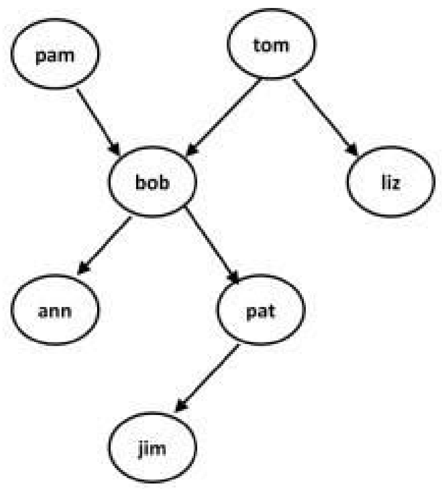
Problem No: 1
Problem Name: Implement the given family tree in the form of Prolog Clauses. Every arrow in the diagram represents a Parent relation. For example pam is parent of bob, tom is parent of bob and so on.



Source Code:

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

Input And Output:

```
?- parent( bob, pat).
true.
```

Problem No: 2
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:

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

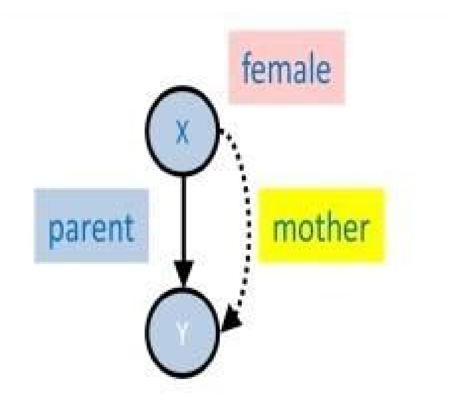
Input And Output:

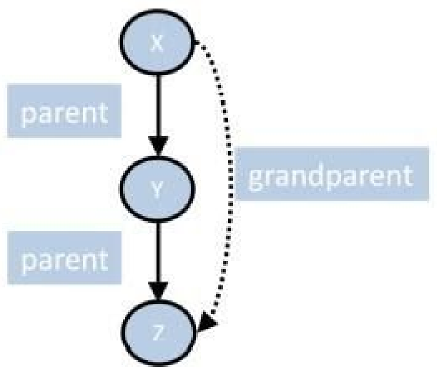
```
?- parent( bob, pat).
true.

?- parent( X, liz).
X = tom.

?- parent( bob, X).
X = ann ;
X = pat.

?- 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.

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

?- parent( Y, ann), parent( X, Y).
Y = bob,
X = pam ;
Y = bob,
X = tom.

?- parent( tom, X), parent( X, Y).
X = bob,
Y = ann ;
X = bob,
Y = pat ;
false.

?- parent( X, ann), parent( X, pat).
X = bob.
```

Problem No: 3
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:
```
parent( pam, bob).
parent( tom, bob).
parent( tom, liz).
parent( bob, ann).
parent( bob, pat).
parent( pat, jim).

female( pam).
female( liz).
female( ann).
female( pat).

male( tom).
male( bob).
male( jim).

offspring( Y, X) :-
        parent( X, Y).
mother( X, Y) :-
        parent( X, Y), female( X).
grandparent( X, Z) :-
        parent( X, Y), parent( Y, Z).
sister( X, Y) :-
        parent( Z, X), parent( Z, Y),
```

female(X), X \=Y.

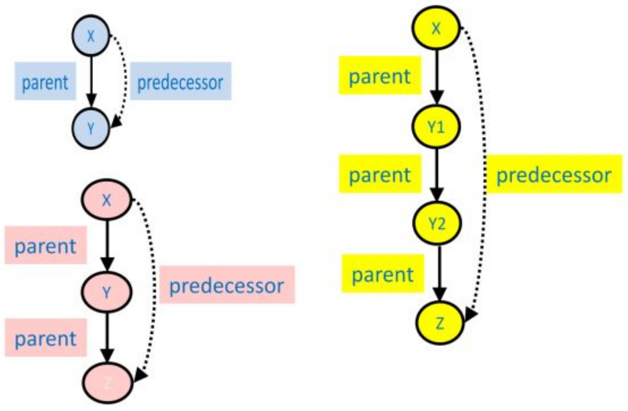




Input And Output:

```
?- offspring( liz, tom).
true .

?- mother(pat,jim).
true.

?- grandparent(bob,jim).
true.

?- sister(ann,pat).
true.
```

Problem No: 4
Problem Name: Define the "predecessor(X, Y)" relation. It should be recursive to satisfy all the conditions depicted below;



Source Code:

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

female( pam).
female( liz).
female( ann).
female( pat).

male( tom).
male( bob).
male( jim).

offspring( Y, X) :-
        parent( X, Y).
mother( X, Y) :-
        parent( X, Y), female( X).
grandparent( X, Z) :-
        parent( X, Y), parent( Y, Z).
sister( X, Y) :-
        parent( Z, X), parent( Z, Y),
        female(X), X \=Y.

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

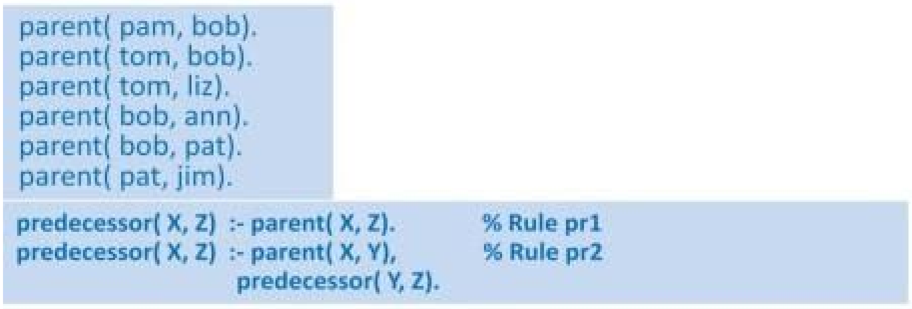
Input And Output:

```
?- predecessor(bob,jim).
```

```
true .

?- predecessor(pat,jim).
true .

?- predecessor(tom,pat).
true .
```

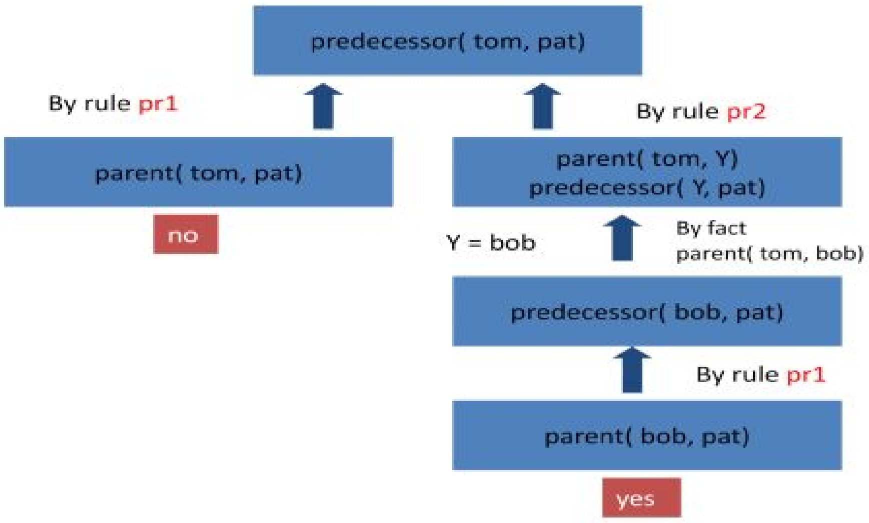
Problem No: 5

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).          % Rule pr1
predecessor( X, Z)  :- parent( X, Y),          % Rule pr2
                       predecessor( Y, Z).
```

Solution:



Input And Output:

```
?- predecessor(tom,pat).
true .
```

Problem No: 6
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:
```
play(asif, tennis).
play(umer, footbal).
play(fahad, footbal).
play(kashif, hockey).

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

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

Input And Output:

```
?- play(X, tennis).
X = asif ;
X = tariq .

?- play(tariq,Sport).
Sport = tennis ;
Sport = cricket ;


?- play(Person,Game).
Person = asif,
Game = tennis ;
Person = umer,
Game = footbal ;
Person = fahad,
Game = footbal ;
Person = kashif,
Game = hockey ;
Person = tariq,
Game = tennis ;
Person = tariq,
Game = cricket .
```

Problem No: 7
Problem Name:
• Create knowledge base in Prolog for data given below;

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

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)

Source Code:
```
phone_no (kashif , 051, 544525).
phone_no (asif, 042, 124536).
phone_no (fahad, 051, 879546).
phone_no (tariq, 021, 112223).
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).

belongsto(Person, City):-
            phone_no(Person, D, _),
            city_code(City, D).
same_city(P1, P2):-
             phone_no(P1, D, _),
             phone_no(P2, D, _),
            P1=:= P2.
```

Input And Output:

```
?- phone_no(kashif,_,M).
M = 544525.

?- belongsto(asif, lahore).
true.

?- belongsto(noman,City).
City = islamabad.

?- same_city(fahad,tariq).
false.

?- same_city(kashif,Person2).
Person2 = fahad ;
Person2 = noman ;
Person2 = aslam.
```

Problem No: 8
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:

```
 type(ungulate,animal).
type(fish,animal).

is_a(zebra,ungulate).
is_a(herring,fish).
is_a(shark,fish).
is_a(whales,fish).
is_a(dolphins,fish).

lives(zebra,on_land).
lives(frog,on_land).
lives(frog,in_water).
lives(shark,in_water).
lives(whales,in_water).
lives(dolphins,in_water).
can_swim(Y):-
        type(X, animal),
        is_a(Y,X),
        lives(Y, in_water),nl,
        write(Y),
        write(" can swim in water").
```

Input and Output:

```
?- can_swim(shark).
shark can swim in water
true.
```

Problem No: 9

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:

```
reference("Albert", "01-123456").
reference("Betty", "01-569767").
reference("Carol", "01-267400").
reference("Dorothy","01-191051").

input:-
        write("Please type a name :"),nl,
        read(The_Name),nl,
        reference(The_Name,Phone_No),nl,
        write("The phone number is: "),
        write(Phone_No).
```

Input and Output:

```
?- input.
Please type a name :
 |: Carol.
The phone number is: 01-123456
true .
```

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:

```
add_em_up(X,Y,Sum):-
        Sum is X+Y.
multiply_em(X,Y,Product):-
        Product is X*Y.

input:-
         write("give first number: "),
        read(X),
         write("give Second number: "),
         read(Y),
         add_em_up(X,Y,Sum),nl,
         write("Addition: "),
         write(Sum),nl,
         multiply_em(X,Y,Product),nl,
         write("Multiply: "),
         write(Product),nl.
```

Input and Output:

```
 ?- input.
give first number: 5.
give Second number: |: 6.

Addition: 11

Multiply: 30
true.
```

Problem No: 11
Problem Name: Write Prolog procedure to calculate factorial of given number.
Source Code:
```
fact1(0,1).
fact1(N,Result) :-
            N > 0,
            N1 is N-1,
            fact1(N1,Result1),
            Result is Result1*N.

factorial:-
           write("give number: "),nl,
            read(X),
            fact1(X,Ans),
            write("The factorial of "),
            write(X),
            write(" is: "),
            write(Ans),nl.
```

Input and Output:
```
 ?- factorial.
give number:
|: 5.
The factorial of 5 is: 120
true .
```

Problem No: 12
Problem Name: How Prolog program can be used to find Greatest Common Divisor of two integers.
Source Code:
```
gcd( X, X, X).

gcd( X, Y, D) :-
  X<Y,
  Yl is Y-X,
  gcd( X, Y1, D).

gcd( X, Y, D) :-
  Y<X,
  Y1 is X-Y,
  gcd( Y1, Y, D).
```

Input and Output:

```
 ?- gcd(39,26,Result).
Result = 13 .
```