19

The **four correct roles** of the **Discrete Cosine Transform (DCT)** in image compression are:

✅ **Role 1:** Transforms the image from the **temporal (spatial) domain** to the **frequency domain**.
✅ **Role 3:** Concentrates most of the image's **energy into a small number of low-frequency components**.
✅ **Role 5: Discards high-frequency components** that are less perceivable to the human eye.
✅ **Role 7:** Provides a **compact representation** of the image for **efficient storage and transmission**.

20.

✅ Final Complete Steps of JPEG Compression:

1️⃣ Split image into 8×8 blocks
2️⃣ Level shift (subtract 128)
3️⃣ Apply DCT
4️⃣ Quantization (divide by Q matrix)
5️⃣ Zigzag scanning
6️⃣ DPCM for DC component
7️⃣ RLE for AC coefficients
8️⃣ Entropy coding (Huffman/Arithmetic)

21.

Feature / Parameter	MPEG-2	MPEG-4 AVC (H.264)
Application	Digital TV, DVD video	Internet streaming, Blu-ray, HDTV, mobile video
Bit rate of HD video	16–24 Mbps	8–10 Mbps
Hardware encoder chip	Widely available	Requires more complex encoder chip
Software HD video processing in PC	Difficult / limited	Possible (efficient compression, suitable for PC processing)

22.
Recommended method: JPEG2000 (Wavelet-based compression)
Reason: Provides high compression efficiency, preserves critical image details, supports both lossy and lossless modes, and is ideal for transmitting high-resolution satellite images over limited bandwidth.

23.

✅ **Correct order of JPEG image compression stages:**

Step 1: Transform the image into a suitable color space (e.g., RGB → YCbCr) and apply chroma subsampling.
Step 2: Apply the Discrete Cosine Transform (DCT) to each 8×8 block of pixels to separate frequency components.
Step 3: Quantize the DCT coefficients using visually weighted quantization tables.
Step 4: Encode the quantized DCT coefficients using a Huffman variable-length coding scheme to reduce redundancy.

✅ **Final Order: 3 → 1 → 2 → 4**

24.

Answer:
Total storage required ≈ **173 MB**

Steps (short):

1. Pixels/page = (9.5×350)×(13×350) = 15,128,750
2. Bytes/page = 15,128,750 × 1 = 15,128,750 bytes = 14.43 MB
3. After 20% compression → 14.43×0.8 = 11.54 MB/page
4. For 15 pages → 11.54×15 = **≈173 MB**

25.

Given:

- $m = [1\ 0\ 0]$, $g(X) = X^2 + X + 1$

Steps (short):

1. Multiply message by X^r: $m(X)X^2 = X^4$
2. Divide by generator $g(X)$ → remainder $r(X) = X$
3. Systematic codeword: $c(X) = m(X)X^2 + r(X) = X^4 + X$
4. Codeword bits: **[1 0 0 1 0]** ✅

Answer: [1 0 0 1 0]

26.

Short Answer with Steps:

Given: (7,4) Hamming code, $g(D) = 1 + D + D^2$, received vectors:

$$y_1 = [1\ 0\ 1\ 1\ 1\ 1\ 0], \quad y_2 = [1\ 0\ 1\ 1\ 1\ 0\ 0]$$

Step 1: Calculate syndrome

$$S = y(D) \mod g(D)$$

- **y₁**: $S_1 = D + 1 \neq 0$ → error detected
- **y₂**: $S_2 = D^2 + D + 1 \neq 0$ → error detected

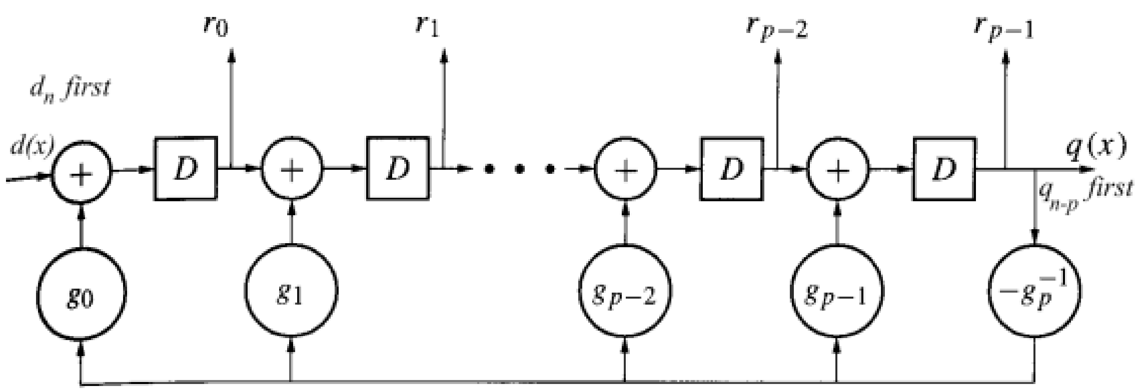
Step 2: Check error correction capability

- (7,4) Hamming → can **correct 1-bit errors only**
- **y_1:** 1-bit error → **can correct**
- **y_2:** 2-bit errors → **cannot correct, only detect**

✅ Answer

- **y_1 → Correctable**
- **y_2 → Only detectable, not correctable**

27.



29.

CRC Codes	Generator Polynomial
CRC-8	$x^8 + x^2 + x + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
CRC-64	$x^{64} + x^4 + x^3 + x + 1$