Long questions

1. What are Intelligent Agents in AI, and how do they function?
2. What is the role of Problem-Solving Agents in AI?
3. Explain Breadth-first Search (BFS) in AI.
4. Describe Depth-first Search (DFS) and its application.
5. How does Uniform Cost Search work, and where is it used?
6. What is Iterative Deepening Depth-First Search (IDDFS)?
7. How does Bidirectional Search enhance search efficiency?
8. Explain Greedy Best-First Search and its uses
9. Describe A* Search Algorithm and its significance.
10. What are Heuristic Functions, and why are they important in AI search algorithms?
11. How do you implement Breadth-First Search (BFS) in Python for graph traversal?
12. Demonstrate Depth-First Search (DFS) with a Python code snippet for tree traversal.
13. Implement Uniform Cost Search (UCS) for a weighted graph in Python.
14. Code an Iterative Deepening Depth-First Search (IDDFS) in Python for a binary tree.
15. Write a Python function to demonstrate Greedy Best-First Search on a grid.
16. Explain the concept of Hill Climbing search in AI.
17. Describe the Simulated Annealing search technique.
18. What is Local Search in Continuous Spaces, and how is it applied?
19. How do you implement Hill Climbing search in Python?
20. Write a Python code snippet to demonstrate Simulated Annealing.
21. Explain the A* search algorithm's heuristic function and its importance.
22. How does the Greedy Best-First Search differ from A* search?
23. Describe the differences between Uninformed and Informed Search Strategies.
24. How is Simulated Annealing applied in solving optimization problems?
25. Implement A* search in Python for a grid-based pathfinding problem.
26. What are the advantages and limitations of using Heuristic Functions in search algorithms?
27. Explain the concept of Bidirectional Search and its application in AI.
28. How does Local Search differ from Global Search strategies in AI?
29. Implement Greedy Best-First Search in Python for finding a path in a graph.

30. What are the key considerations in choosing between different AI search strategies?
31. What is Adversarial Search in AI?
32. How do Optimal Decisions in Games work?
33. Explain Alpha–Beta Pruning in game playing.
34. Describe Imperfect Real-Time Decisions in games.
35. What defines Constraint Satisfaction Problems (CSPs)?
36. How does Constraint Propagation work in CSPs?
37. What is Backtracking Search for solving CSPs?
38. How does Local Search work for solving CSPs?
39. Discuss the Structure of Problems in relation to CSPs.
40. What are Knowledge-Based Agents in AI?
41. Describe The Wumpus World in AI.
42. What is Propositional Logic?
43. Explain Propositional Theorem Proving in AI.
44. What is Proof by Resolution in propositional logic?
45. Explain Horn Clauses and their significance in AI.
46. Describe Forward Chaining in AI.
47. Explain Backward Chaining in AI.
48. Discuss Effective Propositional Model Checking.
49. How are Agents Based on Propositional Logic designed?
50. Demonstrate Alpha–Beta Pruning with a Python example.
51. Create a Python program to solve a simple CSP using Backtracking.
52. Implement Forward Chaining in Python for a rule-based system.
Define a set of rules and facts in propositional logic representing a simple knowledge domain.
53. Code a Propositional Logic Truth Table Generator in Python.
54. Develop a Simple Expert System using Python for diagnosing computer issues.
55. How is Game Theory applied in Adversarial Search?
56. What is the role of Machine Learning in solving CSPs?
57. How does Propositional Logic differ from First-Order Logic (FOL)?
58. Implement a Python program for Alpha–Beta Pruning on a Tic-Tac-Toe game.
59. How is Natural Language Processing (NLP) applied to Constraint Satisfaction Problems?
60. Create a Python program to perform Propositional Logic Resolution.
61. What is First-Order Logic (FOL) in AI?

62. Explain the Representation of Knowledge in First-Order Logic.
63. Discuss the Syntax and Semantics of First-Order Logic.
64. How is First-Order Logic used in AI?
65. What is Knowledge Engineering in First-Order Logic?
66. Implement a Simple FOL Parser in Python.
67. Create a Prolog Program for Family Relationships.
68. Develop a Python-based FOL Theorem Prover.
69. Build a Rule-Based Expert System using PyKnow for Medical Diagnosis.
70. Implement an Ontology with OWL in Python for a Domain of Your Choice. Choose a specific domain for ontology development, such as education, business, or a hobby.
71. How does FOL support Natural Language Processing (NLP)?
72. What is the significance of Herbrand's Theorem in FOL?
73. Implement a Python Script to Convert FOL Sentences to Clausal Form.
74. Explore the Use of FOL in Automated Planning Systems.
75. Create a Python Tool for Knowledge Base Consistency Checking using FOL.