DATA REPRESENTATION PAPER 3

Section A

1) • Explain the **limitations of ASCII encoding** and why Unicode was developed.
2) • Write a step-by-step solution to **convert 183** to binary.
3) • Describe the **algorithm for two's complement addition** with examples.
4) • Convert the binary number **11011100** to both hexadecimal and decimal.
5) • Write a solution for converting **floating-point decimal** to binary.
6) • Explain how **Unicode supports global characters** beyond ASCII.
7) • Describe the **process of binary division** with an example.
8) • Write a **subtraction algorithm** using two's complement for **-37 - 15**.
9) • Convert the decimal number **789** to hexadecimal and binary.
10) • Explain the role of **Big-O notation** in analyzing compression algorithms.
11) • Describe **lossless compression** and its uses in digital storage.
12) • Explain the **application of RLE** in reducing bitmap file size.
13) • Write a **recursive algorithm** for binary-to-decimal conversion.
14) • Explain the differences in **file size** between vector and bitmap images.
15) • Convert the decimal number **450** to an 8-bit binary.
16) • Describe **bit-depth** implications on audio and image quality.
17) • Explain how **two's complement subtraction** works.
18) • Describe the limitations of **Unicode** for future-proofing character sets.
19) • How is **Run-Length Encoding** used in multimedia files?
20) • Describe the role of **metadata** in video file formats.
21) • Convert the **decimal 275** to hexadecimal.
22) • Explain **digitization of analog signals** in audio recording.
23) • Describe **compression artifacts** and their visual impact.
24) • Write an **algorithm to calculate file size** for bitmap images.
25) • Describe **vector graphic transformations** without quality loss.
26) • Explain **audio sampling** and the effects of different rates.
27) • Convert the binary **10011101** to two's complement and find its decimal.
28) • Explain **sampling frequency** and its importance in audio fidelity.
29) • Describe the **differences between lossy and lossless audio compression**.
30) • Write a step-by-step conversion for **D0A** from hexadecimal to decimal.
31) • How does **color depth** influence pixel density?

32) • Explain the process of **digital-to-analog conversion** for audio playback.
33) • Calculate the bit rate for **1920x1080, 24-bit** color depth at 30fps.
34) • Explain how **Unicode encodes emojis and symbols**.
35) • Convert **11110101** to an equivalent 8-bit signed two's complement.
36) • Describe **error correction** techniques in file compression.
37) • Explain the implications of **lossy compression** on high-fidelity audio.
38) • Describe the **structure of vector graphics** in a drawing file.
39) • Convert the decimal **425** to binary and hexadecimal.
40) • Write an algorithm for **decimal-to-hexadecimal conversion**.
41) • Describe **Huffman encoding** and its role in data compression.
42) • Explain the differences between **ASCII, Extended ASCII, and Unicode**.
43) • Write a **lossless algorithm** for RLE on a simple image.
44) • Describe the **structure of a JPEG file** and its compression method.
45) • Convert **10000011** in binary to two’s complement and decimal.
46) • Explain the impact of **compression artifacts** on video quality.
47) • Write an algorithm for **binary to BCD** conversion.
48) • Describe **Unicode’s structure** and its effect on memory.
49) • Convert **FFFF** from hexadecimal to decimal and binary.
50) • Explain the impact of **lossy compression** on multimedia data.

Section B

1) • Write pseudocode to create a **hashing function for alphanumeric data**.
2) • Explain the steps of **normalizing a binary floating-point number**.
3) • Describe **open hashing vs closed hashing** with examples.
4) • Write a **collision handling function** for a hashing table.
5) • Describe the process of **converting a decimal number to binary floating-point**.
6) • Explain **Run-Length Encoding** with a code example.
7) • Describe the difference between **ASCII and Unicode** in hashing.
8) • Write pseudocode to implement **direct access in a random file organization**.
9) • Explain the advantages of **binary floating-point normalization**.
10) • Describe **hashing collisions** and provide handling techniques.
11) • Explain **binary multiplication and division** in two's complement.
12) • Write a program to simulate **sequential file access** with direct access.
13) • Describe **precision errors** in floating-point representation.
14) • Explain the **trade-offs between precision and range**.
15) • Describe how **open hash and closed hash** work together in a system.
16) • Write pseudocode to **convert denary to binary floating-point notation**.
17) • Explain **overflow and underflow errors** with floating-point calculations.
18) • Write a hashing function for **storing customer data** in a file.
19) • Describe **Run-Length Encoding compression** in file systems.
20) • Write pseudocode for **two's complement binary addition**.
21) • Explain the significance of **underflow errors** in large datasets.
22) • Describe **binary arithmetic** for floating-point normalization.
23) • Write an algorithm for **converting binary to hexadecimal**.
24) • Describe **binary floating-point normalization techniques**.
25) • Write pseudocode for **collision resolution using closed hashing**.
26) • Explain **overflow and underflow** with practical examples.

27) • Describe how **ASCII values** are used in hashing tables.
28) • Write a pseudocode algorithm for **Run-Length Encoding**.
29) • Explain **mantissa and exponent calculation** in binary floating-point numbers.
30) • Describe **precision vs range trade-offs** with an example.
31) • Write pseudocode for **binary division using two's complement**.
32) • Explain **hashing with file indexing** in random access files.
33) • Describe **file organization methods** and their applications.
34) • Write pseudocode to normalize a **two's complement floating-point number**.
35) • Explain the impact of **underflow in data processing systems**.
36) • Write an algorithm to convert **floating-point binary to denary**.
37) • Describe the **limitations of ASCII in character encoding**.
38) • Explain the significance of **precision and range** in data representation.
39) • Write pseudocode for **binary multiplication using two's complement**.
40) • Describe the **overflow area concept** in closed hashing.
41) • Explain the difference between **direct and sequential file access**.
42) • Write pseudocode for a **binary-to-decimal converter** for floating-point numbers.
43) • Explain **two's complement arithmetic** and its use in data representation.
44) • Write an algorithm for **detecting underflow in floating-point numbers**.
45) • Explain the **role of mantissa and exponent** in floating-point normalization.
46) • Describe the use of **ASCII values in simple hashing functions**.
47) • Explain **precision loss** in binary floating-point calculations.
48) • Write pseudocode for **decimal to binary floating-point conversion**.
49) • Describe **overflow handling** in data representation.
50) • Explain **file organization and access methods** with practical applications.