1. **What is an algorithm?**
   → A finite, well-defined sequence of steps to solve a specific problem.
2. **What is the role of algorithms in computing?**
   → Algorithms decide how efficiently and effectively a task can be performed by a computer.
3. **What are characteristics of a good algorithm?**
   → Finiteness, definiteness, input, output, and effectiveness.
4. **What is meant by "Algorithm as Technology"?**
   → It means treating algorithms as core technological tools that impact efficiency and scalability.
5. **What is correctness of an algorithm?**
   → An algorithm is correct if it produces the expected output for every valid input.
6. **How to confirm correctness of an algorithm?**
   → Using loop invariants, mathematical induction, and testing.
7. **What is an iterative algorithm?**
   → An algorithm that repeats a set of steps until a condition is satisfied.
8. **What are iterative algorithm design issues?**
   → Loop control, termination condition, and initialization.
9. **What are principles of problem solving?**
   → Understand the problem, plan a solution, execute it, and verify the result.
10. **What are problem-solving strategies?**
    → Divide & Conquer, Greedy, Dynamic Programming, Backtracking, Branch and Bound.
11. **What are classifications of problems?**
    → Simple, complex, and computational problems.
12. **What are different time complexities?**
    → Constant O(1), Logarithmic O(log n), Linear O(n), Quadratic $O(n^2)$, Exponential $O(2^n)$.
13. **What is the Tower of Hanoi problem?**
    → A recursive puzzle involving moving disks between rods following certain rules.
14. **What is the time complexity of Tower of Hanoi?**
    → $O(2^n - 1)$.

---

15. **What is algorithm analysis?**
    → The study of an algorithm's performance in terms of time and space complexity.
16. **What is input size in algorithm analysis?**
    → The measure of the amount of data the algorithm processes.
17. **What are best, worst, and average cases?**
    → Best = minimum time, Worst = maximum time, Average = expected time.
18. **What are dominant operations?**
    → The most frequent or time-consuming operations in an algorithm.
19. **What is asymptotic notation?**
    → Mathematical notation used to describe algorithm growth rates.
20. **Define Big O notation.**
    → Represents the upper bound or worst-case performance.
21. **Define Omega (Ω) notation.**
    → Represents the lower bound or best-case performance.

22. **Define Theta (Θ) notation.**
→ Represents the tight bound or average performance.
23. **Define little-o (o) and little-omega (ω).**
→ o means strictly smaller order; ω means strictly greater order.
24. **What are polynomial problems?**
→ Problems solvable in polynomial time, e.g., $O(n^2)$.
25. **What are non-polynomial problems?**
→ Problems requiring exponential or super-polynomial time.
26. **What is a deterministic algorithm?**
→ Produces the same output for a given input every time.
27. **What is a non-deterministic algorithm?**
→ May explore multiple paths or choices at once.
28. **What is a P-class problem?**
→ Problems solvable in polynomial time.
29. **What is an NP-class problem?**
→ Problems whose solutions can be verified in polynomial time.
30. **What is NP-complete?**
→ Problems that are both NP and as hard as any NP problem (e.g., 3-SAT).
31. **What is NP-hard?**
→ Problems that are at least as hard as NP-complete ones.
32. **What is polynomial reduction?**
→ Transforming one problem into another within polynomial time.
33. **Example of NP-complete problem?**
→ Vertex Cover or 3-SAT.
34. **Example of NP-hard problem?**
→ Hamiltonian Cycle.

---

you make the best possible choice at every step, without worrying about the future.

35. **What is the Greedy strategy?**
→ It chooses the best local option at each step hoping for a global optimum.
36. **What is the principle of Greedy method?**
→ Make the locally optimal choice at every step.
37. **Examples of greedy algorithms?**
→ Job Scheduling, Activity Selection, Fractional Knapsack.
38. **What is control abstraction in Greedy?**
→ The general structure defining the selection and feasibility test.
39. **What is time analysis of Greedy method?**
→ Usually O(n log n) due to sorting steps.
40. **What is Dynamic Programming (DP)?**
→ A method that solves overlapping subproblems and stores results.
41. **What is the principle of DP?**
→ Optimal substructure and overlapping subproblems.
42. **Examples of DP problems?**
→ 0/1 Knapsack, Matrix Chain Multiplication, OBST, Fibonacci.
43. **Difference between Greedy and DP?**
→ Greedy takes local decisions; DP ensures global optimum using stored results.
44. **What is OBST?**
→ Optimal Binary Search Tree minimizes average search cost.

45. **Case study example of DP?**
→ Rail tracks connecting cities.

---

46. **What is Backtracking?**
→ A method to build solutions incrementally and abandon invalid paths.
47. **What is the principle of Backtracking?**
→ Try possible options, backtrack when constraints are violated.
48. **Examples of Backtracking problems?**
→ 8-Queens, Graph Coloring, Subset Sum.
49. **What is control abstraction in Backtracking?**
→ Recursive exploration with constraint checks.
50. **What is the time complexity of Backtracking?**
→ Exponential, usually $O(2^n)$.
51. **What is Branch and Bound?**
→ A method to solve optimization problems using bounding and pruning.
52. **What are FIFO, LIFO, and LC strategies?**
→ FIFO = Breadth-first, LIFO = Depth-first, LC = Least-cost search.
53. **Examples of Branch and Bound problems?**
→ TSP, Knapsack.
54. **Difference between Backtracking and Branch & Bound?**
→ Backtracking finds feasible solutions; Branch & Bound finds optimal ones.
55. **Case study example for Branch & Bound?**
→ Airline Crew Scheduling.

---

56. **What is Amortized Analysis?**
→ It finds the average cost per operation over a sequence of operations.
57. **What are the methods of Amortized Analysis?**
→ Aggregate, Accounting, and Potential methods.
58. **What is Aggregate Analysis?**
→ Total cost divided by number of operations.
59. **What is Accounting Method?**
→ Assigns "credits" to operations to offset expensive ones.
60. **What is Potential Method?**
→ Uses potential function to measure stored work.
61. **Example of Amortized Analysis?**
→ Binary Counter, Stack push/pop.
62. **What is Time-Space Tradeoff?**
→ Balancing between execution time and memory usage.
63. **What are Tractable problems?**
→ Solvable efficiently in polynomial time.
64. **What are Non-tractable problems?**
→ Require exponential time or impossible to solve efficiently.
65. **What are Randomized Algorithms?**
→ Use randomness to optimize performance (e.g., QuickSort).
66. **What are Approximation Algorithms?**
→ Provide near-optimal solutions when exact ones are infeasible.

67. **What are Embedded Algorithms?**
    → Algorithms designed for resource-limited embedded systems.
68. **Examples of Embedded Algorithms?**
    → Power-optimized scheduling, sorting algorithms for embedded systems.
69. **Case study example for Amortized Analysis?**
    → Cutting Stock Problem.

---

70. **What are Multithreaded Algorithms?**
    → Algorithms using multiple threads to perform computations in parallel.
71. **What are performance measures of Multithreaded Algorithms?**
    → Work, Span, and Parallelism.
72. **What is a Race Condition?**
    → When multiple threads access shared data simultaneously causing unpredictable results.
73. **What is a Parallel Loop?**
    → A loop whose iterations execute concurrently.
74. **Examples of Multithreaded Algorithms?**
    → Matrix Multiplication, Merge Sort.
75. **What are Distributed Algorithms?**
    → Algorithms running across multiple interconnected nodes.
76. **Examples of Distributed Algorithms?**
    → Distributed BFS, Distributed MST.
77. **What is String Matching?**
    → Finding occurrences of a pattern inside a text.
78. **What is the Naive String Matching Algorithm?**
    → Compares pattern with text at every possible position.
79. **What is the Rabin-Karp Algorithm?**
    → Uses hashing to compare patterns efficiently.
80. **Difference between Multithreaded and Distributed Algorithms?**
    → Multithreaded works on shared memory; Distributed uses message passing among systems.

Nice bro 🔥 — tu **DAA Lab (Group A)** syllabus share kelास – म्हणजे oral madhe sir basically विचारतो practical related Q&A — म्हणजे concept + output + logic + complexity type.
So खाली मी देतो तुझे **DAA Lab Oral Questions with neat answers (list + one line gap each)** — all 6 assignments + mini project covered 🧠💻

⚙️ DAA Laboratory — Oral Question Bank (Group A)

---

1. Fibonacci Numbers and Step Count

1. **What is the Fibonacci series?**
   → A sequence where each term is the sum of the two preceding ones, i.e., F(n)=F(n-1)+F(n-2).
2. **What are the first six Fibonacci numbers?**
   → 0, 1, 1, 2, 3, 5.
3. **How can Fibonacci numbers be generated?**
   → Using recursion or iteration.
4. **What is the time complexity of the recursive Fibonacci algorithm?**
   → $O(2^n)$.
5. **What is the time complexity of the iterative Fibonacci algorithm?**
   → O(n).
6. **Why is step count used in this program?**
   → To measure how many basic operations the algorithm performs for a given input.
7. **What are the base conditions in Fibonacci recursion?**
   → F(0)=0 and F(1)=1.
8. **Which approach is more efficient, recursion or iteration?**
   → Iteration, because recursion repeats calculations and causes exponential growth.
9. **What is dynamic programming version of Fibonacci?**
   → It stores computed results to avoid recomputation (O(n)).
10. **What is the output for n=6?**
    → 0 1 1 2 3 5.

---

2. Job Sequencing with Deadlines (Greedy Method)

11. **What is the goal of the job sequencing problem?**
    → To schedule jobs to maximize total profit before their deadlines.
12. **Which algorithmic strategy is used here?**
    → Greedy method.
13. **What is the greedy choice property used in this problem?**
    → Select the job with maximum profit that fits within its deadline.
14. **How are jobs arranged before scheduling?**
    → Sorted in descending order of profit.
15. **What happens if two jobs have the same deadline?**
    → The job with higher profit is chosen first.
16. **What data structure is commonly used for time slots?**
    → An array representing job time slots.
17. **What is the time complexity of this algorithm?**
    → $O(n^2)$.
18. **What is the main objective function?**
    → Maximize total profit.
19. **What happens if all slots are full before a job's deadline?**
    → That job is skipped.
20. **Give one real-life example of job sequencing.**
    → Task scheduling in CPU or advertisement slot optimization.

---

3. Fractional Knapsack (Greedy Method)

21. **What is the fractional knapsack problem?**
    → A problem of selecting items to maximize profit, where fractions of items are allowed.
22. **Which strategy is used for fractional knapsack?**
    → Greedy method.
23. **What is the selection criterion?**
    → Highest profit/weight ratio first.
24. **Why is sorting required in this algorithm?**
    → To pick items in order of decreasing profit/weight ratio.
25. **Can items be divided in fractional knapsack?**
    → Yes, items can be divided to fill remaining capacity.
26. **What is the time complexity of this algorithm?**
    → O(n log n) due to sorting.
27. **Difference between 0/1 and fractional knapsack?**
    → 0/1: items not divisible; Fractional: items can be divided.
28. **What is the output of fractional knapsack problem?**
    → Maximum achievable profit.
29. **What is the main property of the greedy method used here?**
    → Local optimal choice leads to global optimum.
30. **Give one application of fractional knapsack.**
    → Resource allocation and cargo loading optimization.

---

4. 0/1 Knapsack (Dynamic Programming / Branch and Bound)

31. **What is 0/1 knapsack problem?**
    → Selecting items to maximize profit without splitting items.
32. **Which algorithmic strategies can be used?**
    → Dynamic Programming or Branch & Bound.
33. **What is the recursive relation used in DP knapsack?**
    → DP[i][w] = max(DP[i−1][w], profit[i−1] + DP[i−1][w − weight[i−1]]).
34. **What is the time complexity using DP?**
    → O(nW), where W is knapsack capacity.
35. **What does each cell of DP table represent?**
    → Maximum profit for given items and capacity.
36. **What is the space complexity of DP solution?**
    → O(nW).
37. **How does Branch and Bound approach differ?**
    → It explores solution tree using bounds to prune branches.
38. **What is bounding function in Branch and Bound?**
    → It estimates the upper bound of achievable profit for pruning.
39. **Why is 0/1 knapsack not solvable using greedy method?**
    → Because local decisions don't guarantee global optimum.
40. **Real-life application of knapsack?**
    → Investment planning, cargo loading, and memory allocation.

---

5. Binomial Coefficients (Dynamic Programming)

41. **What is a binomial coefficient?**
    → It represents number of ways to choose k elements from n items (nCk).
42. **Formula for binomial coefficient?**
    → C(n, k) = C(n−1, k−1) + C(n−1, k).
43. **What is the base condition?**
    → C(n, 0) = C(n, n) = 1.
44. **Which algorithmic technique is used here?**
    → Dynamic Programming.
45. **Why dynamic programming is better than recursion here?**
    → It stores previously computed results to avoid recomputation.
46. **What structure is used to store coefficients?**
    → A 2D array or table.
47. **What is the time complexity of this algorithm?**
    → $O(n^2)$.
48. **What is Pascal's Triangle?**
    → A triangular array of binomial coefficients.
49. **Give real-life application of binomial coefficients.**
    → Probability calculation and combinatorial analysis.
50. **What is the space complexity?**
    → $O(n^2)$.

---

6. 8-Queens Problem (Backtracking)

51. **What is the 8-Queen problem?**
    → Placing 8 queens on a chessboard such that no two attack each other.
52. **Which strategy is used here?**
    → Backtracking.
53. **What are the constraints for valid queen placement?**
    → No two queens share the same row, column, or diagonal.
54. **What is the base case in this problem?**
    → When all queens are placed successfully on the board.
55. **What happens when no position is safe for a queen?**
    → Backtracking occurs to previous queen's position.
56. **What is the complexity of 8-Queen problem?**
    → O(N!).
57. **What does a solution matrix contain?**
    → 1 indicates queen placed, 0 indicates empty cell.
58. **How many solutions exist for 8-Queen problem?**
    → 92 distinct solutions.
59. **What is the advantage of backtracking over brute force?**
    → It prunes invalid paths early, saving time.
60. **Real-life application of backtracking?**
    → Puzzle solving, pathfinding, and constraint satisfaction problems.

Mini Project — (Any One Example)

61. **What is the purpose of the mini project?**
    → To apply algorithmic design techniques to real-world problems.
62. **Give one example of a DAA mini project.**
    → Shortest path finder using Dijkstra's algorithm.
63. **Which data structure is used in shortest path problems?**
    → Graphs (Adjacency List or Matrix).
64. **What is the time complexity of Dijkstra's algorithm?**
    → $O(V^2)$ for matrix, $O((V+E) \log V)$ with priority queue.
65. **Example of a project using backtracking?**
    → Sudoku solver or graph coloring visualizer.
66. **Example of a project using greedy method?**
    → Task scheduling or job optimization.
67. **What is the importance of analyzing algorithm complexity?**
    → To predict efficiency and scalability.
68. **What is meant by correctness in a project algorithm?**
    → It should always produce valid and expected results for all inputs.
69. **What is the importance of testing your project?**
    → To ensure correctness and performance accuracy.
70. **What is the expected output of a DAA mini project?**
    → Optimized solution visualization or computed optimal result (depends on problem).

ML

UNIT I — Introduction to Machine Learning

1. **What is Machine Learning?**
   → Machine Learning is the process where systems learn patterns from data to make predictions or decisions without explicit programming.
2. **How is Machine Learning different from traditional programming?**
   → Traditional programming uses explicit rules, while ML learns rules automatically from data.
3. **What is Artificial Intelligence (AI)?**
   → AI is the broader concept of creating machines that can mimic human intelligence and reasoning.
4. **How is ML related to AI?**
   → ML is a subset of AI focused on systems that learn from data.
5. **Difference between ML and Data Science?**
   → ML builds models to make predictions, while Data Science focuses on data collection, analysis, and visualization including ML.
6. **What are types of learning in ML?**
   → Supervised, Unsupervised, Semi-supervised, and Reinforcement learning.
7. **What is Supervised Learning?**
   → Learning from labeled data to predict outcomes (e.g., classification, regression).
8. **What is Unsupervised Learning?**
   → Learning from unlabeled data to find hidden patterns (e.g., clustering).

9. **What is Semi-supervised Learning?**
   → Uses a mix of labeled and unlabeled data to improve accuracy.
10. **What is Reinforcement Learning?**
    → A learning method where an agent learns by interacting with an environment and receiving rewards or penalties.
11. **What are Geometric Models in ML?**
    → Models that use geometric decision boundaries, e.g., SVM, kNN.
12. **What are Probabilistic Models?**
    → Models based on probability distributions, e.g., Naive Bayes.
13. **What are Logical Models?**
    → Models based on rules and logic, e.g., Decision Trees.
14. **What are Grouping and Grading Models?**
    → Models that classify or rank items into groups or levels.
15. **What are Parametric Models?**
    → Models with a fixed number of parameters, e.g., Linear Regression.
16. **What are Non-Parametric Models?**
    → Models with parameters that grow with data, e.g., kNN.
17. **What is Learnability in ML?**
    → The ability of an algorithm to learn a function from data effectively.
18. **What are Data Formats in ML?**
    → Structured (tables), Unstructured (images, text), Semi-structured (JSON, XML).
19. **What is Statistical Learning Approach?**
    → Approach where algorithms use statistics to find patterns and make predictions.
20. **Case Study (Uber Sales Increase): What is the first step as a data scientist?**
    → Understand the business goal and collect relevant data like customer demand and pricing trends.

UNIT II — Feature Engineering

21. **What is a Feature?**
    → A measurable property or characteristic used as input to a model.
22. **What is Feature Engineering?**
    → Process of creating, transforming, and selecting input variables to improve model performance.
23. **What is Data Preprocessing?**
    → Cleaning and transforming raw data into a usable format for ML.
24. **What is Normalization?**
    → Scaling features between 0 and 1 to bring uniformity.
25. **What is Standardization?**
    → Scaling features to have zero mean and unit variance.
26. **How to handle missing values?**
    → Replace with mean, median, mode, or use predictive imputation.
27. **What is Dimensionality Reduction?**
    → Reducing the number of features while preserving most of the information.
28. **What is PCA (Principal Component Analysis)?**
    → A linear dimensionality reduction technique that converts correlated variables into uncorrelated principal components.

29. **What is Feature Extraction?**
    → Creating new features from existing data using mathematical transformations.
30. **What is Kernel PCA?**
    → A nonlinear version of PCA using kernel functions.
31. **What is Local Binary Pattern (LBP)?**
    → A texture descriptor used in image feature extraction.
32. **What is Feature Selection?**
    → Choosing the most important subset of features for model training.
33. **What is Sequential Forward Selection?**
    → Start with no features and keep adding features that improve performance most.
34. **What is Sequential Backward Selection?**
    → Start with all features and keep removing the least significant ones.
35. **What are count-based features?**
    → Features based on frequency of occurrence of values.
36. **What are statistical features?**
    → Features derived from statistics like mean, median, mode, variance.
37. **What is Multidimensional Scaling (MDS)?**
    → A technique to visualize similarity or dissimilarity between data points.
38. **What is Matrix Factorization?**
    → Breaking down a matrix into smaller matrices for latent feature extraction (e.g., recommender systems).
39. **Why is Feature Engineering important?**
    → It improves accuracy, reduces overfitting, and enhances model interpretability.
40. **Case Study (Marketing Campaign Data): What would be your approach?**
    → Analyze customer behavior, clean data, perform feature selection using correlation or PCA, and train supervised models.

UNIT III — Supervised Learning: Regression

41. **What is Regression?**
    → A supervised learning method to predict continuous outcomes.
42. **What is Bias?**
    → Error due to oversimplifying the model assumptions.
43. **What is Variance?**
    → Error due to sensitivity to small fluctuations in training data.
44. **What is Generalization?**
    → Model's ability to perform well on unseen data.
45. **What is Underfitting?**
    → When a model is too simple and fails to capture patterns.
46. **What is Overfitting?**
    → When a model learns noise and performs poorly on test data.
47. **What is Linear Regression?**
    → A model that finds the best linear relationship between independent and dependent variables.
48. **What is the cost function used in Linear Regression?**
    → Mean Squared Error (MSE).
49. **What is Gradient Descent?**
    → An optimization algorithm to minimize the cost function.

50. **What is Lasso Regression?**
→ Linear regression with L1 regularization that can shrink coefficients to zero.
51. **What is Ridge Regression?**
→ Linear regression with L2 regularization that penalizes large weights.
52. **What is the difference between Lasso and Ridge?**
→ Lasso performs feature selection; Ridge only shrinks coefficients.
53. **What are Regression Evaluation Metrics?**
→ MAE (Mean Absolute Error), RMSE (Root Mean Square Error), R^2 (Coefficient of Determination).
54. **What does R^2 value indicate?**
→ How well the regression line fits the data (0 to 1 scale).
55. **Case Study (Stock Price Prediction): Which model would you use?**
→ Linear regression or LSTM (for time series).

UNIT IV — Supervised Learning: Classification

56. **What is Classification?**
→ Predicting discrete class labels based on input features.
57. **What is k-Nearest Neighbour (kNN)?**
→ Classifies based on the majority class of the k closest data points.
58. **What is Support Vector Machine (SVM)?**
→ A model that finds the optimal separating hyperplane between classes.
59. **What is Ensemble Learning?**
→ Combining multiple models to improve accuracy.
60. **What is Bagging?**
→ Combining predictions of multiple models trained on different data samples.
61. **What is Boosting?**
→ Sequentially combining weak learners to create a strong learner.
62. **What is Random Forest?**
→ Ensemble of decision trees trained on random data subsets.
63. **What is AdaBoost?**
→ Adaptive boosting that focuses more on misclassified examples.
64. **What is Binary Classification?**
→ Classification between two classes (e.g., spam vs not spam).
65. **What is Multiclass Classification?**
→ Classification among more than two categories.
66. **What is Imbalanced Classification?**
→ When some classes have significantly fewer samples than others.
67. **What is One-vs-One Classification?**
→ Builds one classifier per pair of classes.
68. **What is One-vs-All Classification?**
→ Builds one classifier per class against all others.
69. **What is Accuracy?**
→ Ratio of correctly predicted samples to total samples.
70. **What is Precision?**
→ Ratio of true positives to total predicted positives.
71. **What is Recall?**
→ Ratio of true positives to total actual positives.

72. **What is F1-Score?**
→ Harmonic mean of precision and recall.
73. **What is Cross-Validation?**
→ Dividing data into k folds to test model performance multiple times.
74. **Difference between Micro and Macro averaging?**
→ Micro: aggregate across all samples; Macro: average across classes.
75. **Case Study (Thyroid Disorder Prediction): Which algorithm fits best?**
→ SVM or Random Forest for multiclass classification.

UNIT V — Unsupervised Learning

76. **What is Unsupervised Learning?**
→ Learning from unlabeled data to find structure or patterns.
77. **What is Clustering?**
→ Grouping similar data points based on features.
78. **What is K-Means Clustering?**
→ Partitions data into k clusters minimizing intra-cluster distance.
79. **What is K-Medoids Clustering?**
→ Similar to K-means but uses actual data points as cluster centers (medoids).
80. **What is Hierarchical Clustering?**
→ Builds a hierarchy of clusters using bottom-up (agglomerative) or top-down (divisive) approach.
81. **What is Density-Based Clustering?**
→ Clusters are formed based on dense regions of data points (e.g., DBSCAN).
82. **What is Spectral Clustering?**
→ Uses graph theory and eigenvalues of similarity matrix for clustering.
83. **What is Outlier Analysis?**
→ Detecting abnormal or rare data points differing from others.
84. **What is Isolation Factor?**
→ Detects anomalies by isolating observations randomly.
85. **What is Local Outlier Factor (LOF)?**
→ Measures local deviation of density around a data point.
86. **What is the Elbow Method?**
→ A method to determine optimal number of clusters in K-Means.
87. **What are Extrinsic and Intrinsic Evaluation Methods?**
→ Extrinsic uses external labels; Intrinsic measures within-cluster cohesion.
88. **Case Study (Customer Segmentation): Which algorithm would you use?**
→ K-Means or DBSCAN for market segmentation.

UNIT VI — Introduction to Neural Networks

89. **What is an Artificial Neural Network (ANN)?**
→ A computational model inspired by the human brain, consisting of interconnected neurons.
90. **What is a Single-Layer Neural Network?**
→ A network with one input layer and one output layer.

91. **What is a Multilayer Perceptron (MLP)?**
→ A neural network with one or more hidden layers.
92. **What is Backpropagation?**
→ Algorithm used to train neural networks by minimizing error using gradient descent.
93. **What is an Activation Function?**
→ Function that introduces non-linearity into the network (e.g., Sigmoid, ReLU, Tanh).
94. **What is a Functional Link ANN (FLANN)?**
→ ANN with additional functional links between nodes to improve learning.
95. **What is Radial Basis Function Network (RBFN)?**
→ Neural network that uses radial basis functions as activation functions.
96. **What is Recurrent Neural Network (RNN)?**
→ A neural network with feedback connections used for sequence data.
97. **What is Convolutional Neural Network (CNN)?**
→ A neural network designed for image and spatial data processing.
98. **What is the purpose of weights in a neural network?**
→ To determine the strength of connections between neurons.
99. **What is meant by training a neural network?**
→ Adjusting weights to minimize the error between predicted and actual outputs.
100. **Case Study (Neural Network Application): Give an example.**
→ Using CNN for image recognition or RNN for stock trend prediction.

BT

UNIT I — Mathematical Foundation for Blockchain

1. **What is Cryptography?**
→ Cryptography is the science of securing data through mathematical algorithms to ensure confidentiality and integrity.
2. **What is Symmetric Key Cryptography?**
→ It uses a single secret key for both encryption and decryption (e.g., AES, DES).
3. **What is Asymmetric Key Cryptography?**
→ It uses two keys — a public key for encryption and a private key for decryption (e.g., RSA).
4. **Difference between Symmetric and Asymmetric Cryptography?**
→ Symmetric is faster but less secure; Asymmetric is slower but more secure.
5. **What is Elliptic Curve Cryptography (ECC)?**
→ ECC is an asymmetric encryption technique using elliptic curves for smaller, faster, and more secure keys.
6. **What are Cryptographic Hash Functions?**
→ They convert any input data into a fixed-size hash value, used for verification.
7. **What is SHA-256?**
→ A secure hash algorithm that produces a 256-bit hash output, used in Bitcoin.
8. **What is a Digital Signature Algorithm (DSA)?**
→ It's an asymmetric algorithm used for verifying authenticity of digital messages.

9. **What are Merkle Trees?**
   → A hash-based tree data structure used to verify data integrity efficiently in blockchains.
10. **Why are Merkle Trees important in Blockchain?**
    → They help verify transactions quickly without downloading the entire block.
11. **What ensures immutability in Blockchain?**
    → Cryptographic hashing and chaining of blocks.
12. **Case Study: Compare Symmetric vs Asymmetric algorithms.**
    → Symmetric: AES, faster but less scalable.
    Asymmetric: RSA, more secure for digital signatures.

UNIT II — Blockchain Basics and Architecture

13. **What is Blockchain?**
    → A distributed ledger technology that records transactions securely and immutably across nodes.
14. **Difference between Centralized and Decentralized systems?**
    → Centralized has one control authority; Decentralized distributes control across participants.
15. **What are the layers of Blockchain?**
    → Application Layer, Execution Layer, Semantic Layer, Propagation Layer, Consensus Layer.
16. **What is the Application Layer in Blockchain?**
    → It handles user interfaces and decentralized apps (dApps).
17. **What is the Execution Layer?**
    → Executes smart contracts and transactions.
18. **What is the Consensus Layer?**
    → Ensures all participants agree on the blockchain state.
19. **Why is Blockchain important?**
    → It provides transparency, security, decentralization, and immutability.
20. **What are the limitations of centralized systems?**
    → Single point of failure, data tampering, and lack of transparency.
21. **What are the advantages of Blockchain adoption?**
    → Improved security, efficiency, trust, and automation.
22. **Name a few Blockchain applications.**
    → Cryptocurrency, Supply Chain, Healthcare, and e-Governance.
23. **Case Study: What do you study in Blockchain research papers?**
    → Focus on consensus models, security, scalability, and transaction speed.

UNIT III — Blockchain Platforms and Consensus

24. **What are Blockchain Platforms?**
    → Frameworks that enable creation and management of blockchain networks.
25. **Name different Blockchain Platforms.**
    → Bitcoin, Ethereum, Hyperledger, IOTA, Corda, R3.

26. **What is a Public Blockchain?**
→ A blockchain open to everyone (e.g., Bitcoin, Ethereum).
27. **What is a Private Blockchain?**
→ Restricted access blockchain used by organizations.
28. **What is a Consortium Blockchain?**
→ Semi-decentralized blockchain managed by multiple organizations.
29. **What is Consensus in Blockchain?**
→ A mechanism ensuring all nodes agree on the validity of transactions.
30. **What is Proof of Work (PoW)?**
→ A consensus algorithm requiring miners to solve cryptographic puzzles (Bitcoin uses it).
31. **What is the Byzantine Generals Problem?**
→ A situation describing how distributed systems achieve consensus despite unreliable nodes.
32. **What is Proof of Stake (PoS)?**
→ Consensus based on validator's stake (ownership of coins).
33. **What is Proof of Elapsed Time (PoET)?**
→ Consensus based on random wait times (used by Hyperledger Sawtooth).
34. **What is Proof of Activity?**
→ Hybrid of PoW and PoS ensuring participation and verification.
35. **What is Proof of Burn?**
→ Validators prove commitment by burning (destroying) coins.
36. **Why is consensus necessary in Blockchain?**
→ To maintain consistency and prevent double-spending.
37. **Case Study: Compare different consensus algorithms.**
→ PoW: secure but energy-hungry; PoS: efficient; PoET: low power.

UNIT IV — Cryptocurrency and Tokens

38. **What is Cryptocurrency?**
→ A digital currency that uses cryptography and blockchain for secure transactions.
39. **What is Bitcoin?**
→ The first decentralized cryptocurrency, introduced by Satoshi Nakamoto.
40. **What are the basics of Cryptocurrency?**
→ Based on blockchain, uses wallets, decentralized control, and cryptographic proof.
41. **What are the types of Cryptocurrency?**
→ Coins (Bitcoin, Ethereum) and Tokens (USDT, Chainlink).
42. **What is a Crypto Wallet?**
→ A digital tool for storing and managing cryptocurrency keys.
43. **Name some Crypto Wallets.**
→ Metamask, Coinbase, Binance.
44. **What is the difference between a hot and cold wallet?**
→ Hot wallet: online; Cold wallet: offline (more secure).
45. **What is a Private Key in a wallet?**
→ A secret code that authorizes crypto transactions.
46. **What is a Public Key?**
→ A cryptographic address visible to others for sending funds.

47. **What is Gas in crypto transactions?**
→ A fee paid for computation and storage on blockchain (especially Ethereum).
48. **Case Study: Create a wallet using Metamask. What will you observe?**
→ You can send, receive, and view Ether with transaction fee details.

UNIT V — Ethereum and Smart Contracts

49. **What is Ethereum?**
→ A decentralized platform supporting smart contracts and dApps.
50. **What are the types of Ethereum Networks?**
→ Mainnet, Testnet (Ropsten, Goerli, Sepolia), and Private networks.
51. **What is EVM (Ethereum Virtual Machine)?**
→ The runtime environment that executes smart contracts.
52. **What are Smart Contracts?**
→ Self-executing code stored on the blockchain.
53. **Purpose of Smart Contracts?**
→ Automate agreements and remove intermediaries.
54. **What are the types of Smart Contracts?**
→ Financial, Governance, and Utility contracts.
55. **Which language is used to write Smart Contracts?**
→ Solidity.
56. **What is Swarm in Ethereum?**
→ A decentralized storage platform.
57. **What is Whisper in Ethereum?**
→ A decentralized communication (messaging) protocol.
58. **What is Truffle?**
→ A development environment and framework for Ethereum smart contracts.
59. **Case Study: Truffle Development Environment?**
→ Provides tools to compile, deploy, and test Solidity contracts easily.

UNIT VI — Blockchain Case Studies

60. **What are Blockchain Applications?**
→ Blockchain is applied in Retail, Banking, Finance, Government, Healthcare, IoT, and Energy sectors.
61. **Use of Blockchain in Retail?**
→ Product tracking and supply chain transparency.
62. **Use of Blockchain in Banking and Finance?**
→ Fast, transparent, and secure transactions.
63. **Use of Blockchain in Healthcare?**
→ Secure medical record sharing and patient data management.
64. **Use of Blockchain in Government?**
→ Transparent e-voting and land registration systems.
65. **Use of Blockchain in IoT?**
→ Secure communication and data integrity between devices.

66. **Use of Blockchain in Energy Sector?**
→ Smart energy grids and peer-to-peer power trading.
67. **What is Blockchain Integration with Other Domains?**
→ Combining Blockchain with AI, IoT, Cloud, and Big Data for enhanced functionality.
68. **Case Study: How to write a Blockchain use case report?**
→ Include problem statement, platform used, consensus, and advantages.

💻 LAB ORAL QUESTIONS (Blockchain Lab SPPU)

---

69. **What is Metamask?**
→ A crypto wallet and gateway to Ethereum-based dApps.
70. **How do you install Metamask?**
→ Add it as a browser extension and create a new wallet with seed phrase.
71. **What is Ether in Ethereum?**
→ The native cryptocurrency used to pay for transactions.
72. **What is Gas in Ethereum?**
→ The computational cost paid for executing smart contracts.
73. **How can you observe Ether spending per transaction?**
→ By viewing transaction details on Etherscan or within Metamask.
74. **What is a Wallet Address?**
→ A unique public key identifier used for sending/receiving cryptocurrency.
75. **What is a Smart Contract?**
→ Self-executing code stored on blockchain automating tasks.
76. **Write operations of Bank Smart Contract.**
→ Deposit, Withdraw, and Check Balance.
77. **Which language is used for Smart Contracts?**
→ Solidity.
78. **What are the data structures used in Solidity?**
→ Arrays, Structs, and Mappings.
79. **What is a Fallback Function in Solidity?**
→ A special function executed when no other function matches or Ether is sent.
80. **What is Gas used for in Solidity programs?**
→ To pay miners for computation and storage.
81. **What are the types of Blockchains?**
→ Public, Private, and Consortium.
82. **Give real-time examples of Blockchain use cases.**
→ Supply Chain Tracking, Healthcare Data Sharing, and Voting Systems.
83. **What is a dApp (Decentralized App)?**
→ An application that runs on a blockchain network instead of centralized servers.
84. **Example of dApp Mini Project?**
→ E-Voting System using Ethereum Smart Contracts.
85. **What are advantages of dApps?**
→ Transparency, no downtime, and decentralization.

86. **What is Solidity Compiler (solc)?**
→ It compiles Solidity code into bytecode for deployment on EVM.
87. **What is Remix IDE?**
→ A browser-based IDE used for writing, compiling, and deploying smart contracts.
88. **What is a Transaction Fee?**
→ The amount of gas used multiplied by the gas price paid in Ether.
89. **How can you deploy a Smart Contract on a test network?**
→ Connect Metamask to testnet (like Sepolia), compile in Remix, and deploy.
90. **Why is Blockchain suitable for E-Voting?**
→ It ensures transparency, immutability, and security of votes.