Flowcharting & Algorithms

Unit – Problem Solving, Flowcharts, Algorithms

CTAE
RESOURCE NETWORK

Standard – IT-IDT-7

- Students will know how flowcharts are used to map out how an algorithm works.
- Students will understand the use of pseudocode to describe the process of algorithms.
- Students will be able to follow discuss and complete examples of flowchart, pseudocode, and algorithms.

Key Terms

- ***Algorithm*** – a step-by-step procedure or formula for solving a problem
- ***Flowchart*** – a type of diagram that represents an algorithm or process
- ***Pseudo code*** **–** a compact and informal way of describing a program; not a programming language

ALGORITHMS AND FLOWCHARTS

- Computer programming can be divided into two phases:
- ***Problem solving phase***
  - Make an ordered sequence of steps that solves a problem.
  - This sequence of steps is called an ***algorithm.***
- ***Implementation phase***
  - Implement using a programming language.

Steps in Problem Solving

- ***Pseudocode*** is an artificial and informal language that helps programmers develop algorithms. Pseudocode may be in informal English, combinations of computer languages and spoken language. Whatever works for you.
- First produce a general algorithm (one can use ***pseudocode***).
- Refine your steps until you get to an easy sequence. Perhaps use numbers or bullets. The point is to simplify the language to an understandable process or event.

Algorithm

- Algorithms are defined by the step by step nature of them.
- In this way, they are much like a recipe.
- Think of an everyday process you use such as adding numbers or long division.
- Now create a detailed step-by-step guide to completing this task.

Example

- Write an algorithm to log in to school's email account.

Pseudocode:

1. *Go to www.henry.k12.ga.us/elhs*
2. *Click Office 365 for Students and Teachers Site Shortcut*
3. *Enter email ID and password*
4. *Click Sign in*

Pseudocode & Algorithm

- **Example 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 four marks.

Pseudocode & Algorithm

Pseudocode:

1. *Input a set of 4 marks*
2. *Calculate their average by summing and dividing by 4*
3. *If average is below 60*

Print “FAIL”

else

Print “PASS”

Pseudocode & Algorithm

Detailed Algorithm

Step 1: Input M1,M2,M3,M4

Step 2: GRADE ← (M1+M2+M3+M4)/4

Step 3: If (GRADE < 60) then

Print “FAIL”

  Else

Print “PASS”

Endif

The Flowchart

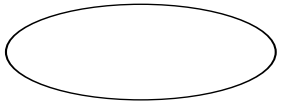
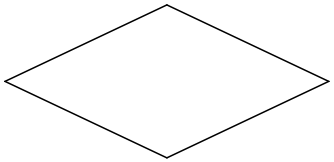
A Flowchart is another algorithm but graphical.

- Shows logic solution
- Emphasizes individual steps and their interconnections
- A flowchart must have a start and stop.
- All steps in a flowchart must connect. Can't leave a step "hanging" with no connection.

e.g. control flow from one action to the next

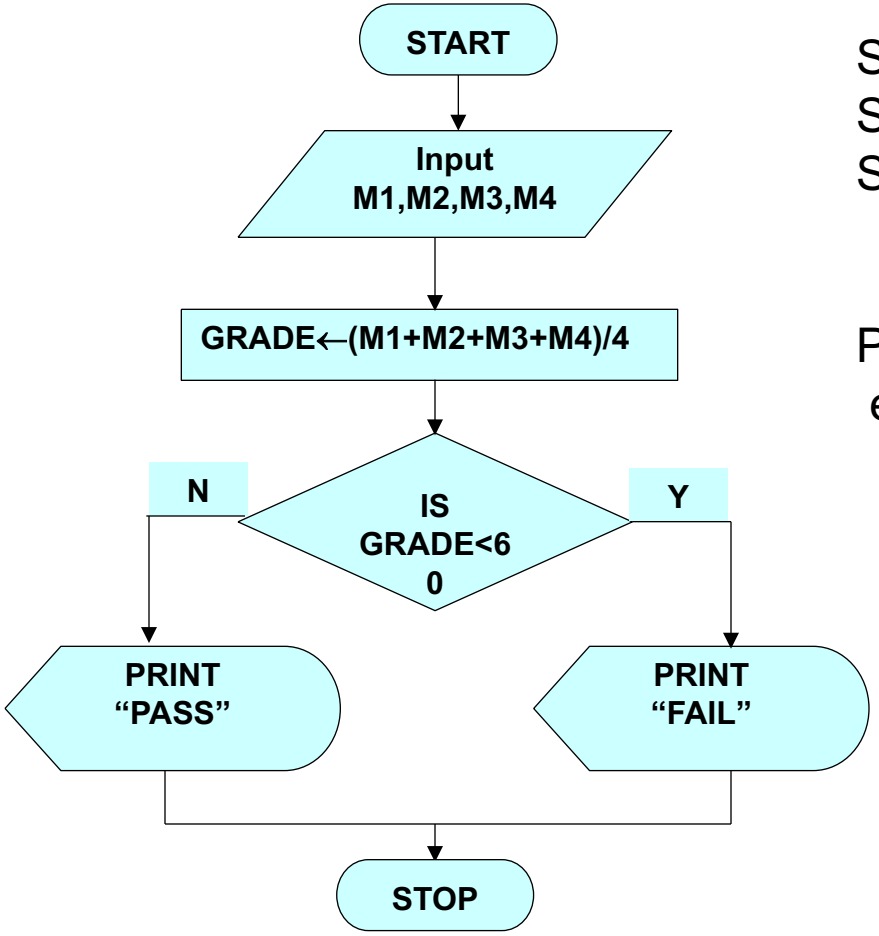
Flowchart Symbols

General Used Symbols

Name	Symbol	Use in Flowchart
Oval		Denotes the beginning or end of the program
Parallelogram		Denotes an input operation
Rectangle		Denotes a process to be carried out e.g. addition, subtraction, division etc.
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes. (e.g. IF/THEN/ELSE)
Hybrid		Denotes an output operation
Flow line		Denotes the direction of logic flow in the program

CTAE RESOURCE NETWORK

Example



Step 1: Input M1,M2,M3,M4
Step 2: GRADE ← (M1+M2+M3+M4)/4
Step 3: if (GRADE < 60) then
Print “FAIL”
else
Print “PASS”
endif

Another Example

Problem: Add 10 and 20

Algorithm (in simple English)

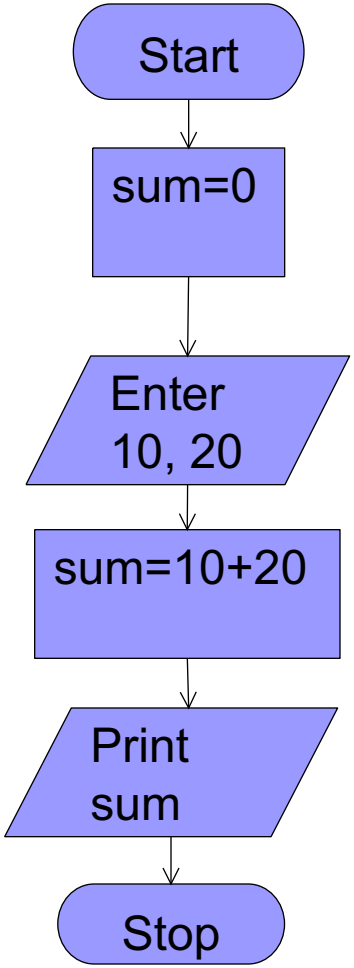
1. Initialize sum = 0 (process)
2. Enter the numbers (input/output)
3. Add them and store the result in sum (process)
4. Print sum (input/output)

Flowchart for problem

Problem: Add 10 and 20

Algorithm (in simple English)

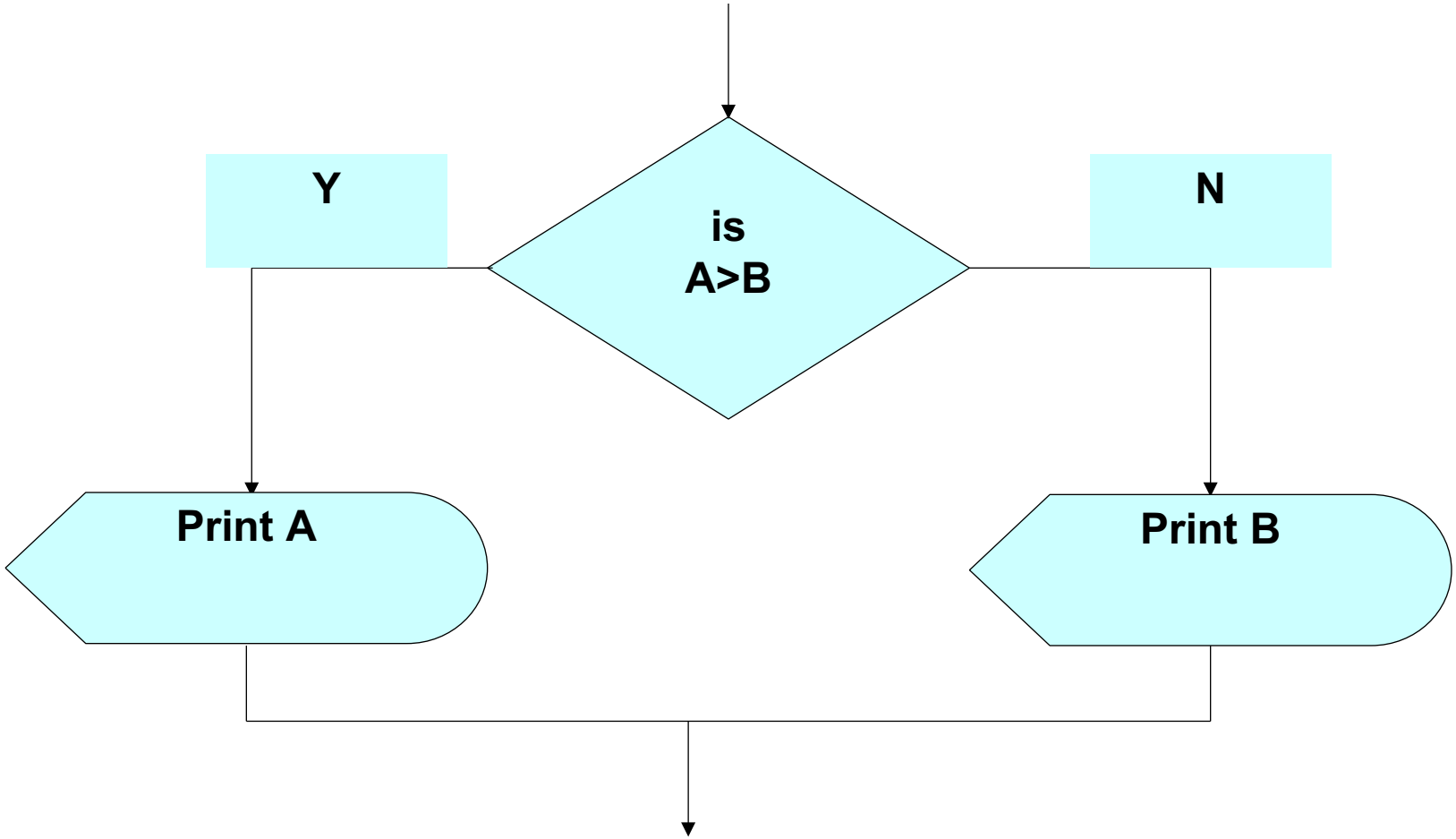
1. Initialize sum = 0 (process)
2. Enter the numbers (input/output)
3. Add them and store the result in sum (process)
4. Print sum (input/output)



DECISION STRUCTURES

- The expression A>B is a logical expression.
- *It describes a **condition** we want to test.*
- ***if A>B is true (if A is greater than B)** we take the action on left.*
- Print the value of A.
- ***If A>B is false (if A is not greater than B)** we take the action on right.*
- Print the value of B.

DECISION STRUCTURES



IF–THEN–ELSE STRUCTURE

- The structure is as follows

If condition then

true alternative

else

false alternative

endif

IF–THEN–ELSE STRUCTURE

- The algorithm for the flowchart is as follows:

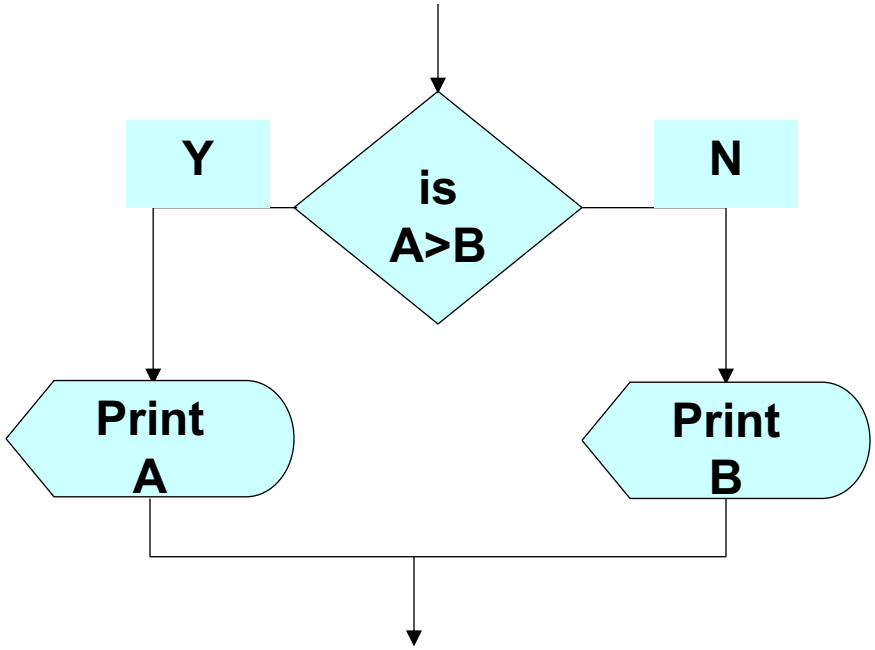
If A>B then

print A

else

print B

endif



Relational Operators

Relational Operators	
Operator	**Description**
>	**Greater than**
<	**Less than**
=	**Equal to**
≥	**Greater than or equal to**
≤	**Less than or equal to**
≠	**Not equal to**

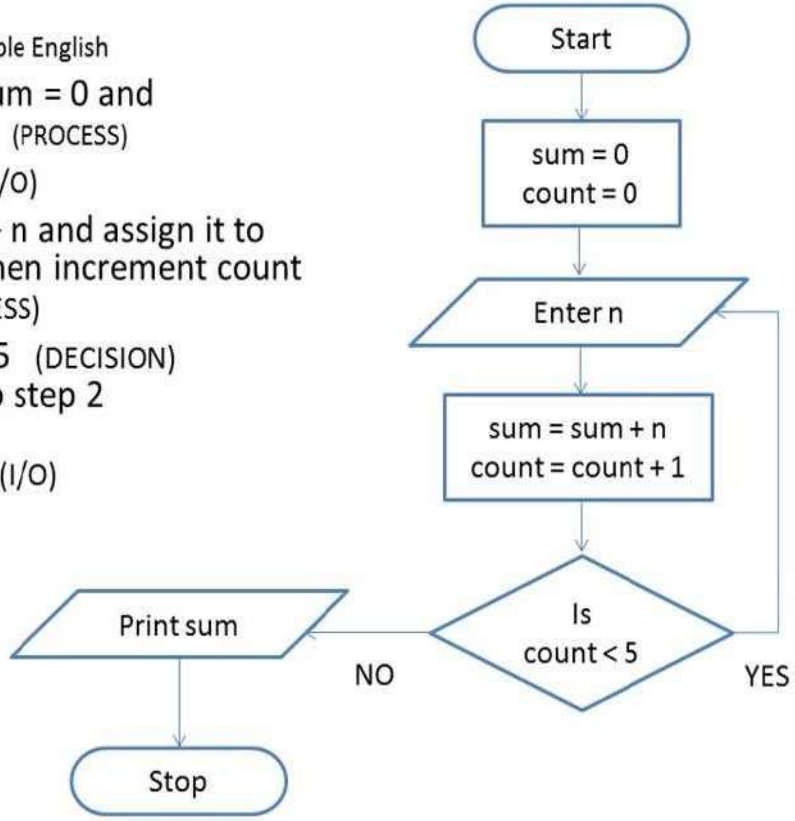
Example of If-Else (decision)

Find the sum of 5 numbers

Algorithm in simple English

1. Initialize sum = 0 and count = 0 (PROCESS)
2. Enter n (I/O)
3. Find sum + n and assign it to sum and then increment count by 1 (PROCESS)
4. Is count < 5 (DECISION)
   if YES go to step 2
   else
   Print sum (I/O)

Flowchart



Flowchart

Note: There are many ways to solve a given problem so there are many ways to draw a flowchart.

- Draw a flowchart for the following:

Pseudocode

Step 1:Go to www.henry.k12.ga.us/elhs
Step 2:Click Office 365 for Students and Teachers Site Shortcut
Step 3:Enter email ID and password
Step 4:If login error then
Re-enter email ID and password
Step 5: Else
Click Sign in
Step 6:Display email account