

## Exercise -1

- Write a program to put facts indicating that a lion, a tiger, and a cow are animals into the database and record that two (lion and tiger) are carnivores. Save your program to a disk file and load it. Check that the database is correct using the **listing**.
- Enter goals to test whether:
  - (a) there is such an animal as a tiger in the database
  - (b) a cow and a tiger are both in the database (a conjunction of two goals)
  - (c) a lion is an animal and also a carnivore
  - (d) a cow is an animal and also a carnivore.
  - (e) list all animals which are also carnivores.

**Code:**

```
animal(lion).  
animal(tiger).  
animal(cow).
```

```
carnivore(lion).  
carnivore(tiger).
```

**Output:**

```
GNU Prolog console
File Edit Terminal Prolog Help
| ?- listing(animal).

% file: D:/Sem 07/AI Lab/Assignment/Lab08/Ex1.pl

animal(lion).
animal(tiger).|
animal(cow).

(15 ms) yes
| ?- listing(carnivore).

% file: D:/Sem 07/AI Lab/Assignment/Lab08/Ex1.pl

carnivore(lion).
carnivore(tiger).

(31 ms) yes
| ?- animal(tiger).

yes
| ?- animal(cow),animal(tiger).

yes
| ?- animal(lion),carnivore(lion).

yes
| ?- animal(cow),carnivore(cow).

no
| ?- animal(X), carnivore(X).

X = lion ?

yes
| ?- animal(X), carnivore(X).

X = lion ?

yes
| ?- animal(X), carnivore(X).

X = lion ? ;

X = tiger ? ;

no
| ?-
```

## Exercise -2

- Type the following program into a file and load it into Prolog.

```
animal(mammal,tiger,carnivore,stripes).
animal(mammal,hyena,carnivore,ugly).
animal(mammal,lion,carnivore,mane).
animal(mammal,zebra,herbivore,stripes).
animal(bird,eagle,carnivore,large).
animal(bird,sparrow,scavenger,small).
animal(reptile,snake,carnivore,long).
animal(reptile,lizard,scavenger,small).
```

- Devise and test goals to find:
  - (a) all the mammals.
  - (b) all the carnivores that are mammals.
  - (c) all the mammals with stripes.
  - (d) whether there is a reptile that has a mane?

### Code:

```
animal(mammal,tiger,carnivore,stripes).
animal(mammal,hyena,carnivore,ugly).
animal(mammal,lion,carnivore,mane).
animal(mammal,zebra,herbivore,stripes).
animal(bird,eagle,carnivore,large).
animal(bird,sparrow,scavenger,small).
animal(reptile,snake,carnivore,long).
animal(reptile,lizard,scavenger,small).
```

```
% (a) all the mammals.
mammals(X) :- animal(mammal, X, _, _).
% (b) all the carnivores that are mammals.
carnivore_mammals(X) :- animal(mammal, X, carnivore, _).
% (c) all the mammals with stripes.
mammals_with_stripes(X) :- animal(mammal, X, _, stripes).
% (d) whether there is a reptile that has a mane?
reptile_with_mane(X) :- animal(reptile, X, _, mane).
```

**Output:**

```
GNU Prolog console
File Edit Terminal Prolog Help
| ?- consult('D:/Sem 07/AI Lab/Assignment/Lab08/Ex2.pl').
compiling D:/Sem 07/AI Lab/Assignment/Lab08/Ex2.pl for byte code...
D:/Sem 07/AI Lab/Assignment/Lab08/Ex2.pl compiled, 7 lines read - 1479 bytes written, 4 ms

yes
| ?- consult('D:/Sem 07/AI Lab/Assignment/Lab08/Ex2.pl').
compiling D:/Sem 07/AI Lab/Assignment/Lab08/Ex2.pl for byte code...
D:/Sem 07/AI Lab/Assignment/Lab08/Ex2.pl compiled, 18 lines read - 2447 bytes written, 23 m

(15 ms) yes
| ?- mammals(X).

X = tiger ? .
Action (; for next solution, a for all solutions, RET to stop) ? a

X = hyena

X = lion

X = zebra

yes
| ?- listing(carnivores_mammals).

yes
| ?- carnivores_mammals(X).
uncaught exception: error(existence_error(procedure,carnivores_mammals/1),top_level/0)
| ?- carnivore_mammals(X).

X = tiger ? ;

X = hyena ? ;

X = lion ? ;

(31 ms) no
| ?- mammals_with_stripes(X).

X = tiger ? ;

X = zebra

yes
| ?- reptile_with_mane(X).

no
| ?-
```

## Exercise -3

- Write a prolog program to infer the following first-order logic statements :
  - The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American. Is West a criminal?

### Code:

```
%Facts
american(colonel_west).
enemy(nono, america).
has_missiles(nono).
sells_to(colonel_west, nono).
% Rules
criminal(X) :- american(X), sells_weapons(X).
sells_weapons(X) :- american(X), sells_to(X, Y),
enemy(Y, america).
% Query
is_criminal(colonel_west) :- criminal(colonel_west).
```

### Output:

```
~~~
| ?- consult('D:/Sem 07/AI Lab/Assignment/Lab08/Ex3.pl').
compiling D:/Sem 07/AI Lab/Assignment/Lab08/Ex3.pl for byte code...
D:/Sem 07/AI Lab/Assignment/Lab08/Ex3.pl compiled, 10 lines read - 1636 bytes written, 4 ms

yes
| ?- is_criminal(colonel_west).

yes
| ?-
```