Certainly! Let's break down **Question 6: Flowchart** in detail.

---

6. Flowchart in C

A **flowchart** is a **diagrammatic representation** of an algorithm, showing the flow of execution step by step. It consists of **symbols** and **arrows** that define the process.

Why Use a Flowchart?

✓ **Easy to understand** – Helps visualize logic before coding.
✓ **Error detection** – Identifies problems before implementation.
✓ **Efficient debugging** – Helps trace issues logically.
✓ **Better communication** – Helps explain logic to others.

Flowchart Symbols

Symbol	Meaning	Example
Oval	**Start / End**	`Start`, `Stop`
Parallelogram	**Input / Output**	`Enter a number`, `Print result`
Rectangle	**Process / Calculation**	`sum = a + b`
Diamond	**Decision (Yes/No or True/False Condition)**	`If num > 0?`
Arrow	**Flow of Execution**	`Indicates direction`

Example 1: Flowchart for Finding the Largest of Two Numbers

Problem Statement

Write a flowchart to **compare two numbers** and print the largest.

Flowchart Explanation

1. Start the program.
2. Input two numbers (`a` and `b`).
3. Compare `a` and `b`.
4. If `a > b`, print `a is larger`.
5. Else, print `b is larger`.
6. Stop the program.

Flowchart Diagram

```
Start
  ↓
Input a, b
  ↓
a > b ?
  ↙      ↘
Yes        No
  ↓          ↓
Print a     Print b
  ↓          ↓
  Stop
```

Example 2: Flowchart for Checking Even or Odd Number

Problem Statement

Write a flowchart to check if a number is **even or odd**.

Flowchart Explanation

1. Start the program.
2. Input a number (`num`).
3. Check if `num % 2 == 0`.
4. If `true`, print **Even**.
5. Else, print **Odd**.
6. Stop the program.

Flowchart Diagram

```
Start
  ↓
Input num
  ↓
num % 2 == 0 ?
  ↙      ↘
Yes        No
  ↓          ↓
Print Even  Print Odd
  ↓          ↓
  Stop
```

Example 3: Flowchart for Calculating Factorial of a Number

Problem Statement

Write a flowchart to **calculate factorial** of a number (`n! = n × (n-1) × ... × 1`).

Flowchart Explanation

1. Start the program.
2. Input a number (`n`).
3. Initialize `fact = 1`.
4. Multiply `fact` by `n`, decrease `n` by 1.
5. Repeat until `n == 1`.
6. Print the factorial.
7. Stop the program.

Flowchart Diagram

```
Start
  ↓
Input n
  ↓
fact = 1
  ↓
n > 1 ?
  ↙      ↘
Yes        No
  ↓          ↓
fact = fact * n
n = n - 1
  ↓          ↓
  Loop      Print fact
              ↓
             Stop
```

How to Draw a Flowchart?

1. **Understand the problem** – Identify inputs, processing steps, and outputs.
2. **Use correct symbols** – Ovals for Start/End, parallelograms for I/O, rectangles for processes, diamonds for decisions.
3. **Connect with arrows** – Show step-by-step logic flow.
4. **Keep it simple** – Avoid unnecessary complexity.
5. **Test it** – Walk through the flowchart with test values.

Key Takeaways

✓ Flowcharts **visualize** program logic before coding.
✓ Use **symbols**: Ovals (Start/End), Parallelograms (I/O), Rectangles (Process), Diamonds (Decision).
✓ Helps in **debugging, planning, and understanding** algorithms.
✓ Always **trace the flowchart manually** before coding.

Would you like a specific flowchart example in **C code**?