 0: START

Step 1: set the initial value of i to 0

Step 2: when i is less than or equal to 15:

2.1 compute  $p = 2 \times i$

2.2 print “ $2 \times i = p$ ”

2.3 increase the value of i by 1

Step 3: STOP



# Example 4 (cont..)



## Pseudocode:

```
INITIALIZE i to 0
WHILE (i<=15):
    COMPUTE p = 2 x i;
    SHOW 2 x i = p;
    INCREMENT i by 1;
ENDWHILE;
```

## Algorithm:

Step 0: START

Step 1: set the initial value of i to 0

Step 2: when i is less than or equal to 15:

2.1 compute  $p = 2 \times i$

2.2 print “ $2 \times i = p$ ”

2.3 increase the value of i by 1

Step 3: STOP

## Keywords:

**Input:** READ, OBTAIN, GET & PROMPT

**Output:** PRINT, DISPLAY, and SHOW

**Compute:** COMPUTE, CALCULATE, DETERMINE

**Initialize:** SET, INITIALIZE

**Add one:** INCREMENT



# CLASS ACTIVITY



Write algorithm, pseudocode and draw flowchart for the following:

1. Circumference of a circle.
2. Determine the maximum of two numbers
3. Determine whether a character is vowel or consonant
4. Determine the minimum number among three entered numbers
5. Print all odd numbers from 1-100
6. Print the series 30 25 20 15 10 5 0



# CLASS ACTIVITY



Read any number from 1-7 and as per the number entered, display its corresponding day of the week as shown below using:

**Score**

**Result**

1

“Monday”

2

“Tuesday”

3

“Wednesday”

4

“Thursday”

5

“Friday”

6

“Saturday”

7

“Sunday”



# Home Assignment



1. Write algorithms for the following
  - a) Print your name on the screen
  - b) Determine the user input year as leap year or not
  - c) Compute sum of all even numbers between 0-100
  - d) Add two 2 x 2 matrix
  - e) Determine the largest number among 10 user entered number



# Home Assignment



- f) Get the marks of a student and grade the student performance as follows

Average	Grade
80-100	Excellent
70-79.9	Very Good
60-69.9	Good
50-59.9	Satisfactory
0-49.9	Fail

- Note:** Student should be awarded from 0-100
2. Translate algorithms that you have designed in question 1 to flowchart and pseudocode.



# Thank you