**PRACTICAL 11**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name:** | Harsh Shah | **Semester:** | VI | **Division:** | 6 |
| **Roll No.:** | 21BCP359 | **Date:** |  | **Batch:** | G11 |
| **Aim:** | **11a -** Understanding the basics and IDE for Prolog Programming  **11b -** Implement any two of the following using Prolog:   * Medical diagnosis of common cold and flu using symptom inputs * Demonstrating list in prolog * Monkey banana problem * Find the factorial of a given number | | | | |

**11a: Understanding the basics and IDE for Prolog Programming**

Prolog is a logic programming language associated with artificial intelligence and computational linguistics. In Prolog, programs are expressed in terms of relations and rules. It's based on a formal system called Horn clauses, which consist of facts and rules. Here's a brief overview:

1. Facts: Facts are statements about relationships between entities. They are represented as predicates.

*Example:*

human(socrates).

1. Rules: Rules define relationships based on conditions. They consist of a head and a body.

*Example:*

mortal(X) :- human(X).

1. Queries: In Prolog, you can ask queries to the knowledge base to retrieve information or verify facts.

*Example:*

?- mortal(socrates).

**IDEs for Prolog Programming:**

There are several IDEs (Integrated Development Environments) available for Prolog programming. Some popular ones include:

* SWI-Prolog: It's a comprehensive Prolog environment with a graphical debugger and IDE-like features. It's available for multiple platforms.
* GNU Prolog: This is a free Prolog compiler with a command-line interface. It provides a basic environment for Prolog development.
* SICStus Prolog: It's a commercial Prolog development system with a comprehensive IDE and advanced features for debugging and optimization.

**11b: Implementing tasks in Prolog**

Medical Diagnosis of Common Cold and Flu Using Symptom Inputs: symptom(fever).

symptom(cough). symptom(sore\_throat). symptom(runny\_nose). symptom(headache). symptom(muscle\_aches). symptom(fatigue).

diagnosis(cold) :symptom(fever), symptom(cough), symptom(runny\_nose), not(symptom(headache)), not(symptom(muscle\_aches)), not(symptom(sore\_throat)), not(symptom(fatigue)).

diagnosis(flu) :symptom(fever), symptom(cough), symptom(runny\_nose), symptom(headache), symptom(muscle\_aches), symptom(fatigue), not(symptom(sore\_throat)).

**Demonstrating Lists in Prolog:**

% Predicate to check if X is a member of the list.

member(X, [X|\_]).

member(X, [\_|T]) :member(X, T).

% Predicate to append two lists.

append([], L, L).

append([H|T], L, [H|R]) :append(T, L, R).