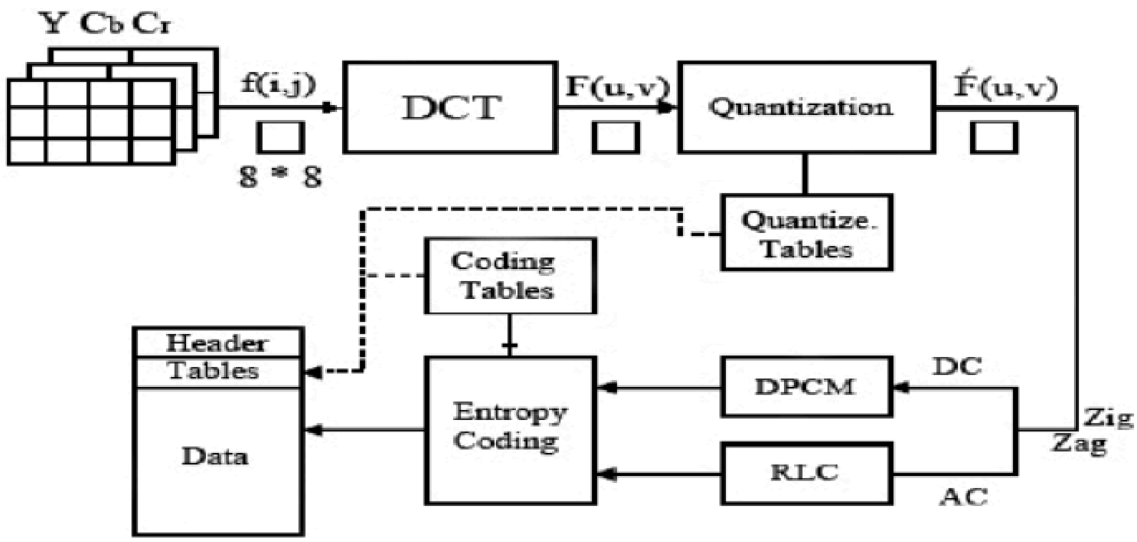
USN | | | | | | | | | |

Sub :	MMC						Sub Code:	21EC745
Date :	16/12/2024	Duration:	90 Minutes	Max Marks:	50		Sem / Sec:	

Answer any FIVE FULL Questions

1 With a neat diagram explain JPEG encoder in detail. [10]



Block diagram of the JPEG encoder [3] .

The JPEG (Joint Photographic Experts Group) encoder compresses images using a combination of lossy and lossless techniques. The encoding process is divided into several stages:

1. **Color Space Conversion**
   Converts the image from RGB to YCbCr (luminance and chrominance) color space for better compression.
2. **Downsampling**
   Reduces the resolution of chrominance components (Cb and Cr) since human eyes are less sensitive to color details compared to brightness.
3. **Block Splitting**
   Divides the image into 8x8 blocks to process smaller chunks, ensuring localized transformation and compression.
4. **Discrete Cosine Transform (DCT)**
   Converts spatial domain data of each 8x8 block into frequency domain data.
5. **Quantization**
   Reduces precision of less significant frequencies (higher frequencies), introducing lossy compression.
6. **Zigzag Scanning**
   Converts the 8x8 quantized DCT coefficients into a 1D array, prioritizing lower frequencies.
7. **Entropy Encoding**
   Uses Huffman or Arithmetic coding to further compress the data, ensuring lossless compression at this stage.

2	a) Determine the encoded version of the following difference values which relate to the encoded DC coefficients from consecutive DCT blocks: 12, 1, -2, 0, -1. b) Derive the binary form of the following run -length encoded AC coefficients: (0,6) (0,7) (3,3) (0,-1) (0,0).	[10]

a) Steps for Encoding DC Coefficients

1. **Difference Calculation**
   The differences are already given: `12, 1, -2, 0, -1`.
2. **Determine Category**
   The **Category** is determined by the number of bits required to represent the magnitude (absolute value) of the difference. The formula for the Category is:
   Category=⌊log2(|Difference|)⌋+1\text{Category} = \lfloor \log_2(|\text{Difference}|) \rfloor + 1Category=⌊log2(|Difference|)⌋+1
   If the difference is `0`, the Category is `0`.
3. **Huffman Code for Category**
   Each Category has a predefined Huffman code (from JPEG's DC Huffman table).
4. **Value Encoding**
   If the difference is non-zero, its binary representation (in the number of bits equal to the Category) is used:
   - Positive values are represented as-is.
   - Negative values are represented using a **ones' complement** encoding (invert the binary digits of the magnitude).

Calculation for Each Difference

Difference	Category	Huffman Code (Example Table)	Encoded Value
12	4	`1110`	Binary of 12: `1100`
1	1	`010`	Binary of 1: `1`
-2	2	`011`	Ones' complement of 2: `10`
0	0	`00`	(No additional value)
-1	1	`010`	Ones' complement of 1: `0`

b)

Step-by-Step Encoding

Run, Size	Value	Huffman Code (Run/Size)	Amplitude (Binary)
$(0,6)$	6	`1011`	$6 \rightarrow 110$
$(0,7)$	7	`1100`	$7 \rightarrow 111$
$(3,3)$	3	`111001`	$3 \rightarrow 011$
$(0,1)$	-1	`010`	$-1 \rightarrow 0$
$(0,0)$	EOB	`1010`	-

	Binary Representation Now, combine the Huffman codes and amplitudes: • $(0,6)$: `1011` + `110` → `1011110` • $(0,7)$: `1100` + `111` → `1100111` • $(3,3)$: `111001` + `011` → `111001011` • $(0,-1)$: `010` + `0` → `0100` • $(0,0)$: `1010`	
3	Explain Lempel-Ziv (LZ) coding and Lempel-Ziv-Welsh (LZW) coding with suitable examples.	[10]

Ans:
Lempel-Ziv (LZ) coding is a lossless data compression technique that identifies repeating patterns in a stream of data and encodes them efficiently. It uses a sliding window mechanism to reference previously seen strings instead of duplicating them.

Working:

Sliding Window:

- Maintains a fixed-size window of recent data. The window slides forward as new data is processed.
- Within the window, repeated substrings are identified.

Encoding Format:

Offset: The distance back to the start of the repeated substring.

Length: The length of the repeated substring.

Next Character: The character that follows the repeated substring.

Example of LZ Coding:

Let's compress the string: `ABABABA`

1. **Initial State:** Sliding window is empty.
2. **Step-by-Step Encoding:**

Processed String	Match	Encoding (Offset, Length, Next Char)
`A`	None	(0, 0, `A`)
`AB`	None	(0, 0, `B`)
`ABA`	`AB`	(2, 2, `A`)
`ABAB`	`AB`	(2, 2, `B`)

Encoded Output: `(0, 0, A) (0, 0, B) (2, 2, A) (2, 2, B)`

4	Explain Graphics Interchange Format and Tagged image format.	[10]

<u>Graphics Interchange Format:</u>

GIF (Graphics Interchange Format) is a raster-based file format commonly used for animated images and static graphics with a limited color palette. It was introduced in 1987 by CompuServe and remains popular for sharing short, looped animations.

Key Features:

1. **Color Palette (Indexed Color):**
   - Supports up to **256 colors** per frame, chosen from a 24-bit RGB color space.
   - Uses a color table to reduce file size.
2. **Animation Support:**
   - GIFs can include multiple frames with timing information, allowing for simple animations.
3. **Transparency:**
   - Supports single-bit transparency, meaning one color in the palette can be marked as transparent.
4. **Compression:**
   - Uses **Lempel-Ziv-Welch (LZW)** compression, which is lossless, ensuring no quality loss.
5. **File Extension:**
   - Typically saved with `.gif`.

Advantages:

- Efficient for small, simple graphics.
- Supports animation without additional software.
- Widely supported on web platforms.

<u>Tagged Image File Format (TIFF)</u>

TIFF (Tagged Image File Format) is a highly flexible and extensible format for storing raster graphics. It was developed in 1986 by Aldus Corporation (later Adobe) and is widely used in professional imaging and publishing.

Key Features:

1. **Color Depth and Range:**
   - Supports 1-bit (black and white), 8-bit (grayscale), and 24-bit or 48-bit (color).
   - Can store images with **high color fidelity**.
2. **Compression Options:**
   - Supports both **lossless compression** (e.g., LZW, ZIP) and **no compression**.
   - Optional lossy compression (e.g., JPEG) is also supported.
3. **Metadata:**
   - Includes extensive metadata storage using **tags** (e.g., EXIF, geotags).
4. **Layer and Transparency Support:**
   - Can store multiple layers and transparency data.
5. **File Extension:**
   - Typically saved with `.tiff` or `.tif`.

Advantages:

- High-quality, lossless image storage.
- Supports large file sizes and resolutions.
- Extensible with custom tags for specialized applications.

5	Write notes on 1) Source encoders and destination encoders 2) Lossless and lossy compression.	[10]

Source Encoders:

- **Definition:**
  A source encoder converts input data from its raw, uncompressed form into a compressed format to reduce redundancy and minimize storage or transmission requirements. It works based on the nature of the source data.
- **Steps in Source Encoding:**
  - **Sampling:** Converts continuous signals into discrete values.
  - **Quantization:** Maps input values to a finite range.
  - **Coding:** Encodes the quantized values using binary codes or other methods.
- **Examples:**
  - JPEG encoder for images.
  - MP3 encoder for audio.
  - MPEG encoder for video.
- **Objective:**
  - Efficient data representation.
  - Reduction in storage and transmission size.

Destination Encoders:

- **Definition:**
  A destination encoder prepares data for the receiving end after compression, ensuring compatibility with playback or rendering devices. It may also involve re-encoding the data for error correction or additional processing.
- **Functionality:**
  - Transcodes data to match device specifications or required formats.
  - Includes error-checking codes to ensure integrity during transmission.
- **Examples:**
  - Video encoding for streaming platforms.
  - Audio encoding to ensure compatibility with playback devices.
- **Objective:**
  - Tailor data for seamless consumption by end devices.
  - Preserve data integrity.

6	Explain 1) Entropy Encoding 2) Source Encoding	[10]

1) Entropy Encoding

Definition:
Entropy encoding is a lossless data compression technique that reduces data size based on the statistical properties of the input data. It replaces symbols with shorter codes for frequently occurring data and longer codes for rare data, minimizing the average length of encoded data.

Types of Entropy Encoding:

1. **Huffman Coding:**
   - Uses a binary tree to assign shorter binary codes to more frequent symbols and longer codes to less frequent ones.
   - Ensures that no code is a prefix of another (prefix-free codes).

    - Example: For a message with symbols `{A: 40%, B: 30%, C: 20%, D: 10%}`, Huffman encoding might assign:
      css
      Copy code
      `A → 0, B → 10, C → 110, D → 111`
        -
2. **Arithmetic Coding:**
    - Encodes the entire message as a single number in a decimal or fractional range [0,1)[0, 1)[0,1).
    - Represents the probabilities of symbols more compactly than Huffman coding.
    - Example:
        - For symbols `{A: 50%, B: 25%, C: 25%}`:
            - Message `ABC` could be represented as a single fractional number like `0.625`.

Advantages:

- Achieves near-optimal compression ratios for data with non-uniform probabilities.
- Works independently of the underlying data format.

Disadvantages:

- Computationally intensive compared to simpler compression methods.
- May be less efficient for small datasets.

Applications:

- JPEG and PNG image compression.
- MP3 audio compression.
- Data transmission in telecommunications.

2) Source Encoding

Definition:
Source encoding is the process of converting a source message (text, audio, video, or other data) into a compressed representation by removing redundancy. It may include both lossless and lossy techniques depending on the application.

Steps in Source Encoding:

1. **Sampling:** (For continuous data like audio or video)
    - Converts continuous signals into discrete time intervals.
2. **Quantization:**
    - Maps input values to a finite range, often introducing some loss (for lossy compression).
3. **Encoding:**
    - Represents the data in a compact binary format using coding methods such as run-length encoding, Huffman coding, or transform coding.

Types of Source Encoding:

1. **Lossless Encoding:**
    - Compresses data without losing any information.
    - Techniques: Run-Length Encoding (RLE), Huffman coding, Arithmetic coding.
    - Example: Text files or PNG images.
2. **Lossy Encoding:**
    - Removes less significant data that may not be perceptible to human senses.
    - Techniques: Transform coding (DCT in JPEG), quantization.
    - Example: JPEG for images, MP3 for audio.

	Advantages: ● Reduces data size for efficient storage or transmission. ● Tailored to the nature of the source data for optimal compression. **Applications:** ● Text compression (ZIP files). ● Multimedia compression (JPEG, MP3, MPEG).	