|  |  |
| --- | --- |
| **Exp. No** | **List of Practicals**   * In answer sheet write theory in brief, Code and Output. * Properly comment the program. |
| 1 | **Program to Simulate an Agent**  Simulate simple reflex, model based, goal based and utility based agent for any definition you take. Clearly describe the agent example. Student should be able to justify the difference in working of each of the agent through logic. Proper comments are expected. |
| 2 | **Implementation of the Uninformed search techniques (Breadth First Search).**  Graph should have at least 5-6 nodes and 6+ edges.  Path must be the output.  Make the graph input dynamic.  Draw the input graph or tree in the answer sheet with nodes and edges labeled. |
| 3 | **Implementation of Informed search techniques (A\* Search**).  Graph should have at least 5-6 nodes and 6+ edges.  Path must be the output.  Make the graph input dynamic.  Draw the input graph or tree in the answer sheet with nodes and edges labeled.  Student should be clearly able to explain h(n) and g(n) function and how it is used and implemented in the program. |
| 4 | **Implementation of Local search techniques (Hill Climbing).**  Number of iterations (anywhere between 10 to 20 depending on program), logic behind global/ local maximum must be known.  Student should be able to explain if the program is moving from local to global maximum or at least climbing uphill value (if not global)  Disadvantages of hill climbing search must be known. Is any of the disadvantage you are experiencing in your program.  You can take any example. (TSP preferred) |
| 5 | **Implementation of Game Playing algorithm (Minmax Algorithm).**  Create a game tree (e.g., Tic-Tac-Toe)  Avoid games like stone paper scissor where players current move don’t affect future moves.  Take the game where player takes alternate turns.  Tree shouldn’t be very small.  Implement Minmax algorithm to calculate optimal move.  Alpha-beta pruning is not required, but you should understand and explain the logic. Draw the full game tree in the answer sheet and mark evaluated values. |
| 6 | **To create knowledge base for simple Facts and Rules in PROLOG.**  Write program for family tree/medical diagnosis/ animal classification etc.  Write all relations in case of family tree.  Take user input for symptoms in case of medical diagnosis.  How prolog uses resolution to make inferences must be known.  Outputs for different types must be taken. |
| 7 | **Implementation for Bayes' Belief Network.**  Take any example like burglary alarm.  Output should be calculation of probability of any given situation.  Bayesian network with CPTs must be drawn in the answer sheet. |
| 8 | **To implement various NLP preprocessing techniques and functionalities.**  Each command with example. Meaning of each preprocessing method, libraries must be known  1. Tokenization  2. Stopword removal  3. Stemming  4. Lemmatization  5. Part-of-speech (POS) tagging  6. Sentence tokenization  7. Word frequency counting  8. Named entity recognition (NER  9. Bag-of-words (BOW) representation  10. Sentiment analysis:  11. Language detection  12. Translation  13. Spelling correction |
| 9 | **Partial order planning**  Define a planning problem with:   * Initial state * Goal state   Implement or simulate the Partial Order Planning (POP) algorithm.  Show how actions are partially ordered rather than totally ordered.  Final output: Valid partially ordered plan achieving the goal.  Draw a diagram or table to illustrate partial ordering.  Don’t take any example where total ordering is required. |