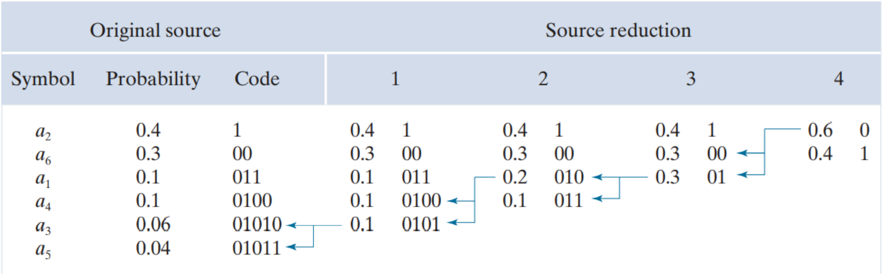
USN ☐☐☐☐☐☐☐☐☐☐

Internal Assessment Test 2 – Oct 2023
SOLUTION & SCHEME

<table>
<tr><td>Sub:</td><td colspan="5">Digital Image Processing</td><td>Sub Code:</td><td>18CS741</td><td>Branch:</td><td>CSE</td></tr>
<tr><td>Date:</td><td></td><td>Duration:</td><td>90 mins</td><td>Max Marks:</td><td>50</td><td>Sem / Sec:</td><td colspan="2">7(A,B,C)</td><td>OBE</td></tr>
</table>

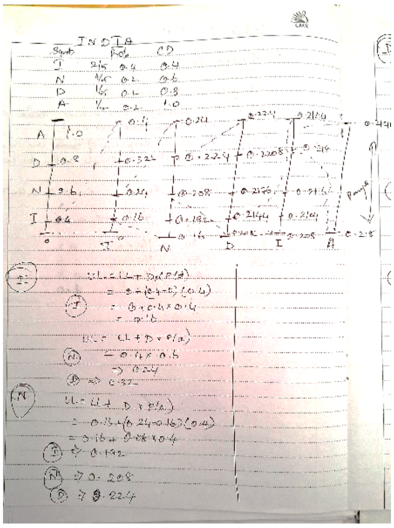
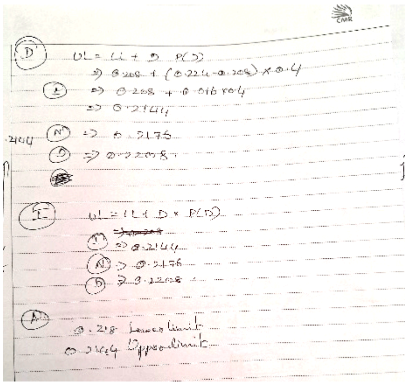
<table>
<tr><th colspan="2"><u>Answer any FIVE FULL Questions</u></th><th>MARKS</th><th>CO</th><th>RBT</th></tr>
<tr><td>1a</td><td>Identify which compression coding method yields the smallest possible number of code symbols per source symbol.
Sol: Huffman coding technique</td><td>2</td><td>3</td><td>L2</td></tr>
<tr><td>1b</td><td>Given a table of source symbols and probability, apply the above identified compression coding and decode the following message “ 010100111100”
<table>
<tr><th>symbols</th><th>probability</th></tr>
<tr><td>a2</td><td>0.4</td></tr>
<tr><td>a6</td><td>0.3</td></tr>
<tr><td>a1</td><td>0.1</td></tr>
<tr><td>a4</td><td>0.1</td></tr>
<tr><td>a3</td><td>0.06</td></tr>
<tr><td>a5</td><td>0.04</td></tr>
</table>
Sol:

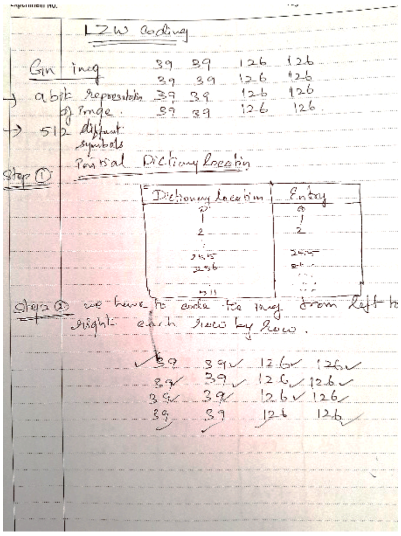
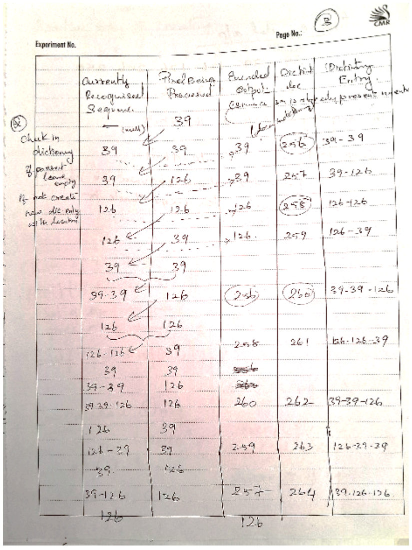
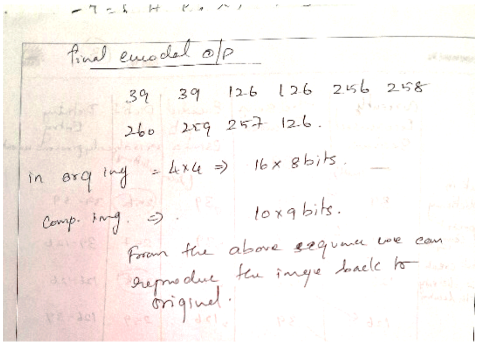
FIGURE 8.8 Huffman code assignment procedure.

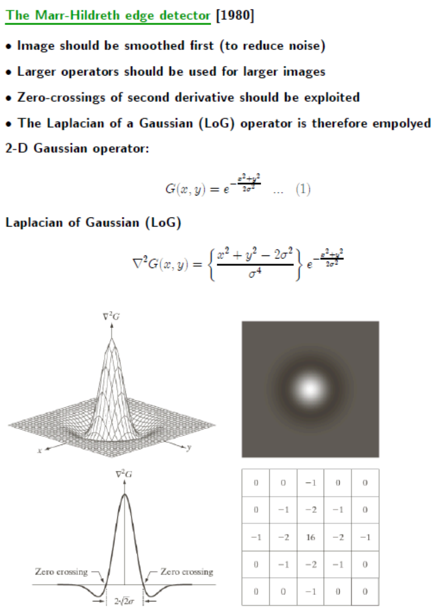
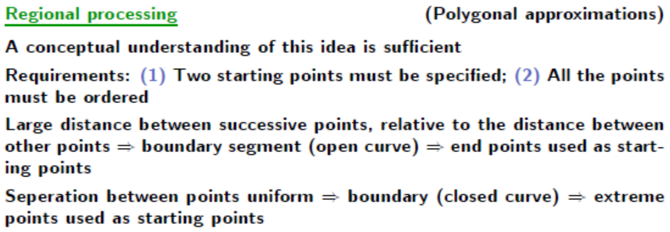
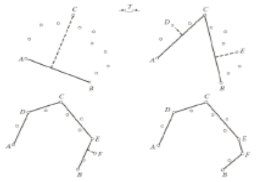


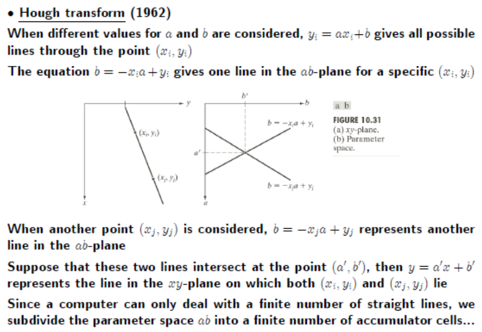
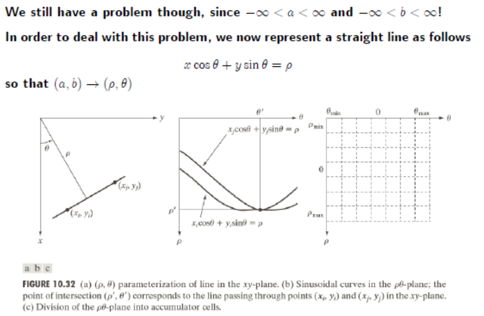
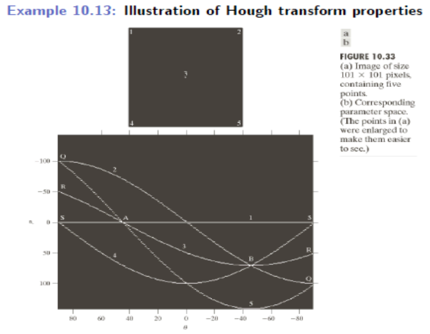
<table>
<tr><th colspan="3">Original source</th><th colspan="8">Source reduction</th></tr>
<tr><th>Symbol</th><th>Probability</th><th>Code</th><th colspan="2">1</th><th colspan="2">2</th><th colspan="2">3</th><th colspan="2">4</th></tr>
<tr><td>a_2</td><td>0.4</td><td>1</td><td>0.4</td><td>1</td><td>0.4</td><td>1</td><td>0.4</td><td>1</td><td>0.6</td><td>0</td></tr>
<tr><td>a_6</td><td>0.3</td><td>00</td><td>0.3</td><td>00</td><td>0.3</td><td>00</td><td>0.3</td><td>00</td><td>0.4</td><td>1</td></tr>
<tr><td>a_1</td><td>0.1</td><td>011</td><td>0.1</td><td>011</td><td>0.2</td><td>010</td><td>0.3</td><td>01</td><td></td><td></td></tr>
<tr><td>a_4</td><td>0.1</td><td>0100</td><td>0.1</td><td>0100</td><td>0.1</td><td>011</td><td></td><td></td><td></td><td></td></tr>
<tr><td>a_3</td><td>0.06</td><td>01010</td><td>0.1</td><td>0101</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>a_5</td><td>0.04</td><td>01011</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Decode message : “ 010100111100” == a3 a1 a2 a2 a6</td><td>8</td><td>3</td><td>L3</td></tr>
</table>

2a	The arithmetic decoding process is the reverse of the encoding procedure. Given a message source sequence “INDIA” generate non block arithmetic code. Sol:  INDIA Symb Prob CD I 2/5 0.4 0.4 N 1/5 0.2 0.6 D 1/5 0.2 0.8 A 1/5 0.2 1.0 I UL = LL + D×(P(A)) = 0+(0.4−0)(0.4) I = 0+0.4×0.4 = 0.16 UL = LL + D × P(a) N = 0.4×0.6 => 0.24 D => 0.32 N UL = LL + D × P(a) = 0.16+(0.24−0.16)(0.4) = 0.16+0.08×0.4 I => 0.192 N => 0.208 D => 0.224  D UL = LL + D P(D) => 0.208 + (0.224−0.208)×0.4 I => 0.208 + 0.016×0.4 => 0.2144 N => 0.2176 D => 0.2208 I UL = LL + D × P(D) I => 0.2144 N => 0.2176 D => 0.2208 A 0.218 Lower limit 0.2144 Upper limit	10	3	L3
3a	Find which compression method assigns fixed length code word to variable length sequence of source symbols. Sol: LZW coding technique	2	3	L2

<table>
<tr><td>3b</td><td>Given an 8 bit image, compress the image with the above method identified.
39 39 126 126
39 39 126 126
39 39 126 126
39 39 126 126
Sol:

LZW Coding
Gn img 39 39 126 126
39 39 126 126
→ 9 bit representation of image. 39 39 126 126
39 39 126 126.
→ 512 different symbols
Step ① Initial Dictionary location
<table>
<tr><th>Dictionary location</th><th>Entry</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>2</td><td>2</td></tr>
<tr><td>⋮</td><td></td></tr>
<tr><td>255</td><td>255</td></tr>
<tr><td>256</td><td>-</td></tr>
<tr><td>⋮</td><td>⋮</td></tr>
<tr><td>511</td><td></td></tr>
</table>
Step ② we have to code the img from left to right each row by row.
✓39 39✓ 126✓ 126✓
39✓ 39✓ 126✓ 126✓
39✓ 39✓ 126✓ 126✓
39✓ 39✓ 126✓ 126✓

Experiment No. Page No.: ③
② Check in dictionary if present leave empty. If not create new dic. entry with location
<table>
<tr><th>Currently Recognised Sequence</th><th>Pixel Being Processed</th><th>Encoded Output</th><th>Dictionary loc</th><th>Dictionary Entry</th></tr>
<tr><td>— (null)</td><td>39</td><td></td><td></td><td></td></tr>
<tr><td>39</td><td>39</td><td>39</td><td>256</td><td>39-39</td></tr>
<tr><td>39</td><td>126</td><td>39</td><td>257</td><td>39-126</td></tr>
<tr><td>126</td><td>126</td><td>126</td><td>258</td><td>126-126</td></tr>
<tr><td>126</td><td>39</td><td>126.</td><td>259</td><td>126-39</td></tr>
<tr><td>39</td><td>39</td><td></td><td></td><td></td></tr>
<tr><td>39-39</td><td>126</td><td>256</td><td>260</td><td>39-39-126</td></tr>
<tr><td>126</td><td>126</td><td></td><td></td><td></td></tr>
<tr><td>126-126</td><td>39</td><td>258</td><td>261</td><td>126-126-39</td></tr>
<tr><td>39</td><td>39</td><td>~~256~~</td><td></td><td></td></tr>
<tr><td>39-39</td><td>126</td><td>~~260~~</td><td></td><td></td></tr>
<tr><td>39-39-126</td><td>126</td><td>260</td><td>262</td><td>39-39-126-126</td></tr>
<tr><td>126</td><td>39</td><td></td><td></td><td></td></tr>
<tr><td>126-39</td><td>39</td><td>259</td><td>263</td><td>126-39-39</td></tr>
<tr><td>39</td><td>126</td><td></td><td></td><td></td></tr>
<tr><td>39-126</td><td>126</td><td>257</td><td>264</td><td>39.126.126</td></tr>
<tr><td>126</td><td></td><td>126</td><td></td><td></td></tr>
</table>

Final encoded o/p
39 39 126 126 256 258
260 259 257 126.
in org img = 4×4 ⇒ 16 × 8 bits.
Comp. img. ⇒ 10×9 bits.
From the above sequence we can reproduce the image back to original.</td><td>8</td><td>3</td><td>L3</td></tr>
<tr><td>4</td><td>Explain the principle types of data redundancies
Sol: detailed explanation on :
1. Coding redundancy
2. Spatial or temporal (interpixel) redundancy
3. Psychovisual redundancy (irrelevant information)</td><td>10</td><td>3</td><td>L3</td></tr>
<tr><td>5a</td><td>Write a formal definition of image segmentation. Explain the various gray level discontinuities in image processing
Sol:
Image segmentation definition- Subdivision of an image into its constituent parts
Segmentation based on discontinuities
(1) Isolated points
(2) Lines
(3) Edges</td><td>5</td><td>3</td><td>L2</td></tr>
</table>

5b	Identify and explain which edge detecting algorithm uses the advantage of Laplacian operator and Gaussian function. Sol: The Marr-Hildreth edge detector  The Marr-Hildreth edge detector [1980] • Image should be smoothed first (to reduce noise) • Larger operators should be used for larger images • Zero-crossings of second derivative should be exploited • The Laplacian of a Gaussian (LoG) operator is therefore empolyed 2-D Gaussian operator: $$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \ldots \quad (1)$$ Laplacian of Gaussian (LoG) $$\nabla^2 G(x, y) = \left\{\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right\} e^{-\frac{x^2+y^2}{2\sigma^2}}$$ 0 0 −1 0 0 0 −1 −2 −1 0 −1 −2 16 −2 −1 0 −1 −2 −1 0 0 0 −1 0 0	5	3	L2
6a	Discuss the need for edge linking and also explain local and regional processing approaches to edge linking. Sol: Edge detection is always followed by edge linking Local processing • Analyze pixels in small neighborhood of each edge point • Pixels that are similar are linked  Regional processing (Polygonal approximations) A conceptual understanding of this idea is sufficient Requirements: (1) Two starting points must be specified; (2) All the points must be ordered Large distance between successive points, relative to the distance between other points ⇒ boundary segment (open curve) ⇒ end points used as starting points Seperation between points uniform ⇒ boundary (closed curve) ⇒ extreme points used as starting points  a b c d FIGURE 10.28 Illustration of the iterative polygonal fit algorithm.	5	3	L3

6b	Explain global processing using the Hough transform. Sol:  • **Hough transform** (1962) When different values for a and b are considered, $y_i = ax_i + b$ gives all possible lines through the point (x_i, y_i) The equation $b = -x_i a + y_i$ gives one line in the ab-plane for a specific (x_i, y_i) a b FIGURE 10.31 (a) xy-plane. (b) Parameter space. When another point (x_j, y_j) is considered, $b = -x_j a + y_j$ represents another line in the ab-plane Suppose that these two lines intersect at the point (a', b'), then $y = a'x + b'$ represents the line in the xy-plane on which both (x_i, y_i) and (x_j, y_j) lie Since a computer can only deal with a finite number of straight lines, we subdivide the parameter space ab into a finite number of accumulator cells...  We still have a problem though, since $-\infty < a < \infty$ and $-\infty < b < \infty$! In order to deal with this problem, we now represent a straight line as follows $x \cos\theta + y \sin\theta = \rho$ so that $(a, b) \rightarrow (\rho, \theta)$ a b c FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.  **Example 10.13: Illustration of Hough transform properties** a b FIGURE 10.33 (a) Image of size 101 × 101 pixels, containing five points. (b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)	5	3	L2