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

**Experiment / assignment / tutorial No. 2**

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

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

|  |
| --- |
| **Title: Implementation of condition-action rules based agent using PROLOG** |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Objective:** Developing a basic level agent program that runs on condition-action rules

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

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **Course Outcome** | **After successful completion of the course students should be able to** |
| **CO1** | Understand the history & various application of AI and choose appropriate agent architecture to solve the given problem. |
| **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. **“Artificial Intelligence: a Modern Approach” by Russell and Nerving, Pearson education Publications**
5. **“Artificial Intelligence” By Rich and knight, Tata McGraw Hill Publications**
6. **“Prolog: Programming for Artificial Intelligence” by Ivan Bratko, Pearson education Publications**

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

**Pre Lab/ Prior Concepts:** Intelligent Agent, Agent Architectures, Rule base Vs Knowledgebase approach

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

**Historical Profile:** Agent programs for simple applications need not be very complicated. They can be based on condition-action rules and still they give better results, though not always rational. The family tree program makes use of similar concept.

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

**New Concepts to be learned:**

Defining rules, using and programming with PROLOG

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

A simple agent program can be defined mathematically as an agent function which maps every possible percepts sequence to a possible action the agent can perform or to a coefficient, feedback element, function or constant that affects eventual actions:

*F*: *P* \* − >*A*

**Algorithm for ‘Condition-Action Rule Table’ Agent function:**

**function**SIMPLE-REFLEX-AGENT (percept) **returns** an action

**Static:** *rules,* a set of condition-action rules

*State*:- nINTERPRET-INPUT (percept)

*Rule*:- *RULE-MATCH (state, rules)*

*Action*:- *RULE-ACTION [rule]*

**Returnaction**

This approach follows a table for lookup of condition-action pairs defining all possible condition-action rules necessary to interact in an environment.

**Example Family Tree/disease-symptom mapping/ City map with their distances between them:**

Here's a family tree for the Ambani family:

Dhirubhai Ambani and Kokilaben Ambani

* Married in 1955

Children of Dhirubhai and Kokilaben:

1. **Mukesh Ambani** (m. Nita Ambani, 1985)
   * Children:
     + Akash Ambani (m. Shloka Mehta, 2019)
       - Children: Prithvi Akash Ambani, Veda
     + Isha Ambani (m. Anand Piramal, 2018)
       - Children: Twins Krishna and Aadiya
     + Anant Ambani (engaged to Radhika Merchant)
2. **Anil Ambani** (m. Tina Munim, 1991)
   * Children:
     + Jai Anmol Ambani (m. Krishna Shah)
     + Jai Anshul Ambani
3. **Nina Ambani** (m. Bhadrashyam Kothari, deceased 2015)
   * Children:
     + Nayantara Kothari
     + Arjun Kothari
4. **Dipti Ambani** (m. Dattaraj Salgaocar)
   * Children:
     + Ishika Salgaocar
     + Vikram Salgaocar

**Base Knowledgebase:**

% Facts: Define individuals and their genders

male(dhirubhai\_ambani).

male(mukesh\_ambani).

male(anil\_ambani).

male(akash\_ambani).

male(anant\_ambani).

male(jai\_anmol\_ambani).

male(jai\_anshul\_ambani).

male(arjun\_kothari).

male(vikram\_salgaocar).

male(prithvi\_akash\_ambani).

male(bhadrashyam\_kothari).

male(dattaraj\_salgaocar).

female(kokilaben\_ambani).

female(nita\_ambani).

female(tina\_munim).

female(nina\_ambani).

female(dipti\_ambani).

female(isha\_ambani).

female(shloka\_mehta).

female(radhika\_merchant).

female(krishna\_shah).

female(nayantara\_kothari).

female(ishika\_salgaocar).

female(veda\_ambani).

female(krishna\_ambani).

female(aadiya\_ambani).

% Marriage relationships

married(dhirubhai\_ambani, kokilaben\_ambani).

married(mukesh\_ambani, nita\_ambani).

married(anil\_ambani, tina\_munim).

married(nina\_ambani, bhadrashyam\_kothari).

married(dipti\_ambani, dattaraj\_salgaocar).

married(akash\_ambani, shloka\_mehta).

married(isha\_ambani, anand\_piramal).

married(jai\_anmol\_ambani, krishna\_shah).

% Parent relationships

parent(dhirubhai\_ambani, mukesh\_ambani).

parent(dhirubhai\_ambani, anil\_ambani).

parent(dhirubhai\_ambani, nina\_ambani).

parent(dhirubhai\_ambani, dipti\_ambani).

parent(kokilaben\_ambani, mukesh\_ambani).

parent(kokilaben\_ambani, anil\_ambani).

parent(kokilaben\_ambani, nina\_ambani).

parent(kokilaben\_ambani, dipti\_ambani).

parent(mukesh\_ambani, akash\_ambani).

parent(mukesh\_ambani, isha\_ambani).

parent(mukesh\_ambani, anant\_ambani).

parent(nita\_ambani, akash\_ambani).

parent(nita\_ambani, isha\_ambani).

parent(nita\_ambani, anant\_ambani).

parent(anil\_ambani, jai\_anmol\_ambani).

parent(anil\_ambani, jai\_anshul\_ambani).

parent(tina\_munim, jai\_anmol\_ambani).

parent(tina\_munim, jai\_anshul\_ambani).

parent(nina\_ambani, nayantara\_kothari).

parent(nina\_ambani, arjun\_kothari).

parent(bhadrashyam\_kothari, nayantara\_kothari).

parent(bhadrashyam\_kothari, arjun\_kothari).

parent(dipti\_ambani, ishika\_salgaocar).

parent(dipti\_ambani, vikram\_salgaocar).

parent(dattaraj\_salgaocar, ishika\_salgaocar).

parent(dattaraj\_salgaocar, vikram\_salgaocar).

parent(akash\_ambani, prithvi\_akash\_ambani).

parent(akash\_ambani, veda\_ambani).

parent(shloka\_mehta, prithvi\_akash\_ambani).

parent(shloka\_mehta, veda\_ambani).

parent(isha\_ambani, krishna\_ambani).

parent(isha\_ambani, aadiya\_ambani).

**Rules:**

% Rules for relationships

father(X, Y) :- male(X), parent(X, Y).

mother(X, Y) :- female(X), parent(X, Y).

sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.

brother(X, Y) :- male(X), sibling(X, Y).

sister(X, Y) :- female(X), sibling(X, Y).

grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

grandfather(X, Y) :- male(X), grandparent(X, Y).

grandmother(X, Y) :- female(X), grandparent(X, Y).

aunt(X, Y) :- female(X), sibling(X, Z), parent(Z, Y).

uncle(X, Y) :- male(X), sibling(X, Z), parent(Z, Y).

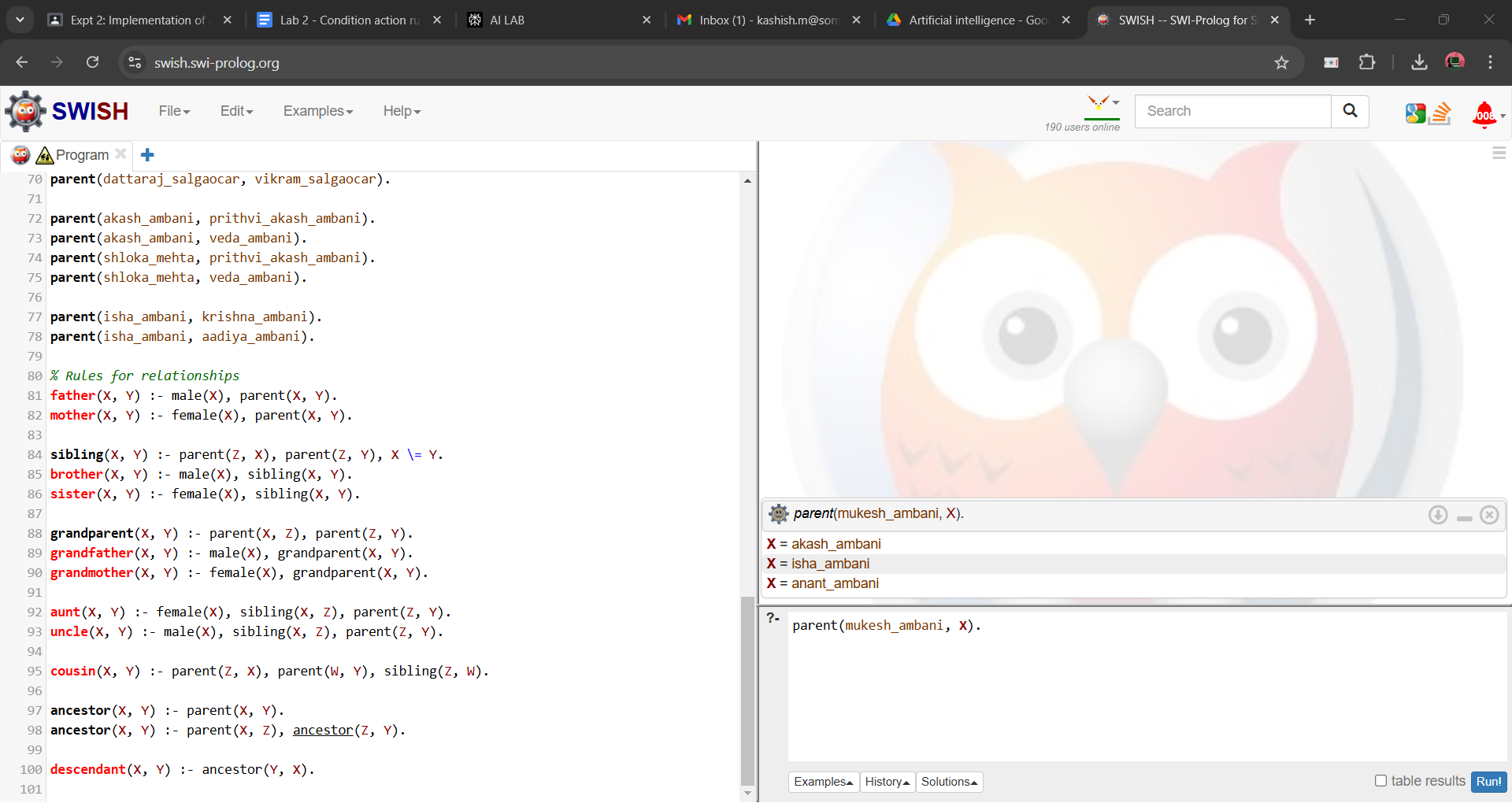
cousin(X, Y) :- parent(Z, X), parent(W, Y), sibling(Z, W).

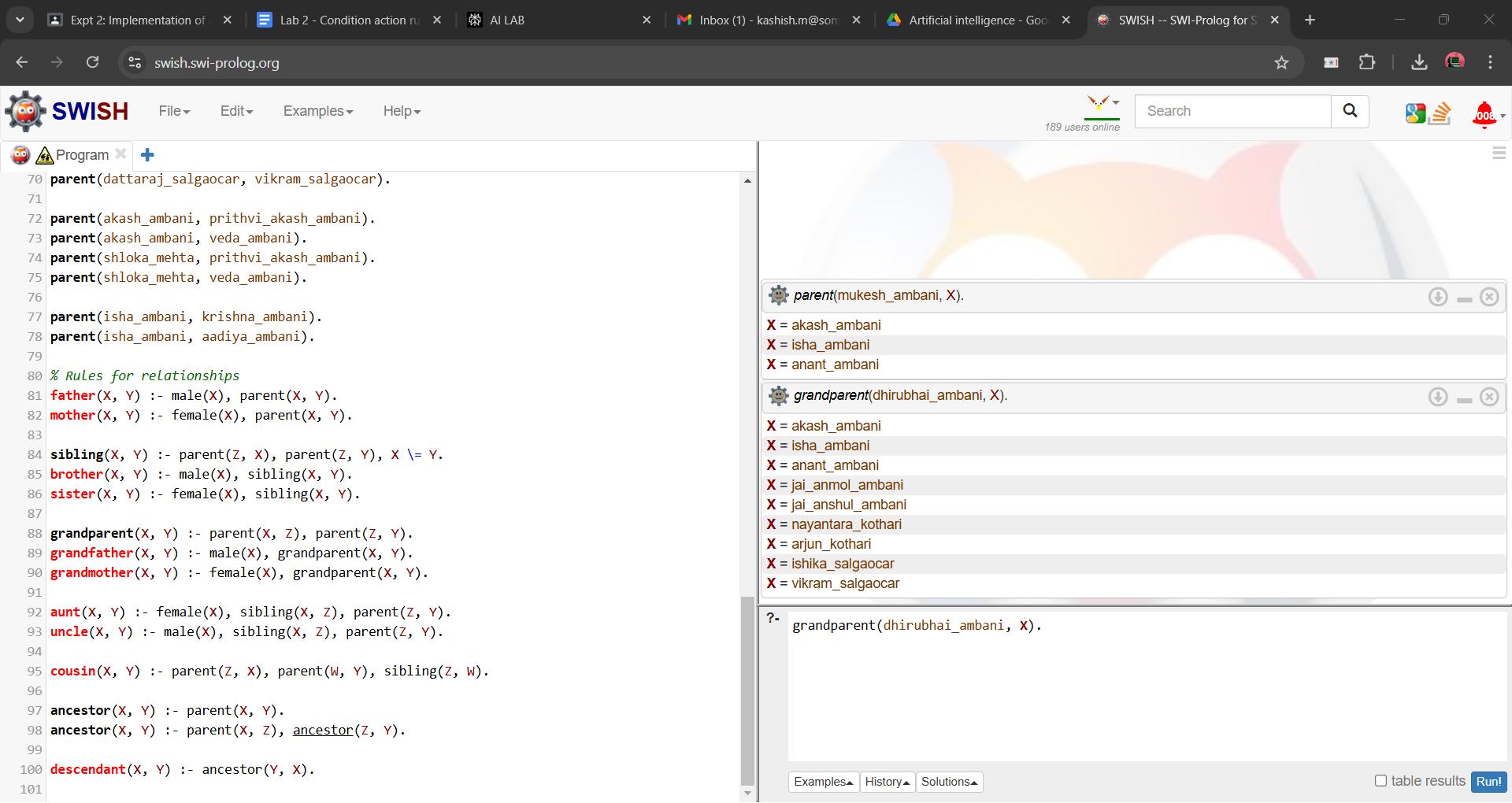
ancestor(X, Y) :- parent(X, Y).

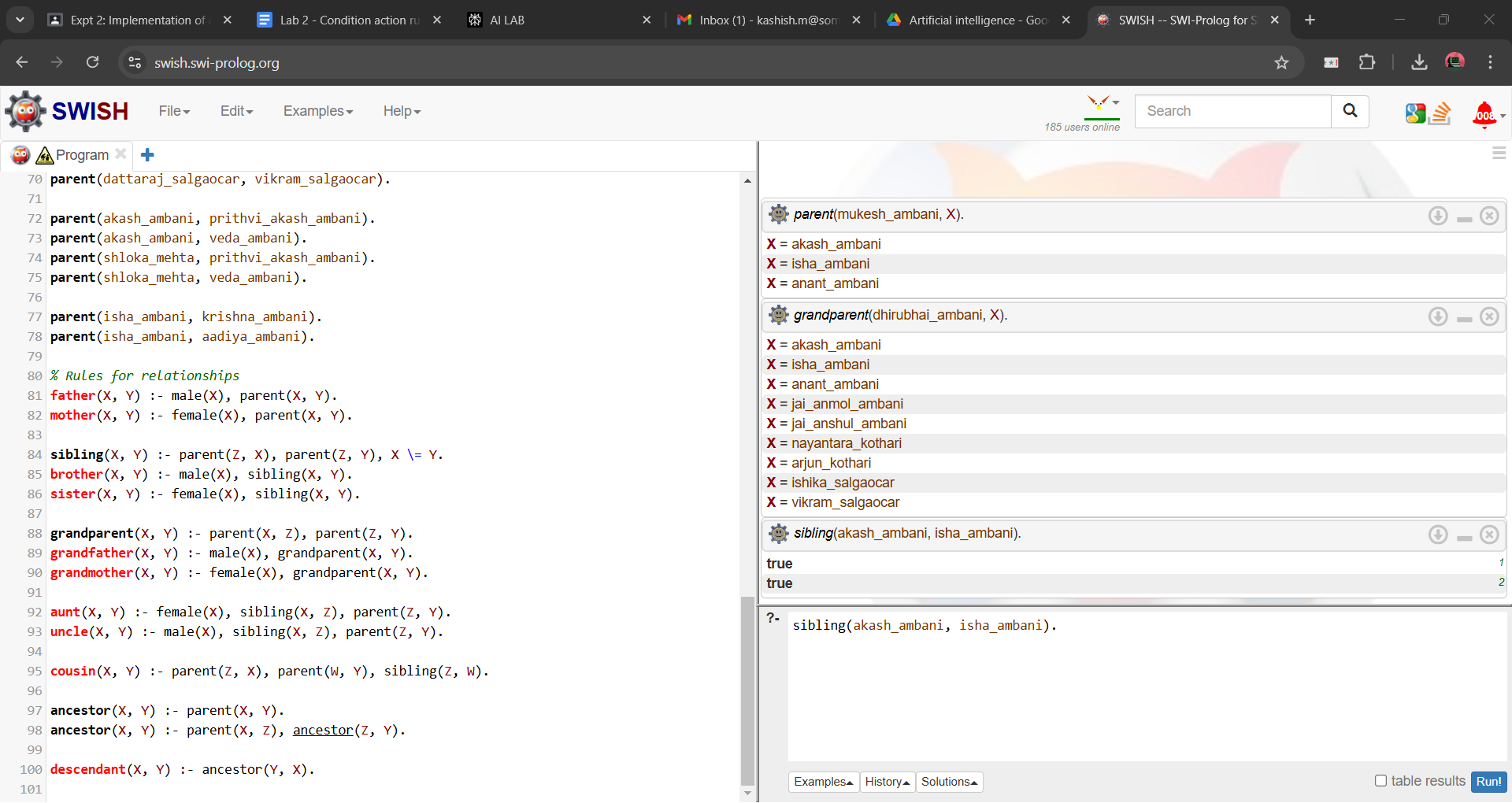
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).

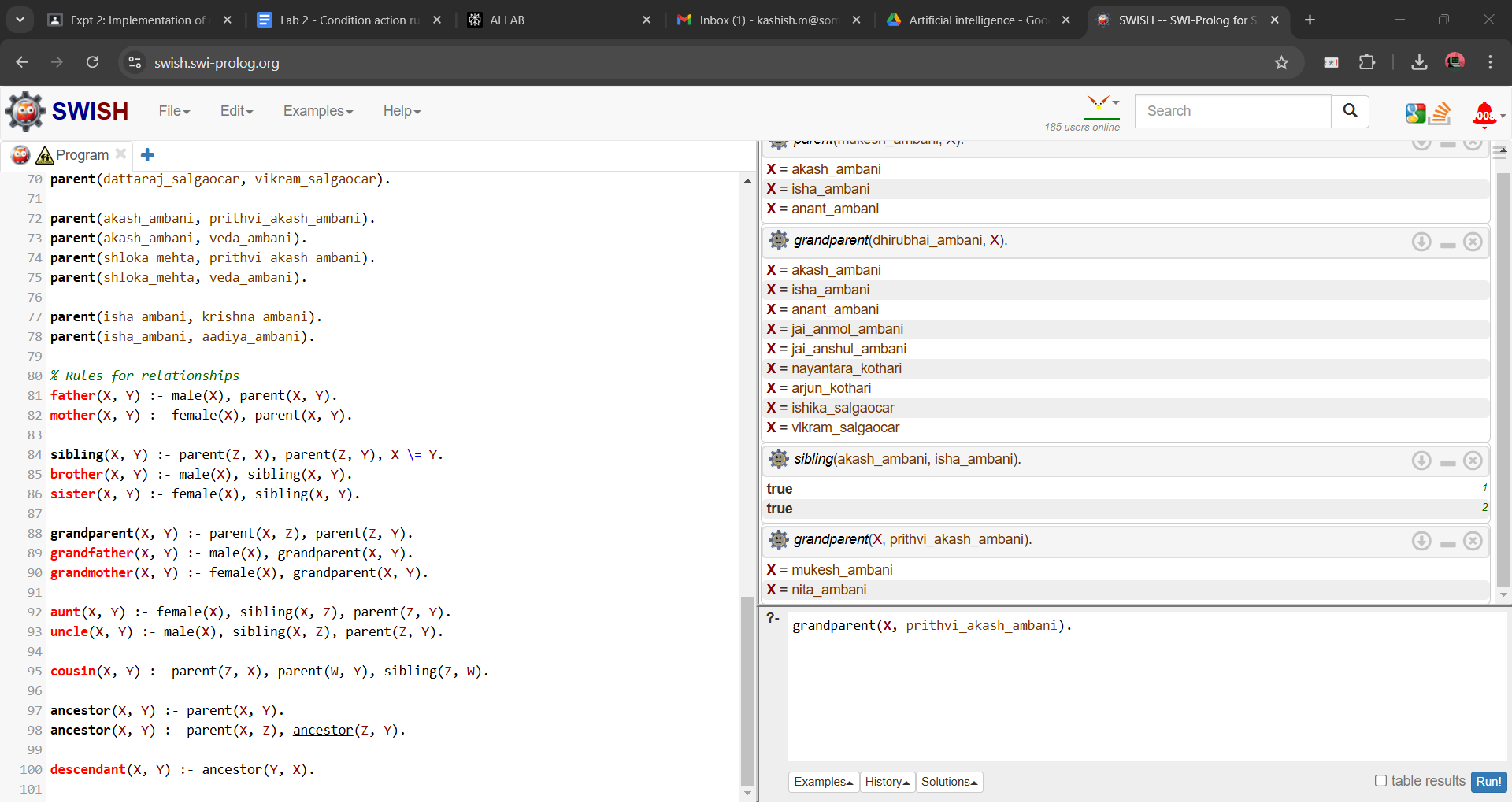
descendant(X, Y) :- ancestor(Y, X).

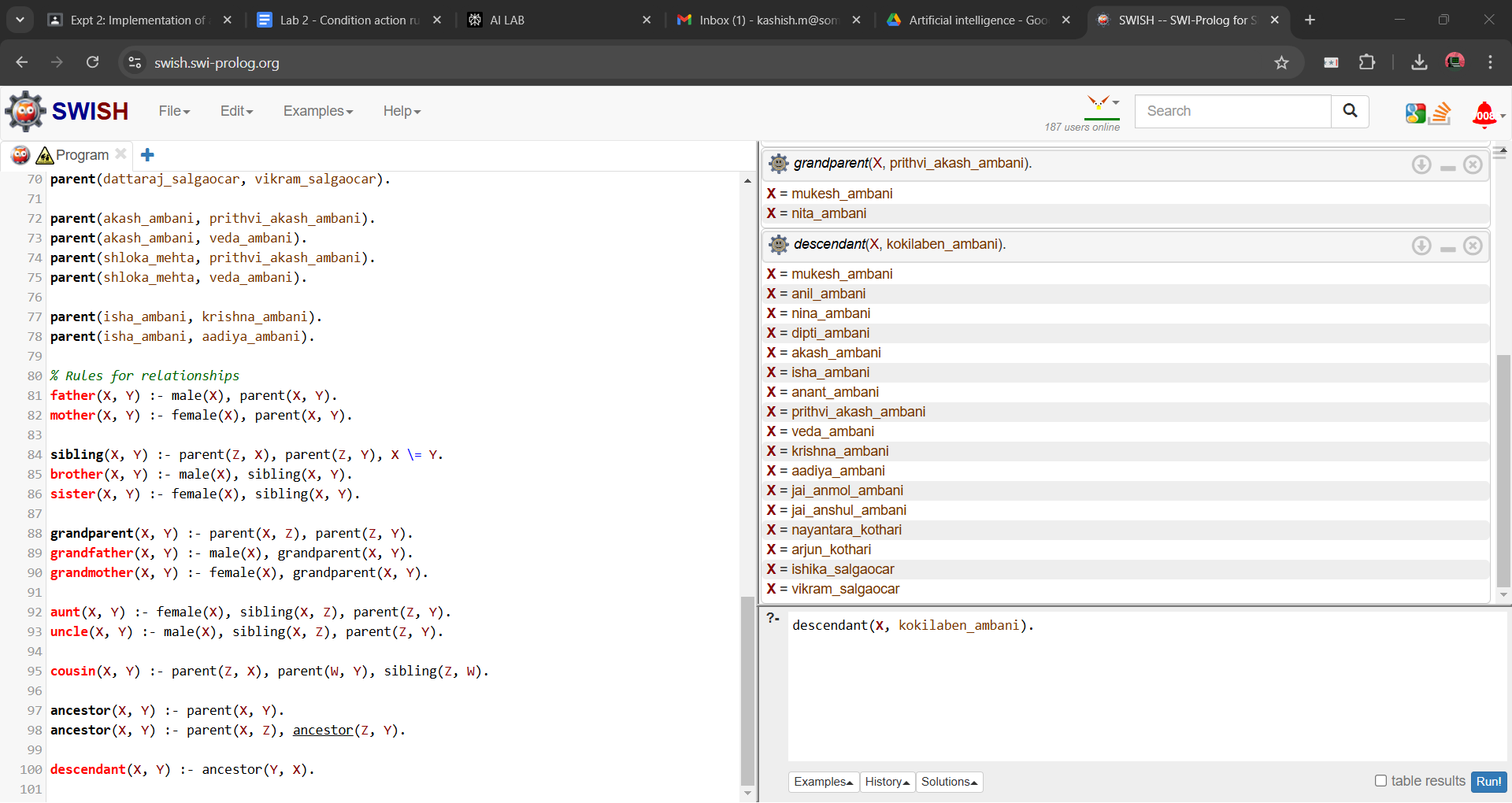
**Some Sample queries and Outputs:**

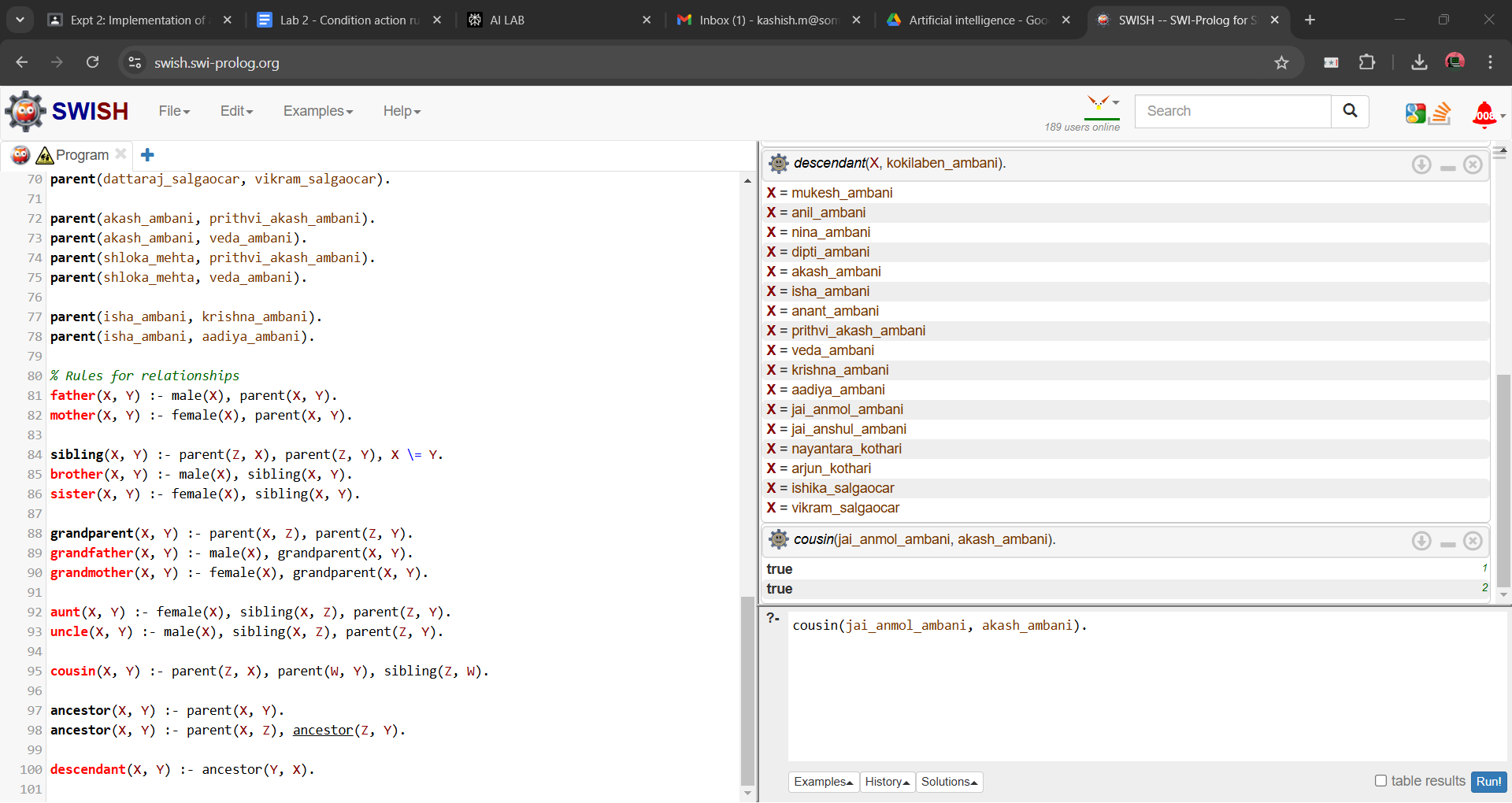
****

****

****

****

****

****

.

**Post Lab Objective Questions**

1. **The PROLOG suit is based on**
   1. Interpreter
   2. Compiler
   3. None of the above

**Answer: a. Interpreter**

1. **State true of false**

There must be at least one fact pertaining to each predicate written in the PROLOG program.

**Answer: false**

1. **State true of false**

In PROLOG program the variable declaration is a compulsory part.

**Answer: false**

**Post Lab Subjective Questions**

1. **Differentiate between a fact and a predicate with syntax.**

**Ans:**

A fact is a statement that is unconditionally true in Prolog. It represents a basic piece of information in the knowledge base. The syntax for a fact is:predicate\_name(argument1, argument2, ...).For example: likes(john, pizza).A predicate is a relation or property that can be true or false. It can be used in both facts and rules. The syntax for a predicate is the same as a fact, but it can also appear in the head of a rule:predicate\_name(Arg1, Arg2, ...) :- condition1, condition2, ...For example: parent(X, Y) :- father(X, Y).The key difference is that facts are always true, while predicates can be defined by rules and may be true or false depending on the arguments and conditions.

1. **Differentiate between knowledgebase and Rule base approach.**

**Ans:**

Knowledge base approach:

* Stores a collection of facts and rules representing domain knowledge
* Focuses on representing information in a declarative manner
* Can include various types of knowledge (facts, rules, relationships)
* Used for general reasoning and inference

Rule base approach:

* Specifically focuses on storing and using rules
* Rules are typically in the form of "if-then" statements
* Emphasizes procedural knowledge and decision-making
* Often used in expert systems for specific problem-solving tasks

While a knowledge base can include rules, a rule base is more specialized and focused on using rules for decision-making and problem-solving.

1. **Differentiate between database and knowledgebase.**

**Ans:**

Database:

* Stores structured data in tables or relations
* Focuses on efficient storage and retrieval of data
* Uses query languages like SQL for data manipulation
* Typically does not include reasoning capabilities
* Designed for handling large volumes of data

Knowledge base:

* Stores facts, rules, and relationships representing domain knowledge
* Includes reasoning and inference capabilities
* Can handle unstructured or semi-structured information
* Often uses logic programming or other AI techniques
* Designed for representing and using complex knowledge

While databases are optimized for data storage and retrieval, knowledge bases are designed to represent and reason about complex information and relationships.

1. **What is a ‘free variable’? Explain with an example.**

**Ans:**

A free variable in Prolog is a variable that is not bound to any specific value within a given context. It can be unified with different values during the execution of a query or rule.Example:  
Consider the following Prolog rule:likes(X, Y) :- enjoys(X, Z), has(Y, Z).In this rule, X, Y, and Z are all free variables. They are not bound to any specific values within the rule itself. When this rule is used in a query, these variables can be unified with different values:?- likes(john, What).Here, 'john' is bound to X, but What (corresponding to Y in the rule) and Z remain free variables. Prolog will try to find values for these free variables that satisfy the rule and any facts in the knowledge base.Free variables allow for flexible querying and reasoning in Prolog, as they can represent unknown or variable information that the system will attempt to determine during execution.