**Batch: A-3 Roll No.: 16010122104**

**Experiment / assignment / tutorial No. 3**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title:** Implementation of Goal based agent architecture using PROLOG. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Objective:** To use the concepts of knowledge engineering to design and solve moderate complex problem.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **Course Outcome** | **After successful completion of the course students should be able to** |
| **CO1** | Design AI solution with appropriate choice of agent architecture |
| **CO3** | Represent and formulate the knowledge to solve the problems using various reasoning techniques |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. **https://www.csupomona.edu/~jrfisher/www/prolog\_tutorial/contents.html**
2. **http://www.csupomona.edu/~jrfisher/www/prolog\_tutorial/pt\_framer.html**
3. **http://www.doc.gold.ac.uk/~mas02gw/prolog\_tutorial/prologpages/**
4. **http://classes.soe.ucsc.edu/cmps112/Spring03/languages/prolog/PrologIntro.pdf**
5. **“Prolog: Programming for Artificial Intelligence” by Ivan Bratko, Pearson education Publications**
6. **“Artificial Intelligence: a Modern Approach” by Russel and Norving, Pearson education Publications**
7. **“Artificial Intelligence” By Rich and knight, Tata Mcgraw Hill Publications**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

Agents, Agent Architecture, Programming with PROLOG

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Historical Profile:**

Knowledge is vast, uncertain and continuously changing. These properties of knowledge make it difficult to arrive at a result. A murder mystery is a kind of situation which depicts the uncertain nature of knowledge and also emphasizes the need of choosing right clauses from entire knowledgebase to make a decision. He goal based agent architecture and some knowledge engineering can help in solutioning of such problems.

The logical agents are complex but they can reason and learn from the actions and new precepts. They are less like acting and think like humans but more like acting and thinking rational agents.

Knowledge and reasoning play a crucial role in dealing with partially observable environments. A knowledge based agent can combine the general knowledge with current percept to infer the hidden aspects of the current state prior to selecting actions.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned:**

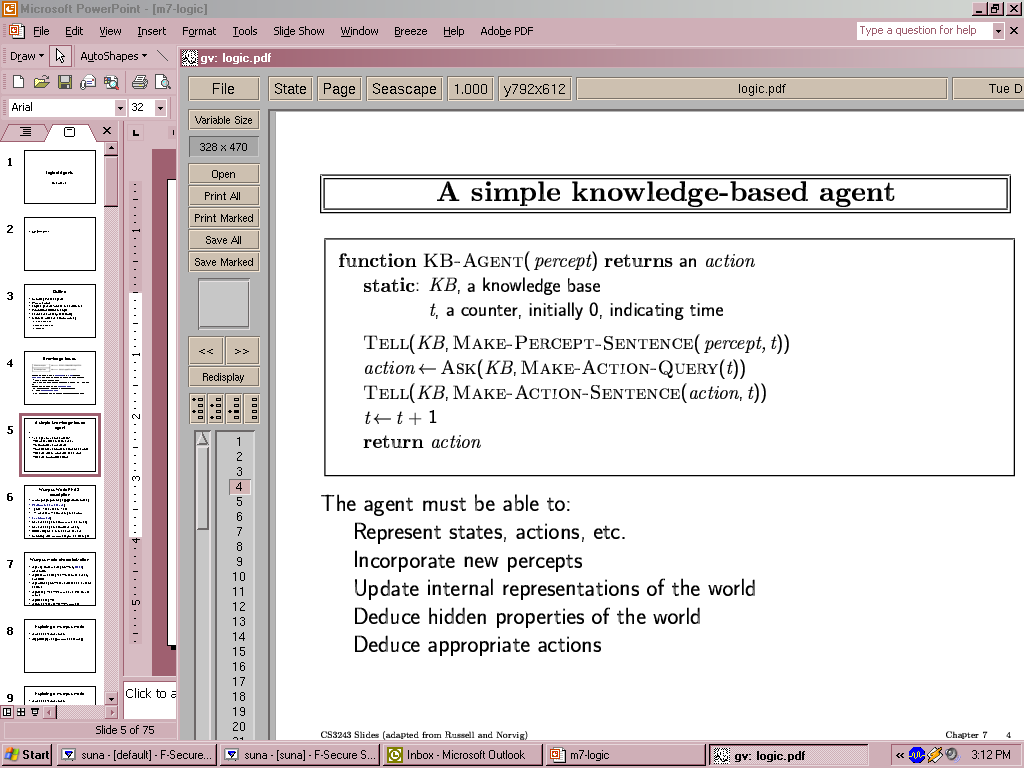
Knowledge engineering, implementing complex agent architecture, uncertainty in knowledge.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**The Knowledge Engineering Process**

1. Identify the task
2. Assemble the relevant knowledge
3. Decide on vocabulary of predicates, functions and constants
4. Encode general knowledge about the domain
5. Encode description of specific problem instance
6. Pose queries to the inference procedure and get answers
7. Debug the knowledge base

**Algorithm for KB-Agent:**



**Problem Statement:**

In the heart of Mumbai’s glitzy South Bombay, the opulent Vikram Mansion hosted a high-profile dinner party to celebrate the Vikram Group's merger with a global conglomerate, bringing together businessmen, celebrities, and socialites for an evening of champagne and grandeur that ended in horror with three prominent figures murdered. Vikram Arora, the ruthless tycoon with a trail of enemies and dark secrets, was found stabbed in his study with a ceremonial dagger, his face frozen in shock; Priya Desai, the scandal-prone Bollywood actress linked to Vikram and burdened by a hidden betrayal, was discovered strangled with a silk scarf in the garden, her phone mysteriously missing; and Rohit Mehra, the ambitious politician marred by whispers of corruption and recently entangled in the merger, was poisoned in the private library, slumped against a bookcase with a glass of wine on the floor and confusion etched on his pale face.

**Knowledge Engineering steps applied to chosen problem:**

% Facts about the victims

victim(vikram\_arora, stabbed, study).

victim(priya\_desai, strangled, garden).

victim(rohit\_mehra, poisoned, library).

% Facts about the suspects

suspect(kiran\_arora, vikram\_wife).

suspect(aishwarya\_mehra, rohit\_wife).

suspect(tariq\_khan, business\_associate).

suspect(simran\_patel, personal\_assistant).

% Motives

motive(kiran\_arora, revenge\_affair).

motive(kiran\_arora, stop\_corruption).

motive(aishwarya\_mehra, political\_gain).

motive(tariq\_khan, jealousy).

motive(tariq\_khan, business\_control).

motive(simran\_patel, blackmail).

motive(simran\_patel, power).

% Access to victims and locations

access(kiran\_arora, vikram\_arora, study).

access(aishwarya\_mehra, rohit\_mehra, library).

access(tariq\_khan, vikram\_arora, study).

access(tariq\_khan, priya\_desai, garden).

access(simran\_patel, vikram\_arora, study).

access(simran\_patel, priya\_desai, garden).

access(simran\_patel, rohit\_mehra, library).

% Knowledge of secrets

knows\_secret(simran\_patel, vikram\_arora, murder\_coverup).

knows\_secret(kiran\_arora, vikram\_arora, affair).

knows\_secret(aishwarya\_mehra, rohit\_mehra, corruption).

% Ability to manipulate others

can\_manipulate(simran\_patel, kiran\_arora).

can\_manipulate(simran\_patel, aishwarya\_mehra).

can\_manipulate(simran\_patel, tariq\_khan).

% Specific skills

has\_skill(simran\_patel, poison\_knowledge).

has\_skill(tariq\_khan, physical\_strength).

has\_skill(aishwarya\_mehra, social\_manipulation).

% Evidence found

evidence(vikram\_arora, dagger, study).

evidence(priya\_desai, scarf, garden).

evidence(rohit\_mehra, wine\_glass, library).

% Rules to determine potential culprits

potential\_culprit(X, Victim) :-

suspect(X, \_),

victim(Victim, Method, Location),

motive(X, \_),

access(X, Victim, Location),

(Method = stabbed -> has\_skill(X, physical\_strength);

Method = strangled -> has\_skill(X, physical\_strength);

Method = poisoned -> has\_skill(X, poison\_knowledge)).

% Rule to determine the mastermind

mastermind(X) :-

suspect(X, \_),

motive(X, power),

access(X, vikram\_arora, \_),

access(X, priya\_desai, \_),

access(X, rohit\_mehra, \_),

knows\_secret(X, vikram\_arora, \_),

can\_manipulate(X, \_),

has\_skill(X, poison\_knowledge).

% Rule to check if a suspect has an alibi

has\_alibi(X) :-

suspect(X, \_),

not(access(X, \_, \_)).

% Query to find all potential culprits for each victim

all\_potential\_culprits :-

victim(V, \_, \_),

findall(X, potential\_culprit(X, V), Culprits),

write('Potential culprits for '), write(V), write(': '), write(Culprits), nl,

fail.

all\_potential\_culprits.

% Query to find the mastermind

:- mastermind(Mastermind),

write('The mastermind behind all murders is: '), write(Mastermind), nl.

% Query to list all suspects with their motives

:- findall(Suspect-Motive, (suspect(Suspect, \_), motive(Suspect, Motive)), SuspectMotives),

write('Suspects and their motives:'), nl,

list\_suspect\_motives(SuspectMotives).

% Helper predicate to print suspect motives

list\_suspect\_motives([]).

list\_suspect\_motives([Suspect-Motive|Rest]) :-

write(Suspect), write(': '), write(Motive), nl,

list\_suspect\_motives(Rest).

% Query to find suspects without alibis

:- findall(X, (suspect(X, \_), not(has\_alibi(X))), SuspectsWithoutAlibi),

write('Suspects without alibis: '), write(SuspectsWithoutAlibi), nl.

% Query to find all evidence

:- findall(Victim-Item-Location, evidence(Victim, Item, Location), AllEvidence),

write('Evidence found:'), nl,

list\_evidence(AllEvidence).

% Helper predicate to print evidence

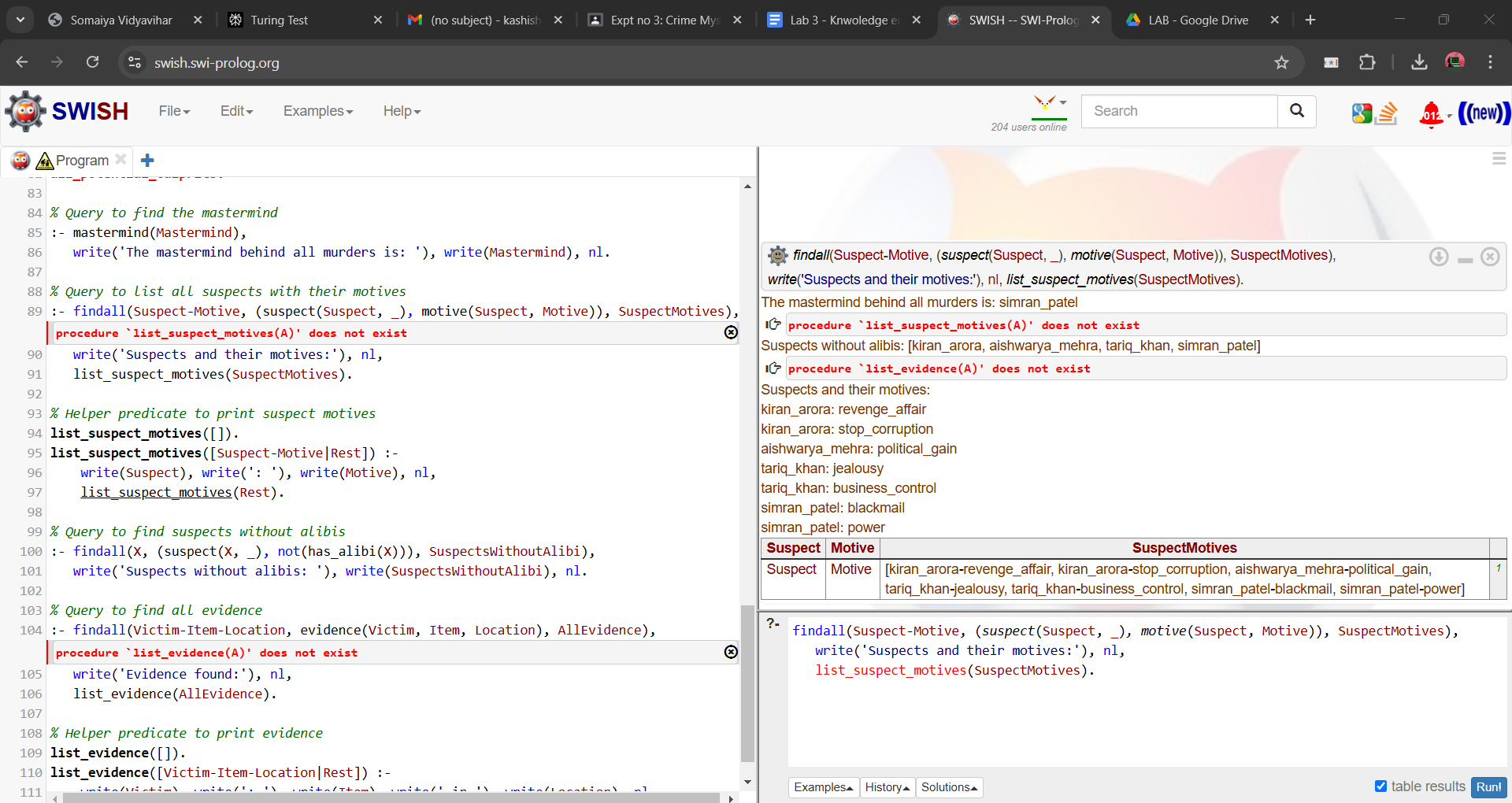
list\_evidence([]).

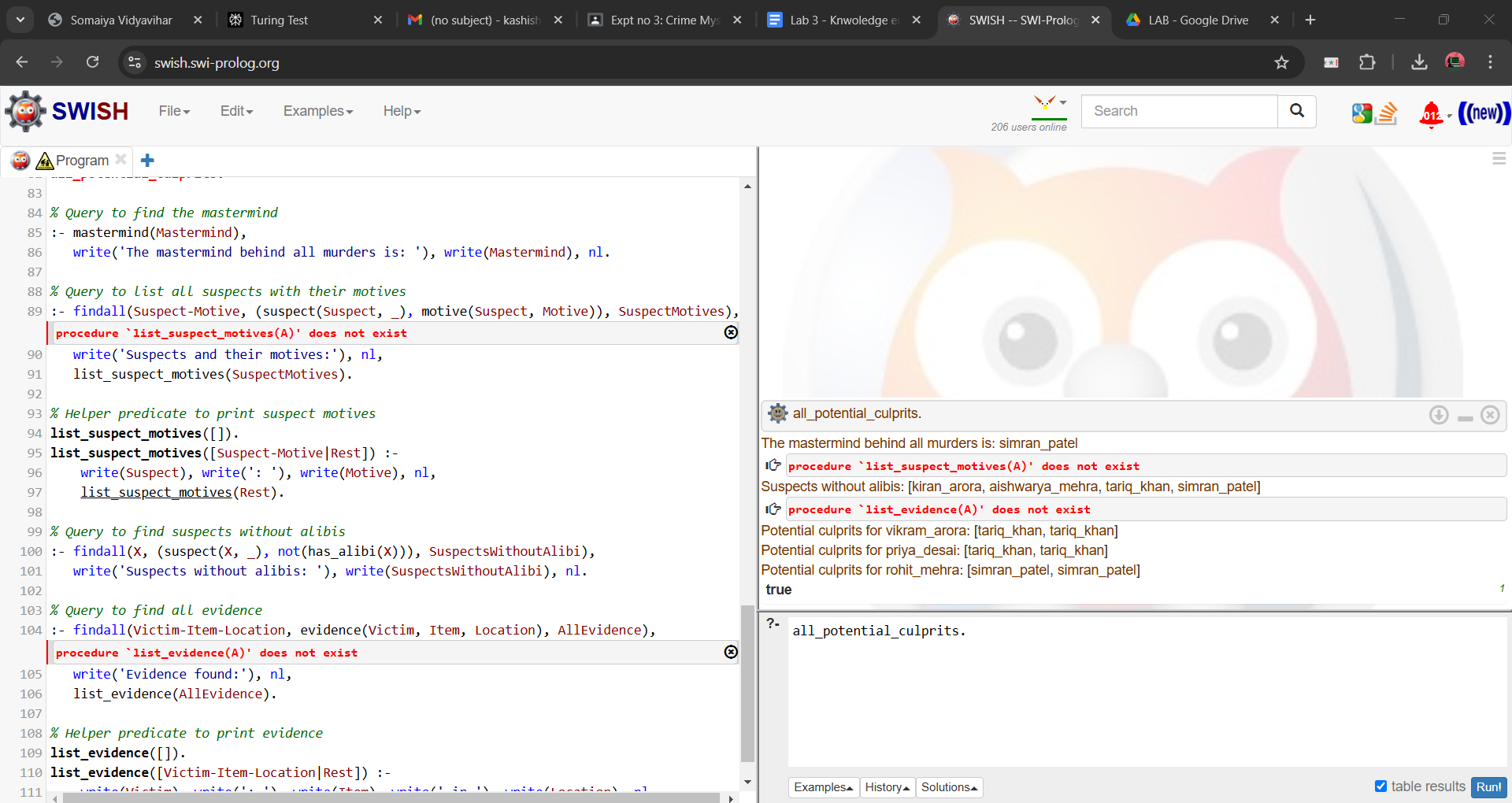
list\_evidence([Victim-Item-Location|Rest]) :-

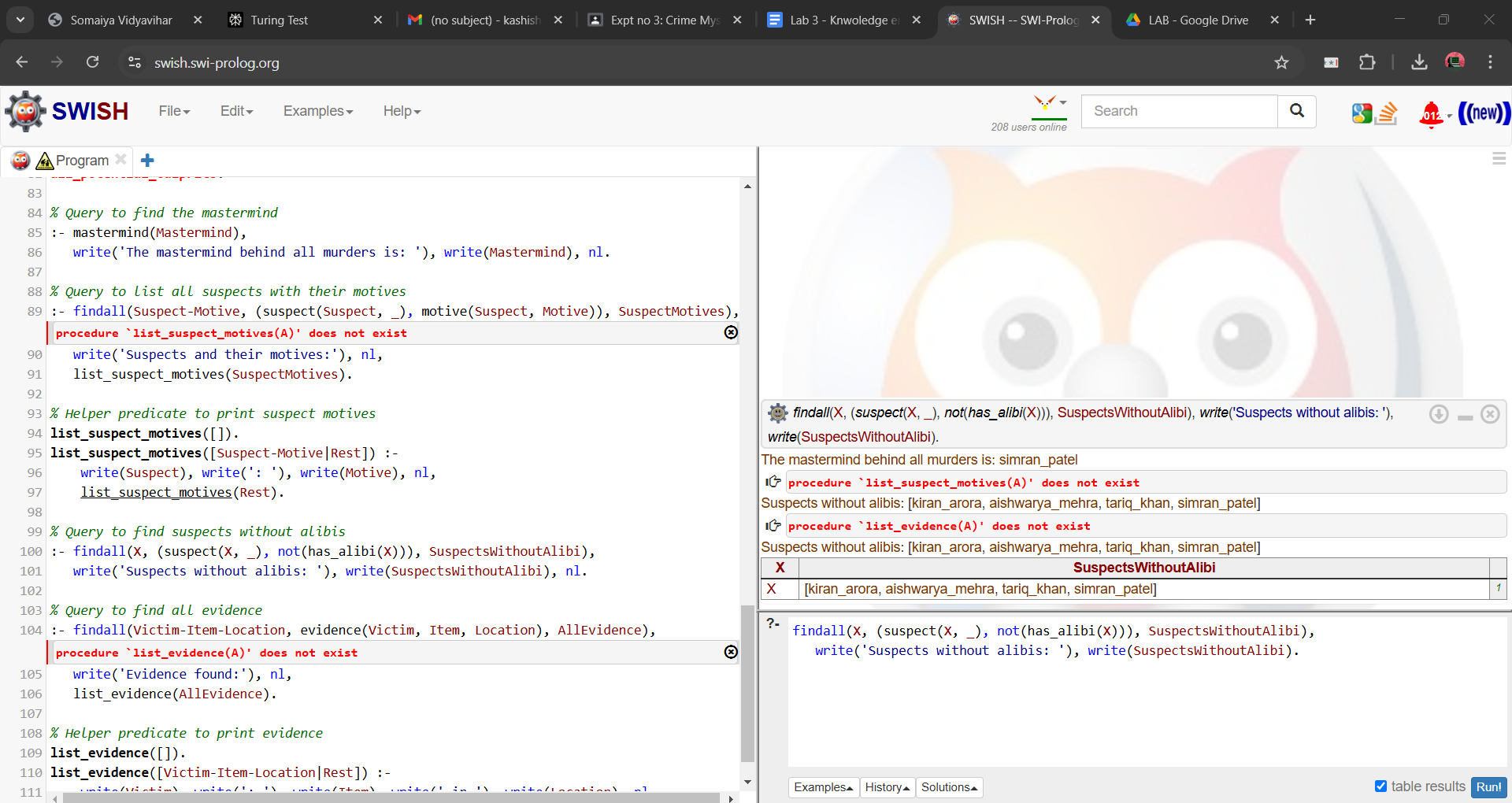
write(Victim), write(': '), write(Item), write(' in '), write(Location), nl,

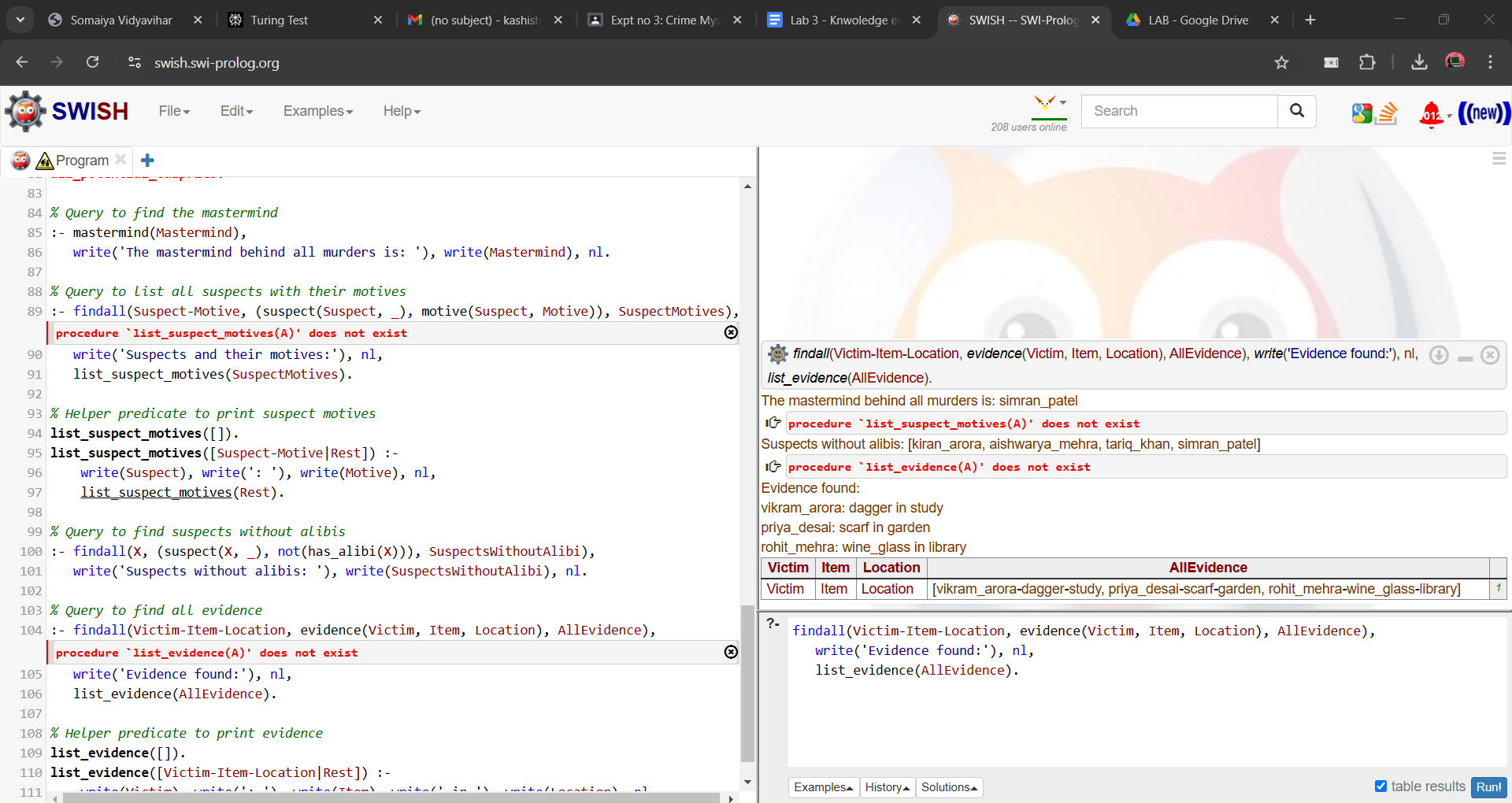
list\_evidence(Rest).

**Agent Architecture** *(Justify the blocks)***:**

****

****

****

****

**Team Members:**

1. **Bhakti Lahane 16010122093**
2. **Kashish Mamania 16010122104**

**Post Lab Objective Questions**

**1. Which is not a Goal-based agent?**

1. Inference
2. Search
3. Planning
4. Conclusion
5. Dynamic search.

**Answer: d. Conclusion**

**2. Which were built in such a way that humans had to supply the inputs and  
interpret the outputs?**

1. Agents
2. Sensor
3. AI System
4. Actuators

**Answer: c. AI System**

**Post Lab Subjective Questions**

**Explain the role of PEAS and task environment in choosing the agent architecture. Justify your answer with an example.**

**Ans:**

PEAS (Performance measure, Environment, Actuators, Sensors) and task environment play crucial roles in choosing the appropriate agent architecture:

Performance measure: Defines the criteria for success, guiding the design of the agent's decision-making process.

Environment: Determines the complexity and nature of the problem space, influencing the required capabilities of the agent.

Actuators: Define the actions an agent can take, affecting the choice of decision-making algorithms.

Sensors: Determine what information the agent can perceive, impacting the knowledge representation and reasoning mechanisms.

Task environment characteristics (e.g., observability, determinism, episodic nature, static/dynamic, discrete/continuous) further refine the choice of architecture.

Example: For a self-driving car:

Performance: Safety, efficiency, comfort

Environment: Roads, traffic, weather (partially observable, non-deterministic, dynamic, continuous)

Actuators: Steering, acceleration, braking

Sensors: Cameras, LIDAR, GPS

Given these factors, a hybrid architecture combining reactive components for immediate safety responses and deliberative planning for navigation would be appropriate. The complex, dynamic environment necessitates real-time processing and adaptive decision-making, while the critical nature of the task requires robust planning capabilities.