Approach

1) Bit Extraction :

Created a utility function extractBits to extract specific bits from a given number. This function is used for extracting fields like opcode, rd, rs1, rs2, funct3, funct7, and immediate values from the instruction.

2) Instruction Decoding Functions:

Implemented separate decoding functions for each format (R, I, S, B, J, U) to maintain modular code and ease of debugging.
Used a mapping approach to easily map combinations of funct3, funct7, and opcode to their corresponding operations.
For immediate values which can be negative, ensured sign extension to get correct offsets.

3) Labeling System for B and J Instructions:

Implemented a two-pass approach:
In the first pass (firstPass), identified the target addresses of all B-format and J-format instructions and mapped these addresses to unique labels.
In the second pass, disassembled the instructions using the labels generated in the first pass.
Labels are generated in a sequential format like L1, L2, etc.

4) Main Execution:

Processed the instructions in a loop, decoding them based on their format, which is determined using the opcode.
Printed labels before the respective instructions.

5) Testing Approach:

Tested individual decoding functions using known machine code and their corresponding assembly instructions to validate their correctness.
For input an array of input values are given in the main function.
For example, for the instruction 0x007201b3, the expected output was add x3, x4, x7.