

---

# [DOCUMENT TITLE]

---

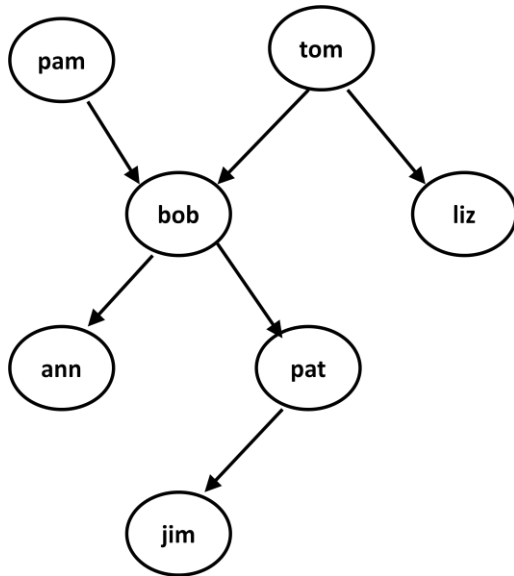
[Document subtitle]



[DATE]  
[COMPANY NAME]  
[Company address]

**Problem no: 01**

**Problem Name:** Implement the given family tree in the form of Prolog Clauses. Every arrow in the diagram represents a Parent relation. For example, pam is parent of bob, tom is parent of bob and so on.



**Source code:**

```
%CLAUSES
parent(pam,bob).
parent(tom,bob).
parent(tom,liz).
parent(bob,ann).
parent(bob,pat).
parent(pat,jim).
```

**Output:**

No output

**Problem no: 02**

**Problem Name:** On the basis of family tree implemented in last Lab, write Prolog statement to query Prolog engine for following questions;

- Is Bob a parent of Pat?
- Who is Liz's parent?
- Who are Bob's children?
- Who is a parent of whom?
- Is bob grandparent of Jim?
- Who is a grandparent of ann?
- Who are Tom's grandchildren?
- Do Ann and Pat have a common parent?

**Source code and output:**

- Is Bob a parent of Pat?

?- parent(bob,pat).

true.

- Who is Liz's parent?

?- parent(X,liz).

X = tom.

- Who are Bob's children?

?- parent(bob,X).

X = ann ;

X = pat.

- Who is a parent of whom?

?- parent(X,Y).

X = pam,

Y = bob ;

X = tom,

Y = bob ;

X = tom,

Y = liz ;

X = bob,

Y = ann ;

X = bob,

Y = pat ;

X = pat,

Y = jim.

– Is bob grandparent of Jim?

?- parent(pat,jim),parent(bob,pat).

true.

– Who is a grandparent of ann?

?- parent(Y,ann),parent(X,Y).

Y = bob,

X = pam ;

Y = bob,

X = tom.

– Who are Tom"s grandchildren?

?- parent(tom,X),parent(X,Y).

X = bob,

Y = ann ;

X = bob,

Y = pat ;

– Do Ann and Pat have a common parent?

?- parent(X,ann),parent(X,pat).

X = bob.

**Problem no: 03**

**Problem Name:** Extend the Family tree program implemented in previous labs by adding the gender information given as follow; Pam, Liz, Ann and Pat are female, while Tom, Bob and Jim are male persons. Using this information define the following relations;

- Define the “mother” relation:
- Define the “grandparent” relation:
- Define the “sister” relation

Also depict the given relations with the help of diagrams.

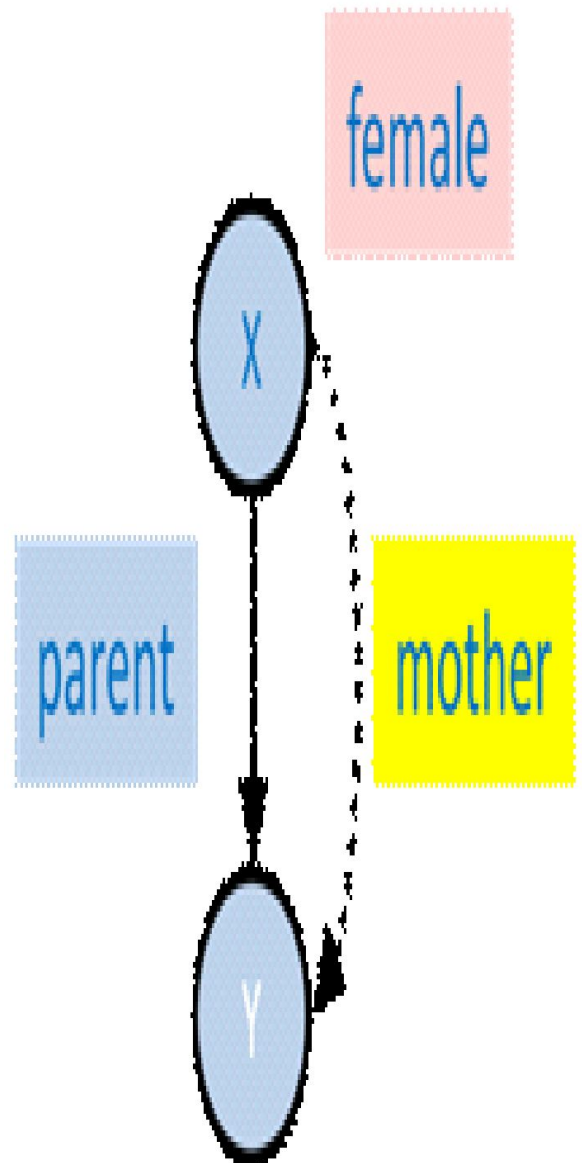
**Source code and output:**

```
%Facts
female(pam).
female(liz).
female(ann).
female(pat).
male(tom).
male(bob).
male(jim).

%Rules
mother(X,Y):- parent(X,Y),female(X).
grandparent(X,Z):- parent(Y,Z),parent(X,Y).
sister(X,Y):- parent(Z,X),parent(Z,Y),female(X),different(X,Y).
```

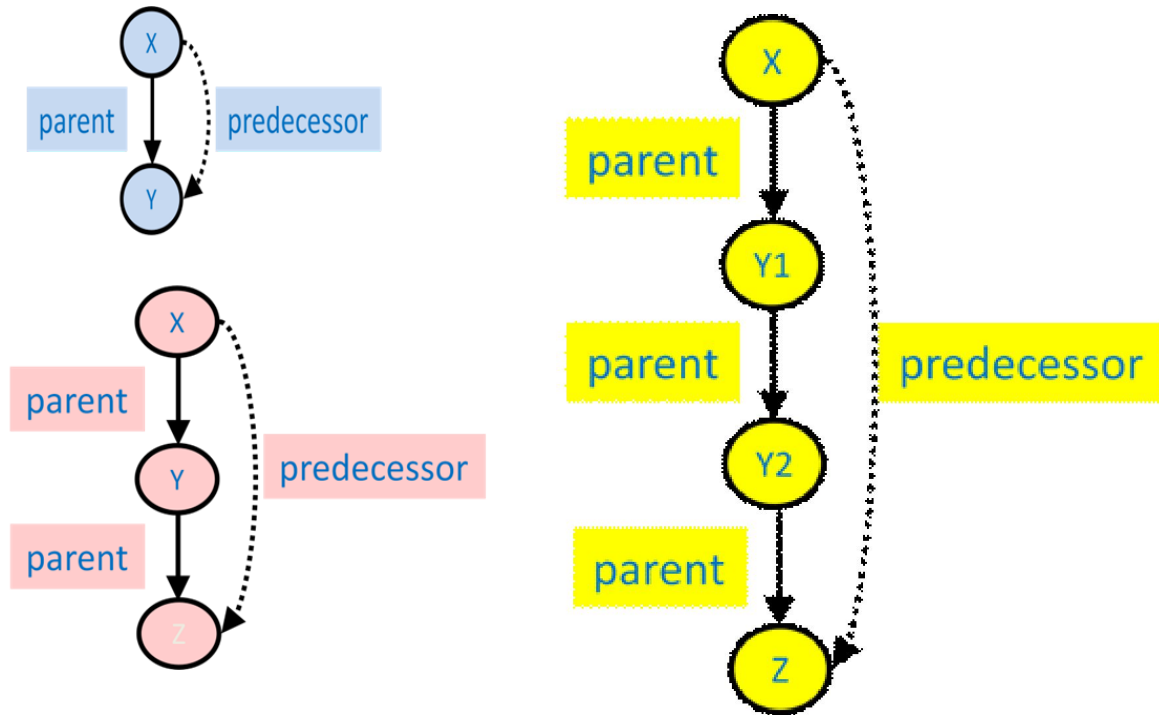
**Output:**

No output.



**Problem no: 04**

**Problem Name:** Define the “predecessor(X, Y)” relation. It should be recursive to satisfy all the conditions depicted below.



**Source code and output:**

```
%Rules
predecessor(X,Z):- parent(X,Z).
predecessor(X,Z):- parent(X,Y),predecessor(Y,Z).
```

**Output:**

No output.

**Problem no: 05**

**Problem Name:** By using Prolog clauses given below, trace the recursive predecessor relation in order to demonstrate how Prolog engine answers queries.

parent(pam,bob).

parent(tom,bob).

parent(tom,liz).

parent(bob,ann).

parent(bob,pat).

parent(pat,jim).

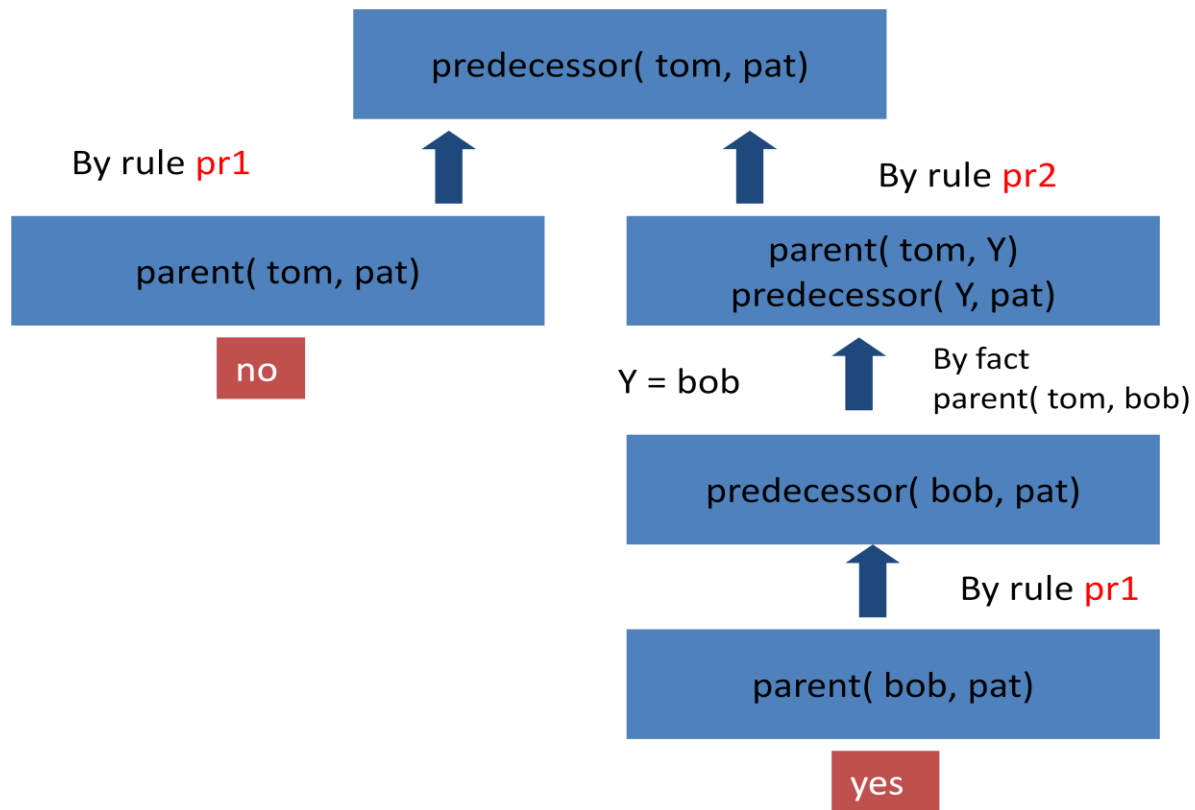
predecessor(X,Z):- parent(X,Z).

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

**Source code and output:**

?- predecessor(tom,pat).

true.



#### Problem no: 06

**Problem Name:** Encode following knowledge in Prolog and answer the given queries;

Asif likes to play tennis, while Umer and Fahad play football. Kashif regularly plays hockey. Tariq, who is friend of Asif plays every sports which Asif plays. Moreover, all of these persons also like to play cricket.

Queries

- Who plays tennis ?
- Which sports Tariq plays ?
- Which game is played by which person ?

**Source code and output:**

```
%CLAUSES
```

```
play(asif,tennis).
```

```
play(umer,football).
```

```
play(fahad,football).
```



play(kashif,hockey).

play(tariq,Sport):- play(asif,Sport).

play(Person,cricket):- play(Person,Sport).

%GOALS

?- play(X,tennis). % Who plays tennis ?

X = asif ;

X = tariq .

?- play(tariq,Sports). % Which sports Tariq plays ?

Sports = tennis ;

Sports = cricket.

?- play(Person,Game). % Which game is played by which person ?

Person = asif,

Game = tennis ;

Person = umer,

Game = football ;

Person = fahad,

Game = football ;

Person = kashif,

Game = hockey ;

Person = tariq,

Game = tennis ;

Person = tariq,

Game = cricket .

### **Problem no: 07**

**Problem Name:** Write Prolog goal statements to answer the following questions on the basis of tabular knowledge given above;

- What is phone number of Kashif ?

- Does Asif belongs to Lahore ?
- Where does Noman lives ?
- Does Fahad and Tariq lives in same city ?
- Who lives in same city where Kashif lives ?

Relations

- Belongs\_To (Person, City)
- Same\_City (Person, Person)

Person Name	City Code	Phone no
Kashif	051	544525
Asif	042	124536
Fahad	051	879546
Tariq	021	112223
Noman	051	555555
Jamil	042	665655
Aslam	051	481526

City Name	Code
Islamabad	051
Lahore	042
Karachi	021

**Source code and output:**

```
%CLAUSES
```

```
phone_no(kashif,051,544525).
```

```
phone_no(asif,042,124536).
```

```

phone_no(fahad,051,87954).
phone_no(tariq,021,112233).
phone_no(noman,051,555555).
phone_no(jamil,042,665655).
phone_no(aslam,051,481526).
city_code(islamabad,051).
city_code(lahore,042).
city_code(karachi,021).
belongs_to(Person,City):- phone_no(Person,Code,Number),city_code(City,Code).
same_city(P1,P2):- phone_no(P1,Code,N),phone_no(P2,Code,M).

%GOALS

?- phone_no(kashif,X,Y).

X = 51,
Y = 544525.

?- belongs_to(asif,lahore).

true.

?- belongs_to(noman,X).

X = islamabad.

?- same_city(fahad,tariq).

false.

```

### **Problem no: 08**

**Problem Name:** Define can swim relation for the animals which live in water. Use output statement in goal portion to display results to the user.

#### **Source code and output:**

```

%CLAUSES

type(ungulate,animal).

type(fish,animal).

is_a(zebra,ungulate).

is_a(herring,fish).

```

```

is_a(shark,fish).
lives(zebra,on_land).
lives(frog,on_land).
lives(frog,in_water).
lives(shark,in_water).
can_swim(Y):- type(X, animal), is_a(Y,X), lives(Y, in_water).
%GOALS
?- can_swim(X), write("A "),write(X) ,write(" can swim\n").
A shark can swim
X = shark.

```

#### **Problem no: 09**

**Problem Name:** Define reference relation to store phone numbers of four persons (use dummy data). And in goal portion write input statements to input query data from user at runtime.

#### **Source code and output:**

```

%CLAUSES
reference("Albert", "01-123456").
reference("Betty", "01-569767").
reference("Carol", "01-267400").
reference("Dorothy","01-191051").
%GOALS
?- write("Please type a name:"),readln(Name).
Please type a name:Carol.
Name = ['Carol', '.'].

?- reference(Name,Phn_no),write("The phone number is: "),write(Phn_no).
The phone number is: 01-123456

```

```
Name = "Albert",  
Phn_no = "01-123456" .
```

**Problem no: 10**

**Problem Name:** Write a Prolog program to demonstrate how Prolog engine can input numbers and process those numbers. As an example you may define simple addition and multiplication procedures in Prolog.

**Source code and output:**

sum :-

```
    write('Write a number: '),  
    read(Number1),  
    read(Number2),  
    process1(Number1,Number2).
```

multiply:-

```
    write('Write a number: '),  
    read(Number1),  
    read(Number2),  
    process2(Number1,Number2).
```

process1(Number1,Number2) :-

```
    C is Number1 + Number2,  
    write('sum of '),write(Number1),write(' and '),write(Number2),write(': '),write(C),nl.
```

process2(Number1,Number2) :-

```
    C is Number1 * Number2,  
    write('Multiplication of '),write(Number1),write(' and '),write(Number2),write(': '),write(C),nl.
```

%GOALS

?- multiply.

Write a number: 4.

|: 6.

Multiplication of 4 and 6: 24

true.

?- sum.

Write a number: 4.

|: 6.

sum of 4 and 6: 10

true.

### **Problem no: 11**

**Problem Name:** Write Prolog procedure to calculate factorial of given number.

**Source code and output:**

```
%CLAUSES
```

```
fact1(0,1).
```

```
fact1(N,F) :-
```

```
(
```

```
    N>0 ->
```

```
(
```

```
    N1 is N-1,
```

```
    fact1(N1,F1),
```

```
    F is F1*N
```

```
);
```

```
write('N should be greater than 0')
```

```
).
```

```
%GOALS
```

```
?- read(X),fact1(X,Ans),write(X),write("Factorial is:"),write(Ans),nl.
```

|: 5.

5Factorial is:120

X = 5,

Ans = 120 .

### **Problem no: 12**

**Problem Name:** How Prolog program can be used to find Greatest Common Divisor of two integers.

#### **Source code and output:**

```
%CLAUSES
gcd( X, X, X).
gcd( X, Y, D) :-
    X<Y,
    Y1 is Y-X,
    gcd( X, Y1, D).
gcd( X, Y, D) :-
    Y<X,
    gcd( Y, X, D).
%GOALS
?- gcd(35,21,Ans).
Ans = 7 ;
```