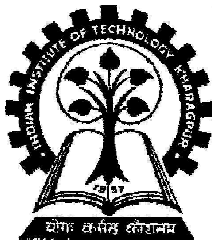


# Basic Programming Concepts

## CS10001: Programming & Data Structures



**Pallab Dasgupta**

**Professor, Dept. of Computer Sc. & Engg.,  
Indian Institute of Technology Kharagpur**

# Some Terminologies

- **Algorithm / Flowchart**
  - A step-by-step procedure for solving a particular problem.
  - Independent of the programming language.
- **Program**
  - A translation of the algorithm/flowchart into a form that can be processed by a computer.
  - Typically written in a high-level language like C, C++, Java, etc.

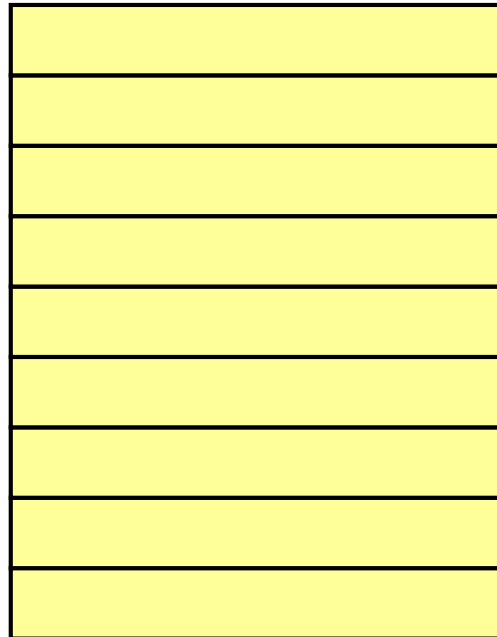
# Variables and Constants

- **Most important concept for problem solving using computers**
- **All temporary results are stored in terms of variables**
  - **The value of a variable can be changed.**
  - **The value of a constant do not change.**
- **Where are they stored?**
  - **In main memory.**

# Contd.

- **How does memory look like (logically)?**
  - **As a list of storage locations, each having a unique address.**
  - **Variables and constants are stored in these storage locations.**
  - **A variable is like a *bin***
    - **The contents of the *bin* is the *value* of the variable**
    - **The variable name is used to refer to the value of the variable**
    - **A variable is mapped to a *location* of the memory, called its *address***

# Memory map



Address 0

Address 1

Address 2

Address 3

Address 4

Address 5

Address 6

**Every variable is mapped to a particular memory address**



Address N-1

# Variables in Memory

Instruction executed

Variable X

Time  
↓

$X = 10$



10

$X = 20$



20

$X = X + 1$



21

$X = X * 5$



105

# Variables in Memory (contd.)

Instruction executed

Time  
↓

$$X = 20$$



$$Y = 15$$



$$X = Y + 3$$



$$Y = X / 6$$



Variable

X

Y

20

?

20

15

18

15

18

3

# Data Types

- **Three common data types used:**
  - **Integer :: can store only whole numbers**
    - Examples: 25, -56, 1, 0
  - **Floating-point :: can store numbers with fractional values.**
    - Examples: 3.14159, 5.0, -12345.345
  - **Character :: can store a character**
    - Examples: 'A', 'a', '\*', '3', ' ', '+'



# Data Types (contd.)

- **How are they stored in memory?**
  - **Integer ::**
    - 16 bits
    - 32 bits
  - **Float ::**
    - 32 bits
    - 64 bits
  - **Char ::**
    - 8 bits (ASCII code)
    - 16 bits (UNICODE, used in Java)

Actual number of bits vary from one computer to another

# Problem solving

- **Step 1:**
  - **Clearly specify the problem to be solved.**
- **Step 2:**
  - **Draw flowchart or write algorithm.**
- **Step 3:**
  - **Convert flowchart (algorithm) into program code.**
- **Step 4:**
  - **Compile the program into object code.**
- **Step 5:**
  - **Execute the program.**

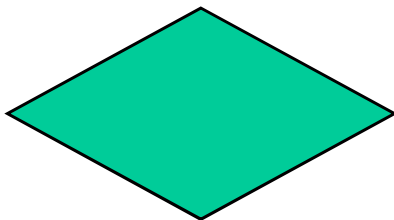
# Flowchart: basic symbols



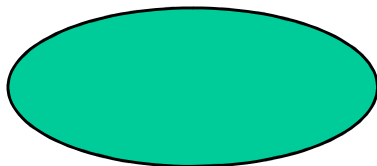
**Computation**



**Input / Output**



**Decision Box**



**Start / Stop**

**Contd.**

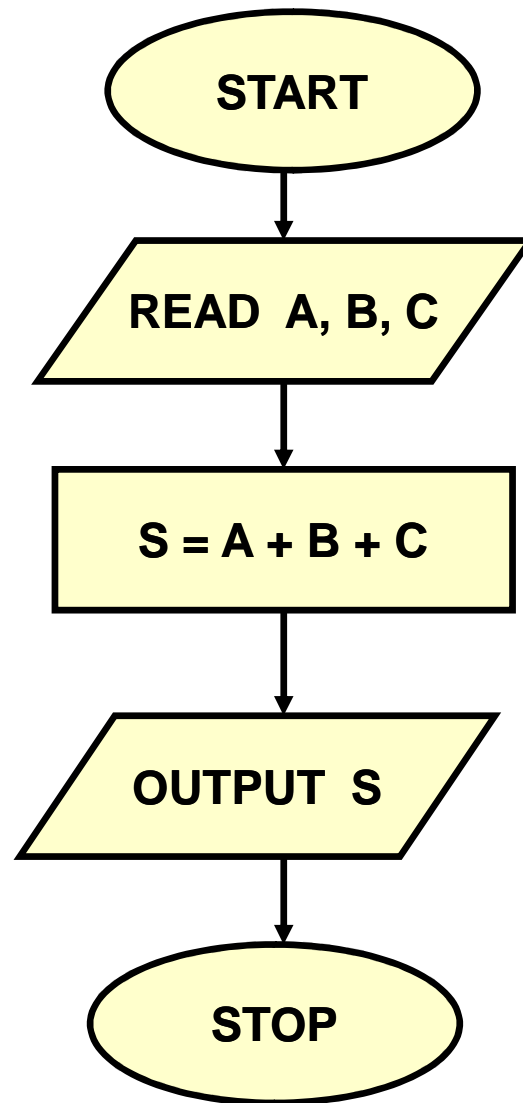


**Flow of  
control**

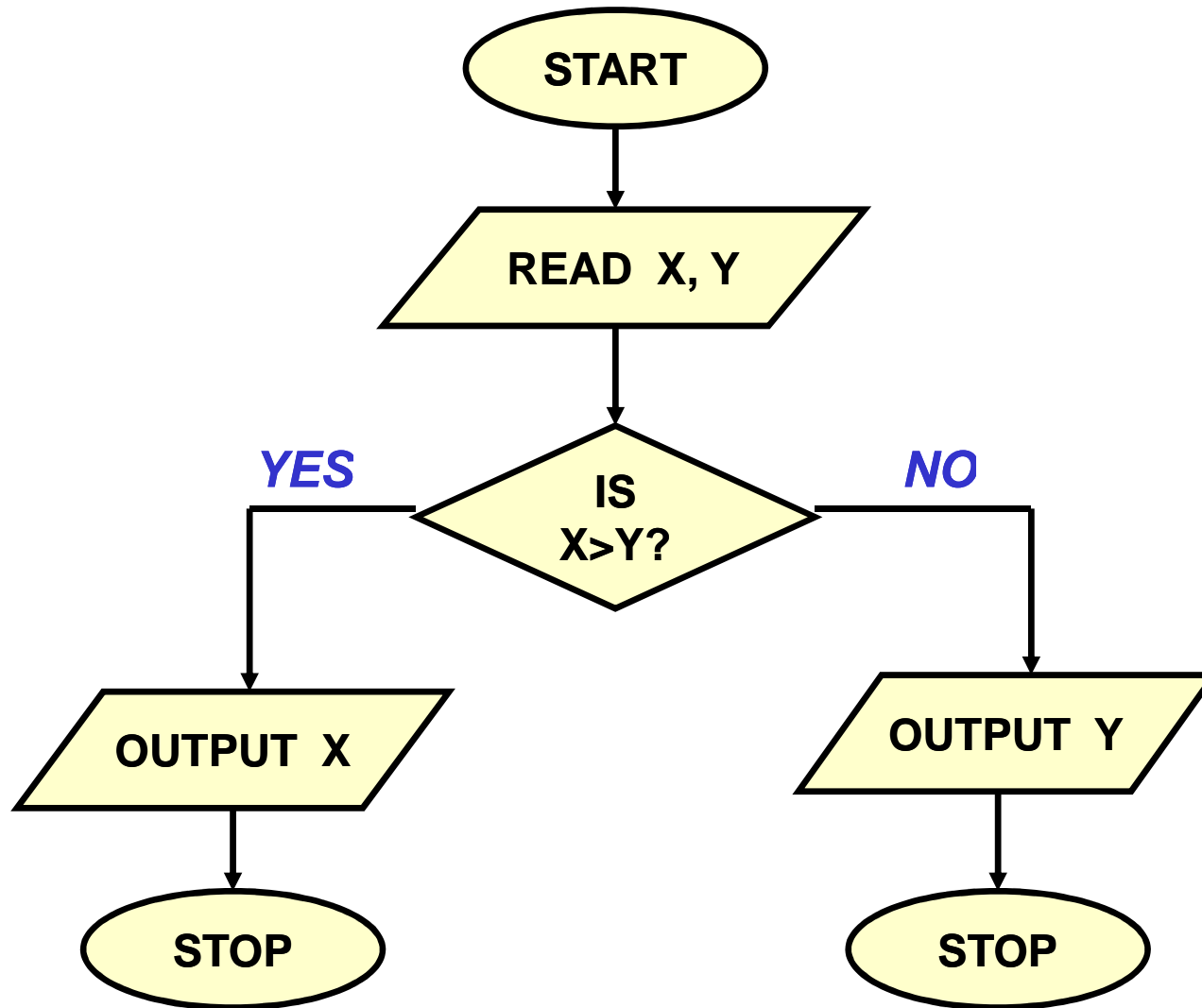


**Connector**

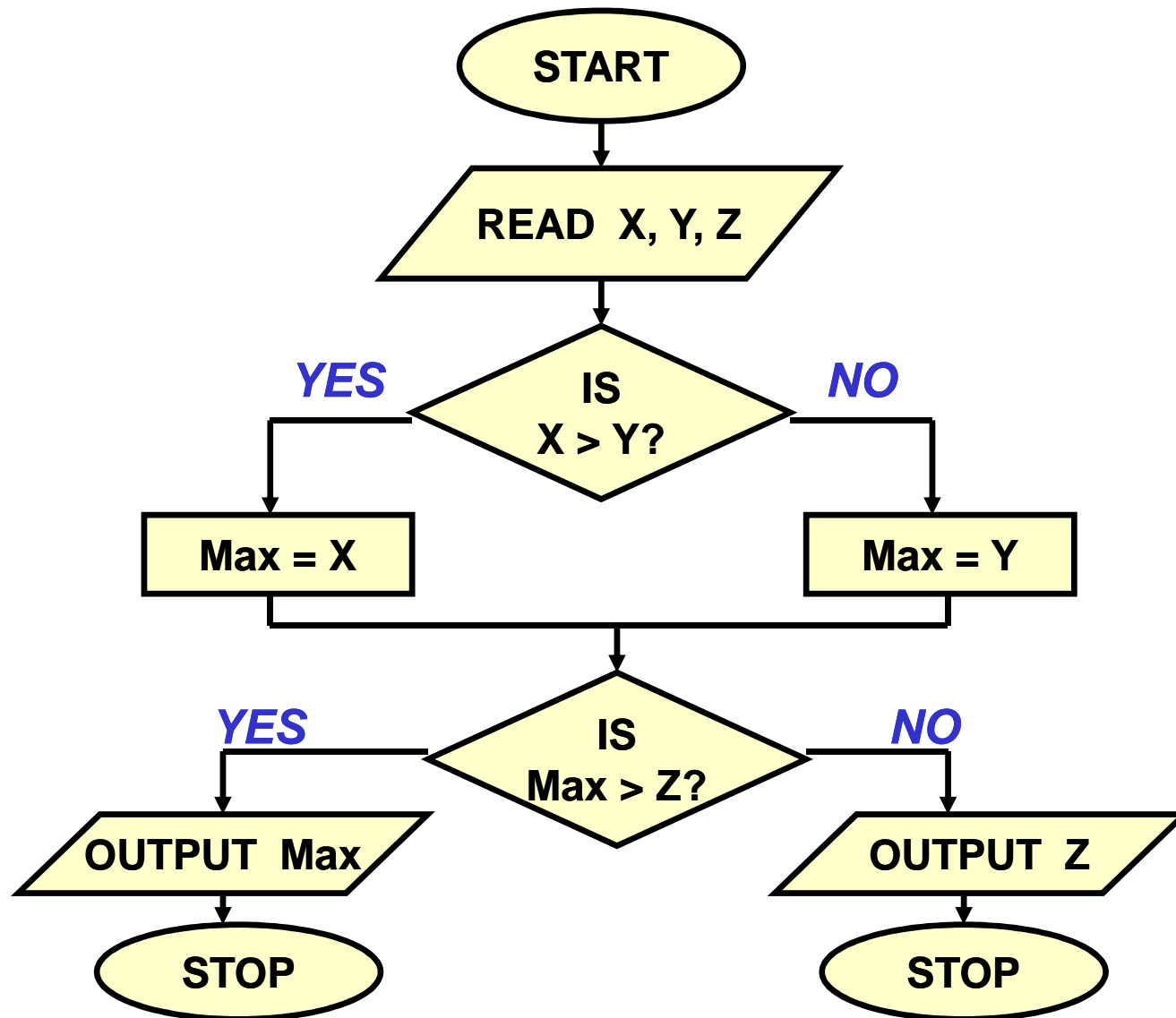
## Example 1: Adding three numbers



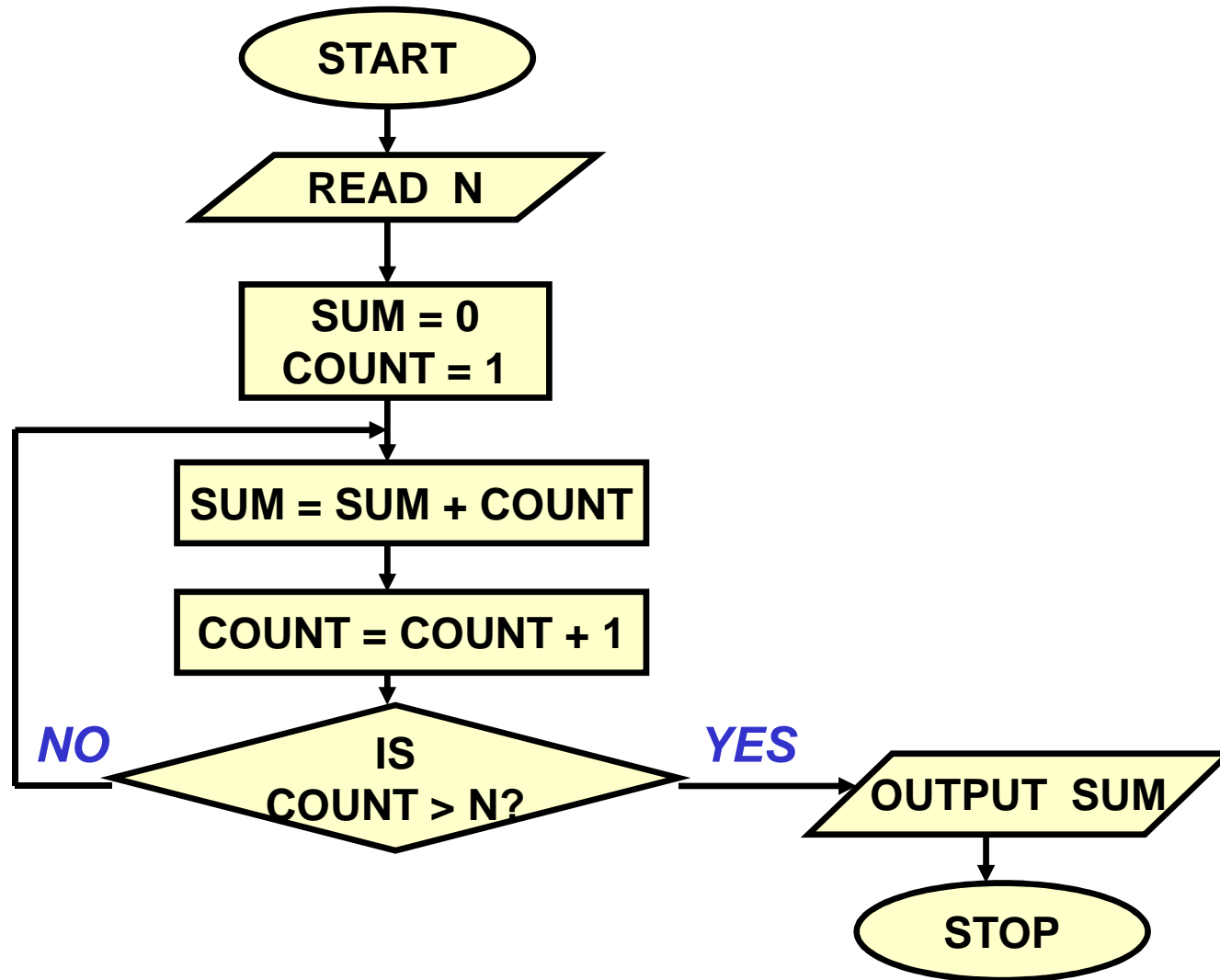
## Example 2: Larger of two numbers



## Example 3: Largest of three numbers

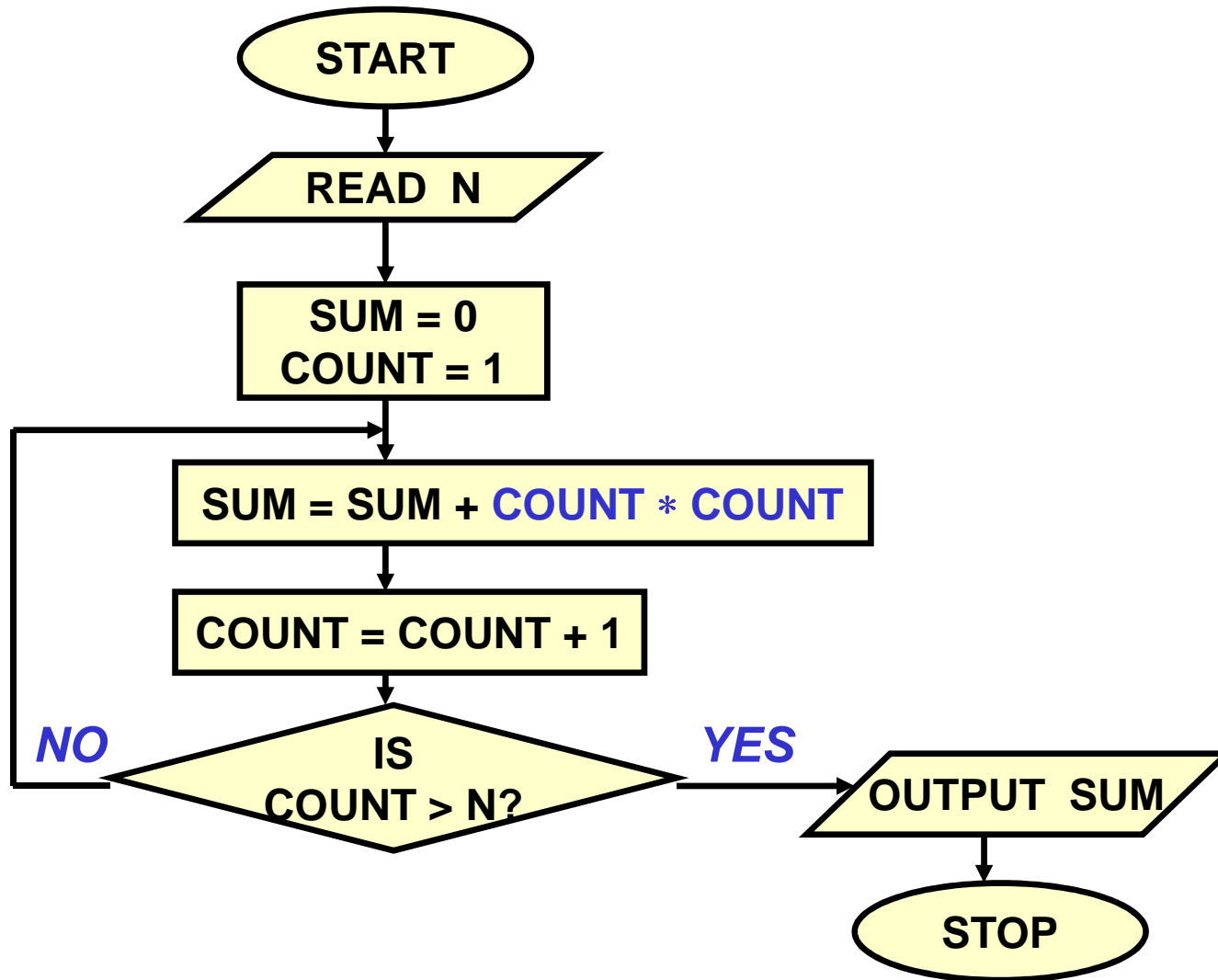


## Example 4: *Sum of first N natural numbers*

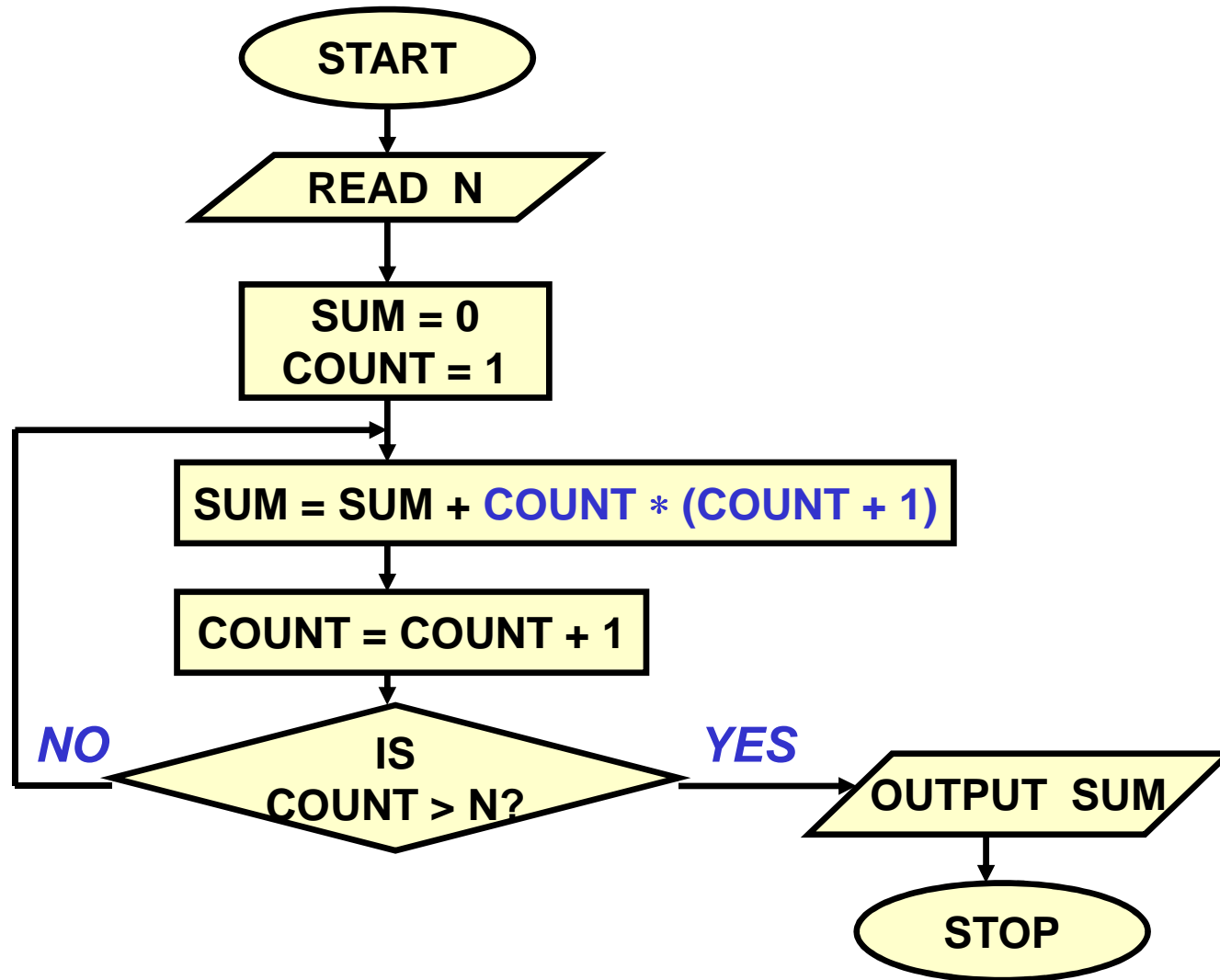




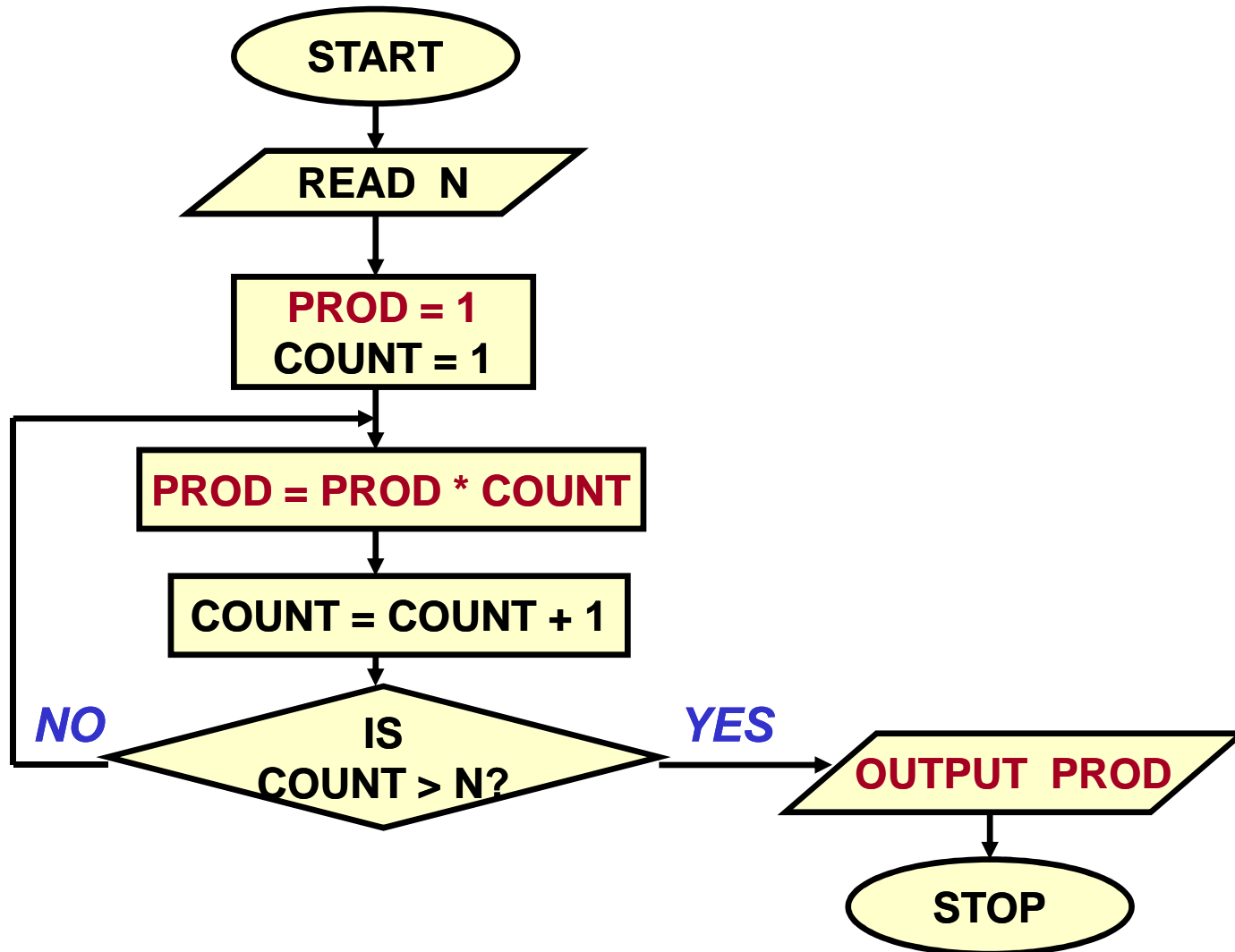
## Example 5: $SUM = 1^2 + 2^2 + 3^2 + N^2$



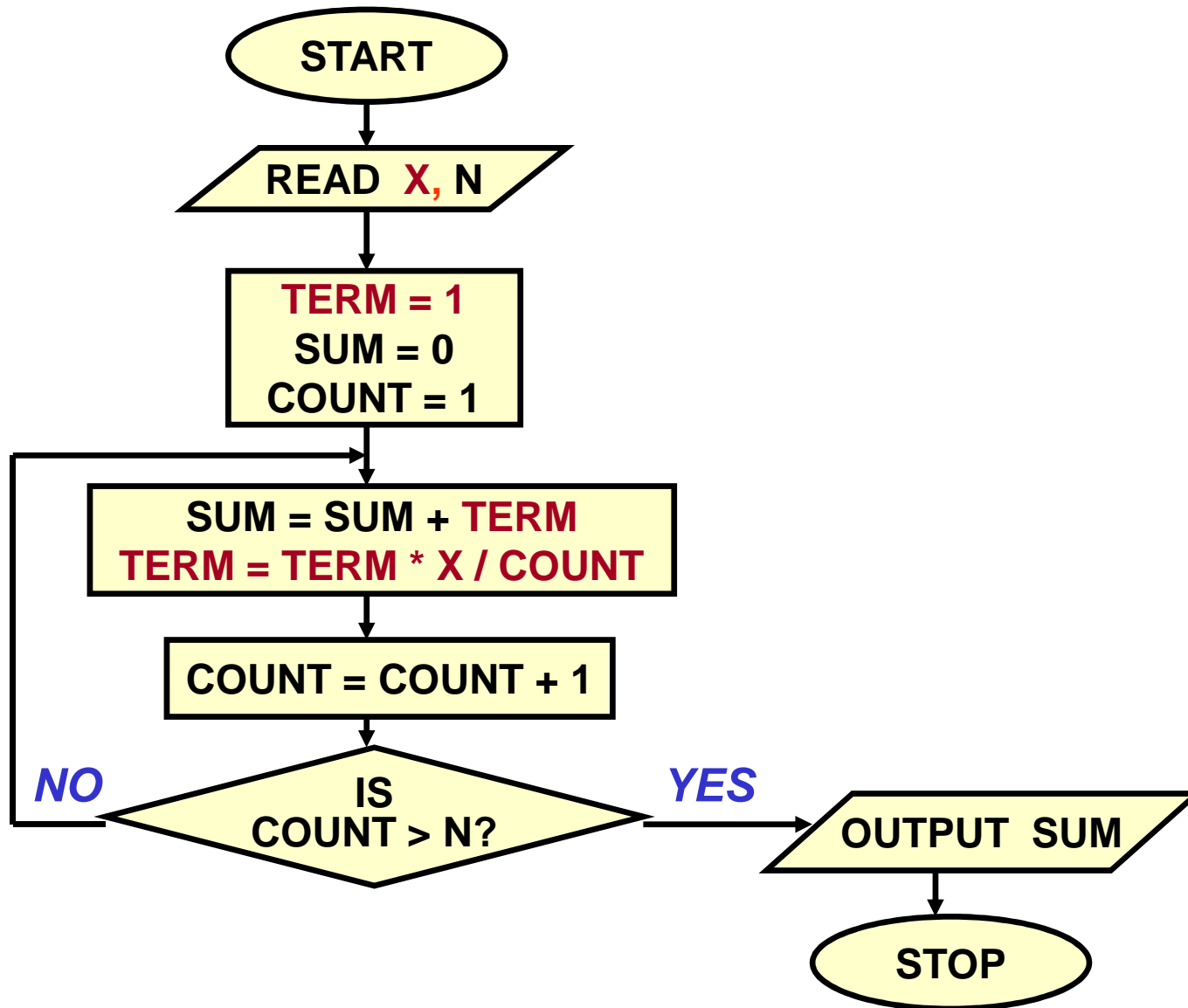
## Example 6: $SUM = 1.2 + 2.3 + 3.4 + \dots$ to $N$ terms



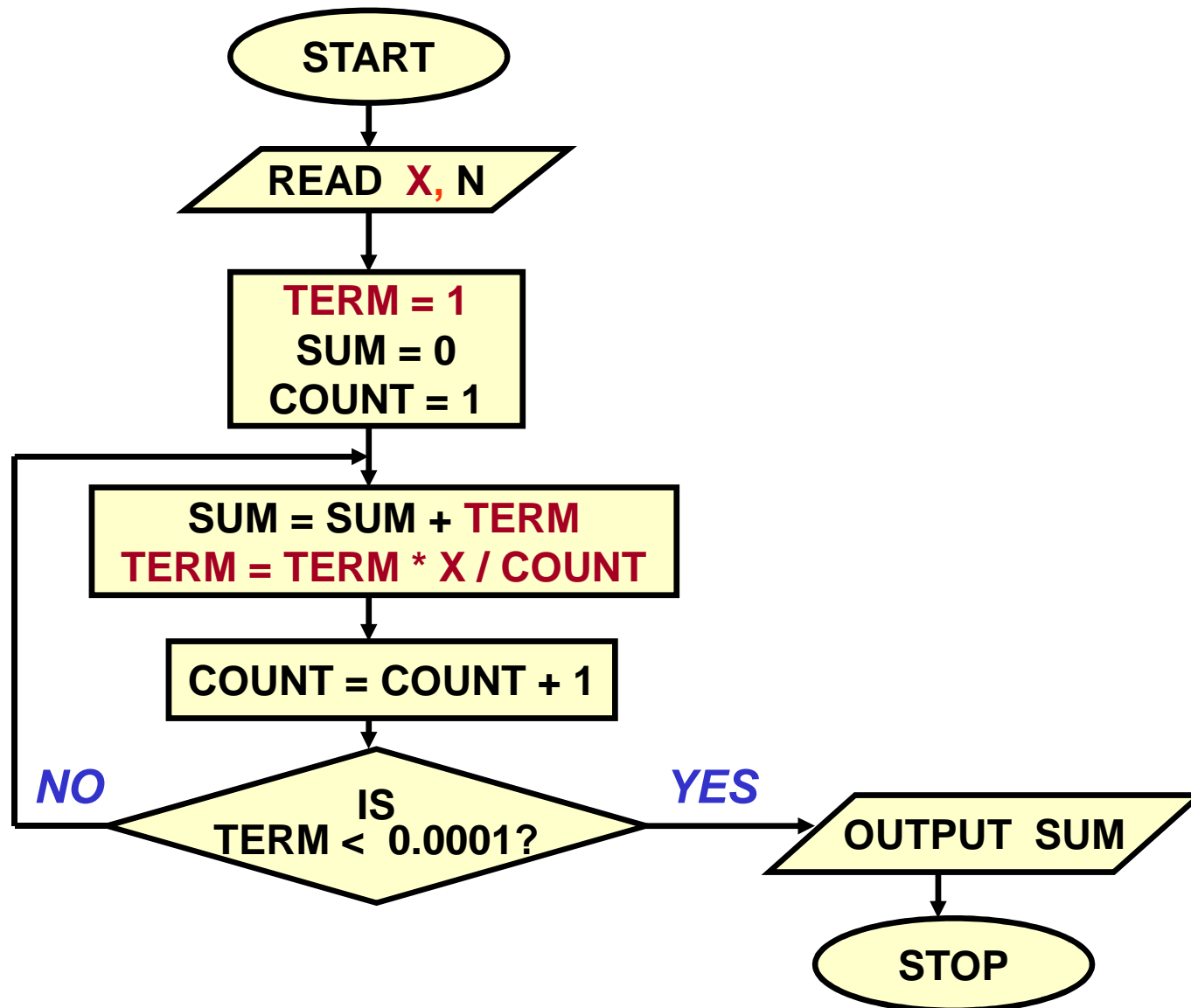
## Example 7: Computing Factorial



## Example 8: Computing $e^x$ series up to $N$ terms



## Example 8: Computing $e^x$ series up to 4 decimal places



## Example 10: *Roots of a quadratic equation*

$$ax^2 + bx + c = 0$$

***TRY YOURSELF***

## Example 11: *Grade computation*

<b>MARKS <math>\geq</math> 90</b>	<b>→ Ex</b>
<b>89 <math>\geq</math> MARKS <math>\geq</math> 80</b>	<b>→ A</b>
<b>79 <math>\geq</math> MARKS <math>\geq</math> 70</b>	<b>→ B</b>
<b>69 <math>\geq</math> MARKS <math>\geq</math> 60</b>	<b>→ C</b>
<b>59 <math>\geq</math> MARKS <math>\geq</math> 50</b>	<b>→ D</b>
<b>49 <math>\geq</math> MARKS <math>\geq</math> 35</b>	<b>→ P</b>
<b>34 <math>\geq</math> MARKS</b>	<b>→ F</b>

# Grade Computation (contd.)

