Department of Computer Science and Engineering,
Anna University, Chennai- 600025

CS23101– COMPUTATION THINKING

Practical
Aug-Dec 2026
Year/Sem/Batch: I/I/P

Exercise: 4	Decomposition and Algorithmic thinking	30-10-2026

Aim: To apply decomposition and algorithmic design to develop an effective solution strategy for a given problem.

Question 1

Searching for a hidden toy:

i. Assume there is a team of 10 people. If the team is asked to search for a toy kept hidden in any of the rooms in the department, design a strategy to search for the toy.

ii. Will the same strategy be a good one if there are 20 people in the team? Can you think of a better strategy?

iii. Will the same strategy be a good one if there are 4 people in the team? Can you think of a better strategy?

Question 2

Plan a birthday picnic: You're planning to hold a birthday picnic for a child and her friends. Break down the preparation into a tree structure of tasks. The facts are:

- You need to send out the invitations to the parents of the other children.
- Food you'll provide: sandwiches (ham, chicken, and cheese), homemade cake.
- Fresh ingredients (meat and dairy) need to be purchased on the day.

- Other things you need to take: disposable cutlery, blankets, games.
- The park where the picnic is held requires you to reserve a spot.
- All guests will get a goody bag with sweets when they leave.

Question 3

Decomposition: Decompose the below problems into smaller problems that would need to be solved. Use a bullet-point for each smaller step.

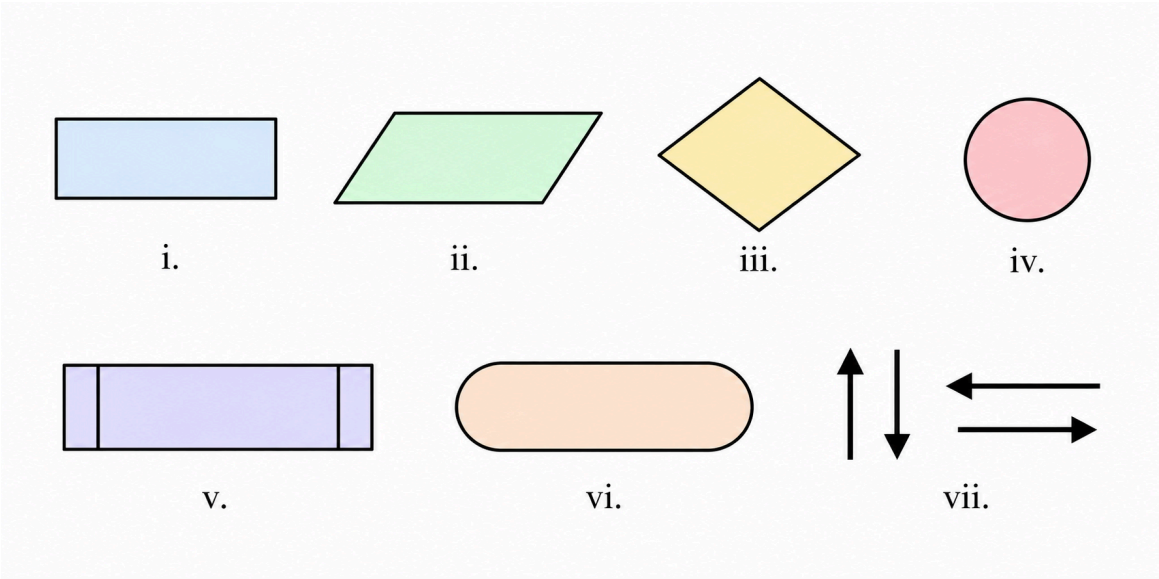
i. Plan your trip to an international destination.
ii. Pick the best smartphone for your grandparents.
iii. Decorate your front room.
iv. Washing a car.
v. Making a family meal

Refer "oral algorithms for everyday tasks" in the study notes if required.

Aim: To explore flowcharts for algorithmic visualisation

Question 4

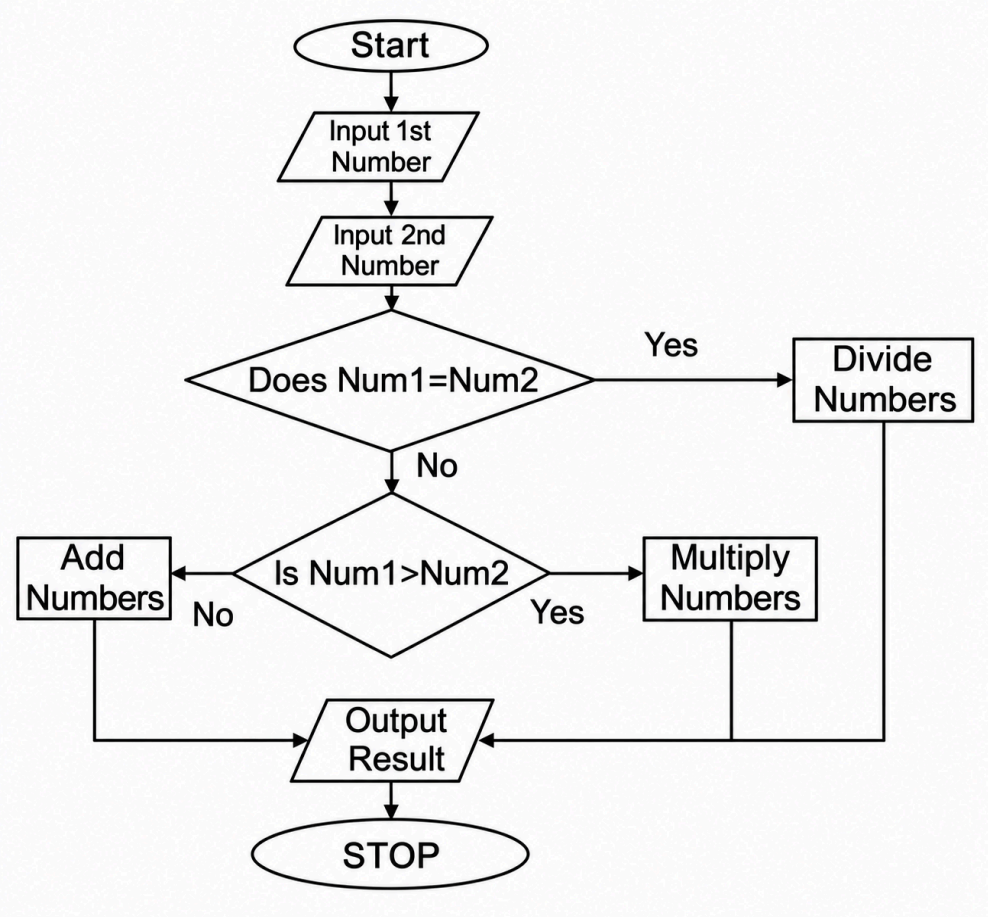
Flow chart symbols: Name the symbols below and what each one means



Question 5

Trace the flow chart: Study the flow charts given below and write the output for the given input values.

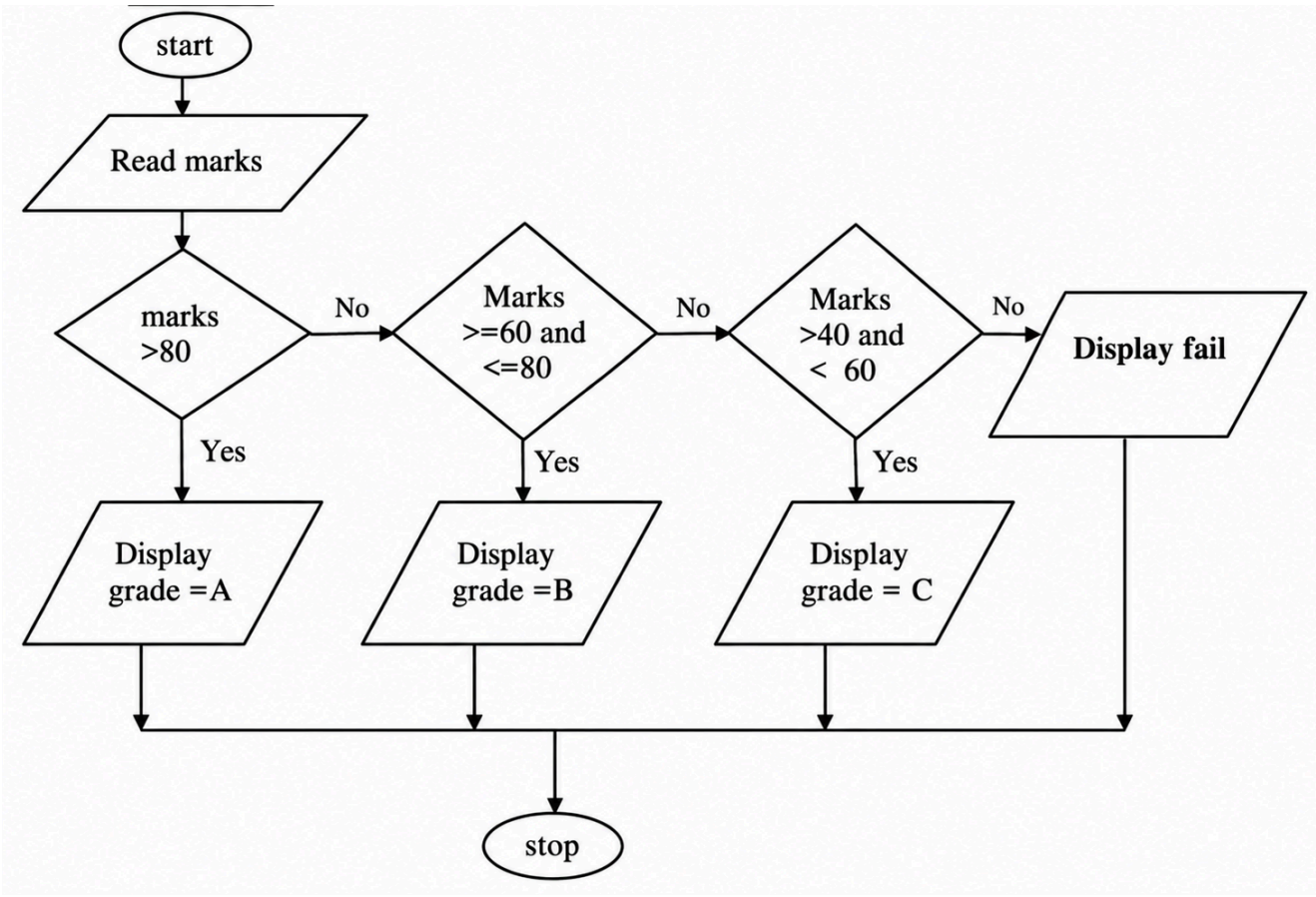
Flow chart 1:



Input values for flow chart 1:

i. Number 1 = 10, Number 2 = 5
ii. Number 1 = 4, Number 2 = 4
iii. Number 1 = 5, Number 2 = 13

Flow chart 2:



Input values for flow chart 2:

iv. Marks = 50

v. Marks = 81

vi. Marks = 30

Question 6

Flowchart: Draw a flowchart for the below tasks mentioned.

i. Print the sum of the first 100 natural numbers
ii. Swap two numbers
iii. Read numbers from the user until -1 is entered. Count how many are positive, negative and zero, then print the three counts. (-1 only stops the input and is not counted.)
iv. Make a move in the "snakes and ladders" game.
(Hint: throw the dice and move the counter by that number. Check whether it landed at the bottom of a ladder (if yes, climb up to the top) or on a snake's head (if yes, slide down to its tail). Then end the move.)
v.Reads a positive whole number N and prints the sum of its digits.
(Hint: Repeat until N becomes zero. For example, N = 539 gives 5 + 3 + 9 = 17.)

Aim: To design algorithms for computational problems and represent them using flowcharts with selection, iteration, functions, and procedures.

Question 7

Design Algorithms: Write pseudocode and draw the flowchart for the tasks below.

i. Sequence: Read the length and breadth of a rectangle and print its area and perimeter.
ii. Selection: Read a year and print whether it is a leap year.
(Hint: a year is a leap year if it is divisible by 4, except century years, which must be divisible by 400.)
iii. Iteration: Read a number N (N > 1) and print whether it is prime.
(Hint: try dividing N by 2, 3, 4 and so on up to N - 1. If any of them divides N exactly, it is not prime.)
iv. Function: Write a function SMALLEST(X, Y) that returns the smaller of two numbers. In the main program, read three numbers and use the function to find the smallest of them. *(Draw the flowchart of the function and the flowchart of the main program separately.)*

v. Procedure: Write a procedure TABLE(N) that prints the multiplication table of N from 1 to 10. In the main program, read N and call the procedure. *(Draw the flowchart of the procedure and the flowchart of the main program separately.)*