**Lab:1**

* **Write a hello world program in Prolog.**

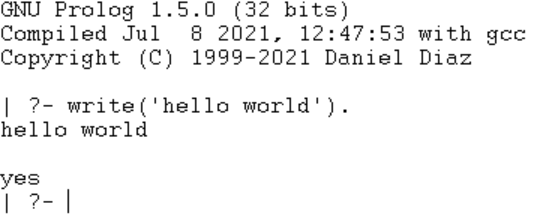
**Objective:** The objective of this task is to write a basic Prolog program

that prints "Hello, World!" to the console.

**Code**:

write (‘Hello World’).

**Output:**



* **Represent the following facts in Prolog.**

a.Tom is a cat

b. Kunal loves to eat Pasta

c. Hair is black

d. Nawaz loves to play games

e. Pratyusha is lazy.

f. Lili dances.

g. Tom is searching for food.

h. Jack loves to play cricket.

i. Bili loves to play cricket.

j. Ryan is free.

**Objective**: the objective of this task is to be familiar with basic facts.

**Code**:

cat(tom).

loves(kunal,pasta).

hair(black).

loves(nawaz,games).

dances(lili).

lazy(pratyusha).

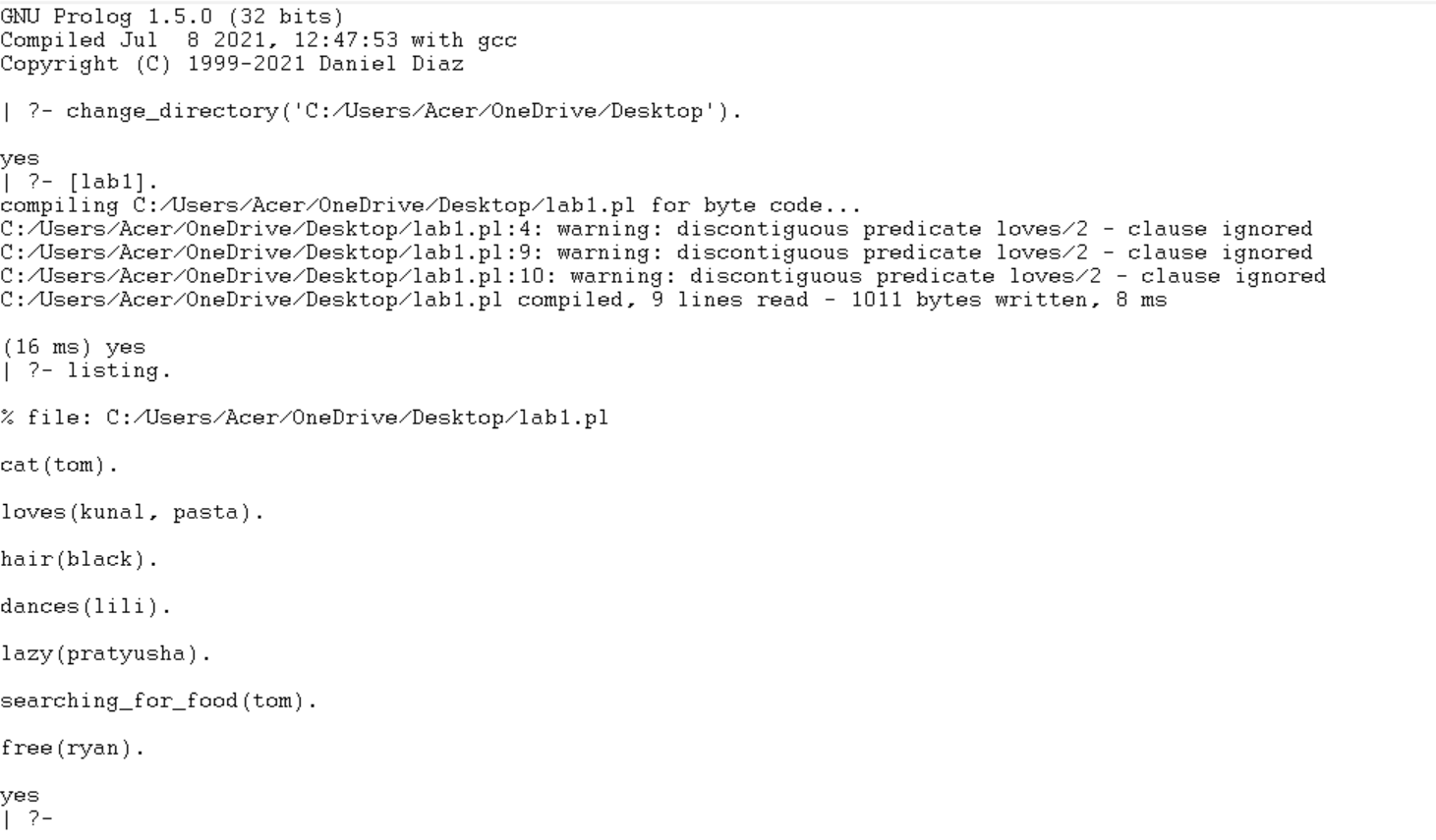
searching\_for\_food(tom).

free(ryan).

loves(jack,cricket).

loves(bill,cricket).

**Output:**

****

* Represent the following relations in Prolog along with the facts from

**Objective**: The objective of this task is to use the previously defined facts to

establish relationships between different entities and write queries to retrieve

information based on these relationships.

**Code**:

cat(tom).

hair(black).

lazy(pratyusha).

dances(lili).

searching\_for\_food(tom).

free(ryan).

loves(kunal, pasta).

loves(nawaz, games).

loves(jack, cricket).

loves(bill, cricket).

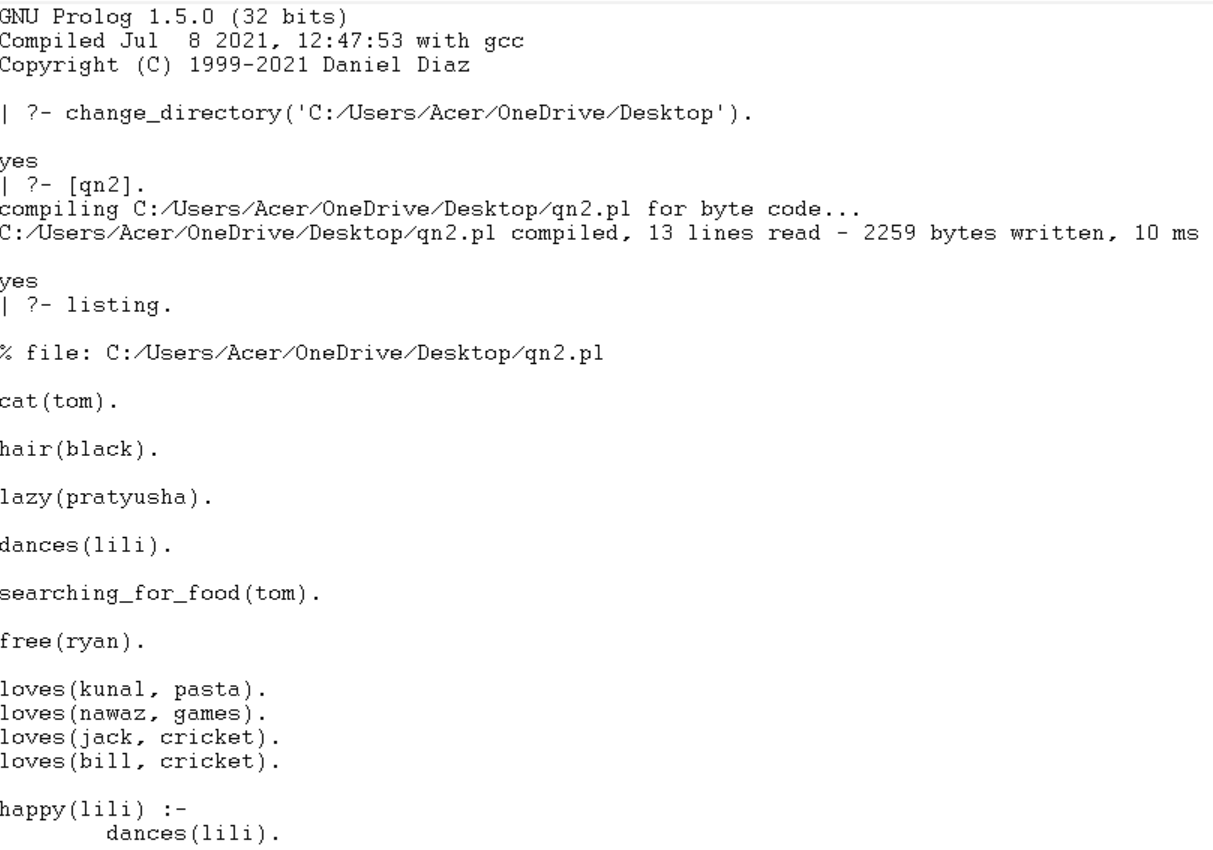
happy(lili) :- dances(lili).

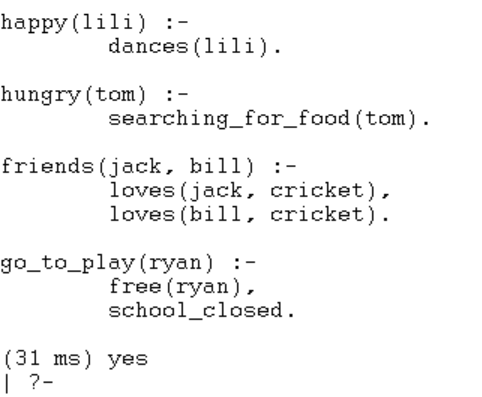
hungry(tom) :- searching\_for\_food(tom).

friends(jack, bill) :- loves(jack, cricket),loves(bill, cricket).

go\_to\_play(ryan) :- free(ryan), school\_closed.

**Output:**

****



1. **Represent the following relations in prolog along with the facts from Q.no 2:**

a. Lili is happy if she dances.

b. Tom is hungry if he is searching for food.

c. Jack and Bili are friends if both of them love to play cricket.

d. Ryan will go to play if school is closed, and he is free.

And make queries to see

● Is tom a cat?

● Does Kunal love to eat pasta?

● Is Lili happy?

● Will Ryan go to play?

**Code:**

% Facts using specified predicates

cat(tom).

loves\_to\_eat(kunal, pasta).

black(hair).

loves\_to\_play(nawaz, games).

lazy(pratyusha).

dances(lili).

searching\_for(tom, food).

loves\_to\_play(jack, cricket).

loves\_to\_play(bili, cricket).

free(ryan).

% Relations

is\_happy(lili) :- dances(lili).

is\_hungry(tom) :- searching\_for(tom, food).

are\_friends(jack, bili) :-

loves\_to\_play(jack, cricket),

loves\_to\_play(bili, cricket).

will\_go\_to\_play(ryan) :-

free(ryan),

school\_is\_closed.

% Additional facts

school\_is\_closed. % Assuming school is closed in this scenario

% Queries

% Query 1: Is Tom a cat?

% Query 2: Does Kunal love to eat pasta?

% Query 3: Is Lili happy?

% Query 4: Will Ryan go to play?

% Query 1: Is Tom a cat?

?- cat(tom).

% Query 2: Does Kunal love to eat pasta?

?- loves\_to\_eat(kunal, pasta).

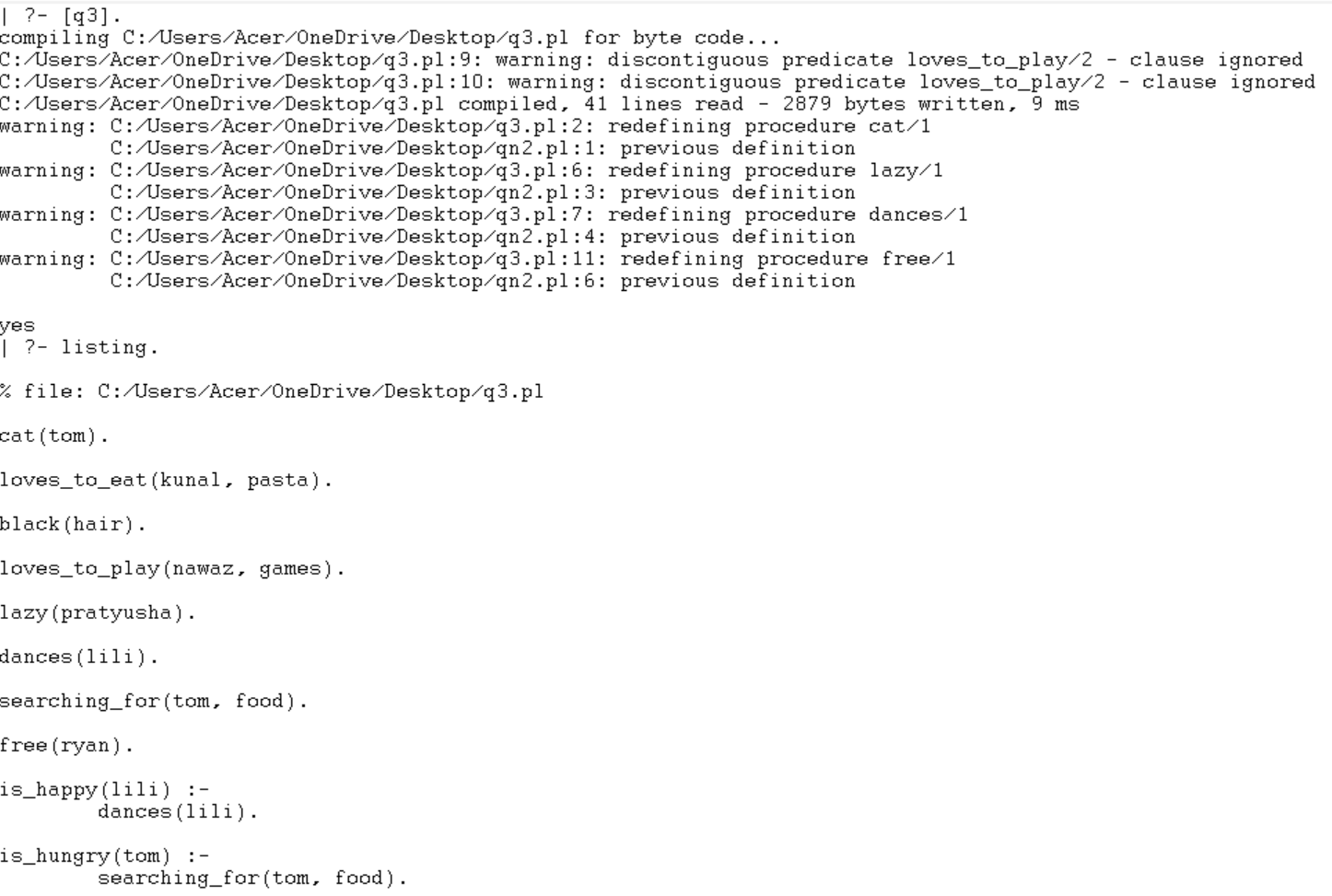
% Query 3: Is Lili happy?

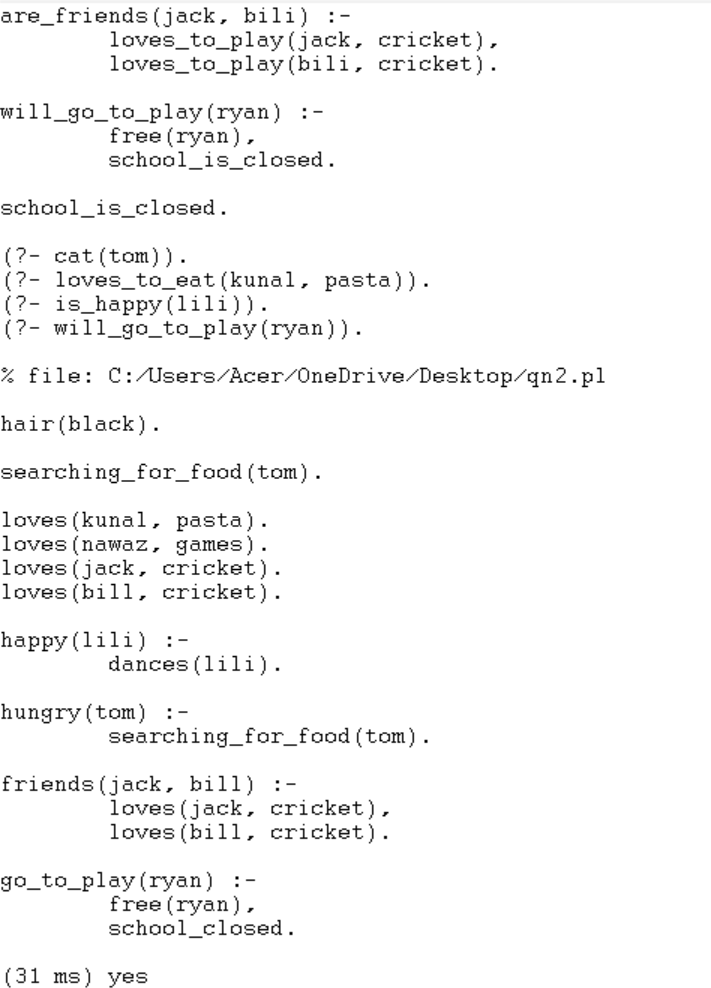
?- is\_happy(lili).

% Query 4: Will Ryan go to play?

?- will\_go\_to\_play(ryan).

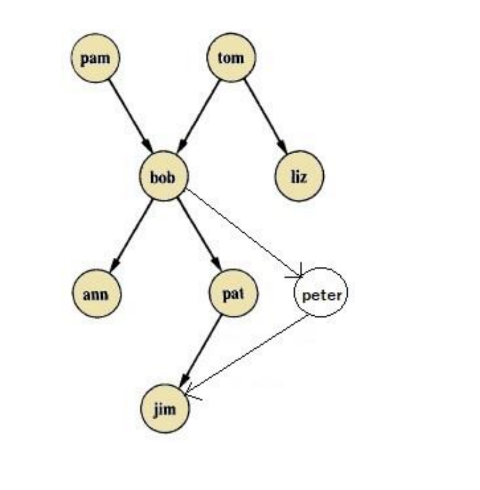
**Output:**





**Q4. Implement the knowledge base of the following family tree and write rules for determining mother, sister, father, grand father, grandmother, grand parent,**

**uncle, wife and husband.**

****

**Objective:** The goal is to develop a Prolog program that displays a family tree and uses facts and rules to establish various relationships such as mother, father, sister, grandfather, grandmother, grandparent, uncle, wife, and husband. This entails creating predicates for parent-child relationships, marital status, and genders, as well as writing rules that use these predicates to infer familial ties. The program should support querying to determine specific relationships inside the family tree based on predefined facts and rules.

**Code**:

mother(pam,bob).

mother(pat,jim).

wife(pam,tom).

wife(pat,peter).

sister(ann,pat).

grandfather(tom,ann).

grandfather(tom,pat).

grandfather(tom,peter).

grandfather(bob,jim).

grandmother(pam,ann).

grandmother(pam,pat).

grandmother(pam,peter).

father(tom,bob).

father(tom,liz).

father(bob,pat).

father(bob,peter).

father(peter,jim).

husband(peter,pat).

uncle(liz,ann).

uncle(liz,pat).

uncle(liz,peter).

female(pam).

female(ann).

female(pat).

male(tom).

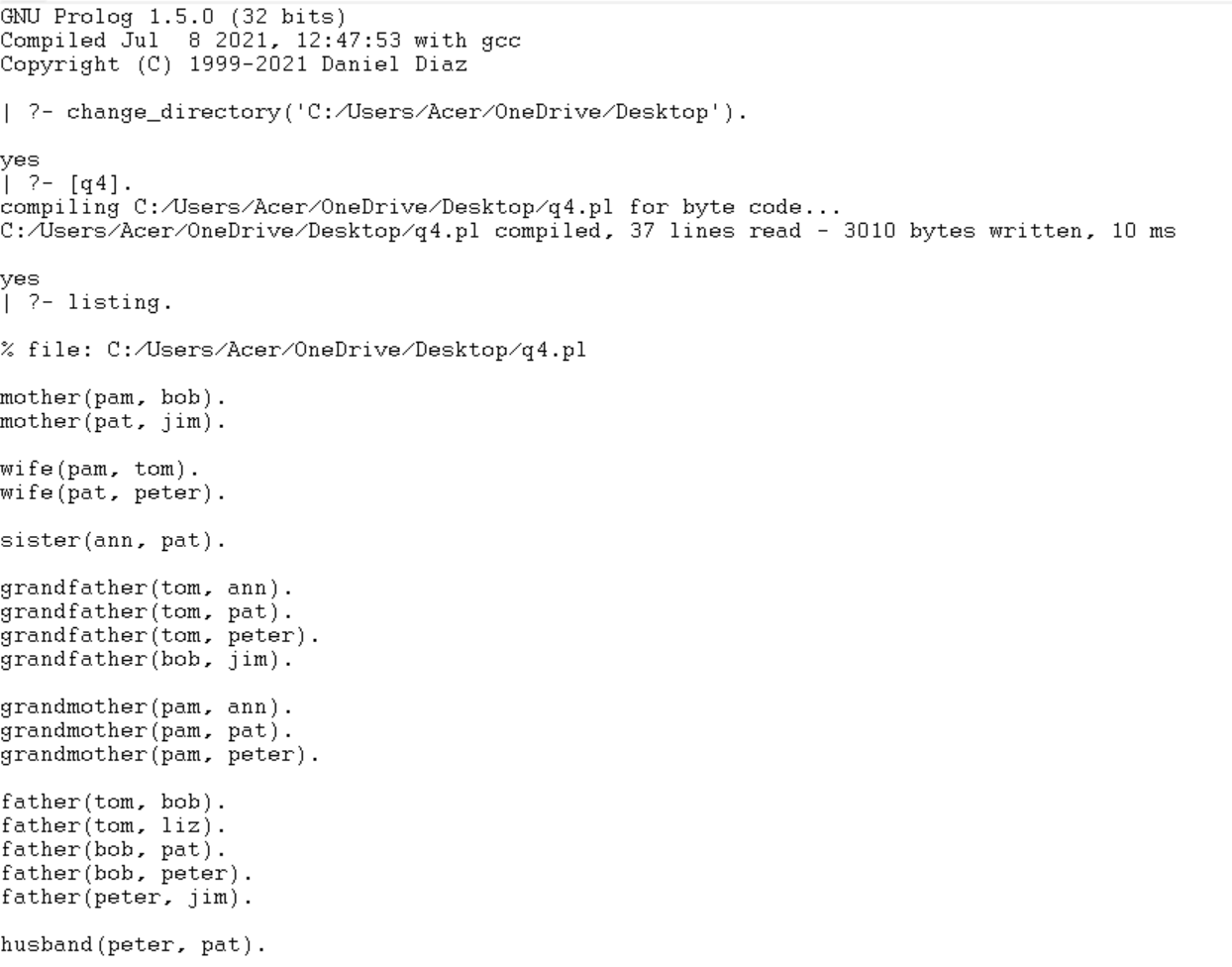
male(bob).

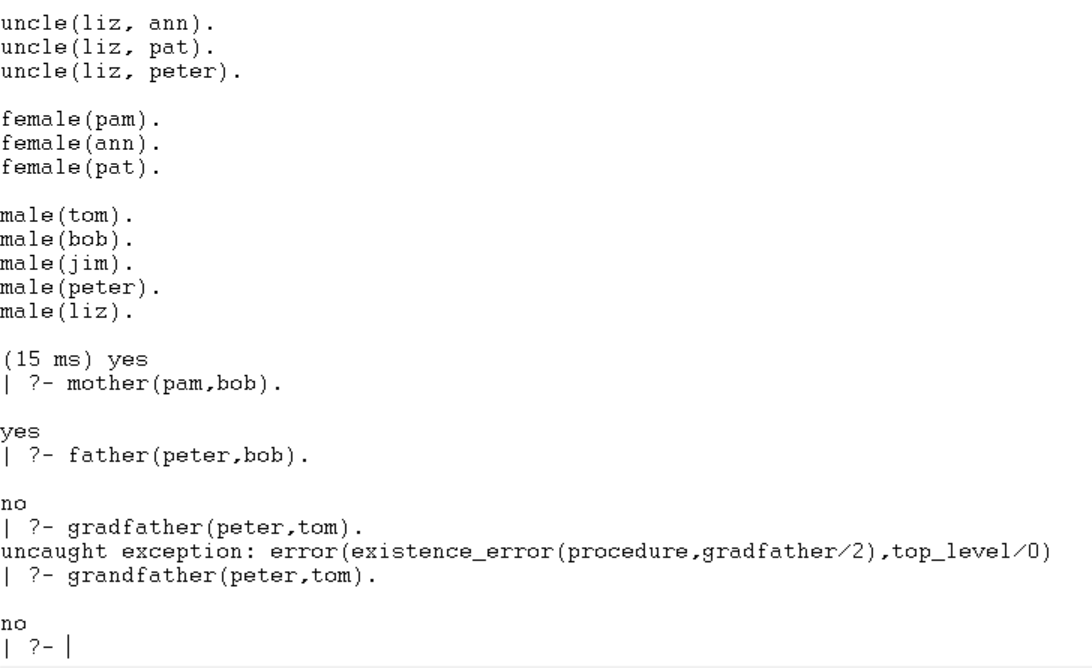
male(jim).

male(peter).

male(liz).

**Output:**





**Conclusion:**

In this Prolog implementation, we’ve successfully represented a variety of facts and relationships using predicates to define individuals attributes like behaviors,and familial connections. From simple assertions like “tom is cat” to more complex relationships such as determining who is a mother, sister, grandfather, or friend based on defined rules. Prologs logical programming paradigm proves effective for expressing and querying structured knowledge. Each predicate and rule serves to build a coherent knowledge base that can be queried to answer specific questions about the relationships and attributes defined within the system, showcasing the versatility and power of Prolog in knowledge representation and reasoning tasks.