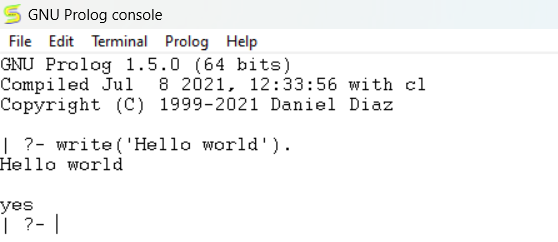
**Qno.1)**

→ **Objective :** To display content on the screen

→ **Task name :** Write a hello world program in prolog.

→ **Code :** write(‘content’).

→ **Output :**



**Qno.2)**

→ **Objective :** To represent facts in GNU prolog

→ **Task name :** Represent given statements as facts in prolog

→ **Code : (to declare facts)**

cat(tom).

loves(kunal, pasta).

color(hair, black).

loves(nawaz, play\_games).

lazy(pratyusha).

dances(lili).

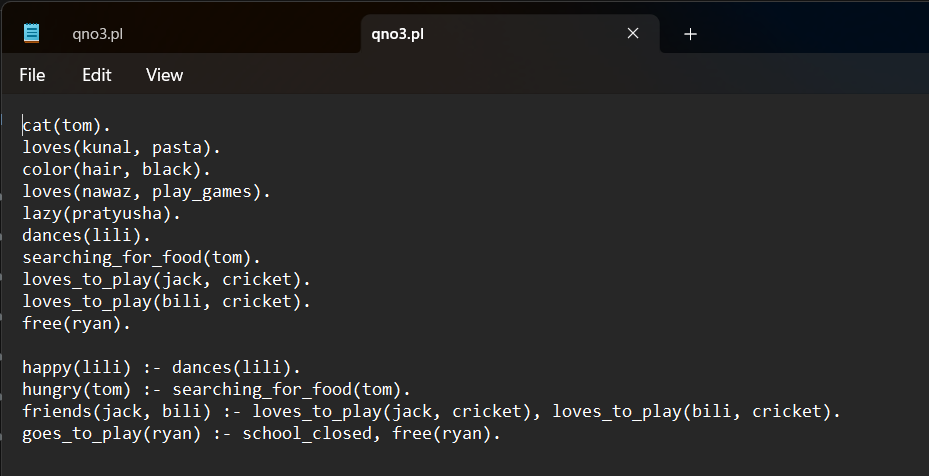
searching\_for\_food(tom).

loves\_to\_play(jack, cricket).

loves\_to\_play(bili, cricket).

free(ryan).

→ **Output :**

****

**Qno.3)**

→ **Objective :** To represent relations along with facts

→ **Task name :** Represent given relations in prolog along with facts from qno2

→ **Code :**

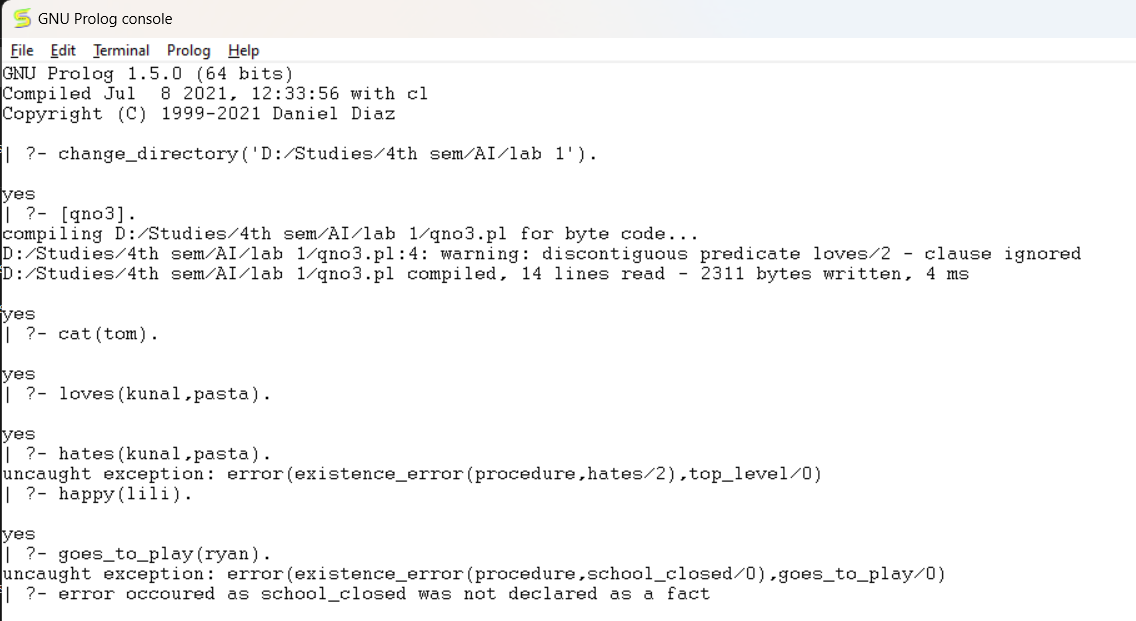
happy(lili) :- dances(lili).

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

friends(jack, bili) :- loves\_to\_play(jack, cricket), loves\_to\_play(bili, cricket).

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

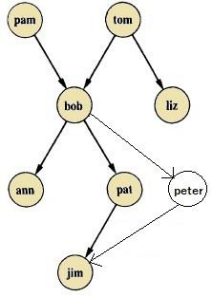
→ **Output :**



**Qno.4)**

→ **Objective :** To implement rules and relations of a family tree

→ **Task name :** 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.



→ **Code :**

mother(pam, bob).

mother(pal, jim).

wife(pam, tom).

wife(pal, peter).

sister(ann, pal).

grandfather(tom, ann).

grandfather(tom, pal).

grandfather(tom, peter).

grandfather(bob, jim).

grandmother(pam, ann).

grandmother(pam, pal).

grandmother(pam, peter).

father(tom, bob).

father(tom, liz).

father(bob, ann).

father(bob, pal).

father(bob, peter).

father(peter, jim).

husband(tom, pam).

husband(peter, pal).

uncle(liz, ann).

uncle(liz, pal).

uncle(liz, peter).

female(pam).

female(ann).

female(pat).

male(tom).

male(bob).

male(jim).

male(peter).

male(liz).

→ **Output :**

